# From Policy *to* Proof

## Building Information Governance That Works in Practice




JASON C. STEARNS, FAI, IGP, CRM

# From Policy to Proof

# Building Information Governance That Works in Practice

Jason C. Stearns, FAI, IGP, CRM

## Contents

Advance Reader Copy

Dedication

AI Use Disclosure

About the Examples

Part I: The Execution Gap

Chapter 1: If It Only Lives in Policy, It Is Not Governance

Chapter 2: Policy Defines Intent. The Environment Determines Whether It Works.

Chapter 3: The Real Cost of the Status Quo

Chapter 4: What You Retain Shapes Incident Scope

Chapter 5: Know Where You Stand

Part II: Structure as the Foundation

Chapter 6: From Spreadsheet to System

Chapter 7: Why Retention Needs Structure

Chapter 8: Know What You Have: The Information Inventory and Systems Map

Chapter 9: Categories, Buckets, and the Classification Model

Chapter 10: Global Retention in Practice

Part III: Execution Across the Environment

Chapter 11: Making Governance Rules Operational Across Systems

Chapter 12: Consistency Across Repositories and Platforms

Chapter 13: From Policy to Action: The Disposition Problem

Chapter 14: Legacy Systems Accumulate Risk

Chapter 15: Legal Hold and the Interruption of the Lifecycle

Chapter 16: Collaboration Platforms and Ephemeral Content

Chapter 17: Third Parties, SaaS, and Information Outside Your Direct Control

Part IV: Defensibility and Evidence

Chapter 18: Defensible by Design
Chapter 19: Retention Is Not Static
Chapter 20: Visibility as Control
Chapter 21: Measuring What Matters
Chapter 22: Test the Control, Not the Documentation
Part V: Governance in an AI Environment
Chapter 23: What AI-Enhanced Classification Changes, and What It Does Not
Chapter 24: Records Retention Has Not Caught Up to AI
Chapter 25: From AI Principles to Evidence
Chapter 26: Information Governance as the Precondition for AI
Part VI: Scaling and Sustaining the Program
Chapter 27: Building the Operating Model
Chapter 28: Making the Case and Funding It
Chapter 29: From Pilot to Program
Chapter 30: Governance at Speed
Chapter 31: The Evolving Information Governance Professional
Chapter 32: Governance as Infrastructure
Appendices
Appendix A: Baseline Diagnostics and Maturity Self-Assessment
Appendix B: Operating Model Roles and Decision Rights
Appendix C: Metrics Library and Sample Governance Scorecard
Appendix D: Questions to Ask Before an AI Project
Appendix E: Skills Matrix for Information Governance Teams
Appendix F: Implementation Artifacts
Appendix G: Evidence and Testing Artifacts
Appendix H: Schedule Structure Reference
About the Author

## Advance Reader Copy

This edition of *From Policy to Proof* is an Advance Reader Copy, prepared for advance reading and review.

## Dedication

*For the people who do the work after the policy is signed.*
*You already knew the hard part was never writing it down.*
*The bubbles are on me.*

## AI Use Disclosure

Artificial intelligence tools were used to help organize this book and to assist with copy editing. AI was not used to draft the book's content. The ideas, analysis, and substantive writing are the author's own.

## About the Examples

All case studies are composites based on recurring patterns observed across multiple organizations. Identifying details have been altered.

Figures, retention periods, and timeframes in examples are illustrative, not benchmarks. Diagnostic sample sizes and suggested cadences are starting points to adapt to the population, exposure, requirements, and available evidence.

# Part I: The Execution Gap

# Chapter 1: If It Only Lives in Policy, It Is Not Governance

Many information governance programs are better than they look from the inside.

The organization may have a current policy library. Its retention schedule may have been reviewed by counsel, mapped to citations, and approved through a process with real rigor behind it. There may be a records management policy, an acceptable use policy, a data classification standard, and a committee that meets on a defined cadence. Someone has done careful, disciplined work, often over several years, and that work is real.

Then a question arrives that the program cannot answer quickly.

It is rarely a hard question. A regulator asks how a category of records was disposed of, and when. Opposing counsel asks what happened to a set of files after a matter closed. An auditor asks for evidence that a three-year retention rule was applied in the system where those records actually live. The policy is clear. The schedule says exactly what should have happened.

What follows is often several weeks of coordinated effort across records, legal, and IT to reconstruct an answer that everyone assumed already existed.

The book focuses on the distance between governance as written and governance as operated.

## Three Layers That Are Frequently Treated as One

The confusion begins with vocabulary. Organizations use the words policy, governance, and control as though they describe the same thing at different levels of formality. They describe different work, done by different people, producing different artifacts.

**Policy expresses intent.** It states what the organization has decided should happen. A retention schedule is a policy instrument. So is a classification standard, an acceptable use statement, or a records definition. Policy answers the question of what the rule is.

**Governance assigns decisions and accountability.** It establishes who sets the rule, who approves changes to it, who owns it operationally, how exceptions are requested and granted, and what happens when the rule and the business collide. Governance answers the question of who decides, and how.

**Control shapes what people and systems actually do.** A control may be a configuration in a content platform, a workflow that routes an approval, an automated disposition rule, an access restriction, or a documented procedure performed by a trained individual and supported by evidence. Control answers the question of what actually happens to information.

Policy without control remains an expression of intent. Control without governance may operate, but its ownership and authority are unclear. Effective governance connects the rule, the decision

process, and the mechanism that carries it out.

### One Rule, Three Layers

The retention period in this example is illustrative, not a recommendation. Actual treatment should follow the applicable requirements and the organization's approved analysis.

The distinction is easier to see through a single rule. Take a common one: employee personnel files, retained for seven years following separation.

At the policy layer, the organization is in good shape. The retention schedule defines the category clearly, states the seven-year period, cites the underlying employment and tax requirements, and distinguishes personnel files from the payroll and benefits records that carry different periods. Counsel reviewed and approved that treatment for the scope of this example.

At the governance layer, the organization is also in reasonable shape. Human resources is the named information owner. Changes to the retention period require legal review and records committee approval. A legal-hold process exists to suspend disposition when preservation obligations apply. The category is also on the annual review calendar.

At the control layer, the picture changes. Separated employee records live in at least four places: the core HR system, a shared drive used by HR business partners, the ticketing platform where offboarding was processed, and the mailboxes of the people who managed the separations. The HR system has no configured disposition capability for the category. The shared drive has no retention applied at all. The ticketing platform retains everything by default. Email is governed by a mailbox-level rule that has nothing to do with personnel files.

Nothing deletes. The seven-year rule is accurate, approved, owned, and reviewed, and it does not operate anywhere.

This pattern is common, and it is worth being precise about the cause. The policy and governance layers may both be functioning as designed. The missing layer is operational control, often because building and maintaining it was never assigned as anyone's defined responsibility.

Programs can invest unevenly across the three layers. Policy development has a clear owner and a recognizable completion point. Operating controls is continuing work, distributed across systems and teams with other priorities.

That distance is the execution gap, a structural consequence of how the work is organized rather than a failure of intent or expertise.

## A Simple Test

A focused diagnostic can reveal something that broad maturity assessments sometimes miss: whether a specific rule operates where the information actually resides. An initial trace may take

about an hour when the necessary people and records are available. A wider review across multiple categories requires a defined scope and a more structured exercise.

Pick one rule from the retention schedule. Choose a category with regulatory weight or litigation exposure, not an easy one. Then trace it.

- Which systems hold information in this category?
- What mechanism enforces the rule in each of those systems?
- Who owns that mechanism, by name?
- When did it last operate or execute, and how was that confirmed?
- If someone asked for evidence that it executed, what would you hand them?

A program may answer the policy and mechanism questions confidently while lacking a current owner or operating evidence. Trace the selected rule and state what the records support before drawing a wider conclusion.

A gap can appear between the stated mechanism and current responsibility for it. A system may have been configured at some point without a recent check that the configuration still matches the rule or a named owner for maintaining it. Missing evidence calls for investigation; it does not by itself establish that the control failed.

This is worth doing before reading further. The rest of the book is more useful with a concrete answer in hand.

## Why the Gap Persists

The distance between policy and control is not an oversight. Several forces keep it open, and most of them are reasonable on their own terms.

**Many governance reviews have historically emphasized policy.** They asked whether a policy existed, had been approved, and had been communicated. Testing whether those policies operated consistently across systems often received less attention. That emphasis is changing, as Part IV discusses, but the habits formed around it remain.

**Ownership often divides at exactly the wrong seam.** Records and legal teams may own the policy while IT and platform teams own the configuration. Both groups can perform their respective work competently while neither owns the connection between them. The weakness becomes visible when a system is upgraded or replaced and the retention configuration does not survive the migration.

**Policy development can be a bounded project.** Policy maintenance and control operation are continuing work. Rewriting a retention schedule has a scope, a budget, and a completion date. Maintaining enforcement across forty systems needs defined ownership, ongoing resources, and review. Where organizations fund the initial project but not continuing operation, the layer needing sustained attention receives episodic attention instead.

A control can stop working without producing a signal the governance program monitors. If a policy is out of date, someone eventually notices during review. If a retention rule silently stops executing after a platform update, the change may go unnoticed. The absence of a signal is interpreted as the absence of a problem.

**Acquisitions and reorganizations can strand control ownership.** Policies usually survive because they are documented. Knowledge of who configured a disposition job, how it was scoped, and why content was excluded may not. Integration programs also tend to prioritize infrastructure consolidation over governance reconciliation. Divestitures create the mirror problem. These events can leave mechanisms without owners and owners without usable controls. They are among the forces most likely to undo work that was implemented correctly.

The gap can remain unnoticed while costs accumulate. An unenforced rule may not produce a signal the program monitors, even as retained volume, discovery scope, and potential exposure grow. The limitation becomes more visible when a question requires an answer under time pressure.

These conditions do not necessarily reflect poor judgment. A program organized around an earlier information environment may need to adapt its responsibilities, maintenance, and evidence to the work now required.

## What Works in Practice

A global manufacturer had a thorough, jurisdictionally mapped retention schedule reviewed annually by counsel. By a documentary standard, the program was strong.

A regulatory inquiry asked a narrow question about the disposition of one record category. Answering it took six weeks. The schedule was correct. The difficulty was that no one could demonstrate the rule had ever been applied in the ERP system where those records actually lived.

The instinct in that situation is to fix the schedule. The team did something more useful. They left the schedule alone and traced twelve categories from policy language through to system enforcement.

The categories were selected using three factors: the strength of the underlying regulatory citation, whether the category had appeared in litigation or an investigation during the previous five years, and the sensitivity of the data. The team also included two categories expected to pass and made the selection jointly with legal and IT. Neither adjustment increased the cost, and both made the results more credible to the participating teams.

For each category they documented the systems in scope, the enforcement mechanism, a named control owner, and whatever evidence existed.

Nine of the twelve had no enforcement mechanism at all. Two had mechanisms that had been configured years earlier and never validated. One worked as intended and could produce evidence.

That result was uncomfortable. It was also the most valuable thing the program produced that year, because it converted an abstract concern into a specific, sized list of work with owners attached.

### What Happened Next

The team did not attempt to close nine gaps at once. They sorted them into three groups and treated each group differently.

**Three involved capabilities that already existed but had never been configured.** The platform supported disposition, but no one had enabled it for the category. These gaps closed within a quarter at minimal cost and were addressed first because early completions built credibility for the harder work that followed.

**Four required controls the systems could not provide natively.** For two, the team implemented a documented manual control with a quarterly cadence, a named reviewer, and a retained attestation. A manual control can be effective when it is documented, consistently performed, reviewed, and supported by retained evidence. The remaining two required integration work and entered the following year's technology roadmap with business cases drawn from the trace results.

**Two were not enforcement problems at all.** Nobody could reliably identify which content belonged to the category in the first place. No mechanism would have helped. These moved into the inventory work in Chapter 8 and classification work in Chapter 9, which is the prerequisite the other two groups did not need.

Twelve months later, the program had not closed every gap and did not claim to have. What it had was a named control owner for all twelve categories, a documented mechanism for each, and a defined evidence artifact. Six could produce that evidence on demand. The rest had a target date and an owner.

Two changes proved more durable than the individual remediation efforts. The team replaced schedule completeness with control coverage as its headline metric, shifting the audit committee conversation from documentation to operation. When the next regulatory question arrived ten months later, the answer took four days rather than six weeks.

The lesson is to connect the two questions: whether the policy is appropriate and whether it operates. A sound schedule is the starting point, not a substitute for implemented controls and evidence.

## Implementation Steps

**Separate the three layers in how you describe your own program.** In your next steering committee update, report policy, governance, and control as distinct lines rather than a single status. The reporting change alone tends to surface where attention has been concentrated.

**Run the trace test on a small, deliberately chosen set of rules.** Ten to fifteen categories can provide a useful initial diagnostic. Choose by regulatory exposure, litigation history, and data sensitivity rather than by ease. The findings describe that selection, not the entire program; timing depends on access to people, systems, and evidence.

**Name a control owner for every traced rule, distinct from the policy owner.** The control owner is accountable for the mechanism operating as intended in a specific system. This is often the most consequential single step, because it creates accountability where none existed. Chapter 27 develops the full ownership model.

**Record what evidence exists today.** For each traced rule, document what you could produce if asked, and how long producing it would take. This becomes the baseline for the evidence work in Part IV and, separately, a credible input to the funding conversation in Chapter 28.

**Turn findings into managed work.** Record the gaps with owners, priorities, target dates, and the evidence needed to verify completion. This gives the program a basis for action and follow-through while preserving the findings and their significance.

## What Good Looks Like

In a mature operating environment, the policy library may look much as it did before. The visible difference is elsewhere.

Each high-priority governance rule has a named policy owner and a named control owner, usually different people. The rule traces to the systems where it applies and to the mechanism used in each one. Configuration changes are reviewed instead of discovered later, and new systems enter an established ownership structure rather than triggering an entirely new governance project.

When someone asks what happened to a category of information, answering is a retrieval exercise rather than an investigation.

## What Proves It

The evidence that this chapter's argument has been acted on is specific and reviewable.

- A rule-to-system-to-owner register covering the organization's highest-exposure categories, with a review date and a named owner for each entry.
- Trace test results for the selected categories, showing enforcement mechanism, last validated date, and available evidence.
- A gap backlog with sequencing, target dates, and accountable owners.
- Steering committee reporting that distinguishes policy status from control status.

None of these are documents produced for their own sake. Each one is a byproduct of work that had to happen anyway.

## Business Value

Closing the distance between policy and control produces returns in four directions.

It reduces response cost. Regulatory inquiries, audit requests, and discovery obligations become bounded work rather than open-ended reconstruction. The difference can amount to weeks of senior legal and IT effort.

It reduces exposure. Rules that actually execute mean less information retained past its useful life, which narrows discovery scope, lowers storage cost, and limits the blast radius of a security incident. Chapter 4 develops that connection.

It improves prioritization. When a program can see which rules operate and which do not, sequencing becomes evidence-based instead of intuitive.

And it makes the program fundable. Executives fund capabilities with measurable outcomes more readily than they fund documentation. A traced, owned, evidenced control environment is a capability. A well-written schedule, on its own, is difficult to distinguish from the version that existed three years ago.

## A Closing Thought

Sound policies and functioning committees can coexist with a control environment that was never built or maintained to match. Where the policy foundation is strong, the priority may be the work connecting it to operation rather than another policy rewrite.

That is a more tractable problem. Much of the foundational policy work has already been done. What remains is to translate those decisions into ownership, controls, and evidence. Operational problems yield to sequencing, accountable owners, and sustained attention. The rest of this book is about that work.

The question is no longer whether the organization has a policy. It is whether the organization can rely on what happens after the policy is approved.

*Next: policy defines intent, but the environment determines whether it works. Why the two drift apart and what widens the distance between them.*

# Chapter 2: Policy Defines Intent. The Environment Determines Whether It Works.

Chapter 1 described a gap. This chapter is about why it widens.

The distinction matters because the two problems require different responses. A missing control can be built. A control that loses effectiveness as the environment changes requires ongoing monitoring and maintenance. A program that treats continuing maintenance as a one-time remediation can return to a similar position after the project ends because the environment continues to change.

The reason is a mismatch in pace. Policy direction should be deliberate and reasonably stable. Individual retention rules still need to change when requirements, business needs, or operating conditions change. A scheduled review supports that process but does not replace it.

The information environment changes on a very different schedule. Content is created continuously. Systems arrive through procurement decisions made across the organization. Business units reorganize, and platforms update themselves. These changes often occur without a governance review, and none of them waits for the annual policy cycle.

Policy remains relatively stable while the environment continues to change. Without ongoing attention, the gap between policy and actual practice grows. Recognizing that drift as an expected operating condition changes how a program plans and measures its work.

## Policy and the Environment Change at Different Rates

Consider the cadences a typical program runs on.

The retention schedule is reviewed annually, and that review is thorough. Counsel checks citations. Business owners confirm categories still reflect how they work. Changes are approved and documented. It is a real control, and the programs that do it well deserve credit for it.

During the same twelve months, the organization may have added new applications, introduced new content types, reorganized business units, changed system ownership, and experienced substantial data growth. A platform update may also have changed retention behavior without triggering a governance review.

The annual review examined the policy. It did not examine any of that.

A schedule review can confirm that approved intent remains appropriate. It does not establish whether that intent is implemented and operating throughout a changing environment. The governance calendar needs to address both questions.

The drift is also difficult to see because policy and the environment can move in opposite directions. Policy may become more refined as categories are subdivided, citations are updated, and definitions

become clearer. At the same time, the environment may become harder to govern as volume grows and content spreads into locations without controls. Each function may be making reasonable improvements, but the result can still be a larger gap between policy and the environment in which it must operate.

## Five Forces of Drift

Chapter 1 covered the structural forces that leave the control layer thin. The forces below are different. They explain why the distance grows even where controls were built correctly.

Creation has outpaced curation. Many retention practices were developed for an environment in which documents were created more deliberately and stored in more predictable locations. Content is now generated as a byproduct of ordinary work. A one-hour meeting can produce a recording, transcript, automated summary, and chat messages without anyone making a separate decision to create or classify each item. The assumptions underlying older retention practices are less reliable in that environment.

**Systems arrive faster than governance can onboard them.** Procurement of software is decentralized in most organizations, and reasonably so, because it lets teams solve their own problems. The consequence is that new repositories enter the environment continuously, each one a new surface where information accumulates outside any retention rule. Nobody intends this. It is a side effect of a procurement model that has other benefits.

Unstructured content often lacks the attributes the schedule depends on. A retention rule may need to know what an item is, when it was created, and when its trigger event occurred. Structured systems are more likely to provide those attributes consistently. In file shares, shared mailboxes, and chat platforms, the necessary context is often incomplete or difficult to apply at scale. The rule may be correct but still lack the information needed for enforcement. This is why inventory and classification generally precede enforcement.

**Organizational change breaks the connection between rules and reality.** Reorganizations, acquisitions, and divestitures change which systems exist, who owns them, and which business processes generate which records. The schedule survives these events unchanged, because it is a document. The operational knowledge behind it often does not. Chapter 1 introduced this as a source of orphaned ownership. It appears again here because it is also a source of drift: after a few such events, the schedule describes an organization that no longer exists in quite that shape.

**Governance can rely too heavily on an annual clock in an environment that changes weekly.** A scheduled policy review cannot replace timely rule changes or ongoing attention to the environment. A control validated in March may stop working in July; a known failure needs action when it is identified, not at the next review.

Most of these forces arise from reasonable decisions made for other purposes. That is what makes drift difficult: there is no single failure to correct.

## The Coverage Question

Chapter 1 offered a trace test that follows one rule to its enforcement mechanism. This chapter offers a companion measure that runs the other direction.

What percentage of the organization’s in-scope information is subject to an operative retention rule?

A first estimate can be completed this quarter using in-scope stored volume as the denominator and the volume subject to an operative rule as the numerator. Where only system-level figures are available, label the result a preliminary proxy and document the assumption about coverage within each system. Do not count a whole system as covered merely because one category has a configured rule. Chapter 11 assesses category and repository pairings; Chapter 21 develops the volume measure and its verification requirements.

Operative uses the single verified-implementation definition in Chapter 21 and Appendix C for both technical configurations and documented manual procedures. A preliminary estimate may rely on provisional attribution, but must identify the volume and assumptions not yet supported by verification. Evidence that a control operated remains a separate question.

Use the first estimate to establish a baseline, with its scope, assumptions, and limitations stated. Refine the method as repository and control information improves, and explain changes that affect comparability. Read the trend alongside the current level and the importance of the uncovered information. Direction is useful, but an improving percentage does not by itself show that material gaps have been addressed.

Direction matters, but its cause needs to be understood. When new volume enters repositories without controls, the covered share can fall even if existing configurations remain in place. Maintaining or improving coverage requires both attention to new information and work on existing gaps.

## What Works in Practice

A regional asset manager had a capable information governance program. Its retention schedule was current, reviewed annually with outside counsel, and supported by experienced records staff. Nothing about the program looked neglected.

Prompted by an examination, the team estimated coverage for the first time. Its first-pass calculation classified thirty-eight percent of stored volume as sitting in systems with configured, executing retention rules. The number itself was uncomfortable. What made it instructive was the comparison. Working from infrastructure records, the team estimated coverage two years earlier at roughly sixty percent.

The program had not regressed. It had done good work in that period. Volume had simply grown by a large multiple, and almost all of the growth had landed in collaboration platforms adopted during

a rapid shift to distributed work. The governed systems were still governed. They had become a smaller share of a much larger environment.

That reframing changed the response. The obvious plan was a remediation project covering the ungoverned platforms. The team recognized that a remediation project alone would have restored coverage temporarily and then watched it erode again, because nothing in the plan addressed how new systems entered the environment.

### What They Did Instead

They created two workstreams, each with its own owner, funding, and success measures.

**Stop the inflow first.** Working with procurement and IT, the team established a retention and classification decision as a condition of provisioning any new repository. The decision was deliberately lightweight: which categories will live here, what rule applies, who owns the control, and does the platform support enforcement. Four questions, answered before go-live, documented in the system register. Where the platform could not enforce a rule, that fact was recorded at onboarding rather than discovered years later.

**Then remediate the backlog, in sequence.** The existing ungoverned platforms were prioritized by volume and sensitivity rather than by ease, and worked through over six quarters. This was the larger and slower effort, and it was deliberately second.

The sequence mattered. Without an onboarding gate, remediation would have improved coverage temporarily while allowing the same problem to continue with every new system. The gate required no technology purchase and took about two months to establish. The remediation work cost considerably more and took two years.

Eighteen months in, measured coverage stood at fifty-one percent, and the decline had stopped. Every system provisioned after the gate went into effect had a retention decision attached. The gate made new requirements and gaps visible; remediation and verified implementation increased coverage. Recording a decision alone did not add volume to the covered population.

The percentages and timeframes in this example are illustrative first-pass estimates, not benchmarks or a substitute for the verified measure in Chapter 21. A first-pass report should identify which assessments come from available system records, owner reports, or direct verification and state what remains unconfirmed. Moving to the rigorous method requires the change in basis to be explained.

## Implementation Steps

**Calculate coverage once, with its limitations stated.** Use in-scope stored volume as the denominator and volume subject to an operative rule as the numerator. If system-level data is used as a first-pass proxy, identify partial coverage and document the assumptions. Keep the method

comparable between reporting periods. Appendix C provides the definition, and Chapter 21 develops verification and reporting.

**Establish a governance gate at system onboarding.** Four questions answered before provisioning: which information categories will live here, what retention applies, who owns the control, and can the platform enforce it. Recording an honest no to the fourth question is more valuable than an optimistic yes, because it creates a known gap instead of an assumed control. Appendix F contains the four questions and a decision record template.

Chapters 24 through 26 extend this gate to AI-use record requirements, system evidence, and source-readiness decisions. They distinguish authorization for controlled validation from approval for operational use.

**Separate the inflow work from the backlog.** The two efforts involve distinct owners, costs, and timelines. When they are combined, the backlog tends to consume the available attention. Fund and establish the gate first because it is less expensive and prevents the problem from continuing to grow.

**Review the environment on its own cadence while keeping the scheduled review of the schedule.** A quarterly review can examine new and retired systems, platform updates, and ownership changes. It supports rather than delays responses to known changes or failures. Chapter 19 develops the change process. The two reviews have different agendas and may need different participants.

**Report coverage over time.** A single measurement often leads to debate about methodology, while a consistent quarterly measure shows whether the program is gaining or losing ground.

## What Good Looks Like

In an organization that has addressed drift rather than only its symptoms, coverage is a known number with a documented method, reported on a regular cadence and trending in the right direction.

No repository is provisioned without a retention decision recorded against it, and the register of those decisions is current enough to be useful. Relevant platform changes enter the governance process when identified, with periodic review checking for missed changes. Schedule review and environment review remain distinct activities, with responsibilities assigned for each.

When volume grows or a new platform is adopted, the program treats it as expected work rather than as a surprise. Growth stops being an event that erodes governance and becomes a condition the program is built to absorb.

## What Proves It

The evidence is quantitative in a way that Chapter 1’s is not, which makes it useful in front of an audit committee.

- A coverage measurement with a documented methodology, calculated at least twice so a trend exists.
- A system onboarding register recording, for each repository added, the retention decision, the control owner, and whether the platform can enforce the rule.
- Minutes or a decision log from the quarterly environment review, showing what changed and what was decided in response.
- A register of systems with no operative rule, each with an owner, a reason, and either a target date or a documented risk acceptance with a rationale, an authorized approver, expiry and any earlier review date, and a link to the open-items view.

Programs are often reluctant to produce the last item, but doing so demonstrates maturity. When a gap is documented, assigned to an owner, and either scheduled for remediation or formally accepted, the organization can manage it. An undocumented gap cannot be evaluated or addressed and is more likely to become a serious finding.

## Business Value

Addressing drift early is generally less expensive than remediating systems after information has accumulated.

Governing a system at onboarding is substantially less expensive than remediating it later. At provisioning, the decision may require an hour and an entry in a register. After three years and several terabytes of growth, the organization may need to identify and classify content created without sufficient metadata, resolve ownership, and account for information subject to legal hold. The cost and complexity increase substantially once information has accumulated and ownership or context has changed.

It also makes the program's position clearer to leadership. A coverage trend answers a practical question: is the situation getting better or worse? Governance reporting often cannot answer that question, which makes it more difficult to explain priorities and support funding requests.

It reduces the risk of surprise. A program that knows where its gaps are can bound its exposure during an examination, an incident, or a discovery obligation. A program that does not know is committing to open-ended reconstruction under time pressure, which is where cost and reputational damage concentrate.

And it makes remediation investment durable. Remediation without a gate produces a temporary improvement and a repeat request for funding in three years. That pattern damages a program's credibility more than the original gap did.

## Treat Drift as an Operating Condition

Policy direction should remain deliberate while individual rules and their implementation respond to relevant changes. The information environment should not be expected to remain static.

Strong programs treat drift as an expected operating condition. They measure it, assign responsibility for it, and include the necessary work in ongoing plans and budgets rather than treating every newly identified gap as an isolated failure.

This approach also changes the conversation with leadership. When a program identifies a declining trend early and presents a plan, it demonstrates that the condition is being managed. The same issue is likely to receive more scrutiny when it is discovered unexpectedly during an audit, investigation, or incident.

Intent and reality will separate. The question is whether the organization notices while the distance is still small.

*Next: What the status quo actually costs, and how those costs accumulate quietly until an audit, an investigation, or a business event brings them forward all at once.*

# Chapter 3: The Real Cost of the Status Quo

Maintaining the status quo carries real costs, even though those costs are not reported in one place.

Organizations operating with the gap described in the first two chapters incur costs continuously through infrastructure, legal, business operations, and IT. Because each function reports only its portion, the total is rarely calculated. The individual amounts may appear ordinary within each budget but can become substantial when combined.

This creates a structural disadvantage for anyone proposing to address the problem. A governance investment reaches the executive committee as a new cost against an apparent baseline of zero. The organization's actual current-state cost is missing because no one has calculated it.

This chapter explains how to assemble a conservative estimate of current-state costs from figures that already exist across several budgets. Chapter 28 addresses how to use that estimate in a broader funding case.

## Where the Money Actually Goes

Four categories provide a practical starting point for identifying current spending and operational cost. A fifth category, risk exposure, should be shown separately because it is contingent and generally should not be included in the annual run-rate calculation.

Storage and infrastructure. Storage is usually the first cost programs identify, but it is rarely the strongest basis for investment. Because unit storage costs have declined, finance leaders may not find raw storage expense persuasive on its own. Include it for completeness, but focus on related effects such as backup windows, replication, migration scope priced by volume, and platform licensing tied to capacity. Those costs can increase even as the unit price of storage falls.

Discovery and response. Collection, processing, and review costs are often tied to volume, which over-retention increases. A matter that might otherwise involve two custodians and four years of records may instead involve nine custodians and fourteen years because the organization cannot support a narrower scope. Depending on matter volume and review requirements, the incremental cost can be substantial and may reach six figures.

Manual effort. Manual effort is often underestimated because it appears as employee time rather than as a separate invoice. People search for records that should be findable. Records staff answer questions that systems should answer. Business users maintain personal filing conventions to compensate for the absence of enterprise standards, and analysts reconcile duplicates. This effort is absorbed into existing workloads and therefore is rarely identified as a governance-related cost.

Decision friction. This category is rarely counted and can be one of the largest. It arises when work cannot proceed because the organization does not know what a repository contains, who owns it, or

whether its information can be disposed of. This can delay system retirement, expand migration scope, complicate acquisition due diligence, and defer platform consolidation. The resulting costs are usually attributed to those projects rather than to the underlying information condition.

Risk exposure. Regulatory findings, contractual exposure, privacy obligations, and security incidents can become more costly when the organization retains information it no longer needs. These exposures may not create a current expense, so present them separately from the annual run rate rather than assigning them a speculative value.

When presenting the model, keep raw storage cost brief and give greater attention to decision friction, which is less likely to have been measured. Identify risk exposure as an important but separate consideration.

## Why the Total Is Never Assembled

The costs are not hidden. Most of them sit in systems of record that finance can query. They go unassembled for reasons that have more to do with organizational structure than with accounting.

**Nobody owns the total.** Each component belongs to a different function, and each function reports it accurately within its own scope. Infrastructure reports storage. Legal reports outside counsel spend by matter. Operations reports headcount. There is no role whose job includes summing them, and information governance rarely has the standing to ask four peers for their numbers without a reason they find compelling.

**The counterfactual is invisible.** You never see the discovery matter that cost less because disposition had run. Savings from governance show up as an absence, and absences do not appear in variance reports. This asymmetry is permanent and worth naming when presenting the model, because the audience will otherwise assume the benefits are speculative and the costs are certain.

**Event-driven costs get attributed to the event.** A costly discovery exercise is recorded as litigation expense. An extended migration is recorded as project overrun. A regulatory finding is recorded as a compliance matter. Each attribution is defensible in isolation, and collectively they ensure that the underlying condition is never charged for anything.

**Falling unit costs mask rising totals.** Storage per terabyte declines while total volume grows faster. Net spend rises modestly year over year, which reads as normal growth rather than as a signal. The metric that would show the problem, cost per unit of business activity, is not one most organizations track.

Deferral can be rational in a single budget period. Disposition requires near-term spending and visible decisions, while many of the costs of continued retention appear later. Someone must decide, and an incorrect decision is more visible than a decision to wait. The resulting costs may fall in a later period and be borne by another team or budget. An individual deferral may be defensible, but the cumulative effect of repeated deferrals is the condition this chapter is pricing.

## What Works in Practice

An engineering and construction firm with approximately six thousand employees faced a data center exit with a fixed lease expiry.

The migration plan encountered forty-three legacy systems, most of them retired applications retained because they held project records that might be needed for warranty claims, disputes, or regulatory inquiry. None could be decommissioned because no one could confirm what was in them or authorize disposal. The default plan was to migrate all forty-three to cloud infrastructure, which was priced at a number the CFO declined to approve.

The CFO's refusal created an opportunity to reconsider the decision. Before estimating remediation costs, the governance lead proposed calculating the cost of maintaining the current state.

### Assembling the Number

Rather than immediately scoping a remediation project, the team spent six weeks assembling the cost of the current state. They did not commission a separate study. They began with existing budget figures and used conservative, documented assumptions only where a cost had to be allocated. Each source figure came from the function that owned it.

Infrastructure supplied storage, backup, and migration figures. Legal supplied three years of e-discovery spending and conservatively estimated the portion attributable to collection scope that disposition might have reduced. IT supplied the annual cost of maintaining the forty-three systems, including licensing, patching, and contractor support. Business operations estimated the time spent searching for project records using a survey conducted for another purpose. The team documented the allocation assumptions and presented the resulting figures as ranges.

The result fit on one page and showed a range rather than a point estimate. It also included an early section identifying information that had a defensible reason to be retained. That represented approximately one-third of the volume. Distinguishing legitimate retention from over-retention improved the credibility of the remaining analysis.

### Pricing the Delay

The figure that changed the conversation was the cost of maintaining the current state for one additional year. The total range remained useful, but it prompted understandable questions about methodology.

An investment is easier to defer when the cost of waiting is not shown. Presenting the annual cost of delay allowed the executive team to compare the proposed investment with the financial effect of maintaining the current state. It did not dictate the decision, but it made the consequence of deferral explicit.

The firm did not fund the full request. It funded a scoped first phase covering the twelve highest-cost systems and agreed to revisit the remainder the following year. The phase was completed on budget. Finance then assumed responsibility for updating the cost model annually, which made the second request part of the regular planning process.

## Implementation Steps

**Start with figures the organization already reports.** If a figure requires a separate study, leave it out of the initial model and document the omission. Where an existing cost must be allocated, use a conservative estimate supplied by the function that owns the underlying number and show it as a range. The initial objective is a conservative, attributable estimate rather than false precision.

**Have each function supply its own figures.** Ask infrastructure for storage, legal for discovery spending, and IT for legacy-system costs. Do not estimate on their behalf. A figure supplied by the function that owns it is more likely to be accepted and can be refined by that function when necessary.

**Publish a range and support the low end.** A point estimate can imply more precision than the model supports. Use the conservative end as the working figure and present the upper end as context rather than as the basis for the request. Overstating the case will undermine the model's credibility.

**Separate legitimate retention from over-retention explicitly.** State what portion of retained information has a defensible basis and should not be affected. This distinction improves the credibility of the analysis and avoids treating all retained information as unnecessary. A model that treats all retention as waste will be read as advocacy.

**Express the result as an annual run rate and a cost of delay.** The run rate shows what the current state costs, while the cost of delay shows the financial consequence of deferring action. Present both on the same page and show contingent risk exposure separately. Chapter 28 explains how to use these figures in the full funding case.

**Ask finance to maintain the model once it is credible.** Finance ownership makes the methodology part of the organization's regular financial reporting rather than a figure used only to support a governance request.

## What Good Looks Like

In an organization that has done this, the cost of current state is a known figure with a documented method, refreshed annually, and it is not owned by the governance team.

Investment proposals are evaluated against the documented current-state cost rather than zero. When a request is declined, record the decision and its known financial and operating consequences. Any required risk acceptance or interim restriction follows its own authority and process; declining the budget does not supply that decision. Project business cases involving

migrations, consolidation, and system retirement draw on the same figures rather than each developing their own.

Use the cost model alongside applicable requirements, exposure, resilience, and dependencies. Work that does not reduce a modeled cost still needs a stated purpose and an appropriate basis for the decision, including work proposed by the governance team. Do not require an unsupported cash-saving claim to justify a necessary control or supported risk response.

## What Proves It

- A one-page cost model with each figure attributed to the function that supplied it and the budget line it came from.
- A documented methodology, including what was excluded and why, so the next refresh is comparable.
- A stated split between defensible retention and over-retention, with the basis for the split.
- An annual run rate and a cost of delay, both expressed as ranges.
- Evidence that the model has been used in a decision outside the governance program, such as a migration business case or a project scoping decision.

The last item is the one that distinguishes a cost model from a presentation. A model nobody has used to decide anything has not been tested, whatever its quality.

## Business Value

The model does not reduce costs on its own, but it creates a more reliable basis for investment and prioritization decisions.

It allows a governance proposal to be evaluated as a potential redirection or reduction of existing spending rather than solely as a request for new spending. That framing is more useful when budgets are constrained.

It makes prioritization defensible. When the cost is understood by category, sequencing follows from the numbers rather than from whoever argued most persuasively. It also makes it possible to decline low-value governance work, which programs are often reluctant to do and which does more for their standing than any individual initiative.

It preserves the financial consequence of a declined proposal. When a request is evaluated against a documented annual cost, that cost remains part of the record and can inform the following year's decision.

It also demonstrates that the program can evaluate its work in financial terms. A conservative model maintained with finance allows the governance team to discuss priorities using the same financial measures leadership applies elsewhere. That capability can be more valuable over time than any single funding decision.

## Make the Cost of Inaction Visible

The cost of maintaining the status quo is distributed across several functions, which is why leadership rarely sees the total.

The underlying figures are usually available and can often be assembled in several weeks using a transparent, conservative method and limited additional spending.

The result may be only one page, and its assumptions should be open to challenge. Its value is that future governance proposals can be evaluated against current operating costs and the additional costs accepted when action is deferred.

*Next: over-retention as a security exposure. Why what an organization retains shapes incident scope, and why disposition belongs in security planning.*

# Chapter 4: What You Retain Shapes Incident Scope

Incident response begins with two questions: how did the attacker get in, and what information was affected?

Security teams typically have more established processes for answering the first question.

Logs, alerts, forensic tools, and incident runbooks support the investigation of entry and containment. Responsibility for identifying the affected information is often less clear. The necessary inventory may be incomplete, and the answer may have to be assembled from system owners, repository records, and forensic findings while counsel evaluates potential notification obligations.

Incidents often expose how little information governance and security have coordinated around retained data, ownership, and disposition.

Chapter 3 treated breach exposure as a contingent risk arising from the status quo. This chapter examines that exposure in more detail. Retention is an information governance practice that also functions as a security control. Unlike controls focused primarily on preventing an incident, defensible disposition can reduce the volume of information exposed when prevention fails.

## Impact Is a Retention Decision

Security controls can reduce the likelihood of an incident, limit its spread, or reduce its impact. Identity management, endpoint protection, patching, and awareness training focus primarily on likelihood. Encryption, access controls, segmentation, backup integrity, and defensible disposition can limit scope or impact when prevention fails.

Security investment often emphasizes prevention and detection, which is understandable. When those controls fail, however, the organization may discover that it has made fewer deliberate decisions about the information available to be affected.

Retention is one of the decisions that shapes incident impact, and it is often made years before an incident occurs.

Notification analysis often depends on the number of affected individuals, the types of information involved, applicable law, and the evidence available about access or acquisition. A larger affected population can also increase regulatory and litigation exposure as well as the cost of forensic and legal review. These costs are driven by the information involved in the incident, regardless of whether that information still has business value.

This creates an important asymmetry. Information may outlive its operational, legal, or business purpose while retaining its sensitivity. A customer file kept after those purposes have ended can still expand the analysis and potential exposure associated with an incident.

Within an affected environment, information may be in scope whether or not the organization still uses it.

## Where the Scope Comes From

Unexpected scope often comes from repositories that receive less active management. Four recurring sources are legacy repositories, copies, externally hosted information, and collaboration content.

**Legacy repositories.** Legacy repositories often retain complete populations of sensitive records after the associated application or business process has ended. Ownership may be unclear, and the organization may not know what the repository contains until an incident requires an answer.

**Copies.** Project extracts, migration staging areas, test environments containing production data, and reports downloaded to local shares may persist after their original purpose ends. These copies often receive less lifecycle attention than the source system and may not appear in a repository inventory.

**Information held outside the perimeter.** Vendor platforms, hosted services, and shared environments may hold organizational information even when internal teams cannot directly scan or manage the repository. The organization may retain legal, contractual, or notification responsibilities despite that limited control. Chapter 17 addresses governance in these environments.

**Collaboration content.** Attachments, exports, meeting recordings, and channel histories can accumulate in platforms adopted for productivity without corresponding lifecycle rules. Chapter 16 addresses this category in detail.

These sources commonly lack clear lifecycle ownership. Someone may own the platform or business process without being accountable for the retention and disposition of the information it contains. An inventory that maps repositories to owners and sensitive data types is a prerequisite for addressing that gap.

## Why the Connection Is Rarely Made

Security leaders generally understand the connection once it is framed in terms of incident impact. The challenge is translating that agreement into ownership, measurement, and funded work.

**The two functions measure different things.** Security commonly reports vulnerabilities addressed, detection and response times, phishing results, and endpoint coverage. Records programs report schedule currency, policy attestation, and training completion. Measures of retained-data exposure may appear in neither scorecard, leaving responsibility unclear.

**Disposition creates visible risk for several functions.** Legal may be concerned about future obligations, business owners about losing useful information, and IT about the effort and accountability involved in execution. Continued retention often requires less immediate justification, so it remains the default.

**The benefit is difficult to observe when the control works.** When disposition reduces the amount of information exposed in a later incident, the avoided population is rarely measured. The program sees the records that require review, not the records that were no longer present.

**Data minimization is often managed primarily through privacy programs.** That can leave policy authority with one function while the systems, budgets, and operational ability to delete information sit elsewhere.

None of this reflects a lack of seriousness about security. It reflects an organizational gap between two programs that may each operate well within their own boundaries.

## What Works in Practice

A public university with approximately twenty-eight thousand students had a deliberately decentralized technology environment, with schools and administrative units operating their own systems alongside central services. The security program was capable and well regarded. The records program had one full-time position and a carefully maintained retention schedule that was applied consistently in only two systems.

Ransomware reached a departmental file server through a contractor account. The security team contained the incident in less than a day. The investigation found evidence of unauthorized access to the server but could not determine precisely which files had been viewed or acquired. That uncertainty required the institution to identify the potentially affected population.

Establishing what the server held took five weeks. It contained student records dating to the late 1990s, scholarship applications with financial information, and medical accommodation files copied during a migration eleven years earlier and never removed. The potentially affected population was several times larger than initially expected. The attacker determined the point of entry, but the institution's prior retention and ownership decisions influenced the amount of information that required analysis.

What the institution did next is more instructive.

The institution did not begin with a comprehensive records project. With one person in the records office, that approach would have taken years and was unlikely to reduce exposure quickly enough. Instead, security and information governance created a joint effort around one measure: the count of sensitive records in repositories with no active business use or clear lifecycle owner.

The team ran sensitive-data discovery and ranked repositories by that count rather than by storage volume or department. Three repositories contained more than half of the identified exposure, and none appeared on a current systems list. The team then worked with general counsel and the appropriate institutional owners to approve disposition of defined classes of aged duplicates and closed-unit content, subject to legal-hold and regulatory checks.

Over eighteen months, the team addressed eleven repositories and reduced the measured sensitive-record count by slightly more than sixty percent. The institution reported the measure quarterly on the security dashboard alongside metrics already reviewed by the audit committee.

Two elements of the approach are broadly transferable.

The first is the choice of measure. The team measured sensitive records in repositories with no active business use or clear lifecycle owner. Unlike storage volume, this measure was directly connected to potential incident scope and changed as repositories were remediated. The second is category-level authorization. Approving defined classes of content allowed the team to execute disposition consistently rather than seek legal review for individual files. Chapter 13 addresses this issue in detail.

The result did not require a rewritten retention schedule, a centralized technology model, or a new platform. The work was funded through the security budget and led jointly by two people who had not previously worked together.

The figures and timeframes in this example are illustrative. The measured reduction relates to the selected repositories and inclusion criteria, not to every sensitive record held by the institution.

## Implementation Steps

**Change the unit of measure before anything else.** Count sensitive records by repository rather than relying only on storage volume. State the repository population and inclusion criteria, including whether the measure covers all sensitive records in scope or focuses on aged or unmanaged information. Sensitive-record counts help identify potential exposure; they are not the same as a notification population.

**Rank repositories by aged sensitive content and unclear lifecycle ownership.** Focus discovery on environments likely to contain both, including decommissioned application data, shares belonging to reorganized units, migration staging areas, and long-lived departmental storage. A first pass might focus on ten to fifteen high-risk repositories, with the selection and size adjusted to the environment. That is a starting example, not a representative enterprise sample. The inventory work described in Chapter 8 can accelerate the analysis, and the results can improve the inventory in return.

**Establish category-level disposition authority with counsel and the appropriate business or records owners.** Present defined classes of content, such as aged duplicates, extracts superseded by a system of record, and content belonging to closed business units, rather than lists of individual files. Document the approved classes, required hold and regulatory checks, responsible owner, and evidence retained after execution. Clear category-level authority prevents the work from becoming a queue of individual approval requests.

**Add a scope question to incident response.** After every incident and tabletop exercise, ask what would have made the affected population smaller. Record the answer as remediation work with an

owner. This creates a repeatable link between incident-response findings and governance remediation with little additional process overhead.

**Report the measure through the existing security cadence, audience, and dashboard.** A measure that appears only in a records update is less likely to be treated as an enterprise-security priority. Chapter 28 develops the broader funding argument.

## What Good Looks Like

In an organization where this connection has been made, the change is visible in operating practices as well as policy.

The incident runbook contains a step for identifying the affected data population. A maintained inventory supports that step and reduces the need for an open-ended repository investigation. The organization can establish an initial scope in hours, giving counsel useful information earlier in the response.

Sensitive-record volume in repositories without clear lifecycle ownership is tracked over time and reviewed by the committee responsible for security posture. System decommissioning includes a disposition decision with a named owner, and the project remains open until that step is completed. New systems identify sensitive data types during onboarding through the gate described in Chapter 2, helping keep the inventory current.

The security and governance roadmaps include shared work such as sensitive-data discovery, access review, and defensible deletion, with coordinated ownership and funding.

## What Proves It

The evidence is concrete because much of it is produced through existing security, legal, and governance activities.

- Sensitive data discovery results by repository, with a trend line across at least three reporting periods.
- Documented category-level disposition authorization from counsel, with the hold check applied before execution.
- Disposition records showing what was destroyed, under what authority, on what date, and by whom.
- A repository register identifying owner, sensitive data types, and retention state, including repositories with no current business owner.
- Tabletop after-action documentation showing the time required to establish the affected data population.
- Roadmap items appearing in both the security and governance plans, with a single funding source.

Track the tabletop result over time. It provides a practical indicator of how quickly the organization can establish scope without waiting for an actual incident.

## Business Value

**The most immediate benefit is a smaller potential incident scope.** Fewer sensitive records in an affected environment can reduce the population requiring analysis, potential notification obligations, regulatory and litigation exposure, and forensic and review costs. Organizations can reduce that exposure in advance, often without purchasing new technology.

**Faster scoping matters as well.** Establishing scope in the case example took five weeks. Similar delays can consume valuable time while legal, regulatory, and disclosure decisions are being made. A maintained inventory and clear ownership improve the quality and timeliness of those decisions.

**The work also opens a different funding conversation.** Information governance programs often compete with other compliance priorities. Framing the work as incident-impact reduction allows it to be considered within the security portfolio, where risk-reduction funding and executive attention may already exist.

**Finally, the work improves the organization's ability to answer external questions.** Cyber-insurance questionnaires, customer security reviews, and acquisition diligence increasingly include questions about data minimization and retention practices. A program that can describe what it holds, where it is located, and how long it is retained can answer from documented records rather than estimates.

## Treat Retention as Part of Security Planning

Disposition completed before an incident can reduce the information exposed when it occurs; later deletion cannot undo that exposure. The potential scope reflects retention, ownership, and disposition decisions made over the preceding years.

The attacker may determine where an incident begins, but the organization's prior information-management decisions influence how much sensitive information is available to be affected.

That makes retention and defensible disposition relevant to both governance and security. These practices complement preventive controls by reducing potential impact and by giving incident-response teams better information about affected repositories.

Security and information governance should evaluate incident impact together, using shared measures, remediation priorities, and evidence.

*Next: a practical maturity lens. Assessing structure, execution, ownership, visibility, and evidence separately, because most programs are strong in some and weak in others.*

# Chapter 5: Know Where You Stand

The first four chapters produced four diagnostics in different units and with different levels of confidence. This chapter places them in a common maturity lens without averaging away the differences.

The value is not the score. It is seeing where the program is stronger or weaker than the overall assessment suggests.

## The Composite Score Problem

A maturity assessment summarized as one overall level is easy to communicate and place on a slide. The convenience of that summary does not establish which capability needs attention.

By itself, however, it offers little guidance about what to do next.

A composite score averages across dimensions that do not average sensibly. A program with excellent structure and no evidence capability is not the same as a program with adequate structure and adequate evidence, even if the two arrive at the same overall level. The first may need a focused evidence workstream. The second requires improvement across several dimensions. The score conceals exactly the information that would tell you which situation you are in.

Composite scores can also obscure where improvement is occurring and encourage investment in areas that are already relatively strong. A program can spend a year improving the dimension its team knows best and see its overall level rise, even though a material weakness remains untouched. That weakness may be the one that matters most during an audit or review.

The alternative is not a more sophisticated model. It is a simpler one that assesses five dimensions separately and treats the differences between them as findings rather than averaging them away.

## Five Dimensions

Five dimensions recur throughout the book, and each should be assessed separately.

**Structure.** Whether the rules are expressed in a form people and systems can apply. A retention schedule with clear categories, maintained citations, and defensible periods scores well here. So does a classification model people can actually use. Part II develops this dimension. Teams with a history of schedule and policy work may begin with greater capability here; the assessment should establish the actual position.

**Execution.** Whether the rules operate in the systems where information lives. Chapter 1 uses the trace test to examine selected controls, and Chapter 2 introduces coverage as an initial view of implementation. Coverage does not by itself demonstrate successful operation. Comparing execution with structure helps identify where approved requirements have not reached practice.

**Ownership.** Whether accountability is assigned, named, and current. The question is not whether a committee exists, but whether a specific person is answerable for a specific control in a specific system and whether that assignment survived the last reorganization. Chapter 27 develops the full model. Here, the assessment considers whether that ownership is already in place.

**Visibility.** Whether the program can see its own environment: what systems exist, what they hold, where sensitive information sits, and what has changed since the last review. Chapter 4's sensitive record count assesses this dimension directly. So does the ability to answer, without launching a project, what is stored where.

**Evidence.** Whether the program can demonstrate that the other dimensions operated as intended. This is the dimension referenced in the book's title; assess its actual condition rather than infer it from the other ratings. Evidence that was not captured when a control operated may be difficult or impossible to reconstruct later. Chapters 18 and 22 develop this dimension.

Assessed separately, the five dimensions produce a maturity profile rather than a single score. A program strong in structure and ownership but weak in execution and evidence requires a different response from one strong in execution and visibility but weak in structure.

## Mapping the Four Diagnostics to the Five Dimensions

The Part I diagnostics provide inputs for execution, ownership, and visibility. Complete the profile with a focused review of rule structure and a separate assessment of evidence availability.

| Diagnostic | Source | Dimension assessed |
|---|---|---|
| **Trace test** | Chapter 1 | Execution and ownership. Whether a rule reaches a system and who is accountable for it. |
| **Coverage measure** | Chapter 2 | **Execution.** How much of the in-scope environment is subject to an implemented rule. |
| **Cost model** | Chapter 3 | Not a dimension. Measures the consequences of weaknesses identified elsewhere. |
| **Sensitive record count** | Chapter 4 | Visibility. Whether the organization can see what it holds and where. |
| **Evidence review** | This chapter | Evidence. What the program could produce today, and how long it would take. |
| **Rule structure review** | This chapter, developed in Chapters 6 and 7 | Structure. Whether sampled rules contain the elements needed for consistent application and controlled change. |

The structure review checks whether sampled rules have clear categories, an authority, an approved retention treatment, any required trigger, and a maintained history. Chapters 6 and 7 develop those requirements. The evidence review uses the Chapter 1 trace results to assess what the program can produce to demonstrate that a control operated and how long production takes. Both reviews can begin with existing records.

The cost model does not map to a dimension. It measures consequence rather than capability. Use it to explain why the identified gaps matter, but do not treat it as a maturity dimension.

## Why Uneven Maturity Goes Unnoticed

**Programs are staffed for their strongest dimension.** A records function built around schedule development will keep producing excellent schedules. A team that came from IT will keep improving system configuration. Capability attracts work, work builds capability, and the dimensions nobody on the team is equipped to address stay where they are. This is not a criticism of anyone. It is what specialization does.

**Reporting often follows the work the program already measures.** A committee shown schedule completeness may not see gaps in ownership, implementation, or evidence. Chapter 20 addresses visibility, accountability, and escalation; Chapter 21 develops the measures and their reporting method.

External assessments tend to produce composites. Maturity reviews commissioned from outside frequently deliver a level and a set of recommendations sequenced by effort rather than by dimension. The findings may be accurate but still fail to show which capability requires attention first.

Weakness in evidence may remain unnoticed until evidence is requested. Controls should create and retain evidence as they operate. A program that has not checked whether those records are complete, retrievable, and usable may discover the gap only during an audit, inquiry, or dispute.

These patterns often develop because the program has spent years building capability in one direction while other dimensions received less attention.

## What Works in Practice

A software company of about twelve hundred people differed from the pattern described in the earlier chapters.

The company was fourteen years old. It had no legacy mainframes, no acquired subsidiaries running unsupported systems, and no filing rooms. Its environment was cloud-native and administered by an engineering team accustomed to automating operational processes. When retention rules were defined, they were implemented in configuration within weeks.

The program could implement retention configurations quickly. Visibility was also strong because the engineering team could produce storage and access reports on request. Those capabilities did

not, on their own, establish whether retention controls had operated reliably over earlier periods.

Structure was a different matter. The retention schedule consisted of eleven categories drafted quickly during a certification effort, several of which did not correspond to how the business generated records. Ownership was limited: the schedule had been written by a compliance manager who had since moved into a different role, and no successor had been named. Retention automation was in place, but it did not create a durable execution record. Historical operation therefore could not be established from retained evidence.

The composite assessment the company had commissioned two years earlier rated it at level two of five and recommended improving policy documentation, expanding the schedule, and establishing a governance committee. The recommendations were reasonable, but none addressed the weaknesses in ownership and evidence that would have become apparent during a detailed review.

### What the Dimensional Assessment Changed

The dimensional assessment distinguished the ability to implement automation from evidence that it had operated. Storage and access visibility were strengths. Ownership and retained evidence were weaknesses, and the schedule structure needed revision. Without the execution records, the program could not give an unqualified assessment of historical control performance.

The response followed the identified gaps. The engineering team added logging and retained disposition records within the existing automation. That work took less than a month and improved the program's ability to document future operation; it did not recreate missing historical evidence. The schedule was rebuilt around how the business worked, reducing eleven categories to nine and aligning each category with actual business records. A named owner was assigned to each category.

Twelve months later, the company completed a customer security and compliance review that had previously required three weeks of manual evidence assembly. The same work took two days. A composite score might not have clearly reflected the overall change, but the program had materially improved the two dimensions most relevant to the review.

The figures and timing in this example are illustrative. They describe changes in the program's ability to manage and produce evidence, not a retrospective validation of periods for which execution records were missing.

## Implementation Steps

**Assess the five dimensions separately and resist combining them.** Use a consistent scale, which may be as simple as strong, adequate, or weak. The scale matters less than the discipline of not averaging the results. Appendix A contains the full self-assessment.

**Use the Part I diagnostics as inputs rather than starting over.** Reuse the trace results, coverage measure, and repository findings, then review rule structure and evidence availability separately. Record where the existing material is insufficient to support a rating.

**Identify the widest material gap and treat it as a central finding.** Strong rule structure with weak execution may indicate an implementation gap. Where automation is in place but operating evidence is missing, investigate whether the problem concerns evidence generation, control performance, or both. The assessment should identify the work needed without assuming the outcome.

**Prioritize the weakest dimension that creates a material constraint or risk.** Do not default to the most familiar or easiest work. If the team lacks the capability to address the priority gap, treat that as a staffing or sourcing issue rather than a reason to focus elsewhere.

**Reassess annually and retain the prior maturity profile.** Compare results on a stated, comparable basis and explain changes in scope, method, or evidence. A single assessment establishes the baseline; later assessments show changes in the assessed position. Use supporting work records before attributing a change to funded work.

## What Good Looks Like

A program that has completed this assessment can identify its strongest and weakest dimensions and explain how the gap developed.

Investment is directed toward the weakest dimension that creates a material constraint or risk, even when the work is unfamiliar. Improvement is reported by dimension, allowing the steering committee to see, for example, that execution improved while evidence did not. Prior assessments are retained and compared, with supporting work records used to explain which changes can be attributed to funded work.

The assessment should also help the program defer initiatives that strengthen an already-capable dimension while more consequential gaps remain.

## What Proves It

- A dimensional assessment with five separate ratings, each with the evidence or diagnostic result it was based on.
- A stated widest material gap, with a short explanation of what produced it.
- An improvement plan addressing the weakest dimension that creates a material constraint or risk, with owners and dates.
- At least two assessments on a stated, comparable basis, with supporting records for any attribution of change to intervening work.
- Reporting to leadership that shows the five dimensions separately rather than as a level.

A second assessment turns the initial baseline into a measure of progress. Comparing the two shows whether the program improved in the intended direction and provides evidence for future funding decisions.

## Business Value

A dimensional assessment requires relatively little additional work because most of the inputs already exist. It provides four practical benefits.

First, it directs investment toward material constraints. A material weakness in one dimension can limit the value of strengths elsewhere. The maturity profile helps prevent unnecessary investment in an already-capable area, such as funding a schedule refresh when the schedule is not the problem.

Second, it gives leadership a clear account of the program's strengths and weaknesses. Reporting each dimension separately is more informative than reporting a single level and ensures that existing strengths are not lost in an averaged score.

Third, repeated assessments show changes in the assessed position. Comparable results and supporting work records give leadership a stronger basis for evaluating progress and the contribution of funded work.

Finally, the maturity profile helps the reader prioritize the rest of the book. Parts II through IV focus on structure, execution, and evidence. Ownership and visibility recur across all three because they support each of those capabilities.

## Using the Assessment

Part I established that governance is judged by what operates, not only by what is written. It also provided four measures for understanding where an organization stands.

Most assessments will identify both strengths and gaps. The composite software company in this chapter could implement automation quickly but lacked evidence of its historical operation. The composite organization in Chapter 1 had a defensible schedule but only one of twelve sampled categories with a verified, functioning control. Neither program was uniformly weak. The dimensional assessment made the gaps and limits of the available evidence visible.

The maturity profile makes the rest of the book easier to use. Part II develops structure, Part III develops execution, and Part IV develops evidence. Ownership and visibility recur throughout. Readers can begin with the capability they most need to strengthen.

*Next: Part II. Why spreadsheet-based retention schedules break down under versioning, maintenance, and jurisdictional layering, and what a governance system has to add.*

# Part II: Structure as the Foundation

# Chapter 6: From Spreadsheet to System

Part I examined the distance between policy and practice. Part II turns to the structure needed to close it, beginning with the retention schedule. Most schedules reflect careful research and review. The problem is that a spreadsheet is now being asked to support work it was never designed to carry.

## The Schedule's Job Description Changed

When most retention schedules were first built, the job was narrow. State the rule. Get it approved. Make it available to anyone who asks. Under that job description, a spreadsheet is a sound tool. It is fast to build, easy to review, and readable by anyone.

The job description has since expanded, mostly without anyone deciding that it should.

The schedule is now expected to support decisions at the point of use and provide rules that content platforms and business systems can apply. It must also accommodate jurisdictional variation, absorb regulatory change, preserve its history, and support disposition decisions that may later be reviewed.

None of that is a formatting problem. The file is not badly built. The model is being asked to carry operational weight that a document cannot carry, and the strain shows up in predictable places.

## Where the Document Model Breaks

Five recurring demands expose the limitations of the document model, and their effects compound.

**Maintenance.** Regulations change, business processes change, systems are replaced, and categories need refinement. In a spreadsheet, each of those changes requires manual editing across rows that may appear on several tabs. The work is slow, difficult to review, and scales with the size of the schedule rather than the size of the change. Programs often respond by batching updates into an annual review, leaving the schedule out of date between review cycles.

**Versioning.** Spreadsheets copy easily, and that is the difficulty. Business units save local copies. Attachments circulate by email. One version sits on the intranet, another in the records team folder, another in the shared drive legal uses. Each looks plausible. Without version controls, users may be unable to establish which is current. A program that cannot state which copy is authoritative cannot state that any particular decision followed the rule.

**Change history.** Which rule changed, when it took effect, who approved it, and why. A spreadsheet typically presents the current state. Unless version history is maintained deliberately, earlier rules and effective dates are difficult to reconstruct. Tracked changes and file timestamps are not a substitute because they record when a file was touched rather than when a rule took effect. The

limitation becomes consequential when the organization must reconstruct, sometimes years later, the rule that governed a record when it was destroyed.

**Jurisdictional layering.** A category governed differently in six jurisdictions has to be represented somehow. In a document, each available option creates a tradeoff. The organization can repeat the category and maintain several copies of the shared elements, add columns until the row becomes difficult to use, or move the variation into notes that may not be applied consistently. Chapter 10 addresses the governance choices behind harmonization and localization. The narrower point here is that the document format makes accuracy and usability difficult to maintain together.

**Access.** A schedule functions as a control only if the people making decisions can use it. A workbook with several hundred rows, four tabs, and a column of legal citations is a reference document for specialists. Employees rarely consult it. They ask someone, guess, or keep everything. An unusable schedule is a significant source of decision friction.

Individually, each demand looks like something a disciplined team could manage. Together, they explain why capable programs with well-researched schedules still cannot answer basic questions quickly.

## Why the Spreadsheet Persists

The document model survives for reasons that are practical rather than careless.

- It works at the moment it is authored. The schedule may be accurate when approved, while later failures remain difficult to detect or attribute.
- Flexibility reads as an advantage. Anyone can add a column, insert a row, or annotate a cell. That same flexibility is why nothing in the file is enforced and why two people can hold different interpretations of the same row.
- Replacement has no obvious owner. Records owns the content. IT owns the platform. Neither carries a budget line for a system that does not sit cleanly inside a records project or a technology project.
- A working file can conceal a weak operating process. The schedule may still open and look authoritative even when its rules are stale or no longer implemented as intended. The absence of a visible failure signal is not evidence that the underlying work is current.
- Migration surfaces disagreements the organization has been working around. Categories that were never fully settled have to be settled. Overlapping definitions have to be reconciled. That work is real, and postponing it is easy.
- The alternative gets framed as a purchase rather than a capability. When the conversation opens with tools, it stalls in procurement. When it opens with requirements, it tends to move.

None of these are failures of judgment. They describe an artifact that quietly outgrew its format while everyone was using it.

## What Works in Practice

A nonprofit health system operating across several states maintained a retention schedule with several hundred entries in a multi-tab workbook. A records manager and a legal department analyst maintained it. The schedule was in good condition: citations were current, the category structure was coherent, and outside counsel had reviewed it recently.

One state then amended a retention requirement affecting a class of clinical records. The change affected dozens of entries. The edit itself took an afternoon, but confirming that everyone downstream was using the updated file took several months and was never completed with confidence.

The consequence surfaced later in a malpractice matter. Counsel asked which retention rule had governed records destroyed under an earlier version of the schedule and whether the destruction had followed the rule then in effect. The current schedule answered the first question misleadingly and the second not at all. Several copies of the workbook were in circulation, and their file dates reflected when someone had opened them rather than when a rule had changed.

The team's response is instructive because it did not begin with technology.

The team first established one authoritative copy, marked the others as reference material with expiration dates, and added a change log recording the affected category, prior and new rules, driver, approver, and effective date. Reconciliation with state compliance leads confirmed that several older copies were still being used. The team then documented its requirements and migrated the highest-exposure categories first, completing the remaining work in later phases.

The change employees noticed was smaller and arrived sooner. Department managers stopped emailing the records manager to ask how long to retain particular records because the answer had become searchable and was written for their role rather than for counsel.

After the new process had operated for more than a year, the same question from counsel could be answered within an hour, with a change history attached.

## What a Governance System Has to Add

The requirements that came out of that exercise generalize. Whether the organization purchases, builds, or assembles the solution from existing platforms, it should provide the following capabilities.

- **A single authoritative source.** Copies are replaced by controlled access rather than by distribution, so the question of which version governs stops being a question.
- **Structured rules rather than text.** Retention periods, triggers, citations, and jurisdictions held as discrete, queryable elements. Chapter 7 develops this in full, and it is the requirement everything else depends on.

- **Change control with a durable history.** Every change carries an author, an approver, an effective date, and a reason, and prior states remain retrievable rather than overwritten.
- **Point-in-time reconstruction.** The ability to state what rule governed a category on a given past date, produced from the system rather than reassembled from memory and email.
- **Jurisdictional layering held once.** Shared elements defined a single time, with variation expressed as related rules rather than as duplicated categories.
- **Role-appropriate access.** What a department manager needs to see is not what a records analyst needs to see. A system can present both from the same source. A single static document is less suited to providing those different views.
- **Machine-readable output.** The schedule has to publish rule sets that systems can consume. Where direct consumption is unavailable, use the reviewed translation process in Chapter 11 and verify the resulting configuration.
- **Reportable evidence.** What the schedule holds, what changed, and who approved it, available without standing up a project. The evidence work in Part IV depends on it.

These are capability requirements, not product requirements. They may be met through a governance platform or a well-governed database and disciplined process. Appendix F contains the schedule requirements checklist and the related implementation artifacts.

## Implementation Steps

**Establish a single authoritative copy before making any technology decision.** Name the owner, publish the location, and mark every other copy as reference-only with an expiration date. This can often be completed before technology selection with limited additional spending and removes a common source of inconsistent application.

**Start the change log now, in the tool you already have.** Date, category affected, prior rule, new rule, driver, approver, effective date. Every entry captured before migration is an entry you will not have to reconstruct afterward, and the log becomes the first evidence artifact the program owns.

**Time three real questions against the current schedule.** Use questions the business actually asks, such as what governs a specific record type in a specific state, what changed in the last twelve months, and what rule applied on a stated past date. Record the elapsed time for each. Those results establish a baseline and provide useful evidence for a future funding discussion.

**Write requirements before evaluating anything.** Start from the capability checklist in Appendix F, add what your environment demands, and mark each item required or preferred. Requirements written after a demonstration tend to describe the demonstration.

**Sequence migration by exposure rather than by tab order.** Move the categories with regulatory weight, litigation history, and sensitive content first. That way the highest-value capability arrives early, and a migration that slows down still leaves the program better protected than it was.

**Review the existing structure before migration.** Preserve defensible distinctions and revise those the organization cannot apply reliably. A new platform does not, by itself, resolve a category-design problem.

## What Good Looks Like

In an organization that has made this move, the schedule stops being a file people look for and becomes something people query.

One source is authoritative, and users know where to find it. Changes move through a defined approval path and leave a record. Prior rule versions can be retrieved where their history was retained, with missing historical material identified. Jurisdictional variation is held once and related rather than copied. Employees can find answers in language suited to their roles. Published rule sets are consumed directly where supported or translated through the controlled implementation process, with the resulting configuration or manual procedure verified.

Leadership may see only a modest change in the schedule itself. The operational difference becomes clear when the organization can answer questions in minutes rather than days or weeks.

## What Proves It

A program should be able to demonstrate the shift through the following evidence.

- A change history for the retention schedule covering at least the previous twelve months, with an approver and an effective date on every entry.
- A point-in-time query result showing the rule that governed one category as of a stated past date, produced in minutes rather than assembled by hand.
- A count of schedule copies in circulation, tracked over time and trending toward one.
- A requirements document with items marked required or preferred, and the evaluation record scored against it.
- A published rule set carrying a version identifier, consumed directly where supported or translated through a controlled implementation record, with the resulting configuration or manual procedure verified.
- Elapsed time to answer the three baseline questions, measured before and after.

These artifacts should arise from operating the program, not from a separate documentation exercise.

## Business Value

Moving from a document to a governed system creates value in four practical ways.

It reduces the cost of managing regulatory change. Assessing which categories a new requirement touches, applying the change, and confirming it reached everyone shifts from a lengthy coordination

exercise to a scoped update with a record attached. For organizations tracking requirements across many jurisdictions, this can produce recurring savings.

It makes disposition defensible. The ability to show which rule was in effect, when it changed, and who approved it converts a disposition decision from an assertion into a demonstrable one. That difference is what separates a bounded response to a regulatory or litigation question from an open-ended reconstruction.

It reduces decision friction. When employees can find the rule that applies to them quickly and in language they understand, they are less likely to retain information by default.

It provides the structured rules that the execution work in Part III depends on. Machine-readable output reduces repeated transcription where platforms can consume it. Where they cannot, a controlled implementation matrix can translate the approved rule into configuration parameters or a documented manual procedure. Each route needs verification.

## A Schedule the Organization Can Rely On

The spreadsheet was never the mistake. It was the right tool for a schedule whose job was to describe intent, and for many years that was the whole of the job.

Expectations have changed. The organization must now preserve the rule, its history, its jurisdictional variations, and the evidence needed to apply and review it. A document can state the current decision; a governed system can maintain the record behind it.

None of that follows from a purchase. A system does not settle a category dispute or create an owner where none exists, and programs that treat migration as a substitute for those decisions arrive at a well-structured version of the problem they started with.

That is the difference between a schedule the organization maintains and a schedule the organization can rely on.

*Next: why retention needs structure. The information model behind a governance system, and how structured rules, reusable attributes, and maintained relationships make a schedule something you update rather than something you rewrite.*

# Chapter 7: Why Retention Needs Structure

A governance system can publish and preserve rules only if those rules are structured. In a document, a retention rule is a sentence. In a structured model, it is a set of connected elements that can be queried and changed one at a time. The practical question is simple: what does it cost to change one thing?

## The Rule Is Not the Row

Open almost any retention schedule and you will find rules expressed as rows. Category name, retention period, a citation, perhaps a jurisdiction column and a notes field carrying whatever did not fit elsewhere.

The row is a reasonable way to display a rule. It is a poor way to hold one, because a retention rule is not one fact. It is several, and they behave differently.

*Figure 7.1. The eight elements of a retention rule.*

| Element | What it holds | Why it has to be discrete |
|---|---|---|
| Record category | What the information is, in terms the business recognizes | Rules attach to it and systems classify against it |
| Trigger event | The event that starts a finite retention period or initiates a lifecycle action | Event-dependent periods and actions need a defined starting point; not applicable must be explicit |
| Retention period | A finite duration tied to a defined trigger, or an explicit permanent designation | An approved retention value must be distinguishable from an unassigned period |
| Jurisdiction | Where the rule applies, and at what level | Variation has to relate to the base rule rather than copy it |
| Citation and authority | The applicable requirement or source authority supporting the treatment | One citation supports many rules and changes on its own schedule |
| Disposition action | The authorized action when a period ends or another defined lifecycle event occurs | Destroy, transfer, anonymize, and review are different controls |
| Effective dates | When this version of the rule governs | Reconstructing a past rule depends on prior versions persisting |

| | | |
|---|---|---|
| Exceptions and suspensions | What changes or suspends normal treatment, with scope, authority, expiry and review dates for time-bound exceptions, or review dates and release criteria for preservation suspensions | Time-bound exceptions need expiry and explicit renewal; preservation suspensions remain until the applicable release criteria are met |

Two of the eight are developed elsewhere. Chapter 9 treats the design of the category model as a governance decision in its own right, and Chapter 13 covers the mechanics of carrying out a disposition action. This chapter is concerned with how all eight relate to one another.

The citation-and-authority element identifies the requirements supporting the rule. It does not replace the decision record explaining the full rationale or the delegation under which the approver acted. Chapter 18 develops that record and its link to the rule version.

In a spreadsheet, all eight live as text in adjacent cells, and their relationships may not be maintained explicitly. The relationship between a citation and the twenty-nine categories that depend on it exists only in the memory of whoever did the research. The relationship between a global rule and its six local variations exists as a footnote. The relationship between a rule and the systems where it has to operate does not exist at all.

A structured model treats the same eight elements as separate, related objects. Citations, jurisdictions, and triggers can each be defined once and related to every rule that uses them.

Everything Chapter 6 asked for follows from that. The eight elements are reproduced in Appendix H as a schedule data model reference.

## Why Schedules Stay Flat

Most schedules are built flat for reasons that are sound at the time.

- The schedule is built to be approved, and approval is a reading exercise. A reviewing committee wants a document it can read in order. Structure is invisible to a reader and essential to an operator, and the reader is the one who signs.
- Flat is faster to produce. Research, draft, review, publish. Modeling the relationships adds weeks to a project whose visible deliverable looks the same either way.
- The relationships often exist only in the knowledge of the people who developed the schedule. A records manager who conducted the citation research may know which categories a requirement touches, but that knowledge remains undocumented and leaves with the person who holds it.
- Nothing forces the model to be explicit until something has to consume it. A person reading a rule fills in the missing trigger without noticing. A system cannot, which is why structural gaps often surface for the first time during a configuration project.
- Granularity gets confused with structure. Adding categories feels like adding precision, so schedules grow wide rather than deep.

The common thread is that a flat schedule works until someone other than its author has to use it. That includes systems, successors, auditors, and the version of the program that exists three reorganizations from now.

## The Trigger Problem

Of the eight elements, the trigger is the one most often left implicit, and it is the one that quietly decides whether a rule is applied correctly.

A finite retention period needs an event that starts it. Seven years after contract termination gives the period an operational starting point; seven years alone does not. The event is frequently the harder half to determine because it lives in a business process rather than in a regulation.

When the trigger is missing from the schedule, it does not stay missing. It gets supplied by whoever configures the system, and the value they supply is often the date the object was created, because that is the date the platform reliably has.

The result is a rule that is wrong in both directions at once. Records tied to a long-running relationship are destroyed while the relationship is active. Records from a short engagement are held for years past any obligation. Neither failure necessarily produces an alert. Both may remain undetected until the organization has to explain or defend the result.

A structured model requires a defined trigger wherever a retention period or lifecycle action depends on an event. The same trigger should mean the same thing in every rule that uses it. An explicit permanent designation does not require an artificial destruction date, but it may still require cutoff, transfer, or review events. A documented not-applicable value must remain distinct from a missing trigger.

## Permanent Retention Requires a Defined Basis

A permanent designation should record an affirmative retention decision. It should identify why the information warrants permanent preservation, who is responsible for its custody, and how it will remain accessible and usable. An unassigned period is a separate condition and should be reported as an unresolved rule.

Permanent preservation also differs from retention pending an event or a future review. If information is kept until a relationship ends or another condition is met, record that condition and the treatment that follows. If the organization has not decided how long to retain it, record the gap, assign an owner, and set a decision date. Neither condition should be converted silently into permanent retention.

The basis may be an applicable requirement or a documented assessment of enduring historical, evidential, or business value. Record the authority, scope, and approval. Where permanence is discretionary, review the justification and the resources needed to support it. A permanent

designation concerns the information; it does not require the same system, format, or custodian to remain in place indefinitely.

Custody requires continuing work. The organization needs arrangements for access, readable formats, and the context that makes the records interpretable. Plan for platform replacement, migration, and transfer to another custodian where appropriate. Record who accepts responsibility at each transition and what evidence demonstrates that the information remains complete and usable.

Review the basis and custody arrangements on a defined cadence and when relevant requirements or systems change. Review does not itself authorize destruction. Any change to the approved treatment must follow the applicable authority and governance process. The schedule should distinguish supported permanent designations from unresolved periods and identify the work needed to maintain each.

## What Works in Practice

A life sciences company with operations in the United States and Europe maintained a schedule with several hundred categories spanning clinical, manufacturing, and commercial records. It had been developed by an outside firm, reviewed by internal counsel, and published as a formatted document. It was accurate.

The difficulty appeared when a European regulatory requirement was revised and the general counsel asked a question that sounded simple. Which of our categories does this change affect?

Answering the question took several months because citations were recorded inconsistently. Some appeared in a notes column, others in an appendix, and some requirements had been applied through reasoning that was never documented. The team eventually produced an answer, but it could not state how complete that answer was.

The second finding came out of the same review and mattered more. A substantial majority of the categories carried a retention period with no recorded trigger. Those categories had already been configured in the document management platform, and the configuration had used creation date because nothing else was available. Nobody had made that decision. It had been supplied by default by someone doing their best with an incomplete rule.

The rebuild did not add categories. It reduced them slightly while adding structure. Citations became objects with their own records and change histories, referenced by every rule that depended on them. Jurisdictions were modeled with scope, and triggers were defined in a small controlled list tied to identifiable business processes. No category requiring a trigger could be published without one.

After the structured model had been implemented, the same question could be answered within a day, with a list of affected categories, the systems where each was configured, and the person accountable for each change.

The team did not describe the outcome as a better schedule. They described it as a schedule they could change.

## What Structure Makes Possible

The value of a structured model shows up as a set of questions the program can answer without a project.

- **Which categories does this regulatory change touch?** Shared citation records identify every rule that relies on the affected requirement. This question often exposes whether a schedule is structured or merely tidy.
- **What governs this record type in this jurisdiction?** A jurisdictional hierarchy shows the applicable global, national, local, or sector-specific requirements.
- **What rule was in effect on this date?** Effective dates and retained versions allow the organization to reconstruct the rule as it existed at that time. Chapter 6 introduced point-in-time reconstruction as a capability; versioned rule objects provide the mechanism.
- **What starts the clock, and does the system know?** A required, controlled trigger identifies the event and allows the organization to determine whether the system captures it.
- **What rule exceptions are active, and when must they be reviewed?** Modeled rule exceptions retain their scope, rationale, approver, and expiry and any earlier review date. Preservation suspensions remain subject to their documented release criteria.
- **What can we hand to a system?** A structured rule can be published as a rule set. Chapter 11 covers what receiving systems do with it.

None of these require a large program. They require that the information behind the schedule be organized so the answer is retrievable rather than reconstructed.

## Implementation Steps

**Run the single-change test.** Take one citation that has been amended in the last two years. Count how many places in the schedule you would have to edit, how long the edit would take, and how confident you are that you found all of them. The level of confidence is itself an important finding because it shows whether the program can reliably identify every affected rule.

**Inventory your triggers before anything else.** Go through the schedule and mark each required trigger as stated, implied, or absent. Record a reason where a trigger does not apply. Then check what the systems are actually using. The gap between those lists identifies structural work that can often be completed without migrating anything.

**Review permanent designations.** Identify each permanent value and confirm its basis, scope, approval, accountable custodian, and preservation arrangements. Assign unresolved designations to an owner with a decision date, and keep them distinct from approved permanent rules.

**Build a controlled trigger vocabulary.** Keep it small enough that every event can be defined in a sentence and tied to an identifiable business process, and give each one a source system that can

supply the date. If a proposed trigger has no system that knows when it occurred, it is a policy statement rather than an operable rule, and it needs either a different trigger or a manual control with an owner.

**Turn citations into objects.** Create one record per citation, with its own effective dates and change history, referenced by every rule that depends on it. This is often a high-value modeling decision because regulatory change is a recurring reason to update a schedule.

**Model jurisdiction with scope rather than as a label.** A rule needs to know whether it applies nationally, sub-nationally, or by sector, and how it relates to the broader rule it varies from. Getting this right is what prevents a schedule from forking into regional copies, which is the failure Chapter 10 examines in full.

**Give time-bound rule exceptions an expiry date.** Record the scope, rationale, authorized approver, expiry, and any earlier review date. A preservation suspension instead remains in place until its applicable release criteria are met. A scheduled review does not extend an expired exception without a new authorized decision, and expiry does not remove the underlying condition from reporting.

**Resist adding categories during the restructure.** The instinct when modeling is to subdivide, because every edge case looks like it needs its own row. Structure is what lets a smaller number of categories carry more variation. Chapter 9 develops the taxonomy side of that judgment.

## What Good Looks Like

In a program with a structured schedule, maintenance stops being an annual event and becomes ordinary work.

When a regulatory requirement changes, the affected categories can be identified promptly. The citation is updated once, and related rules retain the change and approval history. A rule cannot be published without an authority and any required trigger. Permanent designations carry a documented basis and custody arrangements. A new jurisdiction extends the existing model instead of creating a copy. Ownership is assigned through the governance process, and rule exceptions appear on a report as they approach review or expiry.

The schedule still reads as a schedule when someone wants to read it. The difference is that reading is no longer the only way to use it.

## What Proves It

A structured schedule should be demonstrable.

- A citation impact report listing every category affected by a named requirement, produced on request rather than assembled.
- A trigger coverage figure: the percentage of categories requiring a trigger that have a defined trigger drawn from the controlled vocabulary, tracked over time. Document and

review exclusions from the denominator.
• A reconciliation between scheduled triggers and the date fields systems are actually using, with any differences listed, explained, and assigned to an owner.
• A rule history for one category showing prior versions, effective dates, approvers, and the citation change that drove each revision.
• An active rule-exception register with scope, rationale, approver, and expiry and any earlier review date, with preservation suspensions distinguished by their release criteria.
• A published rule set with a version identifier, generated from the model rather than transcribed from it.
• A record of permanent designations showing the basis, scope, owner, approval, custody arrangements, and review date, with unsupported designations listed as unresolved work.

Trigger coverage is a useful structural measure: it shows whether required triggers are explicitly defined in the schedule. Reconciliation and configuration verification establish whether receiving systems can apply them. Report permanent designations separately so they do not obscure missing required triggers. Appendix C carries trigger coverage as a supporting measure alongside the two anchor measures, coverage and sensitive record count.

Keep the category-level unit: where one category has several scoped treatments, count it as covered only when every required trigger is defined. Link missing-trigger findings beneath that category, record justified not-applicable treatments, and do not omit a permanent designation where cutoff, transfer, or review still depends on an event. Appendix C applies the same counting rule.

## Business Value

Structure pays back in maintenance cost, in accuracy, and in the ability to grow.

It changes the economics of regulatory change. In a flat schedule, the cost of an update scales with the size of the schedule. In a structured one, it scales with the size of the change. For an organization tracking requirements across many jurisdictions, that difference produces recurring savings in maintenance time and effort.

It closes the accuracy gap that defaults create. A defined trigger, correctly mapped and verified in the receiving system, helps the rule execute as approved. That gap produces both premature destruction and silent over-retention, and neither is visible until someone looks for it.

It also reduces the program's dependence on individual knowledge. Relationships held in a model survive a departure. Relationships held in a records manager's working memory do not, and the cost of rebuilding them is usually paid at the worst possible moment, under a deadline set by someone else.

It makes scale a configuration question rather than a rewrite. Adding a jurisdiction, a business line, or an acquired entity becomes an extension of the existing model rather than a separate schedule-development effort.

## Necessary, and Not Sufficient

Structure does not decide anything. A well-modeled schedule full of poor judgments is a poor schedule that is easier to query.

Structure does not replace legal interpretation, ownership, disciplined change governance, or sound category design.

What structure does is make decisions durable. It records a judgment and its relationships in a maintained form, supports consistent application, and makes later review and change possible. Implementation and verification are still needed to establish that the approved treatment reached the information.

## Structure Makes Decisions Durable

Nobody sets out to build an unstructured retention schedule. Schedules often remain flat because the format cannot express the relationships uncovered during research. The author holds those relationships in memory, so the loss may not become visible until someone else has to use the schedule.

The loss appears later, in the months required to answer a question the program should own and in configured rules that use the wrong date because the correct trigger was never documented.

Structure is rarely the most visible part of information governance, but it is the foundation for everything that follows. A program can have excellent judgment, competent people, and clear authority, and still be unable to act on any of it if the schedule cannot be changed one piece at a time.

*Next: know what you have. The information inventory and systems map, and why governance cannot control information it cannot locate.*

# Chapter 8: Know What You Have: The Information Inventory and Systems Map

Chapters 6 and 7 built a schedule the organization can maintain and query. This chapter is about the other half of the problem, which is knowing where the information it governs actually lives.

The usual framing is that organizations lack an inventory. That is rarely true. Most have several.

IT maintains an application portfolio or a configuration management database. Privacy maintains a record of processing activities. Procurement maintains a vendor register that identifies which third parties hold company data. Security maintains an asset inventory covering what its tooling can reach. Each is real work, each is reasonably accurate for its purpose, and each was built to answer a question that is not the governance question.

So when someone asks where a specific record category is held, the program has four sources, four partial answers, and no way to reconcile them. That is the operating problem. It is different from having no inventory and often easier to solve once it is framed as a reconciliation problem.

## Four Inventories, None of Them Yours

Each existing inventory is optimized for the job it was built to do, and that optimization is what makes it unsuitable for governance.

- **The configuration database.** Built for incident response and change management, so its unit is the application and its useful fields are technical. It tells you what runs and who supports it. It rarely tells you what information is inside.
- **The record of processing activities.** Built for privacy compliance, so it is organized around processing purposes and personal data. It is often the closest fit, and it stops at the boundary of personal data, which leaves most of the retention schedule outside its scope.
- **The vendor register.** Built for contract management, so it is organized around commercial relationships and renewal dates. It knows which third parties exist. It is usually silent on what they hold and what happens to it at termination, which Chapter 17 takes up in full.
- **The security asset inventory.** Built for coverage of controls, so it reflects what the tooling can see. Anything the agents cannot reach is not absent from the environment. It is absent from the list.

None of these is deficient. Each is a good answer to its own question. The gap appears because governance asks a question none of them was designed for, and because no one owns the reconciliation between them.

## The Unit of Analysis Problem

The reconciliation problem begins with a more basic question: what counts as an inventory entry?

IT counts applications. Governance needs repositories. These are not the same unit, and the relationship between them is not one to one in either direction.

A single application may write to a production database, an archive, a reporting extract, a document store, and a set of scheduled exports that land on a file share. That is one row in the portfolio and five places where records live, each with different access, different lifecycle capability, and potentially different jurisdictions. Meanwhile a single collaboration platform may serve two hundred applications' worth of business process and appear as one line item.

When a program adopts the application list as its inventory, it inherits that mismatch silently. Rules get mapped to applications, configured wherever the application team has authority, and quietly miss the extracts, the archives, and the reporting copies. Chapter 12 addresses what that does to consistency across an environment. The point here is earlier and simpler. Decide the unit before collecting anything, make it the repository, and record the relationship to applications rather than substituting one for the other.

## Why the Map Stays Incomplete

Inventory work is rarely a priority, and several practical constraints leave it incomplete.

- Completeness is treated as the goal, which makes the project too large to finish. Programs that try to inventory everything at once often produce a partial inventory that is difficult to maintain.
- Inventories are funded as projects and consumed as operations. Accuracy has a half-life measured in months, and nothing in a project structure accounts for that.
- The work looks duplicative to everyone who already has a list. Asking IT to help build a fifth inventory is a difficult conversation, and it is the wrong conversation, since the useful move is reconciliation rather than collection.
- Copies do not announce themselves. Extracts, reporting environments, departmental exports, and analytics platforms are created for legitimate reasons by people acting within their authority, and none of those actions triggers a governance event.
- Nobody owns the reconciliation. Each list has an owner, but the relationship between the lists does not.

The result is an environment that is documented four times and mapped once, incompletely, by whoever needed the answer most recently.

## What Works in Practice

A specialty retailer operating in the United States and Canada maintained all four inventories: a configuration database, a privacy processing record, a vendor register, and a security asset inventory.

The question that exposed the gap was routine. A privacy request required identifying every location holding one customer's transaction history. The four inventories produced several plausible answers,

but the team could not demonstrate that any was complete. Answering the request took more than two months and required substantial coordination across the four teams.

The response was not a fifth inventory. It was a change in direction.

Rather than working forward from the system list, the team worked backward from the schedule. They selected a small set of record categories based on regulatory exposure, litigation history, and sensitivity. For each one, they asked where the information was created, stored, copied, and sent at the end of its life. This was the trace test from Chapter 1, applied to repositories rather than rules.

Tracing that small set of high-exposure categories identified more than two hundred repositories. A significant number appeared in none of the four existing inventories.

The composition of those repositories changed how the team worked. Few were unsanctioned tools. Most were extracts from known systems in reporting environments, analytics workspaces, and scheduled exports created for legitimate business purposes. The primary systems were well governed. The copies were invisible.

The inventory they built added three fields that none of the existing lists carried: a named information owner, the record categories present, and the date the entry was last verified and by whom. Everything else they took from the sources that already had it.

Reconciliation became a quarterly obligation with a named owner, and the four-question onboarding gate from Chapter 2 was connected to the inventory so that new systems entered it when they were adopted rather than at the next review. The onboarding gate captures changes as they occur, while the inventory records the current environment. The two controls need to operate together.

After the new process had been operating for more than a year, a comparable privacy request was answered in several days.

## What a Governance-Grade Inventory Holds

The useful inventory is thinner than most people expect. It is not a description of the technology environment. It is the smallest set of facts that lets the program find information, assign accountability for it, and apply a rule to it.

*Figure 8.1. The governance-grade repository record.*

| Field | What it answers |
|---|---|
| Repository identifier, name, and type | Stable identifier, name, and type: system of record, archive, extract, or working store; link source and copy relationships |
| Business function and named owner | Who is accountable for the information, as a person rather than a department |

| | |
|---|---|
| Business processes supported | Why the information exists, which is what makes the category mapping defensible |
| Record categories present | Which rules apply, expressed in the schedule's own vocabulary |
| Sensitive data types present | Sensitive types identified within a stated scope, method, and date; potential exposure rather than a full incident or notification population |
| Hosting location and residency constraints | Where the information is hosted, whether it may move, and where the control must operate |
| Third party and contract reference | Which outside party holds or manages the information and which agreement governs the arrangement |
| Lifecycle capability | How retention, disposition, and holds are supported, including any manual controls; for deletion, the terminal state and the residual period before that state is reached |
| Status | Lifecycle state, such as active, legacy, or planned retirement, with missing or unconfirmed ownership recorded separately |
| Last verified date and by whom | What was checked, when, by whom, and against which evidence, with the result and limitations recorded |

Three fields distinguish a governance inventory from a technology inventory: a named person accountable for the information, the record categories that connect the repository to the schedule, and a dated verification record identifying what was checked, against what evidence, and with what limitations. Identify the decision authority the information owner holds or the authorized decision-maker to whom they refer the matter. Being named in the register does not, by itself, grant approval authority. Chapter 18 develops the authority record.

The record is reproduced in Appendix F, alongside the system-to-policy implementation matrix that it feeds. The two belong together: the register says what exists and who owns it, and the matrix says what rule applies to each entry and how it is enforced.

Part V reuses these repository references when recording AI systems, uses, and source connections. Those are related units, not interchangeable inventory entries, as Chapters 25 and 26 explain.

Container-based classification works only when the organization knows what each container is and which business process it supports. The repository map supplies that connection.

Lifecycle capability should describe what an action achieves as well as whether the platform supports it. For deletion, record the terminal state, when content can no longer be restored through routine administrative action, and the residual period between the deletion instruction and that

state. This allows the disposition record to distinguish an instruction issued from destruction completed. Chapter 13 explains the treatment of versions, replicas, and backups.

Status matters because legacy and orphaned repositories cannot be addressed until they appear on a list with an owner and a decision attached.

## Implementation Steps

**Reconcile before you collect.** Put the existing inventories side by side and identify where they agree, where they disagree, and what each one is missing. A preliminary reconciliation can often be completed using the inventories that already exist, and it usually produces the first defensible findings. It also changes the conversation with the teams that own those lists from a request for more work into a request for access.

**Work backward from categories, not forward from systems.** Choose a small set of record categories based on regulatory exposure, litigation history, and sensitivity, and trace each one to where it lives. The resulting map is partial by design and covers the information that matters most.

**Set the unit of analysis in writing.** The unit is the repository. Applications are recorded as a relationship, not as a substitute. Write this down before the first workshop, because it is difficult to change once people have started submitting entries.

**Follow the copies.** For every repository in scope, ask what is extracted from it, on what schedule, where it goes, and who receives it. Reporting environments, analytics workspaces, and recurring exports often account for repositories that do not appear in the existing inventories, even though they were created for legitimate business purposes.

**Name an owner for every entry, and record the ones you cannot.** A repository with no owner is a finding, not a blank field. Track the count over time as a measure of program progress. Chapter 27 develops the ownership model those names belong to.

For the Appendix C count, use current entries still holding in-scope information or requiring governance, including relevant legacy or orphaned repositories. Distinguish a missing or known invalid holder from overdue confirmation. A valid interim owner may satisfy current accountability while the underlying vacancy remains open. Record any method change and its effect on the trend.

**Give the inventory a verification cadence rather than a completion date.** Verify the highest-exposure entries quarterly and the rest annually, recording the verification date on each entry. An inventory is an operating asset. Funding it only as a project makes another reconstruction effort likely within a few years.

**Connect it to the onboarding gate.** The four-question gate helps keep the inventory from going stale at the source. Without it, the map degrades from the day it is published and the program returns to periodic reconstruction.

## What Good Looks Like

In a program with a working map, the question of where information lives stops being a research task.

A repository register exists with a named owner on every active entry and a verification date that is current. New systems appear in it at adoption rather than at the next review. Extracts and copies are recorded alongside the systems they come from. Legacy and orphaned repositories carry that status openly, with an owner and a decision attached rather than a silence. The register reconciles to the configuration database, the processing record, and the vendor register on a stated cadence, and the differences that remain are explained rather than unnoticed.

The most telling sign appears during an actual response. When a privacy request, a discovery obligation, or a security incident arrives, the first hour is spent on the response rather than on establishing where to look.

## What Proves It

The evidence is the register itself and the discipline around it.

- A repository register with owner, record categories, sensitive data types, lifecycle capability, status, and last verified date populated on every active entry.
- A reconciliation record against the configuration database, the record of processing activities, and the vendor register, showing what was resolved and what remains explained.
- The count of repositories with no named owner, tracked over time and reported alongside the supporting measures in Appendix C.
- An extract and copy map for the highest-exposure repositories, showing what leaves each one and where it lands.
- Verification evidence for the most recent cycle, dated and attributed.
- A traceable link from the register to the coverage denominator, so that the measure Chapter 21 develops rests on something reviewable.

That last item is worth stating plainly. Coverage is the anchor metric of this book, and every coverage figure has a denominator. If the denominator is estimated, report the uncertainty and maintain a comparable method. The inventory provides the basis for improving that estimate.

## Business Value

An inventory is difficult to fund on its own merits and straightforward to justify through what depends on it.

It reduces response cost across three functions at once. Privacy requests, discovery obligations, and incident scoping all begin with the same question, and each function may be paying to answer it separately. A shared register gives all three functions a common starting point and reduces the

research each one must repeat. That is often the argument that reaches a budget owner because the savings appear in legal, privacy, and security budgets rather than only in the records program.

It converts unknown exposure into a bounded list. Chapter 4 made the case that legacy and orphaned repositories drive breach scope. A repository that appears on a register with a status and an owner is a decision waiting to be made. One that appears nowhere represents exposure the organization cannot yet quantify or assign.

It makes measurement more explainable. A reconciled denominator and documented limits give the coverage figure a basis that others can examine and reproduce.

It is also one of the preconditions for the execution work in Part III. Mapping rules to system configurations, achieving consistency across repositories, and executing disposition all assume the program knows what is in scope. Attempting any of them without a map produces work that appears complete but covers only what happened to be visible.

## Reconcile Before You Rebuild

When a program cannot answer where information lives, the usual response is to commission another inventory. In many organizations, several inventories already exist, and the work is more about reconciliation, ownership, and coordination than building another one.

A working inventory needs four things: an agreed unit of analysis, the three governance fields, an owner for reconciliation, and a verification cadence. Accuracy decays; it is never permanently achieved.

A schedule that cannot find its information is a policy. This chapter addresses the other end of that distance: connecting the schedule to the repositories where information actually lives.

*Next: categories, buckets, and the classification model. Why more categories often produce less consistency, and how to build a model that people and systems can both apply.*

# Chapter 9: Categories, Buckets, and the Classification Model

A structured schedule still needs something to attach its rules to. That is the role of the category model: it defines the distinctions that people and systems must apply consistently.

Every retention schedule answers that question through its categories. The number of categories, the distinctions between them, and who must apply them are often treated as research outputs. They are governance decisions because each category creates a distinction the organization must apply correctly and consistently, usually through people who do not work in records.

A schedule with hundreds of categories requires users and systems to make hundreds of distinctions. That is the operating problem, and the design of the schedule often contributes to it.

## Granularity Is Not Precision

Schedules grow granular for an understandable reason. Legal research produces distinctions, and each distinction looks like it deserves its own line. If a regulation treats one class of record differently, the natural response is to create a category for it.

The difficulty is that precision in the schedule and precision in practice are different things, and they move in opposite directions past a certain point.

Every added category creates another classification decision. Past the point where an ordinary employee can reliably distinguish the options, added granularity reduces consistency instead of improving accuracy.

A common failure mode is predictable. When someone cannot tell which of four similar categories applies, they may choose the one with the longest retention period because keeping too long feels safer than destroying too early. Granularity introduced to increase precision can produce systematic over-retention, which is the cost Chapter 3 sized and the exposure Chapter 4 described. The schedule becomes more detailed while the environment becomes less governed.

## The Different-Outcome Test

There is a straightforward way to size how much of a schedule is doing real work.

A useful first test is whether categories produce different lifecycle outcomes. Categories sharing a trigger, retention period, and disposition action are candidates for consolidation, subject to their legal, business, access, and evidentiary requirements. Matching outcomes do not by themselves establish that the categories should merge. Chapter 10 applies a related test to jurisdictional variations.

Sorting a schedule by those three elements usually produces a number far smaller than the category count. In practice, a schedule with several hundred categories may resolve to a much smaller

number of distinct combinations.

That number approximates the size of the schedule that is producing distinct retention outcomes. Much of the difference is vocabulary, which can be preserved through aliases without maintaining a separate rule.

Appendix H carries the category model design rules alongside the schedule data model from Chapter 7, since the two describe the same structure from different ends. Chapter 7 defines what a rule is made of. This chapter defines what the rule attaches to.

## What Big Buckets Actually Do

The argument for consolidating a schedule into a smaller number of broader categories is well established, and it is often described as simplification. That framing undersells it and invites the obvious objection, which is that simplification sounds like a loss of rigor.

Consolidation does not remove the judgment. It relocates it.

In a granular schedule, the hard judgment is made at classification time, by whoever happens to be filing the document, thousands of times, without training, under time pressure, with no feedback on whether they got it right. In a consolidated schedule, that judgment is made once, at design time, by people who are qualified to make it, with the reasoning documented and reviewable.

Consolidation can cause some information to be retained longer than its individual requirement demands. Assess that effect against applicable retention limits, deletion requirements, business needs, and other constraints before approving a shared period. Record the analysis and preserve separate treatment where requirements differ. Choosing the longest period and documenting the choice is not sufficient on its own.

The question is not whether to bucket. It is where the judgment belongs, and how much of it the organization can afford to distribute.

## Three Vocabularies, One Model

A category model must reconcile three vocabularies, but most schedules are written primarily in one.

The schedule speaks records language, precise and formal. The business speaks process language, shorter and specific to a function. Systems speak container language, in sites, libraries, folders, object types, and record classes. A category called Accounts Payable Invoices and Supporting Documentation is unambiguous to its author, is called vendor bills by the people who create it, and exists in the platform as a library with a naming convention.

When these three do not reconcile, the schedule is usable only by the group whose language it was written in. Employees cannot find their content in it, and system owners cannot map it to anything they administer.

The resolution is to separate the label layer from the rule layer. Categories carry the rules. Business vocabulary is held as aliases pointed at those categories, as many as the organization needs, without any of them acquiring a retention period of its own. System containers map to categories the same way. Business units keep their own names, and none of those names becomes a rule the program has to maintain.

## Why Schedules Grow and Do Not Shrink

Category counts tend to increase for structural reasons, not simply because of poor maintenance.

- Adding is cheap and removing is expensive. Adding a category may require only one conversation. Retiring one usually requires evidence that the change will not create a gap, and that work rarely receives the same attention.
- Requests for rule exceptions often arrive as requests for new categories. A business unit with an unusual need asks for a row, and adding a row may be easier than determining whether an existing category already covers it.
- Research output maps one to one. Each citation identified during development looks like it deserves representation, and the schedule becomes a record of the research rather than a design.
- Granularity looks defensible. A detailed schedule appears to demonstrate care. It demonstrates care in authorship, which is not the same as care in execution, and it is the execution that gets examined.
- Classification accuracy may go unmeasured. Without an agreement rate, a schedule that is failing in practice looks identical to one that is working.

That last one is the reason the others persist. A program with no measure of classification consistency has no evidence that granularity is costing anything, and no argument for reducing it.

## What Works in Practice

A property and casualty insurer operating across multiple states and lines of business maintained a schedule with several hundred categories. It had been developed carefully over many years, each addition made for a defensible reason, and it had never been reduced.

The program was competent and the schedule was accurate. What nobody had measured was whether it was being applied.

The team sampled several hundred documents that employees had classified during the previous year and had a records analyst classify them independently. The two classifications agreed in only about one case out of three. Most disagreements were not errors in any meaningful sense. They reflected reasonable people choosing differently between categories that were difficult to distinguish.

They then sorted the schedule by trigger, retention period, and disposition action. The several hundred categories resolved to fewer than one hundred distinct combinations.

The redesign produced slightly more than one hundred categories. The difference between the distinct outcome combinations and the final category count was deliberate, preserving a small number of distinctions the business needed for reasons other than retention, including some tied to litigation history in one line of business.

The move that made it acceptable to the business was the alias layer. Every one of the retired category names was kept as a searchable label pointed at its new category. Claims handlers still search for the terms they had used for a decade. None of those terms carries a rule, so the aliases need to be maintained as mappings rather than researched as separate retention rules.

When the team repeated the exercise more than a year later, the two classifications agreed in roughly four cases out of five. The schedule had become less granular on paper and more consistently applied in the environment.

## Where the Classification Decision Happens

Designing the categories is half the model. The other half is deciding where the classification decision is made, and the options are not equally reliable.

Four common methods are available, and their reliability depends on how and where they are used.

**Container.** Everything in a defined location carries the category. This is often the most reliable default where the container maps clearly to a business process. It depends on the repository map in Chapter 8, which records the business process each container supports.

**Ingest rule.** The system assigns the category at creation, based on source, form, or workflow. It fits structured processes with a system of record, such as claims, invoices, or case files.

**User declaration.** The person creating or filing the content chooses. It is appropriate for cases that need individual judgment, and only where the chooser has the context to choose well.

**Inference.** Content is classified by pattern recognition against a trained model. It fits backlogs and unstructured stores, with human review designed in. Chapter 23 covers it in full.

Use container-based classification where the location reliably identifies the category. It moves the judgment from filing to container setup, where someone with the necessary context makes it once. Ingest rules, user declaration, and inference serve different operating contexts. Choose the method the organization can support and verify rather than defaulting to individual user decisions.

The practical ceiling on granularity depends on the classification methods the organization can apply reliably. Containers can support distinctions tied to location or business process. Ingest rules and validated inference can support additional distinctions. Where those methods cannot resolve a category reliably, a person with sufficient context may need to decide.

## Implementation Steps

**Measure agreement before redesigning anything.** Sample a few hundred classified documents and have a second qualified person classify them independently. The agreement rate provides a useful baseline and can often be developed more quickly than a full schedule review. It gives the consolidation proposal evidence that the current design is producing inconsistent results.

**Run the different-outcome test.** Sort the schedule by trigger, retention period, and disposition action, and count the distinct combinations. Use the result to identify consolidation candidates, then check which distinctions remain necessary for legal, business, access, or evidentiary reasons.

**Set granularity by who has to apply it.** If the applier is a system with a reliable signal, the model can carry more distinctions. If the applier is an employee filing a document between meetings, it cannot. Write down who the applier is for each part of the model, because that assumption tends to be made implicitly and optimistically.

**Separate labels from rules.** Retire category names as categories and keep them as aliases. This makes consolidation easier for business units to accept because they retain their familiar vocabulary without maintaining separate retention rules.

**Define each category with exclusions and a worked example.** A definition that says what is included leaves the boundary open to interpretation. A definition that says what is excluded and shows one real document that belongs and one that does not makes consistent application possible. This step is often under-resourced in schedule projects.

**Choose the method for the category and operating context: container, ingest rule, user declaration, or validated inference.** Record the choice and the evidence supporting its use. Retain working methods where their evidence supports them; Chapter 23 develops the validation and review needed for inference.

**Build a retirement path.** Establish how a category is proposed for removal, who approves it, and what evidence is required. Without a defined path, the count only rises, and the next consolidation becomes another fifteen-year project.

## What Good Looks Like

A working category model is smaller than most practitioners expect and better documented than most schedules are.

Each category has a documented reason for remaining distinct. In most cases, that reason is a different lifecycle outcome. Each category carries a plain-language definition with exclusions and a worked example. Business vocabulary is preserved as aliases that carry no rules. Most content is classified by where it lives rather than by what someone chose. The classification method is recorded for each category, so system owners know what they are configuring. Classification

agreement is measured on a stated cadence and reported rather than assumed. A defined retirement path has also been used at least once.

The visible result is a model that can be explained clearly to a new employee. That is a practical test of whether its distinctions are understandable without specialized records training.

## What Proves It

The evidence for a category model is about application rather than design.

- A classification agreement rate from independent reclassification of a sample, measured on a stated cadence and trended.
- The ratio of categories to distinct retention outcomes, used to identify consolidation candidates while recording the reasons for distinctions that remain.
- The proportion of categories carrying a definition with exclusions and a worked example.
- An alias register showing retired names mapped to current categories, with the date each was retired.
- The classification method recorded for every category, and the proportion of content classified by container rather than by user choice.
- At least one documented category retirement, with the approval and the evidence that nothing was lost.

Classification agreement provides evidence of whether a category model can be applied consistently. A sample reviewed independently by a second qualified person can identify disagreements that a policy review would not reveal. Chapter 22 treats this as a sampling test, with its method, scope, and results retained in Appendix G rather than added to the standing metrics library.

## Business Value

A well-designed category model pays back in places that are not obvious from the schedule itself.

It can reduce over-retention caused by ambiguous categories. Clearer distinctions can reduce the tendency to choose the longest period when the right category is uncertain. The benefit follows when approved treatment is applied and eligible information is disposed of; redesign alone does not reduce stored volume.

It can reduce downstream implementation work. Removing unnecessary distinctions can simplify configuration, cross-repository mapping, and disposition. A model of roughly one hundred categories can require less implementation and maintenance effort than one with several hundred, depending on the variations and controls each must support.

It provides a stronger basis for automation. Container rules, ingest rules, and pattern recognition depend on clear, necessary distinctions and the information available to apply them. A lower

category count does not itself establish better performance. Validation shows which distinctions the chosen method supports within its intended scope.

It can reduce the cost of change. Removing unnecessary categories can mean fewer definitions to maintain and fewer mappings to revisit when requirements change. Preserve distinctions that serve a documented purpose rather than making a lower count the objective.

## Design for the Applier

Schedule design tends to reward addition. Each new distinction appears defensible on its own, even as the accumulated model becomes harder to apply.

Each category creates a distinction that people or systems must apply. The design should account for the information available at that point and the cost of inconsistent classification.

Use clear, necessary distinctions and assign classification to a method or person capable of applying them reliably.

*Next: global retention in practice. Harmonization, localization, data residency, and how to express jurisdictional variation without maintaining a separate schedule for every region.*

# Chapter 10: Global Retention in Practice

Chapters 6 and 7 established the structural requirements for jurisdictional rules. Chapter 6 described the tension between accuracy and usability, while Chapter 7 treated jurisdiction as part of a rule's scope rather than as a label. This chapter addresses the decision that remains: what rule should apply when jurisdictions disagree?

Many multinational programs have not answered that question explicitly and tend to arrive at one of two models.

The first is a single global schedule developed and validated at headquarters but incomplete for some countries, leading local teams to maintain separate lists. The second is a schedule for each country, usually the output of a well-run legal survey, where many separate documents exist and no one can readily explain how they differ or which one an acquired entity should follow.

Both are responses to the same unresolved question. Harmonization is usually treated as a retention decision, a matter of settling on the right period. It is a decision-rights question. What is the default, who may deviate from it, on what evidence, and who approves.

## The Longest-Period Trap

When a program does attempt harmonization, the usual method is to take the longest period across all jurisdictions and apply it everywhere. It is easy to explain, easy to defend in a meeting, and appears to be the conservative choice.

It is not necessarily conservative. It converts a local minimum into a broader retention requirement, including in jurisdictions where that period may not be necessary.

A retention requirement in one country establishes a floor for that country. It says nothing about what any other country wants. Applying it globally imports over-retention into every jurisdiction that had no such requirement, with the storage, discovery, and exposure costs described in Part I following it in.

Harmonizing upward is not neutral where other jurisdictions impose deletion rights, minimization duties, or retention caps. The program exchanges a visible risk of retaining too little for a less visible risk of retaining too long. Without a documented decision, that trade is difficult to defend.

The longest period may be the right answer for a specific category and group of jurisdictions when selected deliberately. It is a poor default because it substitutes a general assumption for a documented decision.

The same default can extend a permanent-retention requirement beyond its intended scope. A requirement that applies in one jurisdiction should not automatically become the rule everywhere.

Chapter 7 explains how to document a permanent designation and the custody arrangements it requires.

## Which Jurisdiction Applies Is Itself a Decision

Before a program can decide what a rule should be, it has to decide which jurisdiction's rule is in play, and this is treated as self-evident far more often than it should be.

A single record can plausibly sit under several. An employment file might follow the employing legal entity, the country where the employee works, the country where the employee resides, or the country where the record is stored. A customer transaction record might follow the contracting entity, the customer's location, or the governing law of the agreement. A shipment document might follow the country of origin, the country of destination, or the country of the carrier.

These questions are often manageable once they are raised, but many programs have not addressed them explicitly. The determination may be made informally and differently by different people. It becomes visible only when two parts of the organization apply different periods to the same kind of record and neither can explain why.

Writing down the jurisdiction determination rule for each group of categories is a relatively low-cost step that can resolve significant ambiguity. Employment records may follow the employing entity. Customer records may follow the contracting entity. Regulated operational records may follow the country of the activity. Whatever the answers, they need to be documented decisions rather than assumptions because every downstream rule depends on them.

## Three Layers, One Schedule

The structural answer to jurisdictional variation is not one schedule or many. It is one schedule with three layers, each with a different owner and a different standard of evidence.

**The global default.** The rule that governs unless something displaces it. It is set centrally from the consolidated category model built in Chapter 9 and applies to every jurisdiction that has not established a variation. In a well-designed model, many categories may remain at this layer because no material jurisdictional variation applies.

**The jurisdictional variation.** A modeled deviation tied to a citation, covering one jurisdiction or a group of them and related to the global default rather than copied from it. It carries an owner, a source, and a review date.

**The local rule exception.** A time-bound departure granted for a specific operational reason, with a rationale, an approver, and a expiry and any earlier review date. It remains distinct from a jurisdictional variation required by an applicable obligation. Expiry prompts review of the approval and any continuing exposure rather than silently ending the underlying work.

The discipline that makes three layers workable is a threshold. Not every difference deserves a variation.

Create a modeled variation when a local requirement changes the applicable treatment. A shared rule can cover different local minimum periods only after confirming that it satisfies all applicable requirements. For example, a seven-year default may also satisfy a six-year minimum, but that conclusion depends on the absence of a conflicting limit or deletion requirement. Record the local authority and analysis even where no separate operational variation is needed.

Setting the threshold before the analysis begins helps prevent the global model from accumulating every local distinction as a separate variation.

## Residency Is a Different Axis

Programs routinely conflate two constraints that behave differently. How long information may be kept is a retention question. Where it must live, and whether it may cross a border, is a residency question.

A residency requirement does not change the retention period. It changes where the control has to operate and which platform has to enforce it. Treating residency as a retention variation causes categories to multiply by hosting location, recreating the same fragmentation in another form.

Residency belongs in the repository register from Chapter 8 because it is a property of where information sits rather than what the information is. The schedule states the rule. The register identifies where the rule must be enforced and by which platform.

## Why Global Programs Fork

The drift toward separate schedules is driven by structural forces rather than by poor judgment.

- Legal surveys produce country reports, and a country report looks like a country schedule. Thirty memos become thirty documents because that is the shape the input arrived in.
- Deviation has no defined path. A local team that finds the global schedule wrong for its jurisdiction has no way to raise the issue and obtain a decision, so it keeps a separate list instead. The resulting list may become operationally authoritative while remaining invisible to the central program.
- Local requirements are expensive to verify and they go stale. Nobody wants to own a claim that requires annual legal spend in thirty countries to keep true.
- Local entities and employee representation bodies may hold authority that headquarters programs do not fully account for. A schedule that cannot accommodate a legitimate local obligation is likely to be worked around rather than followed.
- The global schedule is often validated by the counsel who commissioned it. Headquarters validation alone is not sufficient, and the gap may surface during an acquisition or audit rather than during a scheduled review.

The recurring problem is not a lack of legal research. It is the absence of a defined route from local requirements to documented decisions.

## What Works in Practice

A freight forwarding and contract logistics company operating in several dozen countries had followed the conventional approach. Outside counsel had been engaged in significant markets, and the resulting country memoranda had been reviewed and converted into separate schedules.

What nobody could answer was how they differed.

The question that forced the issue was an acquisition. Several new countries and a substantial employee population entered the organization, and the integration team asked which schedule the acquired entities should follow. There was no basis for answering because there was no model. The organization had many parallel documents with no stated relationship to one another.

The rebuild started with a single global default based on a consolidated model of slightly more than one hundred categories. The team then set a deviation threshold in writing before the analysis began: a local requirement would create a variation only if it changed the trigger, changed the period enough to alter behavior, or imposed a conflicting obligation.

Applying that threshold across categories and countries, only a small fraction of the combinations produced a materially different outcome. Many local requirements were satisfied by the global default after local validation confirmed that no conflicting requirement required separate treatment. Because multiple jurisdictions shared the same deviation, the differences that survived the threshold collapsed into a more manageable set of modeled variations, many of which applied to more than one jurisdiction.

The team also documented the jurisdiction determination rules over several weeks, resolving questions that had remained open for years. Employment records followed the employing entity. Customer and commercial records followed the contracting entity. Customs and transport records followed the country of the movement.

The change that mattered most was procedural. Local counsel gained a defined route to propose a variation, a small standing forum approved or declined it against the threshold, and every approved variation carried a citation, an owner, and a review date. Local deviations moved from informal practice into a documented review and approval process.

The separate country schedules were retired explicitly and kept as references with stated sunset dates rather than left in circulation. When the next acquisition occurred, the team could onboard the new entities by extending the existing model rather than reconstructing the approach.

## Implementation Steps

**Write the jurisdiction determination rules first.** For each group of categories, state which jurisdiction governs and why. Do this before touching a single retention period, because every subsequent analysis assumes an answer to it, and unstated assumptions differ between people.

**Set the deviation criteria before the analysis.** Identify changes in trigger, period, disposition action, or other applicable requirements that require separate treatment. Use materiality to organize the model and prioritize work, not to disregard a binding requirement. Retain the supporting local analysis even where the global rule satisfies it.

**Establish the global default from the category model, not from the longest period.** Where the longest period is the right answer for a category, record that it was chosen and why. Recording the reasoning makes the decision reviewable and distinguishes a deliberate choice from an unexamined default.

**Model variations as related rules.** Each carries a citation, the jurisdictions it covers, an owner, and a review date. Variations shared across jurisdictions are defined once and related to many, which is what keeps the count proportional to the differences rather than to the country list.

**Define decision rights for deviation.** State who may propose a variation, what evidence is required, who approves it, and how quickly. Without a defined route from local counsel to a documented decision, local teams are likely to maintain separate guidance regardless of what the global policy says.

**Keep residency out of the schedule.** Record hosting and jurisdiction in the repository register from Chapter 8. Building residency into categories multiplies the schedule by geography and produces the fork you are trying to avoid.

**Retire the country schedules explicitly.** Mark each as superseded, state a sunset date, and confirm removal from the places local teams actually look. Chapter 6 made the point that a program cannot state which rule governed unless it can state which document did. That applies with more force when many country schedules remain in circulation.

## What Good Looks Like

A global program that has resolved this looks structurally simpler than one that has not, and the simplicity is what makes it maintainable.

One schedule exists, with a global default and a bounded set of modeled variations, each traceable to a citation and an owner. The jurisdiction determination rules are written down and referenced when disputes arise. Local teams have a route to propose a deviation and use it, which means the number of private schedules trends toward zero and the program can see where its pressure points are. Residency is recorded against repositories rather than categories. Variations carry review dates and are re-verified on a cadence rather than assumed to remain true. Adding a country, whether through expansion or acquisition, is a mapping exercise against an existing model.

The clearest indicator is what happens when a requirement changes in one jurisdiction. In a fragmented program, the organization may not be able to determine the full impact. In a modeled program, the citation impact report described in Chapter 7 identifies the affected countries, categories, and owners.

## What Proves It

The evidence is the model itself and the discipline that keeps it current.

- Documented jurisdiction determination rules for every group of categories.
- A written deviation threshold, with the date it was set and by whom.
- A variation register showing each modeled deviation with its citation, the jurisdictions it covers, its owner, its last verification date, and its next review date.
- The ratio of category and jurisdiction combinations reviewed to variations modeled, which shows how much of the difference was material and how much was noise.
- A record of retired country schedules with sunset dates, and confirmation they are no longer in circulation.
- The count of proposed variations, approved and declined, which demonstrates that the deviation route is being used rather than merely published.

That last measure helps show whether the deviation route is being used. Read approved and declined requests alongside the scope reviewed and the reasons recorded. An absence of declined requests does not, by itself, establish either a weak process or successful governance.

## Business Value

A modeled global schedule makes expansion and acquisition more predictable.

Onboarding a new country can reuse the existing rule and category model while validating local requirements and adding the variations needed. It reduces the need to rebuild the schedule from the beginning; it does not replace local legal analysis. For an acquisitive organization, that reuse can improve integration planning and reduce repeated work.

**It can reduce recurring legal spend.** Verifying a bounded set of modeled variations on a review cadence can cost less than maintaining a schedule for every country or reconstructing the organization's position each time a question arises.

It can reduce unnecessary retention caused by a blanket longest-period default and make conflicting requirements visible before they are built into system controls.

And it converts local variation from an invisible risk into a managed one. Private lists do not disappear because a global policy says they should. They disappear when there is a better route, which is the same principle that governs every other control in this book.

## A Default and a Route for Difference

Global retention programs often respond to jurisdictional complexity by producing more schedules. Thirty countries may produce thirty analyses and thirty documents. The legal analysis may be sound, but the resulting collection is difficult to compare and maintain.

The complexity is real. The set of material variations may be smaller than it first appears, and many local differences may not change the operational result.

A global schedule is not an averaging exercise. It is a default, a threshold for departing from it, and a route by which departures are decided on the record. Together, those elements are more maintainable than a separate schedule for every country.

## Closing Part II

Part II established the structural foundation for execution: maintained rules, usable categories, a repository map, and a controlled method for jurisdictional variation. Part III follows how those inputs become operating controls and how the organization verifies the result.

# Part III: Execution Across the Environment

# Chapter 11: Making Governance Rules Operational Across Systems

A retention rule is approved in March. It carries a period, a trigger, a citation, the jurisdictions it covers, an effective date, a disposition action, and the name of the person who approved it. It is correct, current, and available on demand.

In November, someone asks a narrower question. Is that rule operating in the four repositories where the category actually lives? Most programs cannot answer without making phone calls. The schedule is authoritative about what should happen and silent about what is configured, by whom, and whether it has run.

The starting inputs are now in place. For each category and repository, the organization needs to decide how the rule will operate, record the decision, and keep it current as requirements and platforms change.

The step is usually treated as a translation exercise. Hand the approved schedule to the teams who administer the systems and ask them to configure it. That framing causes most of the trouble that follows. Each pairing of a rule and a repository is a decision: how the rule will be enforced, what the platform can and cannot do, who owns the result, and what happens when the honest answer is that nothing can be enforced at all.

## What Changes When a Rule Is Configured

An approved retention rule may contain eight discrete elements. A platform may support only some of them directly.

A content platform may support a period, a starting date, and a label or container that stands in for the category. Other elements may need to remain in the rule register or be implemented through separate controls. A citation may have no configuration field. Jurisdiction may require a separate site or label. Prior rule versions may not be retained in the platform. The available disposition actions may be narrower than those in the schedule, and exception handling may provide only a hold flag without the scope or review information the program needs.

The trigger is often where the translation breaks down. A missing trigger may be silently replaced with creation date during configuration, and even a well-specified trigger can meet the same fate when the platform cannot support the required business event. The rule says the clock starts at contract termination, employee separation, or the close of an audit period. The platform can calculate from the date an item was created or last modified, and nothing else. Someone with administrative rights resolves the difference in a few seconds, and rarely writes it down.

Platform limits do not remove the need to document what was lost or changed during configuration. A rule that reads seven years from contract termination and executes as seven years from upload is

a different rule. Unless the difference is recorded and reviewed, the program may report the approved rule while operating the substitute.

## Four Answers, Not Two

Programs often sort systems into automated and not automated. That distinction does not show how a rule operates or what happens where no control exists.

- **Native enforcement.** The platform holds the rule and executes it: the calculation, the action at the end of the period, and the suspension when a hold applies.
- **Integrated enforcement.** The platform holds the content and something outside it supplies what is missing, usually the trigger date from the system of record or the instruction to dispose. The rule operates across a connection that can break quietly.
- **Manual control.** A documented procedure on a defined cadence, with a named reviewer and a retained attestation. Manual controls can be effective when they are documented, consistently performed, and supported by evidence. Chapter 22 develops how those controls are tested.
- **Not enforced.** No mechanism exists today. The pairing still carries an owner and either a target date or a documented risk acceptance.

The fourth answer makes gaps visible. An honest no about platform capability creates a known gap instead of an assumed control. The unit of assessment is the pairing: one record category in one repository, linked to its applicable rule. A repository may enforce one category correctly and have no mechanism for another because the triggers or disposition actions differ. Assigning one enforcement mode to an entire platform hides those differences.

Where a pairing contains populations subject to different approved treatments, identify or link to each treatment's scope, rule version, and implementation evidence. One verified treatment does not establish coverage for every population in the pairing.

Assess capability by lifecycle action, not only by platform. Can the repository retain information under the rule, preserve the required population when a hold applies, carry out the approved disposition action when permitted, and produce evidence? A platform may support retention without supporting the evidence the program needs. Record the capability and any additional control required for each action.

## Why the Gap Has No Owner

The gap between an approved rule and a running configuration often persists because responsibility for the work is unclear. Technical difficulty can add to the problem, but it does not explain who should resolve it.

- The schedule is written for people and platforms need parameters, and translation between the two belongs to no one. Policy owners do not hold administrative access, administrators are not accountable for the rule, and the work is delegated by access rather

than by accountability. Whoever holds the rights in the console makes the judgment calls, usually while doing something else, and those calls become the operative policy.

• Platform capability is assessed once, at purchase, from a feature list. What a product can do, what the organization licensed, and what is enabled in its tenant are three different questions. Only the third governs behavior.

• Configured is self-reported and undefined. On a project plan it is a checkbox, and it rarely distinguishes a rule that has been loaded from one that has been enabled or one that has run against real content and produced a result.

• Configuration decays. Upgrades change defaults, migrations drop settings, and administrators move on. A control that worked at go-live is not evidence of a control that works now.

• Rule changes do not consistently reach system configurations. An approved change needs a defined route to every affected pairing, with an effective date and verification of implementation.

The result is predictable. The schedule is maintained and the configurations are not. The two come to describe different policies, and the one running in the environment is the one that matters.

## What Works in Practice

A regulated electric and gas utility operating in several states, with roughly nine thousand employees, had completed the structural work. Its category model had been consolidated to ninety-four categories, rules were held as structured objects rather than rows, and its repository register listed just over four hundred entries with named owners. Implementation was then scoped as a technology rollout. The schedule was published to the teams that administered the major platforms, with a request to configure it and a completion date.

Nine months later, four of the eleven largest platforms had reported completion. A spot check found that in two of them every rule calculated from the date an item was created, because neither platform had a field for the business event the rule specified. The categories affected included customer service records and vendor contracts. The work had been done as requested, and the result was a set of rules that were live, documented as complete, and wrong.

The team changed the unit of work. Instead of assigning the schedule to platforms, they assigned pairings to owners. A pairing is one record category in one repository, and it exists only where the register already showed that category present. The full combination would have produced tens of thousands of cells, nearly all of them empty, so scoping came first. Twenty-six categories selected on regulatory weight, litigation history, and sensitivity produced three hundred forty pairings to assess against the four lifecycle questions.

The distribution was more useful than any single finding. One hundred thirty-nine pairings could be enforced natively, forty-one required an integration, and ninety-two were candidates for a documented manual control. Sixty-eight had no available mechanism at all, and twenty-three of

those sat in systems already on the decommissioning list, which moved them into the legacy remediation work in Chapter 14 rather than a configuration backlog.

The trigger reconciliation produced the sharpest correction. Of the one hundred eighty pairings the platforms could execute natively or through an integration, sixty-one required a date the platform could not calculate. Twenty-two were resolved by supplying the date from the system of record. The remaining differences were documented for a decision on a proxy date, with the departure from the business event stated. Recording that choice alone did not establish verified coverage; the alternative needed authorization and verification against the approved treatment.

Two governance changes outlasted the configuration work. The first was recording the control owner and the configuring party as separate names, which ended a recurring pattern in which a control owner believed a change had been made that had never been requested. The second was a propagation path. An approved rule change now generates a pairing-level implementation assessment and a change request against every affected pairing, closed only on verification.

The measure they adopted was configuration lag. The most recent schedule change before the rebuild had reached three of the nine repositories it affected, nineteen months after its effective date. A year into the new process, the median was thirty-eight days and the longest open item four months, with an owner and a date attached.

## The System-to-Policy Implementation Matrix

The artifact that holds this work is a matrix with one row per pairing. It builds on the repository register and uses the structured rule sets published from the retention schedule.

A pairing remains the parent implementation unit. Where populations have different treatments, link their respective versions, modes, responsible roles, and verification history. Shared controls and evidence can be referenced without duplicating volume. Appendix F carries the corresponding scoped field guide.

*Figure 11.1. The system-to-policy implementation matrix.*

| Field | What it answers |
|---|---|
| Repository | Where the information sits, taken from the register rather than created here |
| Record category | Which rule applies, expressed in the category model’s own vocabulary |
| Rule version and effective date | The applicable rule version and effective date for each scoped treatment in the pairing |
| Configured trigger | The event or date field used, or the documented procedure, and how it matches the applicable approved trigger or authorized alternative |

| | |
|---|---|
| Enforcement mode | Native, integrated, manual, or not enforced |
| Retain, preserve, dispose, evidence | Which lifecycle actions the repository can perform for this category and where additional controls are required |
| Control owner | Who is accountable for the rule operating, as a person rather than a team |
| Configuring party | Who holds the access to make the change, which is usually someone else |
| Last verified date and method | When and how the configuration or documented procedure was verified against the applicable approved treatment, with supporting evidence. Link operating-test results separately. |
| **Gap response** | Responsible owner, original date, action or review date, and open-item link for every unresolved condition; acceptance additionally records rationale, authorized approver, expiry, and any earlier review |

Three properties keep the matrix usable rather than merely large.

**It is scoped, and the scope is stated.** Record which categories and repositories are in scope and why. Otherwise a missing row reads as an absence of risk rather than an absence of analysis.

**It creates one controlled point of translation.** Where a platform can use a published rule set directly, it should. Where it cannot, the matrix becomes the controlled translation point, reviewed by the rule owner, and every configuration derives from that row rather than from someone's interpretation of the schedule.

**It has a verification date and method.** A row last verified two years ago may no longer describe the current control. Record what each verification established. Configuration or procedure verification keeps the implementation record current; operating tests establish what the control did within their examined scope.

## Implementation Steps

**Scope by exposure before building anything.** Start from the categories with regulatory weight, litigation history, and sensitivity, and trace them against the register to find the pairings that exist. Start with the information that carries the greatest exposure. A focused matrix that gets implemented is more useful than an enterprise-wide exercise that never gets completed.

**Assess capability by action and test it in your own tenant.** For each pairing, answer whether the repository can retain, preserve information under hold, dispose, and produce evidence. Vendor documentation describes the product. Testing establishes what the organization's licensed and configured environment actually does.

**Record not enforced as a decision rather than a blank, then route it.** Give every unenforced pairing an owner and either a target date or a risk acceptance with rationale, approver, and expiry and any

earlier review date. Link it to the open-items view. Some gaps belong in a technology roadmap, some in the legacy remediation work in Chapter 14, and some in a contract conversation. An acceptance records a response; it does not demonstrate implementation or remove the exposure.

**Name the control owner and the configuring party separately.** Accountability for the rule and access to the console rarely sit with the same person. Recording both closes the space where each assumes the other has acted.

**Reconcile the trigger explicitly for every applicable treatment.** Where the platform cannot calculate from the business event, an integration may supply the date or an authorized alternative may use a proxy. State the difference, obtain approval for the alternative, and verify its implementation. Accepting the risk of an unimplemented requirement does not make that pairing covered.

**Build the propagation path before the next rule change.** Following the pre-approval assessment described in Chapter 19, an approved change should generate a pairing-level implementation assessment, a change request against each affected pairing, and a verification step that closes it. Include citation impacts where legal authorities change. Chapter 19 in Part IV addresses change control; the implementation matrix records how approved changes reach the environment.

## What Good Looks Like

In an organization that has done this work, the question that opened this chapter takes minutes. For any category, the program can state where it lives, how the rule is enforced in each place, who owns the result, when it was last verified, and where it is not enforced at all.

Enforcement mode is a recorded property of each pairing rather than an impression. Manual controls exist deliberately, on a cadence, with named reviewers, rather than as the unspoken remainder. Unenforced pairings are visible, sized, and assigned, and the program can say what closing them would cost.

When a rule changes, it reaches the systems through a defined route and the time it takes is measured rather than assumed. When a platform is upgraded or a tenant is migrated, the affected rows are re-verified rather than presumed intact.

## What Proves It

The evidence connects the approved rule to system behavior and the manual controls used where the platform cannot enforce it.

- The matrix itself, current, with a stated scope so its silences can be interpreted.
- For each pairing, the enforcement mode, the control owner, the configuring party, and the date and method of last verification.
- Exported configuration records for pairings marked native or integrated, produced from the platform rather than asserted in a status report.

- A trigger reconciliation record listing every pairing where the configured trigger differs from the approved trigger, with the rationale and the approver.
- A register of unenforced pairings, each with an owner and either a target date or a risk acceptance carrying rationale, approver, expiry and any earlier review date, and an open-item reference.
- Configuration lag, measured as the elapsed time from the effective date of an approved rule change to verified configuration in each affected repository.

Configuration lag is a supporting measure in Appendix C. Report completed-verification intervals as a distribution, with the completed population stated, and show the age and significance of items still awaiting verification separately. The longest open interval identifies delay, not exposure size by itself. Retain effective and verification dates, identify early completion and changes not yet due, and explain the display convention and any method change. Do not pool unfinished ages with completed durations or combine this clock with decision lag.

## Business Value

The matrix helps answer questions about what actually happened to information. Regulators, auditors, and other reviewers may need more than the policy: which records were affected, what the system did, and under what authority. A maintained record reduces reconstruction under time pressure.

It converts an open-ended technology argument into a costed one. A request to fund a governance capability is difficult to evaluate in the abstract. A statement that ninety-two pairings depend on manual review, that a specific integration would automate a named share of them, and that the rest would still require attestation is a business case with numbers attached.

It also protects the investment already made. A retention schedule that never reaches the systems remains a reference document, regardless of what the organization spent to develop it.

And it keeps policy from being set at the console. Every unrecorded configuration choice is a small, unapproved amendment to the schedule, made by someone who was not asked to make it and would not claim the authority. Recording the choice does not always change the outcome. It puts the decision back with the people accountable for it.

## Where Policy Becomes Behavior

Two organizations can hold the same schedule and run the same platforms while having very different levels of operational control. The difference is whether the program can identify where each rule operates, how it is enforced, who owns it, and when it was last verified.

Treat the move from rule to execution as a governed decision with an owner and a record. Document platform limits and approved alternatives, including manual controls. Governance becomes operational when the approved requirement is carried out through a defined control and the program can verify that it worked.

Next: Chapter 12 examines how to produce consistent outcomes across repositories, including the additional copies that individual system configurations do not address.

# Chapter 12: Consistency Across Repositories and Platforms

A retention rule can be correctly configured in every system that holds the category and still fail.

Chapter 11 treated each record category and repository pairing as a decision with an owner and a record. That is the unit for implementing a control, but it is not sufficient for judging the outcome across the information environment. The same engagement file, case record, or contract may appear in a system of record, a managed repository, several mailboxes, a collaboration workspace, and a reporting extract. Each location may follow its own configured rule while the organization retains more information than intended.

This chapter is about the information environment rather than the individual pairing: why the same rule behaves differently depending on where it is applied, what varies between platforms, and how to produce a consistent policy outcome across systems that were never designed to operate the same way.

## Consistency Is Not Uniformity

The instinct is to make every system behave the same way. That is expensive and often unrealistic because platforms were built for different purposes and enforce rules through different mechanisms.

Consistency is better defined at the level of outcome. Authoritative instances should reach the approved end state within the required period. Convenience copies may follow shorter approved rules, provided that required information and context remain available and preservation obligations are met. Different platforms can use different mechanisms to achieve those outcomes.

That definition changes what counts as a problem. Two repositories using different mechanisms, one automated and the other reviewed quarterly, can be consistent if both meet the applicable outcome and timing requirements. Two using the same mechanism can be inconsistent if one calculates from the approved business event and the other from upload. The comparison must account for the role of the instance and the trigger applied.

Uniformity is a means, and often an unavailable one. Consistency of outcome is the requirement, and it is achievable in environments where uniformity is not.

## The Longest Copy Sets the Period

Copies create a simple governance problem: from a practical risk perspective, information remains retained as long as a copy remains accessible somewhere in the environment.

If a contract is destroyed from the document repository at seven years and a copy of the same contract sits in a shared drive with no rule at all, the organization cannot claim the information is gone at seven years. The copy remains accessible indefinitely, and the disposition record in the

repository describes only what happened in that repository. Inconsistency does not average out. If one copy remains indefinitely, the information effectively remains indefinitely.

Rule design can create a similar problem when organizations harmonize across jurisdictions by applying the longest period everywhere. Across the information environment, the result can be even less deliberate: copies simply outlive the systems or controls that governed them.

The same issue appears in preservation. A hold applied to the authoritative instance does not automatically preserve relevant information in other locations. Chapter 15 addresses how to identify and preserve the required population, including convenience copies that fall within scope.

The repository register identifies where information lives. The extract and copy map extends that view by showing what is extracted, on what schedule, and where additional copies are created.

## What the Rule Attaches To

Platforms are often compared on whether they can enforce retention. A more useful comparison is what the rule attaches to, because that property helps identify likely gaps. The table describes common patterns that need to be checked in the organization's own environment.

*Figure 12.1. How rules attach by platform class, and the resulting failure.*

| Platform class | Rule attaches to | Characteristic failure | What consistency requires |
|---|---|---|---|
| Systems of record | A row or transaction | Deletion breaks referential integrity, so retention becomes archival and disposition never happens | An approved treatment for dependent data, including controlled archival where permitted; archival is continued retention, not destruction |
| Managed content repositories | An item, through a label or class | Works well, but holds a smaller share of the information environment than many programs assume | Placement, so that content which belongs here actually arrives here |
| Collaboration and file sync | A container, site, or workspace | Content that moves between containers silently changes rules | Container-level defaults and a stable answer to what each container is for |
| Communication platforms | A user or mailbox | One record acquires as many retention periods as it has recipients | A named authoritative instance elsewhere, with communication copies governed as copies |

| | | | |
|---|---|---|---|
| Shared drives and endpoints | Often no item-level carrier | No rule carrier exists, so the content is governed only by whoever created it | A supported control, a governed destination, or authorized disposition and decommissioning |

Two of these deserve emphasis because they are the ones programs most often misread.

**In collaboration platforms, the container often determines the rule.** When content moves between containers, its retention treatment can change with it. Container-based classification depends on knowing which business process each container serves. That dependency is what makes this class governable, and it is why containers created ad hoc for a single project or client are among the hardest parts of the information environment to manage consistently.

**Communication platforms often attach rules to users or mailboxes.** Different treatment across recipients may be appropriate where the message is a convenience copy of an authoritative instance elsewhere. The program still needs to establish whether any mailbox contains unique information or context that must be retained.

This chapter addresses how rules attach to platforms. Chapter 16 addresses which transactions within chat, channels, recordings, and transcripts carry record value and how that information should be governed.

## Why Outcomes Drift Across Systems

Most inconsistency is not the product of bad decisions. It accumulates from reasonable ones taken separately.

- Platforms are adopted by different functions on different timelines, each configured by whoever owned that project. Existing platforms may never pass through the governance process applied to new systems.
- Rules are mapped to applications rather than to repositories. One application commonly holds several repositories with different retention behavior, and a rule recorded against the application describes none of them accurately.
- Copies are created for legitimate reasons but may not inherit their source's controls. A reporting extract, an attachment, or a locally saved file can leave a governed location without a process that records the new copy or applies the appropriate treatment.
- Content migrates between platforms during upgrades, consolidations, and reorganizations, and migrations are scoped to move content rather than to preserve its treatment. Retention settings are among the first things lost.
- Nobody owns the outcome across systems. Each pairing may have a control owner, but no one is accountable for whether the category as a whole is being treated consistently. That question falls between the individual owners rather than to any of them.

Each of these decisions may be reasonable on its own. Together they produce an information environment where the policy is uniform on paper and the outcomes are not.

## What Works in Practice

A global professional services firm of roughly forty-five thousand people had done the structural work and the configuration work. Engagement records carried a single rule, seven years from engagement close, natively enforced in the document management platform and verified in the implementation matrix.

A client then asked for written confirmation that its information had been returned or destroyed at the end of the contract. The firm could evidence the document management platform without difficulty. It could not say anything about the other places engagement content lived.

The team did something small and clarifying first. They sampled thirty closed engagements and traced the content rather than the rule, following it wherever it had gone. Engagement material appeared in an average of just under five locations per engagement. The effective retention applied across those locations ranged from ninety days in one communication platform to indefinite in personal drives and in an archive that had been created during a migration and given no rule at all.

The team rejected applying the seven-year rule everywhere. That would have multiplied the volume under formal retention and expanded the population to assess for preservation, while requiring configuration across platforms that could not all carry the rule.

The document management platform was named the authoritative instance for engagement content. Other instances were assessed as convenience copies and given a shorter rule: one hundred twenty days from last access, applied at the container level where the platform allowed it and by scheduled cleanup where it did not. The category rule stayed with the authoritative instance. Convenience copies remained subject to preservation requirements and checks for unique information.

For each platform class they wrote a short standard describing the required outcome rather than issuing one configuration for all of them. The system of record kept its rows and moved closed engagements to a controlled archival state because deletion broke reporting the firm relied on. That retained population needed its own approved treatment; archival did not mean the information had been destroyed.

They also changed the default save location inside the tools people already used to the document management platform, reducing the decisions required to file engagement content correctly. New engagement folders created in collaboration workspaces fell by roughly seventy percent over two quarters.

They also kept a workaround register, on the principle that a workaround points at something that was not working. The most common entry was saving to personal drives because access for newly formed client teams took around nine days to provision. Reducing that to same-day removed more

copies from the information environment than any policy change in the program, and it cost less than the configuration work it replaced.

A year later the firm could answer the client question across the information environment. The authoritative engagement content followed the seven-year rule, convenience copies followed the shorter rule, and retained system-of-record data had a documented archival treatment. The last-access trigger still required oversight because continued access could extend the life of a convenience copy.

## Govern Copies by Their Nature

Many programs apply the record's retention period to every copy. While conservative, that approach can dramatically expand the volume subject to formal retention and make consistent disposition much harder. It treats the proliferation of copies as a filing problem rather than a governance decision.

The alternative is to identify the authoritative instance for each significant category and govern duplicate instances as convenience copies with a short, defined life. Before applying that treatment, confirm that the copy does not contain unique information or context that must be retained. Its rule should include a reliable pointer to the authoritative instance and remain subject to applicable preservation requirements.

This makes three things easier at once. The volume under formal retention falls to something the program can defend and evidence. Hold scoping becomes tractable, because there is a defined authoritative instance and a small, aging population around it. And the outcome becomes consistent without requiring the platforms to become uniform, which is the point of the distinction this chapter opened with.

It also creates two populations with different disposition workflows. The authoritative instance carries category-level authorization, a validated action, and a record. Convenience copies can expire automatically under an approved rule without individual review, provided the control accounts for holds and other applicable restrictions. Chapter 13 develops the authorization and execution process.

It does require honesty about one thing. A convenience copy rule only holds if the authoritative instance is genuinely accessible. If people cannot reach it quickly they will keep local copies, and the short rule will either be circumvented or will destroy the only accessible version. The firm in the vignette found that out through its provisioning delay, which is the ordinary way organizations find it out.

The periods used in these examples are illustrative, not recommendations. Retention periods come from the approved schedule and the legal and business analysis behind it. A convenience copy period should be appropriate to its use and generally shorter than the authoritative record period. A

trigger based on activity, such as last access, also needs a defined approach to copies whose clocks are repeatedly reset.

## Implementation Steps

**Trace content rather than rules for a small sample of closed matters.** Twenty or thirty completed items in a high-exposure category can provide a starting diagnostic, adjusted for the population and available access. Record the selection and locations examined. The result describes that scope rather than every copy in the environment.

**Group repositories by what the rule attaches to.** Sort repositories into classes by attachment point rather than by vendor. Five classes cover most environments, and the class predicts the failure mode more reliably than the product name.

**Name the authoritative instance for each category that matters.** Record it in the implementation matrix and identify any separate populations with distinct requirements. That gives subsequent decisions about copies a documented basis.

**Write an outcome standard per class instead of one configuration for all.** State what reaching the end state looks like in a system of record, a managed repository, a collaboration platform, and a communication platform. Let each class meet it locally.

**Reduce unnecessary copies before configuring them.** Changing the default save location, retiring a duplicate repository, or shortening a provisioning queue can reduce unnecessary copies and the work required to govern them. Address the causes of duplication alongside the required controls.

**Keep a workaround register and treat entries as findings about the program.** Each workaround identifies a control or a service level that blocked legitimate work. Fixing the friction removes the copies. Policing the workaround relocates them.

## What Good Looks Like

In a program that has done this work, the answer to what happens to a category is one answer rather than a list of system-specific answers with different endings.

Every significant category has a named authoritative instance. Convenience copies carry approved shorter rules that account for preservation requirements. Platform classes have written outcome standards, so new systems can be assessed consistently. Migrations include retention treatment as a requirement and verify it afterward. A current extract and copy map shows where additional instances exist.

The clearest sign is what happens when a hold arrives or a client asks for return and destruction. The scope is knowable, the population is bounded, and the answer covers the information environment rather than only the one system the program can speak for.

## What Proves It

The evidence for consistency is comparative. A single system in good order proves nothing about consistency across the information environment.

- A content trace for a sample of completed items in each high-exposure category, showing every location the content reached.
- The authoritative instance named for each significant category, recorded where the implementation matrix can be read alongside it.
- Outcome standards by platform class, with the classes each repository belongs to recorded in the register.
- Convenience copy rules, with evidence that they are running rather than merely published.
- A current extract and copy map for the highest-exposure repositories, maintained rather than rebuilt for each review.
- Migration records showing retention treatment as a scoped requirement and a verified outcome.
- The workaround register, with entries closed by change rather than by instruction.

The measure worth carrying at chapter level is the effective retention spread: for a sampled category, the difference between the shortest and longest periods actually applied across locations. Read that spread alongside each instance's role, trigger, and approved end state. A short convenience copy period and a longer authoritative record period may be intentional; a copy with no limit or a repeatedly reset trigger may indicate a gap. Where an end point is unknown or indefinite, report it explicitly rather than forcing a numeric spread.

## Business Value

Reducing unnecessary copies before an obligation arises can reduce the population that must later be searched, preserved, and reviewed. The benefit depends on where information is held, the requirements that apply, and whether the remaining authoritative instances are complete and accessible.

It makes the scope of disposition evidence clear. A repository-specific disposition record can remain accurate while the organization's wider outcome is incomplete. The credibility problem arises when that record is presented as evidence that the information was removed from the entire environment.

It can reduce the work required for later controls. Named authoritative instances and appropriate rules for convenience copies can reduce duplicated configuration and repeated decisions when platforms change.

And it removes friction the business is already paying for. The provisioning delay, the duplicate repository, and the awkward default save location are costs the organization absorbs quietly.

Addressing them improves governance and daily work at once, which is a rarer combination than it should be.

## One Outcome, Many Mechanisms

The environment will not become uniform. New platforms arrive, teams adopt tools that suit their work, and content moves for reasons unrelated to governance. A program that defines success as uniformity is working against how organizations and technology environments actually operate.

What can be held steady is the approved outcome for each population. Authoritative instances follow the category rule, and convenience copies follow shorter approved treatment where appropriate. The mechanisms may differ, but the program needs to show that required information remains available and that copies do not persist without a reason.

That does not make the work easy. It requires the program to understand where information actually goes, not simply where policy says it should remain.

Next: Chapter 13 addresses disposition itself, including category-level authorization, checks before execution, and evidence that the approved action occurred.

# Chapter 13: From Policy to Action: The Disposition Problem

A program reports broad configuration coverage across the major platforms and a schedule that has been reviewed and approved. Yet eligible information continues to accumulate. The reporting records where retention settings exist without establishing whether the approved disposition action has occurred.

The configuration report may be accurate within that limited scope. Retention is configured and the rules calculate correctly, but eligible information has not been destroyed. Configuration coverage and verified lifecycle execution need to be reported separately.

Organizations are often better at retaining information than at disposing of it. Platform capability is only part of the explanation. Chapters 11 and 12 established how approved rules operate in repositories and how authoritative instances differ from convenience copies. What remains is execution: authorizing the action, carrying it out for eligible information, and showing afterward that it happened. This chapter focuses on destruction, while distinguishing it from transfer, anonymization, archival, and other approved outcomes.

## The Asymmetry That Stops Everything

Consider the position of the person asked to approve a destruction list.

If they approve and nothing goes wrong, nothing happens. No one notices. If they approve and a record turns out to have been needed, the decision has a name on it, a date, and a signature, and it will be examined. If they decline, the cost is storage, which appears in someone else's budget and is never traced back to them.

The organization faces exposure in both directions. Keeping too much creates cost, discovery burden, and breach scope. Keeping too little creates regulatory and evidentiary risk. For the individual asked to approve destruction, the consequences can feel very different. The cost of keeping information is diffuse; the decision to destroy it is attributed to a person.

This is why stronger policy language, training, and repeated reminders do not solve the problem. They ask an individual to accept personal risk in exchange for an organizational benefit. The predictable response is deferral. It avoids the immediate personal risk without requiring anyone to formally refuse.

A quarterly disposition review can reinforce that problem. A business owner receives a list they cannot evaluate individually and is asked to authorize destruction without a clear basis for doing so. If the review repeatedly produces deferral, the program needs to reconsider what question it asks and who can answer it.

The better approach is to move authorization earlier, to the point when the rule is approved. Authorize the disposition action once, at the category level, when the decision is about a class of information rather than a list of files with a deadline attached. That allows the organization to act consistently without seeking a new approval for each disposition cycle.

Once the category is authorized, the recurring question changes shape. It is no longer whether this information may be destroyed. It is whether anything in this eligible population is encumbered, which is a question with a factual answer and a real owner.

## Three Gates, Three Owners

Programs collapse disposition into a single approval, which is why one person ends up being asked several questions they are not equipped to answer. Three separate questions stand between eligibility and an approved disposition action, and each has a different owner and a different form of evidence.

*Figure 13.1. The three gates before disposition.*

| Gate | The question | Who answers | What it produces |
|---|---|---|---|
| Eligibility | Has the applicable finite period elapsed under its approved trigger and transition, or has another authorized lifecycle event occurred? | The system holding the rule | A computed population with the rule version and trigger date applied |
| Encumbrance | Is the population subject to a legal hold, open matter, inquiry, contractual restriction, or approved non-hold suspension? | Legal and the accountable suspension owner, using the relevant registers | A checked population and a record of information withheld, with the restriction and release criteria |
| Dependency | Does anything downstream rely on this information continuing to exist | The technology and business owners for the repository | A recorded assessment per pairing, reviewed at least annually and after relevant changes |

Separating them does two things. It gives each question to someone who can actually answer it, and it makes a delay attributable. When disposition stalls behind a single approval, nobody can say which question is unresolved. When it stalls at a gate, the gate has an owner and a date.

The dependency gate is easy to miss. Reports, integrations, analytics environments, and models may rely on the information. Deleting rows can break a downstream process. Establish dependencies for each category and repository pairing, record them with the implementation matrix, and review

them at least annually and when systems or uses change. This avoids rebuilding the assessment every cycle without treating it as permanently settled.

A real dependency requires an approved response. Options can include retiring the downstream use, retaining a derivative where appropriate, or moving information to a controlled archival state. Any continuing restriction or approved non-hold suspension must also remain visible to the encumbrance check. Archival continues retention and must be consistent with the approved rule; it does not satisfy a requirement to destroy information simply because the business owner prefers it.

Match the process to the population. Authoritative instances need the full disposition process. Convenience copies can expire automatically under approved rules without individual review, provided that preservation and other applicable restrictions interrupt that expiry. The distinction changes the workflow, not the obligation to preserve information within scope.

## Deleted Is a State With a Duration

The word delete implies an event. In most platforms it describes a state that persists for a while and then, usually, resolves.

A delete operation may move content to a recoverable area for thirty or ninety days. It may write a tombstone while the underlying object remains. It may remove the current version while prior versions stay, particularly where versioning was enabled during a migration and never revisited. It may succeed in the primary system while replicas in other regions catch up on their own schedule. In each case the platform reports success, and a disposition record written at that moment describes an outcome that has not yet occurred.

For each repository that destroys information, define the terminal state: the point at which content cannot be restored through routine administrative action. Record the residual period between the deletion instruction and that state, including the treatment of versions and replicas. Keep these details with the repository's lifecycle capabilities. A destruction certificate should reference the terminal state reached; a record of an instruction alone should say that destruction is pending.

Backups need a stated position. A short, defined backup rotation serves a different purpose from an archive, but the program must establish how long information remains recoverable and what happens if a backup is restored. The approved treatment should account for applicable requirements, match actual backup behavior, and distinguish temporary backup persistence from destruction in active systems.

This also matters for the absence testing developed in Chapter 22, which checks for information that should no longer remain within the defined scope. Without an agreed terminal state, recoverable content may be found with no clear basis for determining whether the result is a finding.

## Why Disposition Stalls

Beyond the asymmetry, four structural causes appear repeatedly.

- **Released holds never re-enter the queue.** When a matter closes, the release is recorded and the information returns to normal treatment in principle. In practice, nothing re-evaluates it against the schedule, and populations released years ago sit indefinitely past their periods. This can be a substantial source of over-retention, and it is invisible because the hold register shows the release as complete. Closing a hold is not enough. Released information also needs to return automatically to the eligibility calculation so normal retention treatment can resume.
- **Nobody owns the backlog.** Control owners own their configurations. The question of how much eligible information remains undisposed across the information environment belongs to no one, so it is never sized, never reported, and never funded.
- The first cycle is the hardest and it is scoped as though it were typical. The initial population includes everything that accumulated before the program existed. Teams size the effort from that first run, conclude that disposition is unaffordable, and defer it, which guarantees the next attempt starts from a larger number.
- No evidence means no confidence. A program that has never produced a clean disposition record has nothing to point to when someone asks whether this is safe. Each cycle is argued from first principles, and the argument is exhausting enough that skipping it is easier.

These compound. The backlog grows because nothing is disposed, and nothing is disposed partly because the backlog has become large enough to be frightening.

## What Works in Practice

A telecommunications carrier with roughly eighteen thousand employees and a large consumer subscriber base had done the structural and configuration work. Categories were consolidated, rules were structured, and the implementation matrix recorded native retention capability across the major platforms without separately verifying disposition execution. Storage in governed repositories was growing at the same rate as before the program started.

The cause was visible in the process records. A quarterly disposition review required a business owner to approve each destruction list. Over three years, forty-one lists had been submitted and nine had been approved. The remaining thirty-two were deferred, most with no reason recorded. Nothing had been refused. In practice, repeated deferral had the same effect as refusal, without requiring anyone to make that decision explicitly.

The team moved authorization to the category. Each category's disposition action was approved once by counsel and the accountable business owner at the time the rule was approved, in the same conversation that set the period. The quarterly step stopped asking whether destruction was permitted and started asking whether anything in the eligible population was encumbered.

They then separated the gates. Eligibility was computed by the schedule system. Encumbrance was answered by legal against the hold register, which took days rather than the weeks the previous

review had absorbed, because it was a factual check rather than a judgment about value. Dependency was assessed once for each of sixty-two category and repository pairings and recorded.

The dependency pass produced the most useful finding. Eleven pairings had downstream consumers. Seven were reports that had not been opened in over a year, and retiring them removed the blocker. Three moved to a controlled archival state because deletion broke reconciliation the finance team relied on. Those rows remained retained under a separately approved treatment rather than being counted as destroyed. One retained a derivative with subscriber identifiers removed, which served the analytics use without holding the underlying records.

Two discoveries during validation changed how the program described its own history. In two major platforms, a delete moved content to a recoverable area for ninety-three days. In the object store, versioning had been switched on during a migration years earlier and never reviewed, so prior versions of deleted objects remained available indefinitely. The disposition records for the nine approved lists described destruction that had not fully occurred. The team defined the terminal state for each platform, corrected the versioning configuration, and rewrote the certificate to reference the terminal state rather than the instruction.

The largest single population came from a reconciliation nobody had run before. Records released from legal hold between two and five years earlier had never been re-evaluated against the schedule. Roughly two and a half million subscriber records had been sitting past their periods since the matters that preserved them closed. Adding a step that returns released populations to the eligibility calculation required little additional implementation effort in this case and produced more disposition volume in one cycle than the previous three years combined.

Eighteen months in, the oldest eligible item in the top categories had fallen from over eleven years to fourteen months, and the storage curve for governed repositories flattened for the first time.

## Implementation Steps

**Authorize the disposition action when the rule is approved, not when the list appears.** Category-level authorization from counsel and the accountable owner should be recorded with the rule and carry the same version history. This reduces the need to reconsider the same authorization each cycle.

**Separate the three gates and give each an owner.** Eligibility follows the applicable rule and transition. Legal and the accountable suspension owner check encumbrances against the relevant records. Technology and business owners assess dependency. Record which gate a delayed population is waiting on.

**Answer the dependency question per pairing and record it.** Store the assessment with the implementation matrix and review it at least annually and after relevant changes. The recurring cycle checks that the assessment is current and that any recorded conditions have been met.

**Define the end state for each disposition action.** For destruction, name the terminal state and residual period in each repository. For transfer, anonymization, or archival, record the approved outcome and verify that outcome without describing retained information as destroyed.

**Return released populations to the eligibility calculation.** Verify that no other encumbrance remains, then use the rule version, retention trigger, and any approved transition or fallback applicable to that population. Release does not restart the retention period unless the approved treatment requires it. Run a historical reconciliation to identify information left outside normal treatment after release. Chapter 19 develops rule transitions.

**Size and report the backlog, and scope the first cycle separately.** The initial run clears an accumulation that will not recur. Funding it as though it were the steady state is how disposition programs get cancelled in their first year.

## What Good Looks Like

In a program where disposition works, the recurring cycle is unremarkable. It runs on a schedule, it completes, and it produces records without anyone having to make a case for it.

Authorization sits with the rule rather than with the list, so no individual is asked to accept personal risk on behalf of a policy the organization already approved. Encumbrance checks take days. Dependency answers are on file and current. Released holds return to the queue on their own. When a population is not disposed, the reason is recorded and attached to a gate and an owner rather than to a general sense that the time is not right.

The clearest sign is the age of the oldest eligible item. In a program that is only configuring retention, that number grows every year. In a program that is disposing, it stays close to the length of the cycle.

## What Proves It

Disposition can produce some of the clearest evidence in the program because the action creates a defined, testable record.

- Category-level disposition authorization, dated, attributed, and versioned with the rule it belongs to.
- Encumbrance-clearance records for each cycle, showing the population checked against holds and approved non-hold suspensions, and what was withheld with its reason.
- Dependency assessments per category and repository, with the date of the most recent re-check.
- Disposition records naming the population, rule version, authority, action, date, responsible party, and verified end state. Destruction certificates identify the terminal state reached.
- The residual period for each repository, so a certificate can be read correctly.

- A release-to-requeue record showing that populations coming off hold re-entered the eligibility calculation.
- A list of unresolved disposition items for each cycle: information past its applicable retention period, unencumbered, and not yet disposed of, with the unresolved gate and owner. These are not automatically approved rule exceptions or control-test findings.
- Retain these records under a defined control evidence category in the schedule. Its approved rule should address how long the organization needs to demonstrate the control, rather than simply inheriting the period of the information disposed of. This category also holds evidence received from third parties in Chapter 17.

The measure worth carrying at chapter level is the disposition backlog: the volume of information past its retention period, unencumbered, and awaiting its approved disposition action, reported alongside the time the oldest item has remained overdue. Report held populations separately as encumbrance load, as described in Chapter 15. The backlog volume shows scale; overdue age shows whether execution is keeping pace.

## Business Value

Executed destruction directly reduces the volume the organization retains, affecting storage cost, discovery scope, privacy response effort, and breach impact. Other disposition actions may improve control without reducing volume. Reporting should distinguish those outcomes.

It converts the retention schedule into an operating process. Where eligible populations exist but no disposition has occurred, the program needs to explain the gap. Where no population is yet eligible, testing can establish whether the configured action will operate when required.

It reduces the cost of every future obligation. A hold applied to a category that disposes on schedule covers a bounded population. A hold applied to a category that has never disposed covers everything that ever existed, and the difference shows up in collection cost, review cost, and elapsed time.

And it removes a recurring negotiation. A quarterly review that argues the same case from first principles consumes senior attention every cycle and produces deferral most of the time. Authorizing once and checking encumbrance thereafter returns that attention to work that needs it.

## Authorize Disposition Before the Cycle

The instinct when disposition stalls is to escalate: more attention, firmer policy, a senior sponsor pressing for the lists to be signed. It rarely works, because the problem is not that people are unwilling. It is that the process asks the wrong person the wrong question at the worst possible moment, and then treats their reluctance as a governance failure rather than as a reasonable response.

Move authorization to the category and to the people responsible for the rule. Give the recurring cycle factual checks on eligibility, encumbrance, and dependency. Define the end state for each

action and retain evidence that it was reached.

The answer is not to ask people to be more willing to take the risk. It is to design a control that does not depend on individual courage in the first place.

Next: Chapter 14 applies these decisions to legacy and orphaned repositories, establishing an exit, an owner, and a timetable before migration becomes the default.

# Chapter 14: Legacy Systems Accumulate Risk

Legacy is a governance status, not a technical one.

A twenty-year-old system with a current owner, a known content profile, and an agreed retention treatment is not a legacy problem. It is an old system that is under control. A cloud tenant stood up three years ago and abandoned after a reorganization, still holding customer records, with no owner and no one able to say what is in it, is a legacy problem in every sense that matters.

The difference is not age or architecture alone. It is whether the organization has an owner, a known content profile, and a deliberate treatment for what the system holds. Repositories awaiting retirement need an exit decision as well as the configuration controls in Chapter 11. The disposition authority established in Chapter 13 makes several of those exits possible.

## Choose the Outcome Before Migration

When legacy systems finally reach an agenda, the default plan is to move the content. Lift the content into a modern platform, retire the old hardware, and report the systems as remediated.

Migration can appear easier to approve because it offers a visible technical result without first resolving every decision about content. The risk is that the project moves information before establishing what must be retained, what may be destroyed, and who can authorize those actions.

Migration changes the platform without necessarily resolving the governance problem. An ungoverned population can arrive intact, with greater accessibility and a new operating cost. The project needs an approved treatment for the content as well as a destination.

Migration is appropriate for some populations, including governed information that must be retained in a system the organization can no longer support. The exit should follow from the information requirements rather than from a default assumption that everything must move.

## Four Exits

Every legacy repository ultimately needs one of four outcomes. Defining those outcomes turns an open-ended problem into a set of decisions with clear criteria and prevents migration from becoming the default.

*Figure 14.1. The four exits from a legacy repository.*

| Exit | When it applies | Typical cost factors | What it produces |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Dispose in place, then decommission | All information is eligible for authorized destruction, unencumbered, and clear of unresolved dependencies | Disposition, verification, and decommissioning; complexity depends on the repository | Disposition certificates and a retired system |
| Extract, dispose the remainder, decommission | A defined subset is still within retention or has continuing value | Extraction and validation, disposition of the remainder, and decommissioning | A governed population in a managed repository, certificates for the rest, and a retired system |
| Reduce to read-only with a review date | Continued retention is required and extraction would lose necessary context or is not feasible | Ongoing operation, support, access, and periodic review | A supported system with an owner, a cost, and a review date |
| Migrate | A governed category with a long period in a system that cannot be maintained | Migration and validation, source retirement, and ongoing destination operation | The same information, governed, in a platform that will hold the rule |

Two of these exits depend on being able to destroy information. Without category-level authorization, the first two exits are unavailable, leaving the organization to keep the system or pay to move the content. That is why migration is so often treated as the only realistic option: it may be the only option the organization currently has authority to execute.

These are cost factors, not a fixed ranking. Compare the feasible exits using the repository's content, dependencies, preservation needs, and technical constraints.

The third exit is the one most often taken by accident. A system is left running because no decision was made. Over time, that inaction gets reframed as a read-only strategy. The difference between the accidental version and the deliberate one is that the deliberate version has an owner, a stated annual cost, and a date on which the decision is revisited.

It is also the exit with the strongest legitimate case, and the case rests on context rather than cost. Records extracted from a system frequently arrive without the structure that made them interpretable: the reference data, the relationships between records, the workflow state, the codes that meant something to the people who entered them. A permit file pulled out as a set of scanned images and a spreadsheet of metadata may be technically preserved and practically unusable a

decade later, which is a poor outcome for a category with a thirty-year period. Where that risk is real, keeping a reduced system running is a governance decision rather than a failure to decide, and it should be documented as one. Chapter 7 explains the basis and custody arrangements required for a permanent designation.

## A Decision Date Is Not a Decommissioning Date

Legacy programs are commonly reported by decommissioning date, and most of those dates are aspirational. A system with a target retirement three years out and no intervening milestone is not on a plan. It is on a list.

Two dates are needed. The decision date is when the exit will be chosen, informed by the content profile, constraints, and feasible options. The decommissioning date follows from the chosen exit and its funding. For a read-only system, a review date may replace an immediate retirement date, particularly where retention is long or permanent.

Separating the dates gives the program an earlier milestone it can manage. A portfolio of undecided entries becomes a schedule of assessments and decisions, followed by estimates and funding requests for the chosen work.

## The Orphan Question

Legacy and orphaned describe different conditions. A legacy repository may be superseded or lack an agreed lifecycle treatment. An orphaned repository has no accountable owner. A repository can be both. Ownership needs to be assigned before an exit can be authorized, even if no former business owner claims it.

Orphans need a step the others do not. Before an exit can be chosen, someone has to claim the system, and the most effective way to find that person is to announce an intention. A notice stating that a repository will be scheduled for disposition review, sent to the functions most likely to care, surfaces owners faster than prolonged research. People who may not respond to an inventory question will often respond when told that the repository is being considered for disposition.

What comes back is informative. If nobody claims a repository after a reasonable notice period, that is evidence about its value, and it should be recorded as such. Silence does not authorize disposition. The exit still runs the same three gates: eligibility against the schedule, encumbrance against the relevant records, including holds and other applicable restrictions, and dependency against whatever might still rely on the repository. What the silence changes is the context for the decision, not the process.

No repository should be decommissioned without confirming the authorized treatment of its information. Encumbrance clearance belongs in the decommissioning procedure and includes the hold register and records of other applicable restrictions. Decommissioning can otherwise destroy preserved or restricted information while every status indicator still appears green.

## Why Legacy Accumulates

Legacy accumulates for reasons that have nothing to do with negligence.

- Every project that replaces a system has a budget for the new one and rarely a budget for retiring the old one. The old system stays up for a transition period that is never formally closed.
- Retention periods outlast platforms. A category with a fifteen-year period will outlive the system that created it, and keeping the old system running is the path of least resistance compared with extracting the content and preserving its context.
- The cost is diffuse and someone else's. Licenses, hosting, and support are spread across budgets that nobody reviews together, so the total is rarely seen.
- Nobody is accountable for the legacy portfolio as a whole. Individual systems have owners, or once did. The question of how many legacy repositories exist and what would resolve them belongs to no role, so it is asked only when an external event forces it.

The last one is the compounding force. Because no one owns the legacy portfolio, remediation happens in response to events rather than on a cadence. Each event resolves the systems in its own scope and leaves the rest.

## What Works in Practice

A state transportation agency with roughly four thousand staff across a central office and district locations had accumulated systems over three decades: permitting, construction project management, right-of-way records, an imaging system from the 1990s, and a superseded human resources platform.

The trigger was a public records request that took nine weeks. Staff searched six superseded systems, and three of them required a retired database administrator to be brought back on contract because nobody currently employed could run the queries.

The inventory showed sixty-one repositories carrying legacy or orphaned status. Nineteen had no owner. The initial proposal, already drafted, was to migrate all sixty-one into the shared cloud environment.

The governance lead proposed a different first step: assign an exit to each entry before pricing anything, and set a decision date within ninety days rather than a decommissioning date within three years.

The orphan notice went out first. Fourteen of the nineteen unowned repositories were claimed within three weeks by people who had not responded to two years of inventory requests. Five were never claimed, and running the gates on those five showed that four held duplicated extracts of systems still in production and one held a decade of scanned correspondence already captured in the imaging system.

The extract-and-decommission path then stalled for four months, and the reason is worth recording. Nobody could authorize destruction of the residual content. The agency had category-level authorization for its active records program but had never extended it to superseded systems. Once counsel approved the same authorizations against the legacy categories, the path that had been blocked for four months took three weeks.

The final distribution was not what the migration proposal had assumed. Twenty-two repositories were disposed of in place and decommissioned, because everything they held was past its period and unencumbered. Seventeen had a governed subset extracted into the records repository, with the remainder disposed of and the system retired. Fourteen moved to a deliberate read-only state with a named owner, a stated annual cost, and an annual review, mostly right-of-way and construction records with long or permanent periods. Eight were migrated. The number requiring migration fell from sixty-one to eight.

Two years in, forty-one systems had been retired, including two of the fourteen initially retained in a read-only state. The records request that had taken nine weeks took four days.

The figures and timeframes in this example are illustrative, not benchmarks. The work needed to assess and execute an exit determines the appropriate decision and delivery dates.

## Implementation Steps

**Set a decision date before setting a decommissioning date.** A ninety-day target can provide a useful starting point, with timing adjusted for the work needed to assess content, dependencies, and preservation. The chosen exits then provide the basis for estimating remediation cost.

**Confirm that category-level disposition authorization extends to superseded systems.** Programs commonly authorize their active categories and never extend the same authority to legacy content. Two of the four exits are unavailable until this is done. Extending that authorization is a governance decision, not a technology project.

**Use an orphan notice to help identify the owner.** Notify the functions likely to use the repository and record their responses. If it remains unclaimed, assign an accountable owner through governance. Silence provides context for review but does not authorize destruction or remove the need to run the gates.

**Make the read-only exit deliberate.** An owner, a stated annual cost, and a review date. Without those three it is not an exit, it is the status quo with better wording.

**Put the disposition gates in the decommissioning procedure.** Verify eligibility, clear applicable encumbrances against the relevant records, and confirm the current dependency assessment before the authorized action. The hold-register check is part of that control, not a substitute for other applicable restrictions.

**Give the legacy portfolio an owner and report it as a portfolio.** Track the number of entries, exits chosen, decision dates met, systems retired, and annual cost still being carried. Without that portfolio view, remediation stays event-driven and legacy continues to accumulate between events.

## What Good Looks Like

In a program that has this under control, legacy is a managed portfolio rather than a recurring surprise.

Every entry carries a status, an accountable owner, a chosen exit or decision date, and a decommissioning or review date. Formerly orphaned repositories have an assigned owner even where no business function claimed them. Read-only systems have a stated cost and an annual review. Decommissioning requires preservation checks and verified treatment of the information. Superseded systems enter the portfolio when the replacement goes live.

The clearest sign is what happens during the next replacement project. In a program with an owned portfolio, retirement of the predecessor is scoped and funded as part of the project. In a program without one, it is deferred, and the legacy portfolio grows by one.

## What Proves It

Legacy remediation produces clear evidence because each repository should end with a documented decision and outcome.

- A legacy and orphaned portfolio drawn from the repository register, with status, owner, chosen exit, decision date, and decommissioning or review date on every entry.
- Orphan notices issued and the responses received, including recorded non-responses.
- Category-level disposition authorization covering legacy categories, not only active ones.
- Gate records for each retirement: eligibility, encumbrance clearance against the relevant records, and the current dependency assessment.
- Disposition certificates and extraction records for each retired system, naming the terminal state reached.
- The annual review record for read-only entries, with the cost carried and the decision to continue or retire.

The measure worth carrying at chapter level is the undecided count: how many legacy and orphaned repositories have no chosen exit, read with the age of the oldest undecided entry. It is deliberately not a count of systems retired, because retirement is slow and lumpy while decisions are fast, and a program that measures only retirements will look stalled during the period when it is doing the work that matters.

## Business Value

It removes cost that nobody is defending. Licenses, hosting, support contracts, and the occasional retired specialist brought back to run a query are spread across budgets that are never reviewed

together. The savings may appear across several budgets, which makes them easy to underestimate initially but measurable once systems are actually retired.

It can reduce breach scope. Legacy and orphaned repositories may hold old sensitive information with weak controls. Verified destruction removes that population; extraction or migration changes where it is held and must establish the controls that apply in the destination.

It can shorten response times. In the case study, the nine-week records request depended on searching six superseded systems. The same access problems could also delay privacy responses, discovery, and incident assessment.

And it converts an unbounded liability into a costed plan. An executive asked to fund legacy remediation with no exits chosen is being asked to approve an unknown. The same executive shown sixty entries with four exit types, a cost per type, and a schedule is being asked to make an ordinary business decision.

## Nothing Retires Itself

Legacy systems are not necessarily a failure of technology management. They are often the result of decisions that were deferred when the immediate cost of doing so seemed low. Over time, those deferred decisions become someone else's operational and governance problem.

The work requires both technical assessment and decisions by people with authority over the information. Assigning that authority is part of the remediation plan, which is why legacy work depends on a functioning disposition process.

Two organizations can each have sixty legacy systems and the same operating cost today. If one has made an exit decision for every system and the other has not, only one has a credible path to reducing the problem.

Next: Chapter 15 addresses preservation as an operating process, from scope and implementation through release and verification.

# Chapter 15: Legal Hold and the Interruption of the Lifecycle

Ask a program how many legal holds it has open. Most can answer within a minute.

Ask how much information those holds are preserving, in which systems, and how much of it would otherwise have been destroyed years ago, and the answer takes weeks or does not arrive at all.

That gap is the subject of this chapter. Preservation obligations suspend normal disposition for information that falls within scope, yet the operational mechanics are often poorly defined. Earlier chapters established whether repositories can suspend treatment, whether preservation reaches the relevant copies, and how released information returns to disposition. This chapter covers the process in between: scoping a hold, applying it, keeping it accurate, and ending it.

## The Notice Is Not the Control

A legal hold notice is a communication. It tells named people that a matter exists and asks them not to delete relevant information. Preservation requires an implemented control, whether enforced by a system or performed through a documented procedure with oversight.

Most programs measure the communication. Notices issued, acknowledgments received, reminders sent, and the percentage of custodians who responded. Those metrics primarily measure communication and acknowledgment. They do not establish whether preservation was actually implemented in the repositories holding the information.

The distinction is between preservation enforced by a system and preservation performed by a person. System-enforced preservation prevents destruction of information within scope, either in place or through a preserved copy. Custodian-directed preservation relies on people following a procedure and requires oversight, particularly when they change roles or leave.

When preservation depends entirely on individual behavior, failure may be invisible because the platform continues operating normally. The hold register can continue to show the matter as preserved, and the gap may appear only when someone looks for a specific record and cannot find it.

Custodian-directed preservation is not inherently weak. A manual control can be effective when it is documented, performed consistently, and supported by evidence. The problem is failing to record which method a given pairing relies on, because system-enforced and custodian-directed preservation carry different risks and require different oversight.

Externally operated repositories may require both available tenant controls and a provider’s action. Chapter 17 addresses the contractual and evidence obligations needed to make those arrangements work.

## Reconcile Matter Scope With Repositories

A hold has a scope, and in many organizations it is written twice in two different vocabularies that are rarely reconciled.

Legal scopes by matter. Named custodians, a date range, a subject description, sometimes a list of systems written from memory. Governance scopes by category and repository, which is the vocabulary the retention rules and the implementation matrix use. Neither translation is automatic, and usually neither is attempted.

The result is the same cross-repository problem described earlier, but applied to preservation. A hold issued against named custodians reaches their mailboxes and personal drives, where convenience copies may live, and may never reach the managed repository holding the authoritative instance. Or the reverse: a hold applied to a record category in the document repository leaves convenience copies in a collaboration workspace subject to a rule that will delete them in ninety days. Information may remain available anywhere a copy survives, while preservation reaches only the locations actually included in the hold.

Express the operational hold scope in the vocabulary of the implementation matrix. Map the matter's subject, dates, and other criteria to category and repository pairings, adding custodians where individual accounts matter. Pairings locate the relevant systems; they do not replace the legal scope or mean that every record in a pairing must be preserved.

Recurring hold templates can provide a reviewed starting point. Check each template against the current repositories and adapt it to the facts of the matter. A prior review does not establish that the scope fits every new matter.

## Why Holds Accumulate

The same incentive problem appears in preservation. The personal risk of scoping too narrowly is much more visible than the organizational cost of scoping too broadly.

Scoping a hold too widely increases storage, collection effort, and review time, costs that are rarely attributed to the person who drew the scope. Scoping it too narrowly can create spoliation risk with a clearly identifiable decision behind it. The predictable result is broad scopes that are repeated in later matters and rarely narrowed. Four additional forces keep holds open.

- Closing a matter is nobody's deliverable. The litigation is resolved, the team moves on, and the release step has no deadline attached to it. Matters open a decade ago remain open because closing them was never anyone's task.
- Holds overlap and nobody tracks the overlap. The same records are frequently preserved under several matters. Releasing one matter changes nothing about the information, so release at matter level produces no reduction and no feedback, which makes the exercise feel pointless.

- Departures break custodian-directed preservation quietly. Standard offboarding deletes mailboxes and home drives on a schedule that was designed without reference to the hold register, and the register is not consulted.
- Nobody owns the encumbered population. Legal owns matters. Governance owns categories. The volume of information currently under preservation, and its age, belongs to neither, so it is never reported and never questioned.

The result is an encumbered population that continues to grow. Some of it may already be past its retention period, but it is excluded from the unencumbered disposition backlog defined in Chapter 13. Read the measures together to distinguish information legitimately preserved from information awaiting execution after restrictions have cleared.

## Release Is Half the Control

Applying a hold is usually treated as the serious part of the process, while release is treated as administration. Release deserves the same level of control because a release that does not work can leave information preserved indefinitely.

A release requires three steps. Record the matter's release, verify removal of that matter's preservation control in each affected pairing, and return the population to the disposition eligibility calculation. Where another matter or restriction still applies, preservation must remain in place.

Preservation and release are separate operations and may not mirror each other. A hold generally interrupts disposition, not the passage of the retention period. On release, use the rule version, retention trigger, and any approved transition or fallback applicable to the population. Do not restart the retention period unless that approved treatment requires it. Chapter 19 explains how changes govern information already in place.

Preservation copies can outlive the matter. Many platforms implement a hold by creating a separate copy rather than suspending the rule in place, and discovery tooling may do the same when a collection is taken. Each copy becomes a new location, and the retention rule may not reach it. If release does not address those copies, a record can survive both the hold and its own retention period in a location created by the preservation or discovery process.

Release verification should identify the treatment of every preservation or discovery copy. Where a copy must remain, including as a record of what was produced, record its location and approved rule. Otherwise, return it to the authorized disposition process and verify the required end state. Do not remove a copy while another preservation requirement still applies.

Track encumbrance against the population, not only the matter. If records are held under four matters, releasing one removes only that matter's restriction. The program needs to identify every remaining restriction before normal disposition can resume. When the final restriction clears, re-evaluate the population against its applicable rule and approved transition; release does not by itself make information eligible for destruction.

## What Works in Practice

A regional airline with roughly fourteen thousand employees carried a steady volume of matters: employment claims, passenger claims, and a small number of significant operational investigations. Legal hold was a mature process by the usual measures. Notices went out promptly, acknowledgments were tracked, and reminders were automated.

A routine employment matter exposed the problem. Preservation had been issued to twenty-three custodians and fully acknowledged. Two of them had left the company during the matter, and their mailboxes had been deleted thirty days after departure under the standard offboarding process, which had never been connected to the hold register. The register showed the matter as preserved throughout.

The review that followed found two hundred fourteen open matters. Sixty-one had been open more than five years, nine more than ten, and three had no identifiable owner because the attorney who opened them had left.

The team began by rewriting scope in the vocabulary of the implementation matrix. A matter's scope became a list of pairings plus named custodians rather than a prose description. Doing this once for the standard employment hold template revealed that it covered mailboxes and home drives but not the human resources system of record, where the authoritative instance of most of the relevant material lived. The template had been in use for six years.

They then recorded a preservation method against every pairing in an active matter. Of roughly three hundred forty, sixty-two relied on custodian-directed preservation alone. Those were flagged, reviewed quarterly, and prioritized for platform work. None of them was permitted to be the only preservation covering an authoritative instance.

The offboarding fix was the cheapest change in the program. The departure process now checks the hold register before mailbox deletion. It took a week to implement and closed the failure that had started the review.

Tracking encumbrance at the population level changed the conversation with legal. Roughly forty percent of the volume under hold was preserved by more than one matter, helping explain why closing matters had not reduced encumbered volume. Each release could now be recorded against the affected population, which returned to normal eligibility assessment when the final restriction cleared.

A closure review with counsel took the matters older than five years from sixty-one to twelve. The first reconciliation returned eleven closed populations to the disposition eligibility calculation, which was the first reduction in encumbered volume the program had ever recorded.

## What the Hold Register Holds

The register is the artifact that makes the rest of this operable, and it is usually a matter list with dates. It needs to carry more than that.

*Figure 15.1. The hold register.*

| Field | What it answers |
|---|---|
| Matter identifier and status | What this is and whether it is open, closing, or released |
| Legal owner | Who is accountable for the scope and for closing it, as a person |
| Scope as pairings | Categories and repositories holding the scoped population, with matter-specific criteria retained |
| Named custodians | Which individual accounts are in scope, where account-level preservation is required |
| Preservation method per pairing | System-enforced, including through an integration, or custodian-directed, with required oversight |
| Applied date and verification | When preservation was confirmed for the scoped population in each pairing and by what method |
| Overlap | Which other matters preserve the same population |
| Release date and verification | When the matter's restriction was removed and the remaining preservation and retention treatment verified |
| Requeue confirmation | That the population was re-evaluated under its rule and any remaining restrictions |
| Last reviewed date | Whether the scope is still accurate, since matters change as they develop |

Two fields carry more weight than the rest. Preservation method shows whether the control is system-enforced or dependent on individual behavior. Overlap is what allows release to produce an operational outcome rather than only a status change.

## Implementation Steps

**Map legal scope to the implementation matrix.** Record pairings, custodians where relevant, and the subject and date criteria needed to identify the population. Review the mapping with legal so that operational scope reflects the matter's requirements.

**Use reviewed templates for recurring matter types.** Keep them current as repositories and business processes change, and validate the scope for each new matter rather than applying a template without review.

**Record the preservation method for every pairing.** Prefer system-enforced preservation for authoritative instances. Where only custodian-directed preservation is available, document the procedure, oversight, and evidence, and assess whether an additional preservation mechanism is needed. Treat unresolved capability gaps as assigned work.

**Connect offboarding and system decommissioning to the hold register.** Both are routine processes that can delete preserved information without an obvious warning. Their checks need to identify affected populations before deletion proceeds.

**Give release three verified steps.** Record the release, remove only the relevant matter's preservation controls, and re-evaluate the population for disposition while honoring any remaining restrictions. Closing the matter record alone does not complete release.

**Give matter closure an owner and a cadence.** Review open matters with counsel at a set interval, with the age of each matter and the population it encumbers shown alongside it. Long-open matters often remain open because closure was never assigned as a responsibility.

## What Good Looks Like

In a program where legal hold works as an operating process, the question of what is being preserved has an answer that does not require a project.

Scope is mapped to pairings and reconciled with retention configuration. Every pairing under hold has a recorded preservation method, with evidence and oversight appropriate to that method. Offboarding and decommissioning check the register before deletion. Matters are reviewed with counsel, and each release is reflected against the affected population. The program can report what remains encumbered and why.

The clearest sign is that closing matters can reduce encumbered volume. When it does not, the program can explain which other claims or controls are keeping the information preserved.

## What Proves It

Preservation is often evaluated after the fact, sometimes in an adversarial setting. That makes contemporaneous evidence especially important.

- The hold register in the form of Figure 15.1, current, with preservation method recorded for every pairing in an active matter.
- Scope documented as pairings, with the reviewed template for each recurring matter type and the date it was last reviewed.
- Verification appropriate to the preservation method: platform records for system-enforced controls, and performance records, retained attestations, and oversight checks for custodian-

directed controls.
• Release reconciliation records showing the matter's restriction removed, remaining restrictions checked, and the population re-evaluated under its applicable retention rule.
• Evidence that offboarding and decommissioning check the register, including instances where a deletion was stopped.
• Matter review records showing open matters examined on a cadence with counsel, with closure decisions and reasons.

The measure worth carrying at chapter level is encumbrance load: the volume currently under hold, read with the age of the oldest open matter. Report how much of that volume is already past its retention period, and count populations once even when holds overlap. Alongside the unencumbered disposition backlog, this distinguishes preservation constraints from execution delays.

## Business Value

It reduces the cost of matters. Collection, processing, hosting, and review all scale with the population preserved. A scope based on pairings can be materially narrower and easier to explain than one based on everything a custodian touched.

It reduces one of the most significant preservation risks: information that should be held being destroyed by routine processes that were never connected to the hold. Connecting offboarding and decommissioning to the hold register can materially reduce that exposure.

It unblocks disposition. Preservation is one of the largest legitimate reasons eligible information cannot be disposed of. Reviewing long-open matters and distinguishing what is genuinely encumbered from what remains associated with matters that should have been closed can release significant volumes of information without requiring new technology.

And it gives legal and governance a shared vocabulary. Two functions that have historically worked through prose descriptions and different terminology can use the same list of pairings, reducing friction in both directions.

## Preservation Is Not the Absence of Deletion

It is tempting to treat a hold as simply the state of not deleting something. That framing makes preservation sound deceptively simple.

But preservation is an action with a scope, a method, a verification, and an end. Each can fail independently, and several can fail without producing an obvious warning. A hold that was never applied in the system holding the authoritative instance can look exactly like one that was, right up until someone asks for the record.

The programs that get this right are not more cautious than the ones that do not. They are more specific: about where the information is, about which mechanism is holding it, and about what has

to be true before a matter can be called closed.

Next: Chapter 16 addresses collaboration platforms and ephemeral content, including transactions of record value in chat, channels, recordings, and transcripts.

# Chapter 16: Collaboration Platforms and Ephemeral Content

Organizations often ask whether chat is a record. Chat is a medium, just like email or paper, rather than a record category. The relevant question is whether decisions, approvals, commitments, or other transactions of record value are occurring there, and what happens to that information afterward.

This chapter focuses on identifying those transactions and designing how they move into governed systems. Retention remains part of the answer, but the first task is to make sure the information the organization needs is recorded where it can be found and managed.

## One Platform Can Produce Several Retention Outcomes

A collaboration platform produces several content types with different locations and control capabilities. Unless the program has assessed them separately, some may follow inherited defaults rather than deliberately chosen rules.

*Figure 16.1. Content types in a collaboration platform.*

| Content type | Where it actually lives | Can it carry a rule | Potential outcome to check |
|---|---|---|---|
| Channel messages | The platform, in the channel or team container | Often, through container or tenant controls | Retention without a chosen end point |
| Direct messages | The platform, against the user | Often, through user or tenant controls | Retention without a chosen end point |
| Files shared in channels | The connected storage platform, not the chat platform | Yes, and often already governed | Whatever the storage platform does |
| Meeting recordings | The storage platform or a media service | Usually yes, sometimes only tenant-wide | Expiry after a short default period |
| Transcripts | With the recording or separately | Sometimes, and not always with the recording | Retention that differs from the recording |
| Generated summaries and notes | The platform, often outside any inventory | Depends on the feature and storage location | Retention without separately assessed controls |

| | | | |
|---|---|---|---|
| Reactions, presence, and edit history | Platform metadata | Often limited; verify available controls | Provider-defined behavior that needs assessment |

The table identifies outcomes to investigate, not universal product defaults. Actual behavior depends on the platform, license, configuration, and version. A tenant may delete recordings quickly while keeping transcripts or generated summaries much longer. The program needs to establish what each content type actually does.

Automatically generated summaries and notes add to this problem. Platforms can produce them by default, and they can contain a condensed version of the decisions, discussions, and other material likely to matter later. They may not have existed when the governance design was written. Record the stores holding them in the repository register, with their content types and applicable treatment.

## The Unit Is the Transaction, Not the Message

Attempts to govern chat message by message quickly become unmanageable. A single message can carry record value, but most messages are routine coordination with little lasting value, while a relatively small number document decisions, approvals, commitments, or instructions that matter.

The useful unit is the transaction. Somewhere in the organization, particular things happen that carry consequence: an approval is granted, a commitment is made to a counterparty, a decision is taken that others will rely on, an instruction is issued that changes what someone does. Those are the events the schedule cares about, and they were happening long before chat existed.

The more useful question is which transactions of record value are occurring in the platform. The initial list may be small, perhaps three to six transaction types, such as approvals that used to be an email, confirmations to a counterparty, and terms agreed informally before paperwork follows. The organization needs to establish the list from its own business processes.

This reframing makes the work more manageable. Instead of treating every message as a classification decision, the program identifies the transactions that need a governed record and designs a reliable way to document them.

## Choose Capture or Displacement Based on Requirements

Where legal, regulatory, and business requirements allow, displacement is often more sustainable than broad capture. Some organizations or content types require capture for regulatory, supervisory, evidentiary, or other reasons. The choice depends on those requirements and how the information will be used.

**Capture.** Retain the channel content, index it, and apply rules to it, with the platform or its archive holding the authoritative instance. This can reduce the need for people to move information

elsewhere and may be appropriate or necessary. Broad capture also creates a large population in which material of record value is mixed with routine coordination, increasing the work required to preserve, search, and review it.

**Displacement.** Ensure that transactions of record value are documented in a governed system, with the context needed to understand them. Where requirements allow, channel content can then expire under a short, deliberate retention rule. This requires designing the landing place and making the required step easy to take. It can reduce the volume that must be managed over time.

The argument for displacement is not primarily about cost. An approval should be findable by someone who does not know which channel it happened in, was not a member of it, and joined the organization two years later. Retaining a channel does not by itself make that approval easy to find or establish its authoritative instance. A governed landing place can address both needs.

If a landing step is missed, the channel content may still be recoverable during its retention period, but that depends on the platform and the controls in place. Broad capture can solve the immediate preservation problem while creating a much larger population that must be retained, searched, reviewed, and eventually disposed of. Either approach needs verification that required information is available and that preservation obligations interrupt routine disposition when necessary.

A generated summary may have an approved short life, while an interaction supporting a consequential decision requires the record assessed in Chapter 24. Preserve the required information in its governed location before routine platform expiry. A content-type rule alone does not settle what must be retained to explain that use.

## Why These Platforms Stay Ungoverned

Four forces keep collaboration content outside the program, and only one of them is technical.

- The content feels like conversation rather than information. People may not experience sending a message as creating a record, and governance frameworks written around documents may not adequately address material that is authored collectively, edited after the fact, and read once.
- Rules attach to containers and users rather than directly to record categories. Translating the schedule into those controls requires a mapping, and programs may force one that does not fit or defer the problem.
- The platforms change faster than the governance cycle. Features arrive enabled, content types appear that did not exist at the last review, and a tenant configured deliberately two years ago is now producing three things nobody assessed.
- Nobody wants to be seen governing conversation. There is a real and reasonable concern about surveillance, and programs sense it, so the subject is approached carefully or not at all. The concern is legitimate, and a displacement approach can reduce it because the objective is to retain less conversation, not monitor more of it.

That concern deserves explicit attention because it can shape whether people understand and support the approach.

## What Works in Practice

A media and publishing company of roughly twelve hundred people worked across editorial, commercial, and rights, with most staff remote and most coordination happening in a single collaboration platform.

A rights dispute exposed the problem. A contributor agreement in the contracts system contained terms that the contributor said had been varied during negotiation, and the negotiation had taken place partly in a project channel and partly in direct messages. The channel no longer existed. The project workspace had been archived eighteen months earlier under a default nobody had set, and the archive had removed the message history.

The team began by cataloging what the platform produced rather than by writing a policy. Seven content types, of which exactly one had a retention outcome anyone had chosen, and that one had been chosen for a different reason: files shared in channels lived in the connected storage platform, which was already governed. Everything else was running on a default, and the defaults contradicted each other.

They then asked which transactions of record value were happening in the platform. There were three. Editorial approvals for publication, rights and permissions confirmations, and commissioning terms for freelance contributors.

Each got a landing place and a short required step. Publication approvals were recorded in the editorial system, which already tracked the pieces. Rights confirmations were required as an email to a monitored address that filed automatically into the rights repository. Commissioning terms were issued from a template in the contracts system rather than agreed in a thread. None of the three took more than a minute, and two of them were faster than the chat exchange they replaced, which is why they were adopted.

Channels were then given a design. Every channel required an owner, a stated purpose, and a type. Project channels were archived at project close and retained two years. Standing function channels took the retention of the function they served. Social channels ran ninety days. Channel creation required a short form identifying an owner, purpose, and type. That introduced a small amount of friction at creation but resolved the retention question while the purpose of the channel was still clear.

Recordings and transcripts were set to the same deliberate ninety days, which was an increase for recordings and a substantial decrease for transcripts, and the generated summaries were brought into the inventory and given the same rule. Direct messages were set to one hundred eighty days, announced with the reasoning rather than quietly, alongside the three faster paths that removed most of the incentive to negotiate there in the first place.

Preservation also improved. With channels owned and typed, and with the three transactions landing in governed systems, more of the preserved population was covered by system-enforced controls rather than requests to individual custodians.

Eighteen months later the platform had seven chosen retention outcomes rather than one. The next rights query was answered from the rights repository in a day. The volume retained indefinitely in the platform also declined substantially.

The periods in this example are illustrative, not recommendations. Retention periods come from the approved schedule and the legal, regulatory, and business analysis behind it. What matters here is that each content type has a deliberate outcome, that the decision is documented, and that someone is accountable for it.

## Channels and Direct Messages

**Channel creation is a governance event.** Because the rule attaches to the container, creating a channel also establishes the basis for its retention. In many organizations, that happens without identifying an owner, purpose, or end point. Requiring an owner, a purpose, and a type at creation is a useful control because it asks the question while someone knows the answer. Container-based governance depends on knowing what business purpose the channel or workspace serves.

**Direct messages need a stated position.** They can feel personal while carrying business content, including consequential decisions and commitments. Direct messages are often the path of least friction, not a deliberate attempt to avoid governance. A practical approach combines a stated retention period, communicated with its rationale, with a governed path for transactions of record value that is easy to use. People route around friction, so the design needs to address how work actually happens.

These decisions concern how people work as well as how platforms are configured. Effective controls need to address both.

## Implementation Steps

**Catalog what the platform produces before writing policy.** List each content type, its location, the controls available in the tenant, and its current retention behavior. That assessment establishes which outcomes are deliberate and which need attention.

**Identify the transactions of record value rather than classifying every message.** Ask what decisions, approvals, and commitments happen here. Start with a short list of transaction types and validate it with the people doing the work. This is primarily an interview exercise rather than a technology project.

**Design a landing place for each transaction and make it faster than the alternative where possible.** A required step that is slower than the chat exchange is less likely to be taken consistently. Two of the three in the vignette were adopted because they saved time.

**Require an owner, purpose, and type at channel creation.** Map the type to approved retention treatment. At project close, place project channels into their defined closure process, including retention, preservation checks, and eventual disposition.

**Choose an outcome for every content type, including recordings, transcripts, and generated summaries.** An inherited default is effectively a policy outcome the organization did not deliberately choose and may be difficult to explain or defend.

**State the direct message position openly and explain it.** A period communicated with its rationale is more likely to be understood and accepted. Discovering it later can undermine trust and make the next governance conversation harder.

## What Good Looks Like

In a program with this work in place, collaboration content has defined treatment within the information environment.

Every content type has a chosen outcome with recorded reasoning. Channels have owners, purposes, and types, and project closure triggers the approved lifecycle process. Transactions of record value are recorded in governed systems with enough context to be found and understood by others. Direct message treatment is communicated, and a named owner assesses new content types and capabilities as they arrive.

The clearest sign is what a search looks like during a matter. In a program that has done this work, the systems where transactions landed provide a clear starting point, with conversation searched where the scope requires it. Without that work, finding the same decisions may require reviewing years of conversation. The difference can materially reduce the time and cost required to find and review responsive information.

## What Proves It

The evidence here is relatively straightforward, because most of it records choices that were either made or inherited.

- A content type catalog per platform, with the chosen outcome for each and the date the choice was made.
- The transactions of record value identified, with the landing place and the required step for each.
- Channel inventory with owner, purpose, and type, and the archive record for closed project channels.
- Evidence that the landing steps are being taken, sampled rather than assumed. Chapter 22 provides the related verification approach.
- The direct message position, its retention period, and the record of how it was communicated.

- Stores holding generated summaries and notes identified in the repository register, with the content types and applicable treatment recorded.
- Preservation method per pairing for these platforms, showing what moved from custodian-directed to system-enforced.

The measure worth carrying at chapter level is the inherited default count: how many content types in each collaboration platform are still running on an outcome nobody chose. The target is zero. Each new content type should be assessed so that the count remains a measure of deliberate governance rather than a one-time configuration exercise.

## Business Value

It can materially reduce discovery and privacy response costs. Collaboration content is often high-volume and relatively low-signal within the information environment, so broad collection, preservation, and review can become expensive quickly. Moving transactions of record value to governed systems and allowing routine conversation to expire where requirements permit can reduce that burden.

It makes institutional decisions findable. The business case that does not depend on risk at all is that an approval nobody can locate has to be made again, and the reasoning behind a decision that lives in a deleted channel is reconstructed by guesswork. Landing places serve the organization long before they serve a regulator.

It can reduce preservation gaps. Custodian-directed holds depend more heavily on individual behavior and are therefore harder to verify consistently. Owned, typed channels and transactions in governed systems can move more of that work to system-enforced preservation.

It replaces inherited defaults with deliberate decisions. An organization running seven content types on defaults may have retention outcomes it did not choose and may struggle to explain or defend. Reviewing and documenting those outcomes gives the program a basis for explaining what happens to the information and why.

## Govern Transactions and Set Deliberate Conversation Rules

Applying document-oriented classification and retention practices to every message can create substantial work without making decisions and commitments easier to find. The operating model needs to account for how collaboration content is created and used, as well as the requirements that apply to it.

Where legal, regulatory, and business requirements allow, routine conversation should have a short, deliberate life while transactions of record value move to governed systems. The aim is to preserve the authoritative instance and the context needed to understand it, with a clear landing place, a required step, and evidence that the step is being taken.

Before collaboration platforms existed, decisions were routinely made in meetings and informal conversations. Governance historically focused on documenting the resulting decision or commitment rather than every conversation that led to it. That remains a useful principle wherever applicable requirements allow.

Next: Chapter 17 addresses third parties, hosted platforms, and shared data environments, including the terms and evidence needed where the organization lacks direct control.

# Chapter 17: Third Parties, SaaS, and Information Outside Your Direct Control

There is a short window when governance requirements are easiest and least expensive to secure, and a much longer period afterward when adding them becomes harder and more costly.

That window is before the contract is signed. Requirements for deletion certification, export, preservation, and evidence can be negotiated alongside the other commercial and operational terms. After signature, the same requirements often become change requests or wait until renewal.

Many programs discover their requirements after the agreement is signed. That timing problem helps explain why externally held information is often among the least governed parts of the information environment. This chapter addresses the pairings identified in the Chapter 11 implementation matrix whose configuring party sits outside the organization, including the terms needed for disposition evidence and preservation.

## Three Relationships That Get One Name

Third party is a single phrase covering arrangements with very different control models. Treating them alike can leave important governance requirements unaddressed by standard contract language.

**The provider hosts your information.** A software platform, a hosting environment, a managed service. The provider holds information on the organization's behalf, while the organization may retain significant governance responsibilities. Control comes from the agreement and the configuration available in the organization's tenant. Contract templates often focus on this relationship, but the program still needs to verify what the terms and controls actually support.

**The information is jointly held.** A franchise, a joint venture, an industry utility, a shared data environment. Neither party fully controls the information and both have exposure. Contract language written for a hosting relationship may not fit, because the arrangement is not simply a service performed on one party's behalf. What is needed is an agreement about which party governs which population, but that allocation is often left unwritten.

**The provider maintains its own records.** A payroll provider, benefits administrator, broker, or agent may create and maintain records concerning your organization under its own legal and business requirements. You may need access to those records without controlling their retention or disposition. The governance questions are how to obtain the records and how long the provider will keep them. The provider's retention period may not match the period during which your organization needs access.

Each relationship can fail differently. Hosting arrangements can fail on configuration and evidence. Jointly held information can go ungoverned because responsibility is unclear. Access to a provider's

own records can fail when the provider disposes of them on its schedule before the organization requests them for a matter. Sorting relationships into these three types helps the program identify the controls each one needs.

## Provider Defaults Reflect Provider Priorities

Hosted platforms have retention behavior whether or not the customer configured it. Provider defaults reflect the provider's operational, commercial, and risk priorities, which may not align with the customer's governance requirements.

In some platforms, customer content may be retained longer than the organization needs while logs, audit trails, administrative records, and system metadata may be available for comparatively short periods. That creates a governance mismatch: the organization may retain information it wants to dispose of while losing evidence it may later need to demonstrate what occurred.

That may be the opposite of what the organization's governance program needs. Content needs to follow the applicable retention schedule, and evidence needs to remain available long enough to demonstrate that the required actions occurred. A provider may complete disposal correctly but delete the record of that action before the organization obtains it. Without other evidence, the organization may be unable to demonstrate that disposal occurred.

The program needs to identify both content and evidence retention defaults, negotiate changes where possible, and record any remaining gaps.

## Why Externally Held Information Goes Ungoverned

Commercial and procedural decisions can leave requirements unresolved even where technical controls are available.

- Procurement asks about security and availability, and less often about lifecycle. Security questionnaires are mature, standardized, and expected. Many organizations lack an equivalent set of questions about retention, deletion, export, and preservation, so these requirements are missed when they are generally easiest to negotiate.
- Governance is engaged after the decision. The program hears about a platform when it appears in the inventory or when something goes wrong, by which point the agreement is signed and the requirements have become change requests.
- Nobody owns the renewal calendar from a governance perspective. Renewals are managed by procurement or by the business owner on a commercial timetable, and another opportunity to address governance requirements is missed.
- Audit rights can be mistaken for evidence of control. A contractual audit right can provide useful escalation authority, but it does not itself produce evidence that controls are operating. Specific obligations to provide defined artifacts on a defined cadence make that evidence routine rather than exceptional.

- Distributed purchasing spreads the problem. A platform bought by a single function on a corporate card may bypass contract review, the inventory, and an owner who knows the lifecycle questions. The onboarding gate described in Chapter 2 helps only when purchases reach it.

The last two can compound. Small purchases may never reach a review, while larger ones may pass a review that checks for an audit right without establishing routine evidence obligations.

## What Works in Practice

A franchised restaurant group ran roughly nine hundred locations with thirty people at the corporate entity. The franchisees were independent businesses operating under the brand, employing their own staff and holding their own customer relationships.

A customer privacy request arrived at corporate and could not be answered. Loyalty data sat in a platform corporate had contracted. Order history sat in point-of-sale systems that individual franchisees had contracted independently, across three different products. Complaint records were in email at the locations. Corporate was accountable for the brand and had access to none of it.

A second event sharpened the problem. A terminated franchisee left with their point-of-sale data intact, and the franchise agreement gave corporate no ability to compel either export or deletion.

The team started by counting rather than negotiating. Forty-one pairings from the implementation matrix were held outside the organization, across eight provider relationships and the franchise agreement. Sorting them by relationship type showed six hosting arrangements, two providers holding their own records, and one relationship that was neither, which was where most of the exposure sat.

The contract review against a short list of governance terms was not encouraging. Of the six corporate agreements, two required deletion certification, one specified an export format, none carried any obligation to suspend deletion on request, and four contained audit rights that nobody had ever exercised or costed.

They shifted their negotiating emphasis toward defined evidence obligations that would supplement audit rights. At each renewal, they sought a duty to produce an annual disposition certificate, confirmation of a preservation request within five business days, and a configuration export on request. Four of the six agreed at renewal over eighteen months. Two declined, and those two gaps were recorded as accepted risk with a named approver and a date to revisit. That made the missing evidence obligations visible and assigned responsibility for reviewing them.

The franchise relationship could not be solved by contract alone. An amendment was unlikely to bring nine hundred independent businesses running three products under a configuration standard. Corporate licensed a single point-of-sale and loyalty platform centrally and offered it to franchisees below what they were paying, with the governance configuration already in place. It became the

default for new and renewing locations in the franchise agreement, but the adoption came from the price rather than the clause. Six hundred and ten locations moved within two years.

Where the organization cannot control the system, it can sometimes influence which system gets used through pricing and convenience.

The renewal agreement also gained a data provision requiring export in a specified format and certified deletion on termination. The provision addressed the contractual gap exposed by the departing franchisee; a subsequent termination would still require verification of the applicable outcome. The next privacy request was answered in nine days.

The figures and timeframes in this example are illustrative. Requirements for provider action and evidence should reflect the information, relationship, and applicable obligations rather than these example intervals.

## What the Agreement Has to Say

Use the terms below to assess the requirements that apply to each relationship and population. Hosted content may require return and certified deletion; jointly held information needs an allocation of responsibilities; provider-maintained records may require continued access and a documented retention arrangement. Apply deletion obligations only where they are appropriate to the information and applicable requirements.

*Figure 17.1. Governance terms for a third-party agreement.*

| Term | What it secures | What its absence looks like |
|---|---|---|
| Rights and permitted use | The organization's rights to the information and the provider's permitted uses are clearly defined | Uncertainty or disputes about access, continued use, and retention at termination |
| Retention and deletion | Applicable retention and deletion requirements are defined for the relationship, rather than left to provider defaults | Content retained longer than needed while logs or other evidence expire too soon |
| Deletion certification | Evidence that disposal occurred, in a form your program can retain | A disposition record that stops at the boundary of your own systems |
| Preservation on request | That the provider suspends its own deletion within a stated period when notified | A hold that reaches your systems and not the one holding the authoritative instance |

| | | |
|---|---|---|
| Duty to produce | Named artifacts on a stated cadence: certificates, hold confirmations, configuration exports | Reliance on an audit right without routine evidence that controls operated |
| Export on demand and at exit | A specified format, a specified timeframe, and the context needed to interpret the content | A migration priced as a project because the format was never agreed |
| Termination assistance and the destruction window | What happens to the information after the relationship ends, and by when | Copies persisting at a former provider without an agreed deletion deadline or evidence |
| Subprocessor transparency | Where the information actually goes beyond the party you contracted with | A repository register that omits downstream providers holding the information |

**Supplement audit rights with a duty to produce.** Audit rights remain useful for escalation, investigations, and other reviews, but exercising them can require notice, budget, staff, and scheduling. A duty to deliver specific artifacts on a defined cadence establishes routine evidence production without requiring a separate audit each time. Those artifacts support the control testing discussed in Chapter 22. Providers may also find defined, repeatable evidence obligations easier to agree to and administer.

The table does not provide model contract language. The appropriate wording depends on the jurisdiction, the relationship, and the agreement being negotiated, and counsel needs to review it in that context. Contract wording also changes as services and commercial practices evolve, while the need to obtain evidence that information was destroyed remains. The governance program should bring a clear statement of what it needs and why, so counsel can draft terms suited to the agreement. Starting with a model clause can reverse that process by focusing attention on wording before the organization has defined the outcome it actually needs.

## Implementation Steps

**Build the lifecycle questions into procurement, next to the security questionnaire.** Use a short, standard set covering retention, deletion, certification, preservation, export, and evidence before signature. This is generally the point at which these requirements are easiest to negotiate. The questionnaire makes asking routine rather than dependent on someone remembering.

**Sort the relationships into the three types before reviewing any contract.** Hosting, jointly held information, and records maintained by the provider for its own requirements need different terms. A single hosting template can leave important requirements unaddressed in the other two relationships.

**Own the renewal calendar from a governance perspective.** Every externally held pairing should carry its next renewal date, and the requirements it is missing should be queued against that date. Renewal creates another meaningful opportunity to address requirements that were missed initially.

**Supplement audit rights with a duty to produce.** Name the artifacts and cadence, and retain the evidence under the control evidence category described in Chapter 13. Assign responsibility for reviewing what arrives and following up when it is missing or incomplete.

**Record what you could not get, with an owner and a date.** A provider that declines a preservation obligation is a known gap that can be managed, escalated, or designed around. An unrecorded refusal can leave the organization relying on a control that does not exist.

**Where contractual control is limited, influence the choice of system.** Providing a governed platform centrally at a lower price or with greater convenience can encourage adoption where contract terms alone have limited effect. Commercial incentives can make the governance requirements practical for the parties expected to follow them.

## What Good Looks Like

In a program with this under control, externally held information has the same basic governance visibility as information held internally.

Every externally held pairing carries a relationship type, a contract reference, the governance terms present and absent, a renewal date, and an owner. Lifecycle questions are part of procurement and reach the provider before signature. Evidence arrives on a defined cadence and is reviewed. Where a term could not be secured, the gap has an owner, an approval for any accepted risk, and an expiry, and any earlier review date.

When a relationship ends, the program follows an agreed process suited to the relationship and the information. That may require export and certified deletion of hosted content, an allocation of responsibilities for jointly held information, or continued access to records the provider must retain under its own requirements. The outcome and any continuing obligations are documented.

## What Proves It

Evidence for externally held information is produced by someone else. The agreement should define what must be delivered and when, so the program can obtain it routinely.

- A register of externally held pairings drawn from the implementation matrix, with relationship type, contract reference, owner, and renewal date.
- A terms assessment per agreement showing which of the governance terms are present, which are absent, and what was attempted.
- Disposition certificates received from providers, retained under the control evidence category.

- Preservation confirmations from providers, with the elapsed time from request to confirmation.
- Configuration exports or equivalent artifacts delivered under the duty to produce, dated.
- Termination records showing the applicable outcome: export, verified deletion where required, or documented continued retention and access arrangements, with remaining obligations assigned.
- Accepted risks for terms that could not be secured, each with a rationale, accountable approver, expiry and any earlier review date, and an open-item reference. The next renewal is an opportunity for negotiation, not a reason to defer an earlier review that the exposure requires.

The measure worth carrying at chapter level is the count of externally held pairings with no contractual evidence obligation, read alongside the next renewal date for each. The count identifies the exposure, and the dates identify opportunities to address it. Together they provide a work plan for contract review and negotiation.

## Business Value

It helps close the gap between accountability and access. Outsourcing a service does not necessarily transfer the organization's underlying legal, regulatory, or contractual obligations. The organization may still need to demonstrate how information was retained, preserved, produced, or disposed of even when a provider performs the underlying activity. Contractual evidence obligations help the organization obtain that proof.

It addresses evidence requirements when they are generally easier and less expensive to secure. A deletion certification negotiated before signature may require relatively little incremental effort compared with trying to secure the same evidence during an investigation or after the relationship ends. If the provider has not retained the underlying evidence, it may be unable to produce a reliable certification retroactively.

It reduces exit cost and exit risk. Agreeing an export format, termination assistance, and a destruction window at the start makes a provider change more predictable. It can also improve the organization's commercial position by reducing dependence on a provider whose systems are difficult to leave.

It makes gaps actionable. A program that knows how many externally held pairings lack an evidence obligation, and when each contract renews, can prioritize negotiations and track whether those gaps are being addressed.

## The Obligation Does Not Travel

Information moves outside the organization constantly and for good reasons. Specialist providers can offer capabilities and economies the organization cannot match internally. Shared environments, franchises, and partner models support how many organizations operate and reach their customers.

What may not move with it is the organization's accountability. Depending on the relationship and applicable requirements, the organization may remain responsible for information it cannot see, in systems it cannot configure, held by parties with different operational, commercial, and risk priorities. Managing that responsibility is ongoing work. It requires lifecycle terms negotiated before signature and at renewal, routine evidence, and a register of requirements that could not be secured.

The strongest programs ask lifecycle questions while requirements can still be negotiated, rather than waiting until they have to request capabilities the agreement never required.

## Closing Part III

Part III has focused on the distance between an approved governance requirement and what actually happens to information in the environment. Across these seven chapters, the same lesson has emerged: policy does not become operational on its own.

Rules have to be translated into system controls, applied consistently across repositories, supported by authorized disposition, carried through legacy remediation and preservation, adapted to collaboration platforms, and extended to information held outside the organization. Each step requires clear ownership and decisions made by people with the authority to make them.

Effective governance reduces the number of ad hoc decisions required and assigns the decisions that remain to the right people at the right point in the process, before urgency or individual risk begins to drive the outcome.

Execution also has to be provable. The program needs evidence that configured rules operated, authorized disposition was completed, preservation was applied and released as intended, and providers met their obligations. That evidence allows the organization to verify execution and identify gaps.

Part IV turns to defensibility and evidence: what makes retention decisions explainable, how change control keeps rules current, how unresolved work becomes visible and accountable, how measures describe the position, and how controls are tested.

# Part IV: Defensibility and Evidence

# Chapter 18: Defensible by Design

A reasonable retention decision can be difficult to defend if the organization cannot show how it was reached.

A period may have been carefully analyzed, approved by the right people, and applied consistently. But if the reasoning exists only in their memory, the organization may struggle to explain the decision when it is questioned. By then, those people may have moved on.

Part III followed approved rules into the environment. Part IV follows a governance decision through its rationale, subsequent changes, reporting, measurement, and testing. The objective is to connect what was approved, what was implemented, what remains unresolved, and what the evidence shows. This chapter begins with the decision itself: what the organization decided, why, and on what authority. Chapter 22 addresses whether the control carried it out.

## Two Questions, Two Kinds of Evidence

A reviewer examining a retention outcome, whether a regulator, an auditor, an opposing party, or an internal reviewer, is usually asking two questions, even when they arrive as one.

The first is whether the decision was reasonable. Was the treatment supported by the applicable requirements and a sound business rationale? Did someone with the authority to decide make the decision? Was it made on the information available at the time?

The second is whether the organization did what it decided. Did the control operate on real information, and did it leave evidence behind?

The two questions need different evidence. Evidence of intent describes what the organization meant to do. Evidence of effect describes what happened to information. The second question requires evidence of effect. The first requires a specific kind of intent evidence: a record explaining the decision. A schedule entry that states a period and a citation records the decision's outcome, but not necessarily the reasoning behind it.

A program needs both answers. A reasonable decision that was never carried out offers limited protection, and a control that operated as configured against a period nobody can explain invites the question of why that period was chosen. Answering the first question requires a clearly expressed rule, an accountable decision-maker, and a record of the reasoning. Answering the second requires evidence that the control operated as intended.

## The Citation Is Not the Rationale

Many schedules carry citations, and programs may treat those citations as the complete justification for a period. A citation can establish an underlying requirement, but it does not replace a record explaining how the organization interpreted and applied that requirement.

Approved retention treatments rest on one of three bases. Each needs a documented rationale, with the amount and type of detail proportionate to the decision.

*Figure 18.1. Three bases for a retention treatment.*

| Basis | How the treatment relates to requirements | What the decision record needs to explain |
|---|---|---|
| **Requirement-driven** | The treatment follows an applicable retention requirement | How the requirement was interpreted and applied, including any applicable retention trigger |
| **Requirement-extended** | The period exceeds an applicable minimum | The reason for the additional period, and confirmation that it was weighed against any applicable limit or deletion obligation |
| **Judgment-based** | No specific retention requirement applies | The business need, the alternatives considered, the owner who decided, and when the decision will be reviewed |

For a requirement-driven period, the decision record explains how the requirement applies to the category and its retention treatment, including any applicable retention trigger. The requirement may establish a minimum, a limit, or a deletion obligation. A requirement-extended period exceeds an applicable minimum, so its record also explains why the additional time was chosen and how any applicable limit or deletion obligation was addressed. A judgment-based period needs a documented business need and consideration of reasonable alternatives.

Where a category has different approved treatments, document the basis for each treatment and its applicable scope. Shared reasoning can be referenced rather than duplicated. One explained default does not establish the basis for every variation.

An approved permanent designation also needs a documented basis. Apply the same assessment to determine whether permanence follows an applicable requirement or a documented judgment about enduring value, including any extension beyond a minimum and limits considered. Retain the custody and review arrangements described in Chapter 7. A permanent designation is not an unassigned period or a requirement to invent a destruction date.

These can be sound decisions. A period may extend beyond a minimum because claims can arise later, because a business process depends on the history, or because the category supports another obligation. A judgment-based period may reflect years of operating experience. Those reasons still need to be recorded if the organization is to explain the treatment later.

An extended period carries its own risk. Retaining information longer than required can conflict with limits or deletion obligations that apply to the same information, and it adds to the cost and exposure described in Part I. A rationale for an extension should show that those considerations

were weighed, not only that someone preferred to keep the information longer. Keeping information longer is not automatically the safer choice.

## What a Decision Record Holds

The decision record template in Appendix F provides a consistent structure. Its value depends on the information recorded in it. A retention decision record should state six things.

- **The decision.** The category, the treatment approved, and the rule version it produced.
- **The basis.** The applicable requirements considered, how they were interpreted, and which of the three bases the period rests on.
- **The rationale.** Why this treatment was chosen over reasonable alternatives, including any business need that extends the period beyond an applicable minimum.
- **The authority.** Who approved the treatment and the delegation, charter, or policy provision authorizing them to do so.
- **The information available.** What was known when the decision was made, including open questions, dissenting views, and any risk accepted.
- **The review point.** When, or under what conditions, the decision should be revisited.

These are distinct from the eight elements of a retention rule in Chapter 7. The rule elements describe what the rule is. The decision record explains why that rule was chosen. The two should be linked and versioned together so that point-in-time reconstruction returns both the rule and its documented rationale.

Chapter 25 applies these six decision elements to AI-use decisions, adapting the subject and references. The three bases for retention treatment remain specific to retention decisions; they are not a classification of AI-use risk.

Some of the reasoning may rest on legal advice. Agree with counsel in advance what the governance record will hold and what will remain in privileged communication, so the record can state the basis for a decision without disclosing advice the organization intends to protect. A reference to legal advice with no stated basis may leave the record unable to answer the question it exists for.

## Preserve the Original Decision Record

A contemporaneous record shows what the organization knew, considered, and approved when the decision was made. An explanation prepared later may still be useful, but it should be identified as a later explanation rather than presented as the original rationale.

Where the original reasoning cannot be established, the current owner, working with counsel, can review the category, confirm or revise the treatment, and document the decision using the date of that review.

That record explains the current treatment, the information considered, and who authorized it. It does not suggest that missing historical reasoning has been recovered.

## Consistency Is Part of the Defense

A reasonable decision applied inconsistently becomes harder to defend. Chapter 12 addressed consistency of outcomes across repositories. The concern here is consistency of decisions, and it turns on two questions.

Are similar categories treated alike, or is the difference explained? Where two categories with similar content and requirements carry different periods, a reviewer may ask whether the difference reflects a reasoned distinction or selective retention. The different-outcome test in Chapter 9 can identify categories with the same treatment that may be candidates for consolidation. The reverse check identifies similar categories with different outcomes and no recorded reason.

Are rule exceptions controlled? A rule exception granted with a reason, an approver, and an expiry date is part of a governed program. A pattern of open-ended exceptions for the same category suggests that the stated rule may not be the operative one. At that point the exception has become the rule in practice, and the approved rule needs review through the change process in Chapter 19.

## Why Rationale Goes Unrecorded

The reasoning behind a retention decision may have been clear at the time but left unrecorded for ordinary, structural reasons.

- **Approval workflows may capture the signature without the reasoning.** A process that records who approved a change and when, but not why, leaves an important part of the decision undocumented.
- **The reasoning seems obvious at the time.** The people in the room share the context. Writing down what everyone already knows feels unnecessary, and the need becomes apparent only after those people have moved on.
- **The system may capture citations without capturing rationale.** Where the schedule system provides no clear place for the rationale, the reasoning may end up in email, meeting notes, or memory.
- **Decisions are inherited.** Many schedules carry periods set years earlier, sometimes by predecessor organizations or through earlier versions of the schedule, with no surviving record of why.
- **Legal analysis and governance records sit in different places.** Where counsel's analysis is held only in privileged communication, the governance record may carry the outcome with no statement of its basis.

Institutional memory can carry a program for a time, particularly with a stable team. It does not survive turnover, reorganization, or acquisition well, and those are the events most likely to come before a question about an old decision.

## What Works in Practice

A commercial and residential real estate company with roughly two thousand four hundred employees owned and operated about three hundred properties across several regions. Its schedule held one hundred sixty categories and had been reviewed by outside counsel four years earlier.

The question came through a tenant dispute. Tenant files for a closed lease had been destroyed three years after the lease ended, as the schedule required. The opposing party noted that building maintenance records for the same property had been kept for ten years and asked why the organization kept one and not the other.

The distinction was reasonable. Maintenance records supported building condition assessments and capital planning over the life of the asset, while tenant files carried no comparable need once the relevant claims periods had passed. But nobody could show that this reasoning had been applied when the periods were set. The manager who proposed them had left, and the approval record showed a committee date and a list of attendees.

The team prepared an explanation, and counsel judged it sound. It was still an explanation written after the question, and assembling support for it took several weeks for a decision that had originally taken an afternoon.

Afterward, the team did not try to reconstruct the history of every category. They sorted the schedule by basis. Sixty-one categories were requirement-driven, fifty-four were requirement-extended, and forty-five were judgment-based. The ninety-nine categories in the latter two groups became the priority review population because most lacked a recorded rationale. The requirement-driven categories still needed a documented interpretation of the applicable requirement and trigger.

They reviewed those ninety-nine categories over two quarters, working through them by business function. For each, the information owner and counsel confirmed or revised the treatment and recorded the rationale with the date of the review. Sixty-eight periods were confirmed with a stated reason. Thirty-one were shortened, most to the applicable minimum, because nobody could state a current reason for the additional period. These were changes to approved retention decisions, not evidence that resulting disposition had occurred.

Three changes were introduced to help prevent recurrence. The change approval form gained required fields for basis, rationale, and the authority under which the approver acted, so an approval could not be recorded without them. The rationale was versioned with the rule, so a later reconstruction would return both. And standing rule exceptions were reviewed. Of fourteen found, nine had no expiry and no current reason; those exception approvals were ended.

Ending an exception approval does not, by itself, establish that the standard treatment has resumed. Any resulting implementation work needs an owner and appropriate verification.

When a period was questioned again, during due diligence for a portfolio sale eighteen months later, the team produced the decision record for the category in a day.

The retention periods and figures in this example are illustrative, not recommendations. Actual treatment should reflect the organization's approved requirements and documented analysis. The example shows why different treatments need a documented basis.

## Implementation Steps

**Classify each approved retention treatment by basis.** Use requirement-driven, requirement-extended, and judgment-based to identify the rationale needed and prioritize gaps. Where a category has more than one treatment, retain the basis and applicable scope for each.

**Document the rationale for every approved treatment, with detail proportionate to the decision.** For requirement-driven periods, explain how the requirement was interpreted and applied, including the trigger where applicable. For requirement-extended periods, explain the additional period and confirm that it was weighed against any applicable limit or deletion obligation. For judgment-based periods, record the business need and reasonable alternatives.

**Record authority, not only approval.** Identify the delegation, charter, or policy provision that allows each approver to decide, and keep it current as roles change. Chapter 27 develops the decision rights behind it.

**Version the rationale with the rule.** A change to the treatment should produce a new version of the rationale rather than overwrite the previous one. Point-in-time reconstruction should return the documented reasoning in effect on a given date along with the rule.

**Review inherited decisions where the original rationale is missing.** Have the current owner confirm or revise the treatment and record the rationale using the date of that review. Start with the categories carrying the highest exposure or the longest extension beyond an applicable minimum.

**Make the record a condition of approval.** Build basis, rationale, and authority into the change approval, including approvals for rule exceptions, so a decision cannot be recorded without them. Chapter 19 develops the change process that produces these entries.

## What Good Looks Like

In a program where decisions are defensible by design, a question about why a period was chosen is answered from a record rather than from memory. For each approved treatment within a category, the program can show the requirements considered, how they were applied, the basis and rationale for the treatment, who decided, under what authority, and when. The rationale is linked to each documented rule version. Where historical reasoning is unavailable, the gap is identified along with the date the current treatment was reviewed and documented.

Similar categories are treated alike, or the difference is explained. Rule exceptions carry a reason, an approver, and an expiry date, and recurring exceptions lead to a review of the rule rather than

another exception.

The clearest sign is what happens when a knowledgeable person leaves. When reasoning has been documented, the organization can still explain the decisions that person helped make.

## What Proves It

The evidence for this chapter is evidence of intent: a record of what was decided and why. Its value depends on being specific and accurately dated, with contemporaneous reasoning distinguished from later explanation.

- A basis classification for each approved retention treatment, with its applicable scope and category reference.
- Decision records in the Appendix F form, carrying basis, rationale, authority, the information available, and a review point.
- Rule versions linked to their documented rationale, with gaps identified where historical reasoning is unavailable.
- An authority record showing the delegation under which each approver acted, current as of the decision date.
- Dated review records showing how inherited retention decisions were confirmed or revised.
- Rule exceptions carrying a reason, an approver, and an expiry date, with records of rules reviewed because exceptions recurred.

These artifacts use the structure already in place rather than adding to it. The decision record is the Appendix F template, and the reconstruction test in Chapter 22 exercises the versioned rationale as part of the chain it already tests.

The chapter's priority measure is the count of unexplained periods, reported as the number of categories with at least one requirement-extended or judgment-based treatment lacking a recorded rationale. Count each category once in the total, with the affected treatments and their bases available underneath it. For each basis group, show the number of in-scope categories and the number with unexplained treatments. A category may appear in more than one group, so the group counts are not additive. This identifies a priority review population, not a complete measure of decision-record quality. Requirement-driven treatments also need a documented interpretation and basis. Track the reduction as the identified gaps are reviewed and documented.

## Business Value

It lets the organization show its reasoning rather than argue it. Retention decisions are often questioned long after they were made. A contemporaneous record answers the question directly, which tends to shorten the response and strengthen its credibility.

It reduces dependence on individuals. Recorded reasoning survives turnover, reorganization, and acquisition, and the program no longer needs a particular person in the room to explain its own

schedule.

It identifies retention periods that need to be reconsidered. Reviewing extended and judgment-based periods can identify opportunities to reduce unnecessary retention. Where the revised treatment leads to approved disposition, it can reduce the information the organization continues to carry and the associated cost and exposure.

It makes change faster. When the rationale for the current rule is on file, a proposed change can be assessed against it directly rather than starting from an unexplained number.

## Record the Reasoning When You Decide

Retention decisions are made by people with context: counsel who interpreted the requirement, owners who understand the business, committees that weighed the tradeoffs. That context helps explain the decision. It is also easy to lose unless recorded.

When a decision is questioned, the people who made it may no longer be available. A maintained decision record preserves the reasoning needed to explain it.

Defensibility is not added when scrutiny begins. It is built when the decision is made, by writing down why.

*Next: Chapter 19 treats change control as a governance discipline: how the organization reviews, changes, and communicates rules while keeping the decision record current.*

# Chapter 19: Retention Is Not Static

A requirement governing one category of records took effect in March. It had been published the previous autumn. The records committee reviewed it at its annual meeting in November of that year, approved a longer period, and published a new version of the schedule the following January. The affected platforms were reconfigured over the following spring.

Counsel, the committee, and the configuring teams each completed their assigned work. The process still left the prior rule operating beyond the requirement's effective date, with disposition continuing while the change moved through review, approval, and implementation.

Requirements, business operations, and systems change continuously. Many retention programs change in cycles. This chapter treats change control as a governance discipline: identifying changes, assessing their impact, approving the treatment, and carrying it through to implementation. That includes deciding what happens to information already in place when a rule changes.

## Scheduled Review Is Not Enough

Most programs anchor change on a scheduled review, often annual. The review has real value. It confirms that the schedule still reflects the organization, it surfaces accumulated drift, and it gives the committee a regular point of accountability.

A scheduled review cannot substitute for responding to changes that require action before the next review date. A requirement takes effect on its own date. An acquisition closes when the transaction closes. Control testing may identify a rule that cannot be performed as written. None of these waits for the committee calendar. Routing them all through an annual review can leave outdated rules operating while decisions wait.

Scheduled reviews can also drift toward ratification. A category is reviewed, no change is proposed, and the minutes record that it was confirmed. The record may not show whether anyone checked the underlying requirements, the business need, or the review point in the decision record described in Chapter 18. Without a record of what was checked, the organization may struggle to show that a no-change confirmation was based on a substantive review.

The scheduled review works best as a check on the change process rather than a substitute for it. It can confirm that sources of change were monitored, that decisions due for review were reviewed, and that categories recorded as unchanged were examined.

## Where Changes Come From

Changes arrive from several sources, and each needs an assigned monitoring owner. Without that responsibility, changes may surface only through an audit, inquiry, or dispute.

*Figure 19.1. Sources of change and what they usually affect.*

| Source | Examples | Responsible owner | What it usually changes |
|---|---|---|---|
| **Requirements** | A new or amended retention requirement, limit, or deletion obligation; a changed interpretation | Legal and compliance, assigned by category family | The rule, and sometimes its jurisdictional variations |
| **Business events** | An acquisition, divestiture, reorganization, new product, new line of business, or entry into a new jurisdiction | The business owner, with the program engaged before the event closes | Categories, scope, and ownership |
| **Decision review points** | A review date or condition recorded in the decision record | The information owner | The rule, or a recorded confirmation |
| **Control evidence** | A control that fails repeatedly because the rule cannot be performed as written; recurring rule exceptions | The program, on referral from testing and rule-exception review | The rule |
| **Environment** | A new platform, a system decommissioning, or a platform change that alters retention behavior | Technology owners, through onboarding, ongoing platform changes, and decommissioning | The configuration, rarely the rule |

Control evidence is a source of change that programs can overlook. Chapter 22 recommends referring a rule to change control when repeated failures show that it cannot be performed as written. Recurring rule exceptions for the same category can raise the same question. Both need a defined route into the change process and a recorded response.

## Not Every Change Is a Rule Change

The first decision about any change trigger is what it affects. Some change triggers require a revised rule. Others affect system configuration or classification, or require no action. Treating every change trigger as a rule change can overload the committee. Treating every change trigger as a configuration matter can allow policy decisions to be made during system administration.

Triage each change trigger and record the outcome: a rule change; a configuration change, routed through the System-to-Policy Implementation Matrix in Chapter 11; a classification change, routed through the category model; or no action, with a reason. A single change may also generate linked configuration or classification work. Recording the outcome and rationale shows that the change was considered and identifies what needs to happen next.

## The Transition Is Part of the Change

Discussion of a rule change usually centers on the new treatment: the new period, retention trigger event, or disposition action. The question that often goes unasked is which information the new treatment applies to.

Every substantive change needs an answer. The new rule may apply only to information created after its effective date, or to information whose retention trigger event occurs after that date. It may apply to all existing information in the category or to a defined subset. Each answer has consequences, and they differ by the direction of the change.

- **A shorter period applied to existing information creates a newly eligible population.** That population needs disposition authorization under the new rule version, encumbrance checks covering holds and other approved suspensions, a current dependency assessment, and an estimate of the first disposition cycle's size. The initial cycle may be larger than those that follow.
- **A longer period creates a transition risk** if systems continue disposing of information under the prior treatment after the new requirement applies. The impact assessment should identify that timing gap and determine whether a temporary suspension of disposition is needed. Where necessary, authorize the suspension through the defined expedited process, with a stated scope, owner, and release criteria. Keep it distinct from a legal hold, and verify the revised configuration before ending the suspension.
- **A changed retention trigger event may require a different starting date for existing information.** The new event may not have been captured for information created before the change. The rule needs a stated fallback for that information, approved as part of the change, rather than defaulting to whatever date the platform has available.
- **A changed disposition action changes what the next cycle does.** Moving from destruction to transfer, or from review to destruction with sampled review, needs the receiving process in place before the first eligible population arrives.

Without a stated transition, configuring teams may have to decide how the new treatment applies to existing information. Recording the transition in the Appendix F decision record makes it part of the approved change rather than an implementation assumption.

## Assess the Impact Before Approval

Impact assessment happens twice, and the two answer different questions. Chapter 11 describes the assessment that follows approval: which pairings must be reconfigured, and how the change reaches each one. The assessment here comes first and informs whether and how the change should be approved.

Before approval, the assessment should establish:

- The categories, jurisdictional variations, and related categories affected, including similar categories whose treatment would now differ without a recorded reason.
- The number of pairings in the implementation matrix that will need to change, and any where the change cannot be enforced natively.
- The populations that become newly eligible for disposition or newly subject to retention, and any held populations within them.
- Downstream dependencies recorded under the dependency gate.
- Externally held pairings whose agreements may require notice or amendment.
- The requirement's effective date, the proposed rule effective date and transition, and whether the revised treatment or necessary interim controls can be in place when required.

The assessment does not need to be elaborate for every change. It needs to be proportionate, recorded, and complete enough that the approver knows what the decision will set in motion.

## Classify Changes by Effect, Not Size

Not every change needs the same approval. Triage identifies the work required and the process responsible for it. The change class determines the review and approval required for the proposed rule change, allowing routine changes to move quickly while substantive ones receive full review.

- **Administrative.** Changes that do not alter any lifecycle outcome, such as a corrected citation reference, a clarified description, or an alias added to the category model. Approved by the program with a record of the change.
- **Substantive.** Changes to period, retention trigger event, disposition action, scope, or jurisdictional variation. Approved by the accountable owner and counsel with a complete decision record, including the transition.
- **Urgent.** Substantive changes that cannot wait for the next scheduled forum, typically because a requirement takes effect sooner or a control failure creates ongoing exposure. Approved through a defined expedited path with named approvers and a time limit, then presented to the full forum for ratification.

Classify by effect on outcomes rather than by the size of the edit. A one-word change to a retention trigger event can alter when an entire category becomes eligible for disposition. A rewritten description that changes nothing about treatment may be administrative. Where the class is uncertain, treat the change as substantive.

A wording or alias edit is administrative only where its effect supports that classification. Where a definition, exclusion, instruction, or example is used by a classifier or control, assess the resulting assignment or behavior before treating the edit as non-substantive. Changes affecting assignments, access, scope, or approved reliance need the corresponding assessment and validation. Chapter 23 develops this application for classifiers.

## Why Rules Fall Behind

Change control is rarely neglected on purpose. Several recurring problems follow from treating the schedule as a document to review rather than a set of rules to operate.

- **Monitoring is not assigned between reviews.** Responsibility for requirements may be distributed across legal, compliance, and the business, without a clear obligation to bring changes to the retention program. Changes are noticed only when someone happens to raise them.
- **Change ends at publication.** The committee approves, the schedule is updated, and the change is considered complete. Communication to control owners, configuring teams, and providers, and verification that the environment reflects the change, are treated as someone else's work.
- **The transition is not asked about.** Approval focuses on the new treatment, and the question of what happens to existing information is left to configuration.
- **No-change decisions receive less scrutiny.** Where the process does not require evidence of what was reviewed, confirming an existing category can become easier than evaluating whether its treatment remains appropriate.
- **Change requests arrive in other forms.** A business unit asks for a rule exception or a new category rather than a change to an existing rule. Handled individually, these requests can expand the category model without addressing the rule that needs to change.

The result can be a schedule that is well maintained by its own measures and behind the environment it governs.

## What Works in Practice

A workforce staffing company with roughly three thousand five hundred internal employees placed around sixty thousand workers a year with client organizations across many jurisdictions in two countries. Worker onboarding, eligibility, timekeeping, and placement records carried requirements that varied by jurisdiction and changed often. The schedule was carefully maintained and reviewed annually.

An internal audit examined how one requirement change had been handled. The requirement had lengthened the period for a class of worker eligibility records in several jurisdictions. It had been published five months before it took effect. The schedule changed seven months after the effective date, at the next annual review, and configuration was completed four months after that. For eleven months after the requirement took effect, disposition continued under the prior period, and roughly eighteen thousand records were destroyed that the new requirement would have retained.

Counsel assessed the exposure, and the organization addressed it. The audit also identified a broader pattern. Across the prior three years, every requirement change the team could identify had been approved after it took effect, at a median of six months. Nobody had been assigned to monitor changes between reviews, and transition planning had never been part of approval.

The team made four changes. First, they assigned requirement monitoring by category family to named people in legal and compliance, feeding a monthly intake to the records team. Every item entering intake was triaged within thirty days as a rule change, a configuration change, a classification change, or no action with a reason.

Second, they defined three change classes and an urgent path. Urgent changes were approved by a delegate of the general counsel and the records lead within ten business days, then ratified at the next committee meeting.

Third, every substantive change carried a transition statement. For lengthening changes that needed interim protection, disposition for the affected category was suspended until configuration was verified in each affected pairing. The records lead owned the suspension, with its scope and release criteria documented.

Fourth, changes affecting system behavior were closed only when the approved treatment had been verified in every affected pairing. Other items closed against completion criteria appropriate to their scope. The annual review checked the intake log, revisited decisions whose review points had arrived, and sampled categories recorded as unchanged to confirm that they had been examined.

In the first year, two hundred twelve items entered intake. Thirty-one became rule changes, nine of them through the urgent path. Fifty-eight were configuration changes only, and one hundred twenty-three were closed with no action and a recorded reason. Twenty-four of the rule changes arose from new or amended requirements, and twenty-two of those were approved before the requirement took effect. One lengthening change required a five-week suspension of disposition while two repositories were reconfigured. It was the first time the program had suspended disposition for a rule change rather than a legal matter. Earlier identification, timely approval, and verified interim protection together addressed the sequence of failures the audit had identified.

The volumes and timeframes in this example are illustrative. Intake, approval, and implementation targets should reflect the urgency of the change and any applicable effective date, rather than being governed by the routine review calendar.

## Implementation Steps

**Assign responsibility for monitoring each source of change.** Cover requirements by category family, business events through the business owner, review points through the information owner, control evidence through testing and rule-exception review, and environment changes through technology owners. Record those responsibilities and use a common intake.

**Triage every change trigger and record the outcome.** Identify a rule change, configuration change, classification change, or no action with a reason, including any linked work. Set triage targets that reflect urgency and applicable effective dates so that intake does not become a backlog.

**Define change classes by effect on outcomes.** Administrative, substantive, and urgent, each with its approvers. Give the urgent path named approvers, a time limit, and a ratification step, so speed

does not come at the cost of accountability.

**Require a transition statement for every substantive change.** State which information the new treatment applies to, how newly eligible populations will be authorized and checked, whether interim protection is needed, and what fallback applies to a changed retention trigger event. Record the timing, owner, scope, and release criteria for any suspension in the transition field of the Appendix F decision record.

**Assess impact before approval.** Identify affected categories, variations, pairings, populations, dependencies, and external agreements. Compare the applicable requirement's effective date with the proposed approval and implementation timing, and identify any necessary interim controls. Hand the approved change to the implementation process in Chapter 11.

**Close changes on verification, not publication.** For changes affecting system behavior, closure requires verification of the approved treatment in every affected pairing. Other changes should close against documented completion criteria appropriate to their scope. Use the scheduled review to check the change process itself, including a sample of categories recorded as unchanged.

## What Good Looks Like

In a program that manages change well, approval is planned alongside implementation and any necessary interim controls, with the applicable effective date in view. Change triggers enter a known intake, are triaged within a defined period, and close with a recorded outcome and reason.

Approvals state what the change applies to as well as what it changes. Newly eligible populations are authorized and checked before the first cycle runs. Where interim protection is needed, its scope is defined and the revised treatment is verified before the suspension ends. The committee focuses on substantive changes, while administrative changes move through the program and urgent ones follow a defined path.

The clearest sign is that the schedule's currency no longer depends solely on its annual review date. Changes are handled when they need to be, and scheduled reviews check that the process is working.

## What Proves It

Change control produces evidence at each step, and much of it lands in records the program already keeps.

- An intake log of change triggers, showing the source, the date received, any applicable requirement effective date, the triage outcome, and the reason for each item closed with no action.
- Decision records in the Appendix F form for each substantive and urgent change, including the transition statement and the impact assessment.

- Approval records by change class, including urgent approvals and their ratification by the full forum.
- Suspension records for lengthened periods, showing authorization, scope, owner, start date, release criteria, and the dates configuration was verified and the suspension ended.
- Communication records showing that control owners, configuring teams, the hold function, and affected providers were notified.
- Closure records linking changes that affect system behavior to verified treatment in every affected pairing, with completion evidence appropriate to the scope of other changes.
- Scheduled review records showing what was checked, including the sample of categories recorded as unchanged.

The chapter-level measure is decision lag: the elapsed time from a change trigger entering intake to an approved change or a recorded decision to take no action. Read it alongside the count of rule changes approved after the requirement behind them took effect.

Configuration lag, the Chapter 11 measure, is the elapsed time from the effective date of an approved rule change to verified configuration in each affected repository.

Decision lag shows how quickly the program reaches a documented decision after an item enters intake. Configuration lag shows how implementation relates to the approved rule's effective date. Together they help distinguish decision delays from implementation delays, but they do not reveal changes that have not yet been identified. Read both in the context of any applicable requirement effective date.

## Business Value

It narrows the interval between a requirement taking effect and the corresponding treatment operating in the environment. Timely approval supports that outcome, but implementation and any necessary interim controls are what carry the decision into practice.

It protects information during transitions. A targeted, time-limited suspension can protect affected information while the approved change is implemented. Its scope, operational impact, and release criteria should be understood and documented.

It reduces the committee's workload without reducing its authority. Change classes move routine work to the program and give urgent changes a defined path, so the full forum spends its time on the decisions that need it.

It makes the schedule a current account of the organization's rules. Leadership, auditors, and control owners can rely on the version in effect, rather than on the version that was current at the last review.

## Current Is a Practice, Not a Date

A retention schedule is sometimes described as current as of its last review. That states when the schedule was examined, not whether it still matches the requirements, business operations, and systems it governs.

Change control addresses that gap through assigned monitoring, proportionate decisions, and verified implementation. Change triggers have owners, rule changes state what they apply to, and changes affecting system behavior are complete when the approved treatment is verified in the environment.

Programs that manage change this way do not avoid outdated rules altogether. They find them sooner, and they can show what they did about them.

Next: Chapter 20 turns to visibility: how an open-items view, linked source records, ownership, and escalation turn hidden implementation gaps into managed work.

# Chapter 20: Visibility as Control

The quarterly governance report showed coverage at eighty-seven percent, twelve workstreams on track, and no escalations. Each figure was accurate. None of them told the committee which gaps were open, who owned each one, or how long any of them had been waiting.

Most governance programs know more about their gaps than their reporting shows. Across the program, these lists include repositories without owners, pairings with no enforcement, eligible and unencumbered populations awaiting disposition, holds awaiting release review, agreements missing evidence terms, periods nobody can explain, changes awaiting triage, and test exceptions awaiting closure. Each list is useful where it was created. Together they describe the program's open work, and few organizations ever see them together.

Chapter 5 treated visibility as whether the program can see its own information environment. This chapter extends the idea to whether the program can see its own gaps, and whether that visibility reaches people who can act on them. Making a gap visible and assigning an owner creates the basis for managing it. Progress still depends on action, follow-through, and verification. A gap that is hidden, or visible only to someone without the authority to act, tends to stay open.

## A Percentage Is a Summary of a List

Governance reporting tends to favor aggregates: coverage percentages, completion rates, counts of controls in place. Aggregates are useful for direction. They are less useful for action, because nobody can close a percentage.

Every aggregate should be traceable to the underlying information and open work. A coverage shortfall should lead to the in-scope information or pairings that lack an operative rule. A disposition backlog should identify information that is eligible, unencumbered, and still awaiting disposition. An improving trend can conceal a single high-exposure item that has not moved in two years, because the items that did move outweigh it.

The unit of visibility should be the open item: one gap, with a description, location and source reference, current owner, due date, status, verification method, and exposure rating. A figure on a governance report that cannot be traced to the items behind it is harder to act on and harder to verify. Chapter 21 develops which measures belong on that report and how they are calculated. The point here is narrower: the underlying items need to be identified, owned, and accessible.

## Reported Status and Observed Status

Much governance status is reported by the people responsible for the work. A workstream lead marks a task complete. A control owner confirms a configuration. A business unit attests that its review was performed. Self-reported status has a place, and it is often the only source available early in a program.

Self-reported status may reflect assumptions about completion. A task marked complete when a configuration was requested may stay complete after the request stalls. Over time, the distance between reported status and actual status can grow without anyone intending it.

Where verification evidence is available, use it to establish status. A completed field in a register is not enough on its own; the entry should identify the evidence supporting it and when verification occurred. Where status remains self-reported, label it accordingly.

For a pairing, the implementation matrix should link to the evidence supporting its recorded status. Configuration verification establishes whether the implemented treatment matches the approved rule. Execution records and testing establish what happened when the control operated.

Disposition is complete when the information reaches the approved outcome and the disposition record documents it. For destruction, that includes the defined terminal state. Hold release requires evidence that the relevant suspension was removed and the population re-evaluated under its applicable rule and transition, with other restrictions preserved. A test exception needs evidence appropriate to its recorded response or resolution. Observed status depends on that evidence, not simply on where the status is recorded.

## One View of Open Work

Open items arise in many places across the program, usually in lists owned by different functions and held in different tools. That fragmentation has two costs. Nobody can see the total, so nobody can prioritize across sources. And the same underlying gap often appears on several lists, counted and worked separately.

*Figure 20.1. Where open items come from.*

| Source | The open item | Typical owner | Developed in |
|---|---|---|---|
| **Onboarding gate** | A system provisioned with a recorded gap in enforcement | Technology owner | Chapter 2 |
| **Repository register** | A repository with no current owner or no lifecycle decision | Information owner | Chapter 8 |
| **Implementation matrix** | A pairing not enforced, or a configuration not yet verified | Control owner | Chapter 11 |
| **Disposition cycle** | A population eligible, unencumbered, and still awaiting disposition | Control owner | Chapter 13 |
| **Legacy register** | A repository without an exit decision or past its decision date | Information owner | Chapter 14 |

| | | | |
|---|---|---|---|
| **Hold register** | A hold overdue for release review, or a release not requeued | Legal | Chapter 15 |
| **External pairings** | An agreement missing evidence or lifecycle terms | Relationship owner | Chapter 17 |
| **Decision records** | A period with no recorded rationale, or a rule exception past expiry | Information owner | Chapter 18 |
| **Change intake** | A change trigger awaiting triage or an approved change not yet verified | Program | Chapter 19 |
| **Control testing** | A test exception requiring a response or follow-through, including accepted exposure and unresolved referrals | Control owner | Chapter 22 |

The answer is rarely a single system. It is a single view: an index that draws open items from each source and presents them with consistent fields. Consolidate duplicate entries into one underlying open item while retaining links to the source records, responsibilities, and work already completed. The source lists remain where the work happens. The view lets the program ask questions no single list can answer, such as which owners carry the most overdue items or which repositories appear on several lists.

Items affecting the same repository are not necessarily duplicates. Combine entries only where they describe the same underlying gap.

Consistent fields matter more than consistent tools. Apply the same minimum to every item: description, location and source reference, current owner, due date, status, verification method, and exposure rating. Distinguish resolution dates from next-action dates and risk-acceptance review dates so readers know what is due. Missing owners or dates are accountability deficiencies to flag against the existing item and count, not an automatic reason to create another entry.

## Accepted Is Not Closed

A documented risk acceptance may be an appropriate response to a gap: a provider may decline an evidence term, a control may not be corrected within the year, or a rule exception may be approved. The acceptance needs a rationale, an authorized approver, an expiry date, and any earlier review date. A scheduled review does not extend an expired acceptance without a new authorized decision.

For reporting purposes, distinguish approval of a risk acceptance from resolution of the underlying gap. The source workflow may record the acceptance decision as complete, but the open-items view should keep the accepted exposure visible while the condition remains. Show accepted items as a separate class, with their approver and expiry date, and flag acceptances approaching or past expiry.

An expired acceptance that receives no review leaves the organization carrying exposure without a current documented decision. The item remains open and requires review. Renewal needs documented approval rather than an automatic extension, and it does not remove the underlying exposure from the view.

## Different Views for Different Decisions

The same open items support different decisions at different levels, and a single report rarely serves all of them well.

- **The owner view.** Each owner sees their own items, the actions or reviews due, and what is blocking them. This is the view that moves work, and it should be available at any time, not only when a report is produced.
- **The program view.** The program sees all items across sources, with aging, exposure, and blockages. This is where duplicate entries are consolidated, priorities are set, and dependencies between owners are identified.
- **The committee view.** Leadership sees the items that need its decision or support, the aging of high-exposure items, accepted risks approaching or past expiry, and changes across reporting periods.

Committee reporting is more useful when it uses a format leadership already understands and works from. Audit and risk committees are used to reviewing findings with owners, target dates, closure aging, and risk acceptances. Governance reporting in that form can be considered alongside other risks on the agenda.

Set escalation criteria by exposure, urgency, and age. Routine overdue items may move through the management chain, while items requiring an urgent decision should reach the appropriate authority without waiting for the standard aging threshold. Define the routes in advance and record the action taken. Agreed criteria make escalation consistent without preventing someone from raising a concern earlier.

## Why Gaps Stay Hidden

Most programs do not hide their gaps deliberately. The forces below are ordinary, and several of them reward reporting that looks better than the underlying position.

- **Dashboards are built from the data available.** Reporting often starts with what a system can already count, rather than with the questions leadership needs answered. The result measures activity readily and gaps rarely.
- **Status is self-reported.** Without verification, status may reflect assumptions about completion rather than evidence of the result.
- **Lists are fragmented.** Each function maintains its own list in its own tool. No one sees the total, and items that fall between functions may appear on no list at all.

- **Items leave the list without being resolved.** Accepted exposure drops out of reporting when an acceptance is approved, stalled items are reclassified, and items tied to departed owners disappear from view.
- **Visible gaps invite questions.** Showing leadership a long list of open items can feel like a risk to the program. Reporting that emphasizes progress feels safer, and it can leave the committee unaware of the items that most need its support.

The result can be a program that reports steady progress while its open exposure is unchanged or growing.

## What Works in Practice

A hotel and resort group with roughly twenty-two thousand employees operated about one hundred forty owned and managed properties in several countries. Its governance program reported quarterly to the risk committee, and for two years the report had shown steady progress: coverage rising, workstreams on track, and no items requiring escalation.

A newly appointed chief risk officer asked a simple question: what were the ten largest open retention and disposition gaps, and who owned each one? The program could not answer from its report. The information existed, but it was spread across nine lists held by records, legal, IT, procurement, and the property operations team.

The team built a single index before changing anything else. The nine lists contained one thousand four hundred thirty entries. Consolidation removed three hundred ten duplicate entries, leaving one thousand one hundred twenty distinct open items. Thirty-eight percent had no owner, or an owner who had left. Two hundred twelve were more than a year past their original due date. Of sixty-one accepted risks, forty-four had no expiry date and had been counted as closed.

The team compared reported and observed status for the forty workstream items marked complete. Eleven lacked sufficient verification records: configurations had been requested but not confirmed, or confirmed in some properties and not others. They were marked open pending verification, distinguishing incomplete evidence from a confirmed failure to perform the work.

Four changes followed. Every item received a current owner and a due date, with ongoing monitoring of items missing either and zero as the target. Status for pairings, disposition, and holds was based on verification evidence linked through the implementation matrix, disposition records, and hold register, rather than on owner updates alone.

Escalation criteria were set in advance. Items ninety days past due moved to the property’s general manager or the functional lead, and high-exposure items still unresolved at one hundred eighty days past due moved to the risk committee. The committee report was redesigned in the format already used for audit findings, with the top items by exposure, aging, accepted risks approaching expiry, and the trend.

The first redesigned report showed more open exposure than any report before it. The committee's response was to ask what the program needed, and it funded two additional roles to work the backlog. Twelve months later, open items had fallen to five hundred forty, all with owners and due dates. Items more than a year past due had fallen from two hundred twelve to nineteen, each with an escalation record. Every accepted risk carried an expiry date.

The figures and escalation timeframes in this example are illustrative, not recommended thresholds. Actual escalation criteria should reflect the exposure, urgency, and decisions required. Aging thresholds are backstops, not reasons to delay escalation of a high-exposure item or an urgent decision.

## Implementation Steps

**Build one index of open items before redesigning any report.** Draw items from every source list and consolidate entries only where they describe the same underlying gap. Retain the source links and work history. Apply consistent fields: description, location and source reference, current owner, due date, status, verification method, and exposure rating. Identify whether each date concerns resolution, the next action, or review of an acceptance.

**Flag missing owners and due dates against the existing item.** Count items without either and monitor the count on an ongoing basis, with zero as the target. Explain material changes, including newly identified gaps, and record the action taken. An item nobody owns is a gap in accountability as well as a gap in control.

**Derive status from verification evidence where it is available.** Use the implementation matrix, disposition records, hold register, and test records to link to that evidence and record when verification occurred. Label status that remains self-reported. A completed field alone does not establish that the underlying work is complete.

**Keep accepted risks visible as their own class.** Record the rationale, approver, and expiry date. Flag acceptances approaching or past expiry and require a documented decision to renew them. Keep the exposure visible until the underlying condition is resolved, regardless of how the source workflow labels the acceptance decision.

**Define escalation criteria before they are needed.** Set the thresholds by exposure, urgency, and age, identify the authority each route reaches, and record the outcome. Routine aging thresholds should not delay concerns that need an earlier decision.

**Design views by decision.** Give owners their items continuously, give the program the full view, and give the committee a short report in the format it already uses for audit and risk findings. Chapter 21 supplies the measures that report should carry.

## What Good Looks Like

In a program that uses visibility as a control, every known gap is tracked with consistent fields, an accountable owner, and a way to verify closure. Dates distinguish resolution, next actions, and acceptance reviews. Owners can see their own items at any time. The program can see the full view, including new items, stalled work, and the reasons for delay.

Status reflects verification evidence where available, with self-reported status identified. Accepted risks remain visible while the underlying exposure remains. If an acceptance expires without review, the item remains open and is flagged for a decision. Renewal requires documented approval rather than an automatic extension.

Routine escalation follows agreed criteria rather than depending on someone choosing to raise an issue. The process also allows concerns requiring immediate attention to be escalated earlier.

The clearest sign is how the program answers the question the chief risk officer asked. Fragmented lists can require a lengthy reconstruction that still leaves gaps. With a maintained single view, the program can answer directly, including the current owner and the relevant date.

## What Proves It

The evidence for visibility is the view itself and the record of what happened because of it.

- The open-items index, with links to source records, the consolidation method, and consistent fields.
- A record of items without current owners or due dates over time, with explanations for material changes and the action taken.
- Status sources for each item type, linking to verification evidence and dates and distinguishing observed from self-reported status.
- An accepted-risk view with rationales, approvers, expiry dates, and review decisions, including expired acceptances awaiting a decision.
- Escalation criteria and a log of escalations, including urgent referrals made before routine aging thresholds, with actions and outcomes.
- Committee reports across reporting periods in the format used for audit and risk findings, allowing changes in the position to be compared.

The chapter’s measures are the count of open items without a current owner or a due date and the count of items that have met an escalation criterion without a recorded escalation. Apply the criteria for exposure and urgency as well as age. The first measure identifies accountability gaps. The second shows whether escalation requirements are being followed. Explain material changes rather than treating a falling count as the only evidence of progress. Neither replaces the measures Chapter 21 develops; both indicate whether the underlying work is being managed.

## Business Value

It helps turn known gaps into managed work. Gaps that exist only in scattered lists tend to wait. Owners, dates, verification requirements, and escalation paths give the program a practical basis for action and follow-through.

It gives leadership a more complete picture. A committee that sees open exposure, aging, and acceptances approaching or past expiry can direct support where it is needed. A committee that sees only progress may learn about the gaps from an auditor, a regulator, or an incident.

It reduces duplicated effort. Consolidating entries can prevent the same gap from being tracked and worked separately by different functions while preserving the source history. The view also shows where one repository or owner is associated with several distinct items.

It makes the rest of Part IV usable. Defensible decisions, controlled change, measurement, and testing all generate open items. Visibility connects those items to owners, decisions, and evidence of follow-through rather than leaving them in separate files.

## What Gets Seen Gets Decided

Visibility does not close a single gap. What it changes is which gaps get attention, who is asked to act, and how long an item can wait before someone with authority has to decide about it.

Programs often treat reporting as the last step, a summary prepared after the work is done. It works better as part of the control itself: the mechanism that keeps every known gap in front of someone accountable for it.

A gap nobody can see is a risk the organization carries without knowing it. A gap that is visible, owned, and dated is a decision waiting to be made, and that is a position most organizations can manage.

*Next: Chapter 21 turns from visibility to measurement: which measures show that governance is working, how coverage is calculated with enough rigor to withstand scrutiny, and what sustains funding over time.*

# Chapter 21: Measuring What Matters

A measure is a promise about what a number means. It states what was counted, what it was counted against, and how someone else could count it again and reach the same result.

Many governance measures keep only part of that promise. Training completion, policies reviewed, and schedules approved are easy to count and accurate as far as they go. They describe what the program did, but say little about what happened to information as a result. Coverage figures move closer to that question, but may depend on calculations or source records that others cannot reconstruct. Their value depends on being able to explain how they were produced.

Chapter 2 introduced coverage as a first estimate and left three questions for this chapter: what counts as an operative rule, how to treat partial coverage within a system, and how to build a method that survives scrutiny. This chapter answers them, places the book's measures against the five maturity dimensions from Chapter 5, and explains how to keep a small set of measures useful and explainable. Chapter 20 addressed how gaps become visible, owned work. This chapter addresses the numbers that summarize that work.

## Activity Is Not Outcome

Activity measures count effort: sessions delivered, attestations collected, categories reviewed, meetings held. They are useful for managing a program's own work, and some are required. Their limitation is that they can rise while the underlying position stays the same. A program can review every category in the schedule during a year in which no configuration changed and no eligible information was disposed of.

Outcome measures address what changes in the information environment, such as reduced exposure or completed disposition. Measures of control implementation and testing help explain that position: how much information is subject to verified retention treatment, how far behind disposition is running, how long changes take to reach systems, and how much has been tested. These measures are useful together, but they describe different things. Implementation or test coverage does not, by itself, establish that the intended lifecycle outcome occurred.

Most programs need both activity and outcome measures. Activity measures belong primarily in the program's operating detail, although some may need leadership attention. The committee view described in Chapter 20 should lead with outcomes and the supporting measures needed to interpret them, rather than activity alone.

## What Counts as an Operative Rule

Coverage asks what share of in-scope information is subject to an operative rule. The answer depends on what operative means, and programs often count at different points without saying which.

Chapter 22 separates three questions about a control: does it exist, does it match the rule, and did it operate? Coverage is counted at the second. For each scoped treatment within a pairing, the enforcement mode is native, integrated, or manual, and its configuration or documented manual procedure must have been verified against the approved rule, including the retention trigger where applicable. Coverage does not, by itself, establish successful operation.

A rule loaded but not verified has been configured, not confirmed. A rule using a proxy retention date counts only where the difference from the approved retention trigger has been documented and authorized as an alternative treatment, and that treatment has been verified. This differs from accepting the risk that the required treatment has not been implemented. That acceptance does not, by itself, establish verified coverage. A pairing recorded as not enforced is uncovered, regardless of its target date.

Verification applies to the rule version and configuration or procedure examined. A subsequent change or finding that invalidates that match requires the coverage status to be reassessed. An operating failure is reported through the test result and open item; it should not be confused with a configuration-verification result.

Testing and execution evidence establish whether the control operated. A program can report broad configuration coverage while eligible information continues to accumulate. Retention coverage is volume-based; control test coverage is control-based. Report their units and denominators explicitly. They are not interchangeable percentages, and multiplying them does not establish the share of information subject to effective controls. Test results show what the testing found.

## Coverage in Bands

A single coverage percentage hides what the rest of the environment contains. Report in-scope volume in four bands that together account for all of it. Each unit of in-scope volume belongs in one band only.

- **Covered.** Volume in pairings with a verified operative rule, whether implemented through system configuration or a documented manual procedure.
- **Configured, not verified.** Volume in pairings where a configuration or documented manual procedure is in place but has not yet been verified against the approved rule.
- **Known, not covered.** In-scope volume in registered repositories that is not subject to a verified operative rule and is not included in the configured-but-unverified band. Missing ownership, dates, or risk decisions remain visible as management gaps; they do not remove the volume from the measure.
- **Unreconciled.** In-scope stored volume that cannot yet be traced to an entry on the repository register.

The fourth band makes a limitation in the inventory visible: storage within scope that cannot yet be attributed to a registered repository. Reporting it separately prevents coverage from appearing higher simply because unreconciled volume was left out of the denominator.

The bands also change how progress reads. As unreconciled volume is traced to the register, the known-not-covered band may grow even while the covered share increases. That is improved visibility, not necessarily remediation. The newly identified gaps can then be assigned owners, dates, and appropriate action through the process in Chapter 20.

## Partial Coverage Within a System

A whole system should not count as covered because one category has a configured rule. The rigorous method measures at the level of the pairing: one record category in one repository.

Where one pairing contains populations with different approved treatments, allocate its volume to those scoped treatments and their respective verification status. Count each unit of information once. A verified default or one verified jurisdictional treatment does not establish coverage for the other populations in that pairing.

For each repository, attribute volume to the categories it holds and count as covered only the volume subject to verified treatment within those pairings. A container default counts as covered only when it applies under an approved rule and has been verified against that rule. While an applicable configured default awaits verification, its volume belongs in the configured-but-unverified band. Unclassified content in a known repository remains in the denominator and otherwise belongs in the known-not-covered band.

Precise attribution is not always available. Where category volume within a repository can only be estimated, use a documented sample or classification proportions, label the result as estimated, and record the method. Measured, estimated, and proxy figures can appear in the same report, provided each is labeled and the share of the total resting on each is visible. State the basis and limitations of each component rather than treating the label alone as a confidence assessment.

A proxy figure uses another observable measure as a substitute where direct measurement is unavailable. Record why the substitute is appropriate and its limitations. This is distinct from a proxy retention date, which substitutes for the event used to start a retention period.

State the unit and keep it consistent. Stored bytes reflect storage volume. Item counts give each counted item equal weight, regardless of size. Either can support a trend, provided the unit and its limitations are stated. If the unit changes, restate comparable prior periods where possible or identify the break in the series.

## A Method Someone Else Can Reproduce

A coverage figure can be reproduced when an independent person, following the written method, reaches the same number from the same source information. Five practices support that standard.

- **Trace the denominator.** Reconcile the repository register and infrastructure storage records at each reporting period to establish total in-scope volume. Include volume not yet

reconciled to the register as the unreconciled band. Report approved exclusions separately, and account for each unit of in-scope volume once.

- **Trace the numerator.** Covered volume comes from pairings with verification evidence referenced in the implementation matrix, whether for a configuration or a documented manual procedure. A recorded status or owner confirmation alone does not establish verified coverage.
- **Record exclusions.** Any volume excluded from scope carries a reason and an approver, and the total excluded is reported. A change in scope can improve a figure without improving the underlying controls, so its effect should be visible.
- **Label the basis and limitations.** Mark each component as measured, estimated, or proxy. Record the method, supporting assumptions, and limitations so readers can assess what the figure supports.
- **Version the method.** When the method changes, restate prior periods where possible, or mark the break in the series so a trend is not read across it.

Retain the source extracts, reporting cutoff, verification references, and assumptions used for each reported figure. Where estimates are used, retain the sample or classification proportions behind them so the calculation can be reproduced.

Appendix C holds the single definition of coverage, with a labeled note for the first-pass version from Chapter 2. The first-pass estimate and the rigorous measure address the same question, while the reporting makes their different levels of verification and precision clear.

## Measures Mapped to the Maturity Dimensions

Chapter 5 argued against composite maturity scores, because averaging across dimensions hides the gap between them. The same applies to measures. A scorecard that combines coverage, test results, and training completion into a single index conceals which dimension needs attention.

The book has introduced measures in two tiers. Appendix C holds a small library: two anchor measures and three supporting measures. Chapter-level measures and diagnostics support the work where it happens without automatically becoming standing committee measures. Figure 21.1 maps the principal measures developed through this part to the five dimensions; their definitions and limits remain in the owning chapters.

*Figure 21.1. Measures by maturity dimension.*

| Dimension | Library measures (Appendix C) | Chapter-level measures and diagnostics |
|---|---|---|
| **Structure** | Trigger coverage (Chapter 7) | Unexplained periods (Chapter 18); inherited default count (Chapter 16) |

| | | |
|---|---|---|
| **Execution** | Coverage, anchor (Chapters 2 and 21); configuration lag (Chapter 11) | Effective retention spread (Chapter 12); disposition backlog and overdue age (Chapter 13); encumbrance load and oldest open matter (Chapter 15); decision lag and requirement-driven changes approved after the requirement took effect (Chapter 19) |
| **Ownership** | Unowned repository count (Chapter 8) | Undecided legacy repository count and oldest undecided entry (Chapter 14); open items without an owner or due date, and missed escalations (Chapter 20) |
| **Visibility** | Sensitive record count, anchor (Chapter 4) | Unreconciled volume, reported as a coverage band (Chapter 21) |
| **Evidence** | None in the library | External pairings without contractual evidence obligations, read with renewal dates (Chapter 17); classification agreement (Chapters 9 and 22); control test coverage and test exception aging (Chapter 22). Testing measures and methods remain in Appendix G. |

The library is small on purpose. Every standing measure needs a written method, an owner, a data source, and a reporting cadence, and each one added competes for the committee's attention. This book keeps the standing library to five measures. Chapter-level measures support specific operational decisions without automatically becoming standing committee measures.

Evidence measures remain with the testing program in Appendix G. Chapter 22 defines control test coverage and test exception aging, including how to distinguish the extent of testing from the results it produces.

Read these diagnostics in their stated scope. Effective retention spread accounts for the instance's role, trigger, and approved outcome. Encumbrance load remains separate from the unencumbered disposition backlog. The third-party count describes an evidence-obligation gap, not proof of control failure.

## Pair Each Measure With Its Counterweight

Measures change behavior. A team reporting coverage has a reason to narrow scope. A team reporting sensitive record count has a reason to reclassify. A team reporting backlog volume has a reason to clear easy populations and leave hard ones. None of this requires bad intent. It follows from what gets praised.

Report each measure alongside companion information that can help reveal its most likely distortion. These counterweights provide context; they are not complete safeguards against every way a figure can be misunderstood.

- Coverage alongside excluded volume and the unreconciled band.

- Sensitive record count alongside the number of repositories scanned, so readers can assess whether scanning scope has changed.
- Disposition backlog volume alongside the age of the oldest overdue item.
- Configuration lag as a distribution, with the longest open item.
- Control test coverage alongside test exception aging.

Repository count helps explain scanning scope, but it does not show whether the sensitivity criteria changed. State the repository population and inclusion criteria used for the sensitive-record count, including any focus on aged information or unclear lifecycle ownership. Report material changes to those criteria, classification criteria, or scanning scope alongside the count.

These companion measures help distinguish improved governance from changes in scope, classification, or the work selected for completion.

## Measures That Support Funding Decisions

Sustaining funding requires more than the original business case. Leadership needs evidence of what the investment has changed and how that relates to cost, exposure, and operational performance.

Three connections are worth maintaining. Coverage and disposition measures can support the cost model maintained with finance, where actual storage reductions or reduced discovery scope are demonstrated. Sensitive record count informs discussions of security posture and incident scope. The time needed to produce evidence for a traced rule connects to the cost of responding to audits, inquiries, and disputes. None needs to be a standing library measure. Refresh these connections often enough to support the funding conversation developed in Chapter 28.

Use the measures to support the connection, not assume it. A higher coverage percentage does not itself establish that information was disposed of or costs were reduced. Those outcomes need their own supporting evidence.

## Why Measurement Stays Weak

Most programs know their measures could be stronger. Several ordinary forces keep them where they are.

- **Activity data is easier to obtain.** Training systems and workflow tools count activity automatically. Measures of implementation and outcomes may require a register, a matrix, and reconciliation, which take work before the first number exists.
- **A stronger method can change the baseline.** A more rigorous method may produce a lower coverage figure than the estimate it replaces. That can reflect a more complete denominator or stricter verification, rather than deterioration in the underlying controls. The report should explain the difference.
- **Methods live with individuals.** The person who built the calculation knows its assumptions. When that person moves on, the next figure is calculated differently, and the

trend breaks without anyone noticing.

• **Measures accumulate.** Each initiative adds a metric, few are retired, and the scorecard grows until nothing on it stands out.

• **Composites are requested.** Leadership often asks for a single rating. Supplying one is easier than explaining why the dimensions should be read separately.

## What Works in Practice

An agricultural cooperative with roughly three thousand two hundred employees ran grain handling, crop inputs, and farm supply operations across a large rural region, owned by its member farmers. Its governance program reported coverage of seventy-two percent to the board's audit committee, and the figure had risen steadily for two years.

Internal audit was asked to validate the figure and could not reproduce it. The denominator came from infrastructure storage totals two years old. The numerator counted every system with any configured retention policy as fully covered, including an enterprise resource planning system where two of fourteen categories had rules. The person who had built the calculation had since moved to another role.

The team rebuilt the measure at the pairing level. It reconciled the repository register with current storage reports to establish the full in-scope denominator, including volume not yet traced to a registered repository. Verification records referenced by the implementation matrix supported the numerator. Forty-one percent of in-scope volume was covered by a verified operative rule. Fourteen percent was configured but not verified. Twenty-seven percent sat in known repositories without verified coverage. Eighteen percent could not be traced to an entry on the register.

The forty-one percent figure became the baseline for the revised method. The report distinguished that methodological change from changes in the underlying controls, rather than presenting the difference from seventy-two percent as an operational decline.

The team expected the committee to focus on the lower coverage figure. The committee focused instead on the method. It was documented, an internal auditor had reproduced it independently, and it showed where the remaining volume sat. The committee asked for the four bands each quarter and for a trend using the revised baseline.

Three further changes followed. Exclusions were listed with a reason and an approver, and the total excluded was reported with the bands. Activity measures, including training completion and categories reviewed, moved from the front page to the program's operating detail. And each measure on the scorecard was reported beside its counterweight.

Four quarters later, the covered share of in-scope volume stood at fifty-eight percent. The configured-but-unverified share had fallen to six percent and the unreconciled share to seven. The known-not-covered share had risen to twenty-nine percent because much of the storage traced to the register during the year was not covered by a verified rule. The committee read that increase as

improved visibility: gaps that had been invisible now had owners and dates. These percentages described the distribution of in-scope volume, not changes in the absolute volume of each band.

The figures in this example are illustrative, not benchmarks. The purpose is to show how a consistent method can distinguish verified coverage, known gaps, and unresolved scope.

## Implementation Steps

**Separate activity from outcome.** Keep activity measures primarily in the program's operating detail and lead leadership reporting with outcomes and the supporting measures needed to interpret them.

**Define operative at the verified tier.** Count volume as covered only where its configuration or documented manual procedure has been verified against the applicable approved treatment, including the retention trigger where applicable. Report configured-but-unverified volume separately. Use control testing and its results to establish whether the control operated successfully.

**Report coverage in four bands.** Covered, configured but not verified, known but not covered, and unreconciled must account for all in-scope volume once. Missing owners or dates do not remove known gaps from the measure.

**Measure at the pairing level.** Attribute volume within repositories to categories and any distinct scoped treatments, count container defaults as covered only when approved and verified, and keep unclassified content in the denominator. Label estimates and proxy figures with their basis, method, and limitations.

**Write the method so someone else can reproduce it.** Reconcile the register and infrastructure records for the denominator, trace the numerator to verification evidence, record exclusions with approvers, and version the method. Retain the source extracts, reporting cutoff, verification references, and assumptions. Ask someone independent to reproduce the figure at least once.

**Report by dimension, with counterweights.** Use the Appendix C library and chapter-level measures, pairing each with the information that helps explain it. Present the dimensions separately and explain where a composite score would conceal a gap leadership needs to see.

## What Good Looks Like

In a program that measures what matters, the front page of the governance report leads with outcomes and supporting measures by dimension, each beside companion information that helps explain the result. Coverage appears in bands that account for all in-scope volume, including what cannot yet be traced to the register.

Every figure has a written method, a data source, and an owner. Estimates and proxy figures are labeled with their limitations. Source records and reporting cutoffs are retained. When the method changes, the report identifies the change and any break in the series. Measures are few enough that the committee can see which moved and ask why.

The clearest sign is what happens when the person who built the measures leaves. In a program whose methods live with individuals, the next figure is calculated differently and the trend quietly breaks. In a program with documented methods, the next person follows the method and the series continues.

## What Proves It

The evidence for measurement is the method and the record that it was applied consistently.

- A written method for each library measure, versioned, with its data sources, unit, exclusions, and the basis and limitations of measured, estimated, or proxy components.
- A reconciliation between the repository register and infrastructure storage for each reporting period, supported by retained source extracts, the reporting cutoff, verification references, and the assumptions or samples used.
- An exclusions list with reasons, approvers, and the total excluded volume.
- Coverage reported in four bands across comparable reporting periods, with restatements or breaks in the series identified where the method changed.
- A record of at least one independent reproduction of the coverage figure.
- A governance scorecard in the Appendix C form, reported by dimension without a composite, with each measure beside its counterweight.

The chapter-level measure is the unreconciled band: the share of in-scope stored volume that cannot be traced to the repository register. It sits within coverage rather than beside it. Reducing unreconciled volume improves visibility into the denominator. The reliability of the remaining figures still depends on their sources, verification, and method.

## Business Value

It gives leadership numbers it can rely on. A figure that an auditor can reproduce is worth more than a higher one that cannot be explained. It also protects the program's credibility, which is harder to rebuild than any metric.

It helps target investment. Reporting by dimension and coverage band shows where gaps are concentrated and provides a clearer basis for prioritizing the work.

It keeps the trend intact through turnover. Documented methods let the series continue when people change roles, so the program can show progress over years rather than restarting its baseline.

It connects governance to the terms leadership uses. Linking measures to demonstrated changes in cost, security posture, and response times gives funding discussions evidence rather than assertion.

## A Number Someone Else Can Reproduce

The strongest governance measures are rarely the most impressive ones. They are the ones that mean the same thing each quarter, can be traced to the records behind them, and can be recalculated by someone who was not involved in producing them.

A stronger method may change the reported baseline. What matters is that the organization can explain the change and use the resulting figures to decide where to invest, show leadership what has changed, and answer an auditor's question without rebuilding the calculation.

Measurement alone does not improve a program. It provides a more reliable account of what is changing, where gaps remain, and what decisions need to follow. Improvement still depends on the action taken.

*Next: Chapter 22 turns from measurement to testing: how the program confirms that a control operated, and what that confirmation leaves behind as proof.*

# Chapter 22: Test the Control, Not the Documentation

A reviewer asks one question: show us that records in this category were disposed of when the schedule required it.

The program responds quickly with the policy, the schedule showing the period, trigger, and citation, the approval record for the most recent change, and an exported configuration from the platform where the category lives. Every item is accurate and current.

Those records establish the approved treatment and the control configured to carry it out. They do not establish that the required disposition occurred.

Chapter 18 asked whether the retention decision was reasonable and documented. This chapter addresses the other question: whether the organization carried it out. Chapter 11 established how approved rules reach systems. Testing now examines whether those controls operated on real information and what the available evidence supports.

## Three Questions About Every Control

Testing should distinguish three questions so that each finding identifies what needs attention.

- **Does the control exist?** Something has been configured, or a manual procedure has been written and assigned.
- **Does it match the rule?** The configuration or documented manual procedure corresponds to the applicable approved treatment, including any required retention trigger and lifecycle action. This is the verification step developed in Chapter 11.
- **Did it operate?** The control acted on real information during the period under review, and the evidence supports a conclusion about whether it operated as intended.

All three questions require evidence. The first two establish whether the control exists and reflects the approved rule. The third establishes whether it operated as intended during the period examined. Evidence answering one question should not be presented as answering all three.

A missing control is a gap. A control that does not match the approved rule is a configuration or procedure defect. Where execution evidence is missing, operation cannot be established for the period examined. The control may have operated, failed, or operated only partly. Further evidence is needed before distinguishing among those outcomes.

The five maturity dimensions in Chapter 5 remain distinct: structure, execution, ownership, visibility, and evidence. Testing helps establish what happened without assuming that strength in one dimension demonstrates strength in the others.

The test follows the approved outcome. Permanent retention calls for evidence of continuing preservation, accessibility, and custody arrangements, not a search expecting the information to be

absent. Transfer, anonymization, archival, and destruction likewise require criteria appropriate to their approved treatment.

## Test Whether Eligible Information Remains

Evidence of intent explains what the organization decided and why: the policy, schedule, and approval record. Configuration records show how that treatment was implemented. Evidence of effect describes what happened to information, through execution records such as disposition logs, provider certificates, and verified hold-release records. These sources answer different questions and should be read together.

An absence test checks whether information that should no longer remain is still present within the defined search scope. A zero-result search is useful evidence within that scope, but it does not independently establish when or how disposition occurred. Read it alongside execution records and document the limits of the search. Use other tests where the approved outcome is continued retention, transfer, or another state not demonstrated by absence alone.

Define the applicable rule version, required retention trigger, any approved transition or fallback, eligibility boundary, and expected disposition state. Account for relevant holds and approved suspensions. Use the terminal state and residual period defined for the repository, as Chapter 13 describes, so recoverable content is assessed against the approved treatment rather than assumed to have been destroyed.

Identify all relevant category and repository pairings for the selected category. Test each repository within scope and record any locations that could not be examined. A search covering three of five repositories is a partial test, and its conclusion must remain limited to that scope.

Testing before an external request gives the organization an opportunity to identify and address unexplained retention or missing evidence on its own timetable.

## You Cannot Sample What You Cannot Enumerate

Selecting twelve categories and checking their configurations can establish whether those configurations match the rules. It does not establish operation. Testing operation requires examining the information the control should have acted on during a defined period, which first requires a population that can be identified.

Use the repository register and implementation matrix to identify the controls and locations in scope. Establish the information population from relevant source records, eligibility records, or execution records for the period examined. Retain the basis used to select the sample. Sampling only information that remains today cannot establish what happened to everything that should have been processed earlier.

An incomplete or unreliable population is a scope limitation. Record it and address it before drawing conclusions about the full population. Testing the information available may still be useful, provided

the result states what was and was not examined.

Select at the pairing level, because the same rule can operate differently across repositories. Identify each scoped treatment examined within a pairing. Weight selection by exposure and include controls expected to pass. Record the population, selection method, sample size, and date. Conclusions should reflect the population and method examined; a small, deliberately selected sample does not establish the condition of every control.

Classification agreement tests an input the other tests depend on. Compare assigned categories against a qualified reviewer's judgment using a documented sample. A control can exist, match its rule, and operate correctly while acting on information categorized incorrectly. This test remains with the testing material in Appendix G rather than becoming another measure in the standing library.

## What Makes a Manual Control Testable

A manual control can be effective when it is documented, consistently performed, and supported by retained evidence. A signed statement that a review occurred provides limited information about its scope or result.

A useful attestation identifies four things: the population reviewed and how it was identified, the work performed, the exceptions found, and the action taken. Where no exceptions are reported, the record should still make the scope and method clear. Testing can then examine the review and its supporting records rather than relying on the signature alone.

## Why Testing Does Not Happen

The barriers to testing are not only technical. Ownership, skills, evidence availability, and incentives also affect whether it happens.

- Findings create work before funding is available. The prospect of identifying gaps without resources to address them can discourage testing or delay its expansion.
- Reviewing a schedule and testing a control require different skills. Governance teams may have a well-developed review process but limited assurance experience. Chapter 31 returns to the skills question.
- Execution evidence is not retained. Logs or supporting records may expire under platform defaults before anyone examines the period they describe.
- Testing relies only on control owners. Self-assessment is useful, but its source and limitations should be clear. Independent testing provides a different level of challenge.
- Internal audit is treated as the program's testing function. Audit works to its own plan and criteria. The program still needs a process for testing and monitoring its controls between audits.

These conditions can leave a program able to describe its controls but unable to establish how they operated during the period in question.

## What Works in Practice

A mining and metals company with roughly six thousand employees and operations in several countries ran a mature assurance program, though not for information. Safety and environmental compliance had sampling plans, findings registers, corrective actions with owners and closure evidence, and quarterly reporting to a board committee.

An external review of a joint venture then asked for evidence that a category of operational and personnel records had been disposed of on schedule. The program produced the schedule, the policy, and configuration exports. The reviewer asked for a disposition log. For that platform, none had been retained.

The team adapted the assurance model the business already used rather than creating another structure. It retained three tiers: annual self-assessment by control owners, independent testing by a small assurance function on a risk-weighted cycle, and internal audit on its own plan.

The unit tested was the pairing from the implementation matrix. Of roughly three hundred pairings in scope, forty were selected for independent testing in the first year, weighted by exposure and including six expected to pass. Three test types were defined: configuration match, operation, and absence.

Results were reported separately by test type. Configuration match failed in seven pairings. Operation could not be established in fifteen because the necessary execution records had not been retained. Absence testing found over-retained populations in nine. These counts were not combined into a total of distinct affected pairings.

The fifteen evidence gaps were not treated as proof of successful operation or control failure. The team assigned work to enable logging, define where the logs would be stored, and give the evidence an approved retention treatment. In this example, that work cost substantially less than rebuilding the controls. Most evidence-generation issues were addressed within two quarters, with subsequent cycles tested to verify that evidence was being produced and retained. The missing historical evidence remained a documented limitation; later testing did not recreate it.

Three pairings could not be corrected within the year. Each received a documented risk acceptance with an expiry date and an approver accountable for the exposure. The underlying gaps remained visible, and the acceptances required review rather than automatic renewal.

One finding led to a review of the rule. A high-volume category required review before destruction, but that review had never been performed at volume. Through change control, the team assessed the applicable requirements and obtained approval to move to destruction with sampled review. The approved change included a transition statement and verification of the revised treatment. Operational difficulty prompted the review; it did not by itself justify changing the requirement.

Findings and test exception aging joined the quarterly assurance report already going to the board committee. In the second year, the team repeated eleven first-year tests and compared their results,

providing evidence of changes within the scope examined.

The figures and timeframes in this example are illustrative, not benchmarks. Testing scope, frequency, and remediation priorities should reflect the controls and exposure being examined.

## What a Control Test Record Holds

A consistent test record makes the scope, method, evidence, and conclusion available for review and comparison. Figure 22.1 uses the control test record held in Appendix G.

*Figure 22.1. The control test record.*

| Field | What it answers |
|---|---|
| **Control tested** | The category and repository pairing, its implementation-matrix reference, and the scoped treatments examined. |
| **Rule version and effective date** | The applicable approved rule version, effective date, and any transition or fallback governing the population examined. |
| **Test type** | Configuration match, operation, or absence; identify or link classification-agreement and reconstruction tests. |
| **Test criterion and procedure** | The expected result and the steps used to examine it. |
| **Population and selection** | The population source, reporting cutoff, sample size, selection method, and scope limitations. |
| **Tester and assurance tier** | Who tested, their relationship to the control, and whether the work was self-assessment, independent testing, or audit. |
| **Test date and period covered** | When the test ran and which operating period it examined. |
| **Result** | The observation, supporting evidence, limitations, and conclusion against the criterion. A pass/fail summary does not replace these. |
| **Test exceptions and response** | The finding and original recorded date, chosen response, owner, action or review deadline, and closure or review evidence. Acceptance and referral are distinct from verified resolution. |
| **Evidence retained and where** | The retained artifacts, their location, and the approved retention category and treatment, including any required trigger. |

The three control questions organize the assessment. Configuration match, operation, and absence are primary test methods used to answer them. Classification-agreement and reconstruction tests

should be identified in the same record or linked to it, with their scope and results retained.

The pairing identifies a retention-control test. Chapter 25 adapts the subject and approval references for AI-system or connection testing while retaining the criteria, population, tester, timing, results, and evidence treatment. Retention-rule and pairing references remain where applicable.

Define three responses to a test exception: correct the identified problem, approve a documented risk acceptance, or refer the underlying rule for review through change control. Referral does not predetermine a rule change. Any resulting revision follows Chapter 19's process. Substantive changes require the corresponding impact assessment and transition statement. Changes affecting system behavior require verified implementation; other changes close against proportionate completion criteria.

Record the response separately from the condition being managed. Accepted exposure remains visible while it persists, as Chapter 20 describes. A referral stays linked to the open item until a decision is made and any resulting work meets its completion criteria. A completed acceptance or referral action is not evidence that the underlying gap has been resolved.

Use the control-evidence retention category discussed in Chapter 13. Assign it an approved retention treatment, including any required period and trigger, with a documented basis for how long the organization needs to demonstrate past operation and reconstruct decisions. Do not leave that treatment to platform defaults. The structural definition remains in Appendix H, with a corresponding note in Appendix C; this is a schedule requirement, not an additional testing-program artifact.

The reconstruction test connects the evidence across the process. For a stated past date, it asks what rule governed the category, why it was approved, what was configured or procedurally required, and whether the recorded outcome matched. It draws on the versioned rationale from Chapter 18, implementation history, and retained execution evidence. Where historical material is missing, identify the limitation rather than presenting a later explanation as the original record. Appendix G retains this test as part of the existing evidence structure.

## Implementation Steps

**Separate the three questions in the test design.** State whether each procedure examines existence, alignment with the approved rule, or operation. Define the expected result and retain evidence supporting the conclusion.

**Establish the information population before selecting the sample.** Use the register and matrix to identify locations and controls, then use source, eligibility, or execution records to establish the population for the period. Record incomplete scope rather than testing only what is visible and generalizing from it.

**Give control evidence an approved retention treatment.** Use a defined category and approved retention value, with any required trigger and a supporting rationale. Confirm that the required

evidence can be retained and retrieved for that period.

**Define the three responses to a test exception.** Correct the problem, accept the exposure with an approver and expiry, or refer the rule to change control. Assign an owner and date, distinguish response from resolution, and verify corrective action before closing the underlying gap.

**Define assurance tiers and coordinate with internal audit.** Agree common terminology and evidence expectations where appropriate, while preserving each function's responsibilities and independent judgment. Record who performed each test, their relationship to the control, and the assurance tier. Agreement on terminology does not guarantee identical conclusions.

**Include absence testing in the risk-based cycle.** Prioritize high-exposure categories and set a documented frequency reflecting the exposure and rate of change. Define eligibility and the expected state, test the relevant pairings, and retain the scope and limits alongside the result. Record overdue testing and the action required.

## What Good Looks Like

In a program that tests its controls, requests for evidence draw on records already produced and retained. Controls are tested within their defined risk-based cycles, with independent testing identified separately from owner self-assessment. Overdue testing and gaps in evidence are visible rather than treated as successful results.

Test exceptions have owners, dates, and documented responses. Accepted exposure stays visible and is reviewed before acceptance expires; an expired acceptance remains an open decision. Rules that cannot be performed as written are reviewed through change control, not informally revised to remove a finding.

The program can explain what was tested last quarter, what the evidence showed, what remained inconclusive, which corrective actions were verified, and what is still open. Comparisons with earlier periods identify changes in scope or method as well as changes in results.

## What Proves It

The testing program should retain evidence of its scope, methods, results, and follow-through.

- A test plan identifying controls, expected results, population sources, selection methods, assurance tiers, and the risk-based testing cycle.
- Completed test records in the form of Figure 22.1, with supporting evidence, scope limitations, and conclusions retained under the approved treatment.
- A test exception register linked to the open-items view, with owners, target dates, responses, verified closure evidence, and aging.
- Risk acceptances with rationale, accountable approvers, expiry dates, any earlier review dates, and review records, while accepted exposure remains visible.

- Absence-test and reconstruction-test records showing the scope examined, evidence obtained, limitations, and plans to address missing capabilities or records.
- Control test coverage and test exception aging across reporting periods, with the findings and method changes needed to interpret the results.

A corrected finding with a verified retest demonstrates follow-through. Consistently clean results should also be read in the context of test scope, selection method, and evidence retained. Neither a finding nor a clean result explains the condition of the whole program on its own.

The chapter-level measure is control test coverage: the share of in-scope operative controls independently tested in the preceding twelve months. Define the denominator and reporting cutoff, and count each control once. Here, operative refers to verified configuration or a verified documented manual procedure, consistent with Chapter 21, not demonstrated effectiveness.

Count completed tests with retained scope, methods, and results, regardless of whether they identified a failure. Incomplete tests remain outside the numerator. Report evidence-limited or inconclusive results separately so completed testing is not mistaken for confirmed operation. Also show testing overdue against its risk-based cycle.

Measure the age of an unresolved test exception from its initial recorded finding to the reporting date. Retain the original finding date and the verified resolution date for closed items. Track lateness against the relevant action or review deadline separately. Acceptance, referral, or a change of owner does not reset the age of the underlying unresolved condition.

Report historical evidence limitations separately from current corrective actions. Verifying that evidence is generated and retained now can close the current corrective action, but it does not recreate missing historical records. Keep the related limitation and any acceptance or review decision visible, with a link to the completed action.

A pass rate alone does not explain scope or demonstrate program effectiveness. Read control test coverage alongside test exception aging and the findings needed to interpret the result. Coverage describes verified implementation, test coverage describes the extent of testing, and results establish what the evidence supports. Control test coverage and test exception aging remain with the testing material in Appendix G, not as additions to the five-measure library.

## Business Value

Testing can identify problems before an external request exposes them. Earlier identification gives the organization more opportunity to assess the issue, assign responsibility, and manage its response. It does not change the underlying facts or remove the need to address them.

It helps scope remediation. Separating demonstrated control failures from gaps in evidence helps the organization investigate and plan the response appropriately. Missing evidence may require additional testing as well as changes to how execution records are generated and retained.

It supports decisions about resources. Findings, aging, verified corrective actions, and risk acceptances give leadership a basis for discussing priorities in the format already used for assurance and risk reporting.

It reduces reliance on assumptions. Testing gives the program a better-supported account of what worked, what did not, and what remains uncertain. Improvement depends on how the organization responds to those results.

## Proof Is Something a Control Leaves Behind

Policies, schedules, and configuration records remain necessary. They explain the approved treatment and how the control was intended to work. Execution records and testing show what happened to information within the scope examined.

The purpose is not to replace documentation with testing. It is to connect the approved rule, the implemented control, the observed result, and the evidence needed to explain that result. Where the evidence is incomplete, the limitation should be visible.

Policy establishes the requirement. Proof comes from evidence that the organization carried it out, supported by testing that makes both the result and its limits clear.

## Closing Part IV

Part IV has addressed two connected questions: was the governance decision reasonable, and did the organization carry it out? A decision record explains the requirements considered, the rationale, and the authority behind the approved treatment. It establishes the basis for examining what followed.

Change control keeps that treatment current and carries approved revisions into the environment. Visibility connects known gaps to owners, decisions, and follow-through. Neither process is complete simply because an approval or a status update has been recorded.

Measurement gives the organization a reproducible account of its position. Testing examines whether controls operated and identifies the limits of the available evidence. Together, these practices allow the program to explain its decisions, distinguish implementation from performance, and direct attention to what needs to happen next.

Part V, Governance in an AI Environment, applies this operating model to AI-enhanced classification, AI-related information, and rapidly changing technology. It examines how scope, taxonomy quality, and human review shape classification; how schedules address prompts, outputs, and other AI-related records; what evidence demonstrates governance at the system level; and how information readiness, access, provenance, privacy, security, and lifecycle controls support responsible AI adoption. The aim is to extend the existing model, not create a separate evidence structure for AI.

Next: What AI-Enhanced Classification Changes, and What It Does Not. Where pattern recognition adds value, and why scope, taxonomy quality, and human review remain part of the operating model.

# Part V: Governance in an AI Environment

# Chapter 23: What AI-Enhanced Classification Changes, and What It Does Not

AI-enhanced classification can change the scale and cost of applying a category model. It does not replace the governance decisions behind that model.

Chapter 9 described four ways the classification decision can be made: by container, by ingest rule, by user declaration, and by inference. Inference uses a classifier to assign content to categories by recognizing patterns. It can reach shared drives inherited through acquisitions, legacy stores awaiting an exit decision, and unstructured content that never passed through a governed workflow. It can make classification practical for populations that would be difficult to review item by item.

Part IV described how a governance decision is explained, kept current, made visible, measured, and tested. Part V applies that operating model to AI, beginning with classification. This chapter examines where AI-enhanced classification can add value, how scope and category quality shape its results, and why human review belongs in the operating model rather than being added after the first run.

These chapters separate four connected areas of work, not four successive stages of deployment. Define the use, assess its information, establish record requirements, and assemble the supporting evidence before relying on the system within the approved scope.

These chapters focus on classification, AI-assisted business work, system-level evidence, and retrieval connections. They do not provide a detailed treatment of model development or training-related information.

## What Changes

Inference can change how classification is performed in several useful ways.

- **Reach.** A classifier can examine content where no container maps to a business process and no ingest rule applied at creation. Backlogs can be sorted into populations for further assessment and approved action.
- **Cost per decision.** A validated classifier can reduce the effort required to classify additional items. The operating cost still includes human review, validation, and ongoing maintenance.
- **A signal of uncertainty.** Where a classifier supplies a usable confidence value, the program can use it as a routing input. The signal helps distinguish assignments needing review; it does not replace validation of their reliability.
- **Consistency.** Automated assignment can reduce variation, but that consistency needs to be demonstrated within the approved scope. Ambiguity in the category model or instructions can also affect assignments across a large population.

These benefits depend on clear distinctions and validation of how reliably they are applied.

## What Does Not Change

Several parts of the governance model remain the same when a classifier is introduced.

- **The category model.** Introducing a classifier does not resolve overlapping categories, missing exclusions, or definitions that depend on unavailable context. It is another applier, in Chapter 9's terms. Validation establishes which distinctions the classifier can support within the tested scope.
- **The retention decision.** A validated assignment helps identify the approved treatment that applies. The program still needs to establish the relevant scope, rule version, and any required retention trigger. Classification does not create a new treatment or authorize a lifecycle action.
- **Disposition authorization and checks.** Apply the approved disposition authorization and the eligibility, encumbrance, and dependency checks appropriate to the population, as Chapter 13 describes. Classifier output is an input to that process, not a substitute for it.
- **Accountability.** An owner remains accountable for each category and for the decision to rely on automated assignments. Delegating the classification step does not delegate responsibility for its result.
- **Evidence.** The program still needs to show how assignments were made and what supports reliance on them. Chapter 22 treats classification agreement as a sampling test because a control can operate correctly on information assigned to the wrong category.

A classifier operates within these elements. Its output needs an approved use, accountable owners, and evidence supporting the decisions made from it.

## Scope Before Sophistication

Define the question before evaluating the technology. A limited set of well-defined distinctions can be easier to validate and improve than hundreds of overlapping categories. The scope should reflect the information available in the content and the decisions the result will support.

Use the different-outcome test in Chapter 9 to identify consolidation candidates. Preserve distinctions needed for legal, business, access, or evidentiary reasons. Definitions, exclusions, and worked examples then become classification inputs: what belongs, what does not, and where the boundary lies. Gaps in those inputs need to be addressed rather than left for the classifier to resolve.

Narrow the first question. A first pass can identify candidate duplicate or transitory content, or sort information into a small number of category groups. Validate the broader distinctions before adding finer ones. Document how each group maps to the approved categories and treatments; grouping for classification does not itself change the schedule. A candidate label does not establish that information may be disposed of.

Identify constraints on where processing can occur before selecting an approach. Some information cannot leave the organization's environment, or can be processed only by approved people or systems. Those constraints affect which approaches are available and should be recorded with the scope. Chapter 26 develops source-readiness assessment and distinguishes controlled validation from operational connection and reliance.

## Design the Human Review

Human review is part of the operating model, not a phase that ends when the classifier is deployed. People define the scope, resolve ambiguity in definitions, review uncertain assignments, validate results, and decide when the classifier's output can be relied on. Each of those roles needs an owner and capacity.

Route assignments using validation, available confidence signals, and the consequences of reliance. Assess both premature disposition and retention beyond approved requirements. An assignment can be corrected, but action taken in reliance on it may not be reversible. Higher-consequence uses need stronger evidence and proportionate review, including individual review where required.

Where confidence values are used, define what high, moderate, or low means for the category and support the thresholds with validation. Without a usable score, use approved non-score criteria supported by validation, or route the item to review or unresolved status. Do not invent a confidence value.

*Figure 23.1. Routing automated assignments.*

| Route | When it applies | What happens | What is retained |
|---|---|---|---|
| **Assign and sample** | Meets validated, approved reliance criteria; no individual or specialist review is required | Apply the assignment and perform the specified sampling and downstream checks | Assignment record, criteria reference, and sample results |
| **Assign after review** | Falls outside automatic-assignment criteria, or the approved treatment requires individual review | A reviewer confirms or corrects the assignment before it takes effect | Proposed and applied assignment, reviewer, and correction |
| **Refer to a specialist** | Restricted, sensitive, or potentially held content, or a category requiring specialized judgment | A qualified reviewer decides with the required context and restrictions | Referral, decision, basis, and required follow-through |

| | | | |
|---|---|---|---|
| **Leave unresolved** | Insufficient confidence or context, or no suitable category | Do not force a category; review and document interim treatment and restrictions | Population and size, interim treatment, owner, and next-action date |

Mandatory specialist review and preservation action take priority over confidence-based routing. A classifier flag supports, but does not replace, the formal encumbrance check. Apply any required protection while the referral is being resolved.

An unresolved route makes uncertainty visible instead of forcing a complete-looking result. Record the population as an open item with an owner and a next-action date. Existing treatment can continue when that is the reviewed decision; otherwise, document the interim treatment and any necessary restrictions. Unresolved does not mean automatic suspension or indefinite retention.

Set reliance criteria from validation and the consequences of error. Use the pilot to estimate the review workload those criteria create. Where capacity is insufficient, narrow the scope, stage the work, or provide additional capacity. Any threshold change needs validation and approval, not merely a smaller queue.

Reviewer corrections help identify recurring confusion. Examine whether the cause lies in the classifier, the instructions, missing context, or the category definitions, and use that analysis to plan the next validation cycle.

## Validate by Category, Not by Average

A provider’s overall accuracy figure can be useful context, but it does not establish reliability for the organization’s information or intended decisions. Confidence is a routing input, agreement is an observed validation result, and reliance is the organization’s approved decision about how the output may be used.

Use the classification agreement test from Chapters 9 and 22. Draw a documented sample from the scoped population, have a qualified reviewer classify it independently, and compare that result with the classifier’s original assignment. Report final assignments after human correction separately. For each category or group, retain the number examined and agreeing, the selection method, and the basis for allocating sampled items to groups. Broad-group agreement does not validate every category distinction within those groups.

Report agreement by category or group, not only overall: large populations can conceal weaker results in a small, consequential category. Record which categories are confused. Concentrated disagreement can indicate overlapping definitions; broader disagreement may reflect missing context, classifier limitations, or an over-broad scope. Investigate rather than assume the cause.

Refine definitions, exclusions, examples, and instructions deliberately, recording what changed between cycles. Approve reliance criteria before automatic assignment, including the required

agreement and review for each category or group. Where those criteria are not met, retain individual review or an unresolved route. Agreement supports the decision within the tested scope; it does not make the decision on its own.

Each assignment needs a traceable record: the item or population reference; proposed and applied category or group; classifier and category-model versions; date; confidence where available; approved route; and any reviewer, decision, or correction. Link the system and use references and separately versioned instructions and routing criteria. For approved automatic assignments, link the applicable sample records rather than imply individual review. The record need not copy the source content, but it should connect the information, assignment, and conditions under which it was made.

## Classifiers Change

Validation applies to the scope and conditions examined. Content changes with business processes, category definitions change through Chapter 19's process, and providers update classifiers. Each can affect the basis for reliance.

Treat a classifier update as a change to how rules are applied. Assess its effect before reliance and retain the result. A prior sample provides comparison across versions; record whether its results were used to refine the classifier or instructions. Use independent resampling to support broader reliance, and state the scope of each conclusion.

Record whether a change applies only to future assignments or requires review of an already-classified population. Carry any resulting treatment change through the existing approval and verification process. Apply Chapter 17's approach to defined provider obligations by establishing the notice and evidence needed to assess classifier changes before reliance.

Resample on a defined cadence even when nothing has visibly changed. Gradual changes in agreement may otherwise remain unnoticed until an operational decision depends on the result.

The classifier is also an AI system in its own right. Chapter 25 addresses the system-level record that demonstrates governance of AI systems generally, including approval, risk classification, and monitoring. This chapter addresses the classification-specific evidence that record should draw on.

## Where It Fits

AI-enhanced classification can help assess large unstructured populations, backlogs without reliable containers, and legacy stores needing an exit decision before migration. It can identify candidate duplicate or transitory content and help locate sensitive information within a defined scope, supporting Chapter 4's exposure work.

Without approved treatment and accountable owners, results should inform assessment rather than authorize lifecycle action. Classification also needs ongoing review, not just deployment. Where

container or ingest methods already work reliably, retain them unless there is a demonstrated reason to change. Inference can address content those methods do not reach.

## Why Classification Projects Disappoint

Classification projects depend on more than pattern recognition. Scope, category design, review capacity, integration, and change management can each limit the result.

- **The initial scope is too broad.** A first run against the full schedule can produce results that are difficult to evaluate and improve.
- **Accuracy is treated as a single number.** An overall figure, or one measured on someone else's content, replaces validation by category in the organization's own environment.
- **Review capacity is not planned.** Human review is assumed rather than resourced, and thresholds are lowered to fit available reviewers rather than supported by validation.
- **The tool is treated as the program.** Classification is expected to resolve problems in the category model, ownership, or authorization that it depends on instead.
- **Outputs are disconnected from the lifecycle.** Candidate populations bypass authorization or the eligibility, encumbrance, and dependency checks, or assignments lack records that explain how they were made.
- **Changes are not managed.** Classifier updates, category changes, and shifts in content are not revalidated, and their effect on prior assignments is not assessed.

These conditions can produce a complete-looking result without enough evidence to support its intended use.

## What Works in Practice

An aerospace components supplier with roughly eight thousand employees held about thirty-eight million files on shared drives, much of it inherited through two acquisitions. Engineering, quality, program, and commercial content were mixed across folders that no longer mapped to current business processes. A migration to a new platform was planned, and the team wanted to move only what the organization needed.

Before the pilot, the team established that access-restricted technical data had to be processed within the organization's own environment. Processing was kept within that environment, and flagged files required specialist review. These constraints formed part of the pilot scope.

The provider reported encouraging accuracy on its own test set. The team then piloted the classifier against its full schedule of one hundred forty categories using its own files. In an independent review of six hundred files, agreement with the classifier was just over half. Most disagreements fell among about thirty engineering and quality categories whose definitions overlapped.

The team narrowed the question. The first pass identified candidate duplicate or transitory files and assigned other content to one of twelve category groups sharing a lifecycle outcome. The group definitions gained exclusions and worked examples drawn from the disagreements. Their

relationship to the approved categories and treatments was documented; the grouping did not itself revise the schedule. Restricted-content flags remained a specialist referral regardless of confidence.

Reliance criteria and routing thresholds were set by group from validation and the consequences of error. Where assignments would support disposition, the criteria required stronger evidence, higher confidence, and reviewed samples before the cycle. Six part-time reviewers were assigned; their capacity determined the scope and pace of the work, not the reliance criteria. Unplaced items remained unresolved with a reviewed interim treatment, an owner, and a next-action date.

After three validation cycles, agreement ranged from about seventy percent in the most difficult group to over ninety percent in the clearest. Groups below the approved level stayed in review. Of the thirty-eight million files, about nineteen million were candidate duplicate or transitory content, fifteen million were assigned to category groups, and four million remained unresolved. These were group-level results, not validation of all one hundred forty category distinctions.

The nineteen-million-file candidate population did not go straight to deletion. It was assessed under the existing disposition authorization and the applicable eligibility, encumbrance, and dependency checks. About one million files within two active holds were withheld. Roughly sixteen million were disposed of during the first year under the approved process; about two million awaited review because they did not meet the automatic-assignment criteria.

Midway through the year, the provider released a new classifier version. The team ran the prior validation sample before adopting it. Agreement fell in two groups, so the update was held until the instructions for those groups were revised and the sample rerun. That comparison showed what changed on those files; it did not establish performance across the full population.

The figures and timeframes in this example are illustrative, not benchmarks. Agreement levels, thresholds, and review capacity should reflect the organization’s categories, content, and the consequences of the decisions that depend on them.

## Implementation Steps

Prepare the category model first. Confirm definitions, exclusions, and worked examples. Use shared outcomes to identify consolidation candidates while preserving necessary legal, business, access, and evidentiary distinctions.

Narrow the first question. Start with candidate duplicate or transitory content or a defined set of groups. Record the scope, processing restrictions, and mapping to approved categories and treatments before the pilot.

Design routing by confidence and consequence. Validate the reliance criteria and any confidence thresholds. Give mandatory specialist or preservation action priority, define routing without a usable score, and give unresolved items a reviewed interim treatment, owner, and next-action date.

Size and assign review capacity before scaling beyond the pilot. Estimate workload from the pilot and name reviewers and specialists. Stage or narrow the work, or add capacity, where needed. Do not lower reliance criteria solely to fit available reviewers.

Validate by category and approve reliance. Compare original assignments with independent review, retain the sample and category-level counts, and report corrected results separately. Record the approved use, required evidence, review route, and approver.

Connect outputs to the lifecycle. Apply the approved disposition authorization and the eligibility, encumbrance, and dependency checks appropriate to the population. Retain traceable assignment records and links to the actions taken in reliance on them.

Revalidate after change and on a cadence. Use prior samples for comparison and independent resampling for broader reliance. Record whether prior assignments need review and carry resulting treatment changes through approval and verification.

## What Good Looks Like

In a program using AI-enhanced classification well, the classifier works within the category model. Definitions can be applied by reviewers, classification groups map to approved categories and treatments, and validation supports the distinctions being used.

Routing reflects validation and consequence, with confidence used where it is meaningful. Review capacity supports the approved criteria. Unresolved items have a reviewed interim treatment, an owner, and a next-action date. Disposition remains subject to authorization and the applicable eligibility, encumbrance, and dependency checks.

Reliance is recorded for each category or group and intended use, with the limits of validation stated. When content, the classifier, or the categories change, the program revalidates and decides whether prior assignments need review.

## What Proves It

The evidence should connect assignments to their validation, review, and intended use, including limitations.

- A scope record identifying the categories or groups, the question asked, processing constraints, and the relationship to approved treatments.
- Approved routing criteria, validated thresholds or non-score criteria, specialist priority, and assigned review capacity. Unresolved items have interim treatment, owners, and next-action dates.
- Validation records retaining population, selection method, reviewer, original assignments, counts examined and agreeing by category or group, corrections, and changes between cycles.

- Reliance decisions identifying the intended use, required evidence and review, limits of validation, and accountable approver.
- Assignment records linking the information to proposed and applied categories, versions, date, available confidence, routing criteria, any reviewer decisions or corrections, and linked sample records where the approved route uses sampling.
- Disposition records linking classifier-identified populations to authorization and the applicable eligibility, encumbrance, and dependency checks and verified outcomes.
- Change records identifying revalidation, sample reuse, conclusions within scope, and the approved treatment of affected prior assignments.

These records use the testing structure already in place. Validation applies the classification agreement test held with the testing material in Appendix G. Appendix D carries the questions to answer before a classification project begins.

The chapter's priority measure is classification agreement by category or category group for original automated assignments. Agreement provides evidence for reliance within the population and scope tested. Read it alongside the consequences of disagreement, the review route, and validation limits, with corrected-assignment results identified separately.

Report the share routed to human review and the unresolved share against the same scoped population and cutoff, counting each item once by its recorded primary route for that cycle. The human-review route includes individual and specialist referrals, not samples of automatically applied assignments. Report review completion separately from routing volumes and work still awaiting a decision.

## Business Value

It can make backlog decisions more practical. Classification can sort large populations for assessment where item-by-item review would be difficult. The benefit depends on the classification, review, validation, and maintenance effort required.

It can reduce migration and storage effort. Identifying candidate duplicate and transitory content can narrow what needs to move. Where approved destruction follows, it reduces retained volume; classification alone does not establish that result.

It can reduce variation in assignment. Well-defined categories provide a stronger basis for consistent application. Validation establishes whether that benefit is being achieved within the intended scope.

It supports decisions the organization can explain. Category-level validation, recorded routing, and traceable assignments help show how content was classified and why the output was relied on, consistent with Part IV.

## Automation Applies the Model It Is Given

AI-enhanced classification can extend the program's reach and make large populations more practical to assess. Its value depends on a defined scope, categories that can be applied reliably, and evidence supporting decisions made from its output.

Clear definitions provide a stronger basis for classification, but their effectiveness still needs to be tested. Ambiguity, missing context, and unsupported assumptions remain issues to resolve rather than conditions the technology can be expected to remove.

The operating model connects those tasks: define the question, validate the assignments, provide the review capacity, and record what the organization relied on. Classification then becomes a governed input to lifecycle decisions, not a substitute for them.

*Next: Chapter 24 turns from classifying existing information to the records AI creates: prompts, outputs, revised drafts, interaction logs, and system documentation, and how an existing schedule can be extended to govern them.*

# Chapter 24: Records Retention Has Not Caught Up to AI

An analyst asks an AI assistant to summarize a forty-page submission, edits the summary, and pastes it into a review memo. The memo is filed with the case record and governed by the retention schedule. The prompt, the generated summary, and the analyst's edits are held wherever the assistant keeps them, for as long as its default settings allow.

Each step is ordinary. The organization can show what it decided but may be unable to show what the system produced, what review was recorded, or how the analyst used or changed the output before relying on it.

An existing retention schedule may cover the business record without expressly addressing the prompts, outputs, revisions, logs, and system documentation surrounding an AI interaction. This chapter examines how to extend the schedule deliberately, without rebuilding it around the technology. Chapter 23 addressed AI as a way of classifying existing information. This chapter addresses the information AI use creates.

## Retention Follows Purpose and Consequence, Not Technology

AI involvement alone does not determine retention treatment. Begin with the business purpose and applicable requirements rather than a category defined solely by the technology. The consequences of the decisions the information supports also inform what needs to be retained.

A report produced with AI assistance remains governed by the category for reports of that kind. A convenience summary may qualify for transitory treatment where it has no continuing business or evidentiary purpose and no applicable requirement calls for retention. The decision follows the information's use, not whether AI or a person produced it. Retention, privacy, preservation, and deletion obligations still need to be addressed.

The existing schedule provides the starting point. Check how its categories apply to AI-generated and AI-assisted records, what needs to be retained from interactions supporting consequential decisions, and how system documentation and derived content are treated.

## The Draft Rule Does Not Always Transfer

An approved draft rule may allow working versions to be disposed of once the final record adequately captures what the organization needs to retain. Before applying that rule to an AI interaction, assess whether the generated output, the review, or changes made before reliance carry information the final record does not preserve.

An AI-generated draft records what a system produced before a person exercised judgment. The final business record may not preserve that contribution or the review that followed. The extent of editing alone does not establish whether a review was adequate, and an unchanged output does

not establish that no review occurred. Where a particular review is required, retain evidence appropriate to that requirement.

Where AI output informs a legal, regulatory, employment, financial, safety, or customer-facing determination, the organization may need to explain the system's contribution and the review before reliance. Assess the particular determination and use, not every document created within one of those functions. Applying a working-draft rule without that assessment can remove information needed to answer the question.

Those determinations provide a practical baseline for consequential uses. Use the organization's AI risk classification, discussed in Chapter 25, to refine that baseline rather than maintain a separate retention list. A use rated high risk should generally be treated as consequential here, and the baseline determinations should be reflected in the risk classification.

This does not make every AI-assisted draft a long-term record. It means the interaction's treatment is a deliberate decision about purpose, requirements, and consequence, rather than an automatic extension of a draft rule.

## Govern the Use, Not the Tool

The useful unit for this analysis is the AI use: a tool applied to a defined purpose. The same assistant may help draft internal meeting notes in one team and summarize applications that inform funding decisions in another. The tool is the same. The consequence of relying on its output is not, and neither is the appropriate treatment.

An inventory of AI uses, rather than tools alone, lets the program assess those differences. Chapter 25 addresses the broader system-level record. For retention, identify each use's purpose, supported business process and record category, consequence classification, what the tool retains and where, and the accountable owner, with stable system and use references linked through Chapter 25's record.

Drafting, research, and summarization can be incidental or consequential, depending on how the output is used. Classify the defined use and the decision it supports, not the task name alone. Both groups need deliberate treatment.

Reassess when the purpose or intended reliance changes. An interaction created for an incidental use should not keep that designation automatically when it later supports a consequential decision.

For each use, identify how treatment will be implemented through available settings, capture into a governed business system, or another documented process. Where a tool serves uses with different requirements, confirm that the arrangement supports each one. Required business-record capture must occur before the provider-held copy expires.

## Where Each Kind of Information Gets Its Treatment

The rows in Figure 24.1 describe the roles information plays, not mutually exclusive formats. A summary adopted as a business record or retained as part of a consequential interaction follows that approved treatment. A convenience derivative needs a documented relationship to its source and a defined lifecycle.

*Figure 24.1. AI-related information and where its treatment comes from.*

| Information | Examples | Where its treatment comes from |
|---|---|---|
| Generated content that becomes a business record | A report, letter, analysis, or determination adopted as the organization's record | The approved category and treatment for records of that kind |
| Interaction records for incidental uses | Prompts, outputs, and revisions supporting routine drafting or convenience | Approved transitory treatment where appropriate, subject to applicable requirements and restrictions |
| Interaction records for consequential uses | Prompt or request, input references, output, system details, date, review route, and recorded actions | The supported business record's approved treatment, including any required trigger and restrictions |
| Derived content | Summaries, extracted data, and search indexes built from existing information | Purpose and source relationship: source-linked convenience treatment or a separately approved treatment |
| System documentation | Instructions, configuration, version history, and evaluation results for an AI use | An approved documentation retention value and any required trigger, with a basis for continued availability |
| Collaboration outputs | Generated meeting summaries, transcripts, and notes | The content-type and business-purpose treatment developed in Chapter 16 |

The consequential interaction record is a central addition in this approach.

Attach it to the business record it supports and apply that record's approved retention treatment, including the applicable scope, rule version, any required trigger, and preservation restrictions. This keeps retention tied to purpose rather than creating a separate AI category with its own period.

Attachment means a maintained relationship, not necessarily another physical copy. A reference to retained source information can avoid duplication when it identifies the version used and remains usable for the required period. If an interaction supports several business records, document the relationships and applicable treatment rather than let the first filing location determine retention.

The record should identify the prompt or instructions for the interaction; source information or traceable version references; generated output; interaction date; system and available version details; the approved review route and any reviewer, review date, and recorded action; and the business record supported. Reference relevant system-level configuration separately. Record unavailable required information as a limitation rather than imply it was captured.

Where approved automation operates with sampling rather than individual review, link the assignment or batch to its applicable criteria and sample records, as Chapter 23 describes. Retain evidence of the review required and performed without implying that every output had an individual reviewer. Existing assignment or batch records can supply the interaction evidence when they contain the required information; link them rather than duplicate them.

Where a linked record also serves as control evidence, preserve its applicable evidence treatment and the usable reference, following Chapters 13 and 22. Assess both the business-record and control-evidence purposes. The end of the business record's period does not, by itself, authorize disposal of evidence retained under a separate approved rule.

Derived content can preserve information after the source instance has been disposed of. Identify the stores, what they contain, and whether they serve only the source's purpose or have a separately approved purpose. Before applying Chapter 12's convenience-copy treatment, check for unique information or context that must be retained.

Convenience derivatives should not outlast their source unless continuing retention is separately approved. Where content draws on several sources, relate it to the relevant source populations and treatments rather than assume one common end date. Register derived stores under Chapter 8 and ensure source disposition triggers the applicable action, using Chapter 13's authorization and eligibility, encumbrance, and dependency checks.

A disposition record limited to the source repository can remain accurate while a derivative continues to exist. The problem is presenting that record as evidence of removal from the entire information environment. Report the scope and explain any approved continuing retention.

System documentation also needs an operable rule. Assign an approved retention value, any required trigger, and a documented basis. Identify the system and use versions it describes and how its treatment supports interpretation of retained outputs and decisions. An update or retirement should not automatically remove documentation still needed to explain those records.

## Extend the Existing Controls

Extend the existing processes to AI uses and their information stores. Explicit scope is the starting point; the program also needs to confirm that required capture, preservation, export, and disposition can be performed.

- **Legal hold.** Identify relevant interaction records and derived stores using the matter-specific scope, pairings, and custodians described in Chapter 15. Verify that preservation reaches the required population. A notice naming an AI store does not establish that its contents were preserved.
- **Onboarding.** Apply Chapter 2's gate to AI uses and their tools. Ask what is retained, where, for how long, and whether the required capture, retention, export, preservation, and deletion are supported. Record the implementation route rather than assuming the capability exists.
- **Provider terms.** Apply Chapter 17's approach to retention, deletion, export, preservation on request, and defined evidence obligations. Establish what the provider will supply and who will review it. Contract terms and settings need to be read alongside evidence of the required actions.
- **Change control.** Use Chapter 19's process for new or amended treatments, supported by Chapter 18's decision record. State whether the change applies to future information, existing interactions and derived populations, or both. Account for system-documentation versions still needed to explain retained records.

A provider default can remain when it has been assessed and deliberately adopted. The gap is unexamined or unsupported treatment, not use of a default value. Where configuration or export cannot meet the approved requirement, record the gap, owner, next-action date, and any interim decision in Chapter 20's open-items view. Use Chapter 25's shared-finding approach where the gap affects more than one use or connection.

## Why Retention Falls Behind AI Use

The gap between AI use and the schedule tends to widen for reasons that are familiar from earlier chapters.

- **Tools are adopted faster than requirements are assessed.** Uses spread through teams and new features appear. Where changes wait for a scheduled review, information can accumulate before its treatment is considered.
- **Provider defaults become unexamined treatment.** A tool that keeps interactions for thirty days, or indefinitely, determines what remains unless the organization assesses and deliberately adopts or changes that setting.
- **The decision is left to individual users.** Without guidance, each person decides whether to save an output, and practice varies by team and tool.

- **The draft rule is applied by habit.** Generated drafts are treated as working notes regardless of how much weight the output carried.
- **Ownership falls between functions.** Records management owns the schedule, while another group may own AI governance. Without a defined connection, the two may not exchange the information needed to keep treatment current.
- **Some use is outside approved channels.** It can leave information outside the inventory and established controls, limiting the program's visibility and ability to apply approved treatment.

Two failure patterns can result. Some organizations retain everything because no one has decided what may be disposed of. Others dispose of interaction records inconsistently, tool by tool. The first can increase volume and exposure; the second can leave consequential decisions without the supporting record the organization needs.

## What Works in Practice

A private grantmaking foundation with roughly three hundred fifty staff reviewed several thousand funding applications a year. Program staff had adopted an AI assistant to summarize applications, draft review memos, and prepare correspondence. The assistant's provider retained interactions for thirty days by default. Some staff saved outputs to their own folders; most did not.

An applicant declined for a significant grant asked, through the foundation's appeal process, whether AI had been used to evaluate its proposal and whether staff had read the full application. The foundation could produce the final review memo and the committee's decision. It could not show what the assistant had produced, how the reviewer had changed it, or whether the reviewer had worked from the summary alone. The interactions had expired under the provider's default.

The team began with an inventory of uses rather than tools. They identified fourteen uses across three tools. Nine were assessed as incidental in the foundation's context, including internal-message drafting, meeting-note summaries, and general research. Five were consequential: application summaries used in review, review memo drafts, due diligence research summaries, drafts of award terms, and analysis of grantee reports informing renewal decisions.

For the five consequential uses, the team defined an interaction record: the prompt or request, the application or document provided or a traceable version reference, the generated output, the interaction date, available assistant-version details, and the reviewer's dated note of what was changed or relied on. It was saved to the grants management system with the grant or application file and took that file's applicable approved treatment.

For the nine incidental uses, the foundation's review supported transitory treatment, and provider retention was reduced to the shortest period that review found appropriate. Keeping consequential interaction records in the grants system separated their business-record treatment from the provider copies. That arrangement depended on capturing the required records before provider expiry.

Generated application summaries retained as part of consequential interaction records followed the application file's treatment. Declined applications and their corresponding summaries therefore shared the applicable end date, subject to preservation restrictions. The assistant's instructions, configuration, and version history were retained as system documentation.

The foundation amended two existing category definitions to state that AI-assisted versions of review memos and correspondence fall within them, and added one treatment for the system documentation. It did not create a separate AI category. The hold template was updated to name the assistant's stores, and the provider agreement was amended at renewal to cover export and preservation on request. Each change went through the foundation's change process with a decision record.

When a similar question arose the following year, the foundation could produce the generated summary, the recorded reviewer action, and the final business record together. That improved the record available for the later decision; it did not recover interactions that had expired before the first appeal or independently establish every aspect of the review.

The figures and timeframes in this example are illustrative, not benchmarks or recommended periods. Treatments for AI-related information should reflect the organization's requirements, the purpose of each use, and the consequence of the decisions it supports.

## Implementation Steps

Inventory uses and assess consequence. Record purpose, supported process and category, tool and information locations, and owner. Apply the baseline determinations and the organization's AI risk classification, and reassess when purpose or intended reliance changes.

Confirm how existing categories apply. Amend definitions where needed so AI-generated and AI-assisted business records fall within the categories governing records of that kind. Classify by purpose, not by a task label or technology alone.

Define consequential interaction records and their capture. Specify the contents and relationship to the supported business record, including shared interactions, the applicable review route, and usable version references. Apply the full approved treatment, not merely the same duration, and capture required records before provider copies expire.

Set deliberate treatment for incidental interactions. Use transitory treatment where the assessment supports it. An assessed provider default may remain; an unexamined default is not an approved decision. Confirm that the implementation arrangement supports each use sharing the tool.

Govern derived content by purpose and source relationship. Register stores and relate their contents to the applicable source populations and treatments. Distinguish convenience derivatives from separately governed records, and verify the approved action when source disposition affects them.

Give system documentation an approved retention rule. Specify the retention value, any required trigger, and rationale. Identify the system and use versions described, and confirm that documentation needed to explain retained records remains retrievable after updates or retirement.

Extend and verify the existing controls. Update hold scoping, the Appendix F onboarding questions, provider terms, and change control. Verify capture, preservation, and disposition within a stated scope, using Chapter 22's approach. Track unresolved capability or provider gaps through Chapter 20's open-items view.

## What Good Looks Like

In a program that has extended its schedule to AI use, each known use has a recorded treatment and an implementation route. AI-assisted business records follow their applicable categories. Consequential interactions are linked to the records they support and can be produced with them, including the approved review route, recorded actions, and any limitations.

Incidental interactions receive transitory treatment where appropriate. Convenience derivatives follow source-linked treatment, while separately governed derivatives have an explained basis. System documentation follows an approved rule and remains retrievable for the required period. Shared tools support the different uses they serve, and unresolved gaps remain visible with owners and dates.

The clearest sign is the response to a question about a consequential decision. The organization can identify the use, produce the system's output and evidence of its approved review route, and explain the retention treatment. It distinguishes what those records establish from what remains uncertain.

## What Proves It

Use Chapter 22's testing structure to connect approved treatment, implementation, and observed results, with scope and limitations stated.

- An inventory of AI uses, with purpose, supported category, consequence classification, information locations, owner, and implementation route.
- Decision records for category amendments and treatments, using Appendix F. Include the approved system-documentation retention value, any required trigger, rationale, and evidence that relevant versions can be retrieved.
- Defined consequential interaction records, with sampled evidence of capture and retrieval alongside the supported business records, including the approved route, review actions or linked sample records, and limitations.
- Provider settings and capture arrangements mapped to approved use-level treatments, with configuration-verification records and separate evidence of operation within the scope tested.

- Derived stores in the repository register, linked to source populations and approved treatments. Disposition records identify the action taken, any separately approved continuing retention, and the verified outcome within scope.
- Hold scoping and implementation records identifying relevant AI information and showing whether preservation reached the required population, with gaps recorded.
- Provider terms and evidence covering retention, deletion, export, and preservation, or recorded gaps with owners, next-action dates, and any interim decisions. Acceptance does not itself resolve a capability gap.

The chapter's priority measure is the count of in-scope AI uses without an approved interaction-record treatment. Read it alongside the count of consequential uses whose required capture is absent, incomplete, or not yet verified. The first identifies decision gaps; the second identifies gaps in implementation or its supporting evidence.

State the known inventory scope and reporting cutoff. Show the first count beside all in-scope uses and the second beside consequential uses requiring capture. Distinguish absent, partial, and unverified capture; a documented process or one successful example does not establish complete implementation.

Count each use once within each measure. The populations may overlap, so do not add the counts together. An assessed, approved default is not an undecided treatment. Newly identified uses can increase the counts without deterioration in existing controls; explain those changes. Neither measure accounts for uses not yet identified.

## Business Value

It supports explanation of consequential decisions. The interaction record connects the system's contribution, the applicable review evidence, and the final business record. Together, they provide context while making the limits of the retained evidence clear.

It can limit unnecessary retention. Where shorter approved treatment for incidental interactions is implemented and disposition occurs, it can reduce accumulated interaction records.

It helps prevent derived stores from escaping lifecycle controls. Source relationships, approved treatment, and verified actions make continuing retention visible rather than allowing summaries, extractions, and indexes to persist without an explained basis.

It extends the existing program rather than creating a parallel one. Working through existing categories, holds, onboarding, provider terms, and change control lets the organization govern AI information with processes it already operates.

## Purpose Still Determines Treatment

AI use can create interaction records, derived content, and system documentation that an existing schedule does not expressly address. The retention decision still begins with purpose and applicable

requirements.

Where an interaction supports a consequential decision, retain the record needed to explain the system's contribution and the review evidence required for the approved route. Where information has no continuing purpose or applicable retention requirement, give it deliberate treatment rather than allowing accumulation by default.

The work is to connect those decisions to the business records, tools, stores, and providers involved, then verify that the approved treatment operates.

*Next: Chapter 25 turns from the information AI use creates to the evidence that AI systems are governed: the inventories, risk decisions, testing, approvals, monitoring, and ownership needed to move from AI principles to proof.*

# Chapter 25: From AI Principles to Evidence

An organization can describe its AI principles in detail and still struggle to show how they were applied to a particular system on a particular date.

An acceptable use policy, a responsible AI statement, and a committee discussing risk provide a starting point. Questions following an audit, inquiry, complaint, or dispute are more specific. Was this system reviewed before deployment? Who approved its uses, on what basis? What information does it use? What was tested and monitored, and did the required review occur before reliance? Answering requires records about the system and use in question.

Chapter 23 addressed AI used to classify information, and Chapter 24 addressed the information AI use creates. This chapter addresses system-level governance evidence. Part IV's decision records, change control, open items, measurement, and testing provide the structure. The gap addressed here is connecting those records to each system and its uses, with someone responsible for maintaining the connection.

## Principles Describe Intent

An AI policy states what the organization expects. A responsible AI statement states what it values. General committee discussion can show that risk received attention. These program-level records do not, by themselves, establish what happened with a particular system.

Minutes can provide evidence of a system-specific decision when they identify what was approved, by whom, on what basis, and subject to which conditions. They still do not establish that the resulting controls operated. A policy requiring human review does not show that review occurred for the outputs in question. A provider's published evaluation does not establish performance on the organization's information.

Chapter 18 distinguished the basis for a decision from evidence that it was carried out. AI governance needs both: a record explaining the approved uses and conditions, and records showing what happened in operation. The value depends on what each record establishes, not its document type.

## The System and Its Uses

The system organizes the evidence, with defined uses within it. Build on Chapter 24's use inventory: identify the system, its uses and approval status, information sources, provider where applicable, and accountable owners. State what constitutes one system entry, assign it a stable reference, and link embedded features and uses to it. A product name alone does not define the entry.

Use stable references to connect the system, its defined uses, the source connections, and the assignments or interactions being evidenced. These can be related views of existing records;

approval or evidence for one scope does not automatically extend to another.

Include AI features and uses within the program's defined scope, including those added to existing platforms. Chapter 16 described how new features can appear without a separate adoption project. Assessment determines the depth of the record; a brief record for an incidental use is different from leaving that use unidentified.

Reconcile the inventory against procurement records, platform release notes, and the repository register, applying Chapter 8's approach. Check discrepancies rather than assume each discovery bypassed intake. Completing an assessment closes that task, not any approval, configuration, or evidence gaps it identifies. Keep those gaps linked as open items until their own completion or review criteria are met.

## Risk Classification Sets the Depth

Risk classification allows scrutiny to reflect consequence while keeping lower-risk records brief. Record the classification and requirements for each use, and explain how they determine shared-control oversight and the system's reporting tier. Approval or testing of one use does not establish approval or performance for every other use. Reference shared evidence without duplicating it.

Incorporate Chapter 24's baseline: uses informing legal, regulatory, employment, financial, safety, or customer-facing determinations belong at least in a tier requiring review evidence and retained interaction records. Assess the particular use and reliance. The organization may add criteria such as information sensitivity, automation, people affected, and reversibility. A common risk classification helps align retention and AI governance decisions; its documented application still matters.

Risk classification is itself a decision. Use the Appendix F decision record to retain its basis, information considered, approver's authority, approved scope, and review point. Review changes in purpose, information, or automation through Chapter 19's process.

Tiering scales the depth, frequency, and independence of the work. Use-specific requirements and stated reliance criteria still apply. Risk tier determines scrutiny; Chapter 22's assurance tier identifies who performed the work. The two are not interchangeable.

Every tier retains the applicable approval basis, authorized scope, review point, and evidence requirements. The figure shows how depth and scrutiny vary; a shorter cell does not remove a common or use-specific requirement.

The organization defines its tiers, criteria, and thresholds. Figure 25.1 illustrates rather than prescribes requirements. Consequential uses are at least middle tier here. Use-specific requirements, including Chapter 23's classifier revalidation, remain applicable at every tier.

*Figure 25.1. How the per-system record can scale by risk tier.*

| Record element | Lower tier | Middle tier | Highest tier |
|---|---|---|---|
| Approval | Brief basis, scope, authorized approval, and review point | Decision record with basis, authority, conditions, and review point | Full decision record; approval at the designated high-risk level |
| Testing before operational use | Configuration or procedure checked; use-specific evidence supports reliance | Verified implementation; evaluation in the organization's environment | Verified implementation; use-level evaluation; approved reliance criteria and results |
| Testing in operation | Proportionate testing cycle and relevant changes | Defined testing cycle and relevant changes | Shorter testing cycle and relevant changes, including independent testing |
| Human review evidence | No consequential uses in this tier; retain any other required review evidence | Approved review route and timing for reliance; evidence linked to the business record | Approved route and timing; criteria-based sample checks; evidence linked to the business record |
| Monitoring | Owner review against defined criteria on the assigned cadence | Defined measures reported on a cadence | Defined measures, companion information, and escalation criteria |
| Record review | Longer defined interval and relevant changes | Intermediate interval and relevant changes | Shortest interval and relevant changes |

## Reuse the Part IV Structure

Existing processes provide a structure for the evidence. Assessments, tests, and approval conditions still need to fit the AI use. Figure 25.2 maps eight evidence elements to their source processes.

Reuse the decision and testing structure, adapting the subject and references to the decision or control examined. An AI-use decision identifies the system, use, approved conditions, and relevant versions. The six decision elements from Chapter 18 remain useful, but its three retention bases apply only where a retention treatment is being decided. An AI-use risk classification is a different decision.

A connection test identifies the source population, connection, criteria, and approval conditions examined, using Chapter 22's test discipline. A system-to-source connection is not a category and repository pairing. Keep retention-category and rule references where applicable, and link shared evidence without extending its conclusion beyond the scope tested.

*Figure 25.2. AI governance evidence and the structures that produce it.*

| Evidence needed | What it shows | Structure that produces it |
|---|---|---|
| Intake record | When and how the system or use was identified, what was assessed, and any original-intake gap | The onboarding gate (Chapter 2), with its AI questions |
| Approval and risk classification | Approved uses and conditions, basis, authority, tier, actual decision date, and review point | The decision record (Chapter 18, Appendix F) |
| Change record | Assessment, decisions, and transition for versions, uses, information sources, and provider updates | Change control (Chapter 19) |
| Evaluation and testing | Match to approval and performance against stated criteria within the tested scope | Control testing (Chapter 22); classifier evidence (Chapter 23) |
| Monitoring | Observed performance for a stated use, population, and period | The measurement discipline (Chapter 21) |
| Review evidence | Approved review route, recorded actions and timing, with any criteria-based checks and limitations | Interaction records linked to business records (Chapter 24); testing (Chapter 22) |
| Issues, incidents, and limitations | The unresolved condition, response, owner, dates, and accepted exposure | Open items (Chapter 20) |

| | | |
|---|---|---|
| Retention of the above | Each source's approved treatment, location, applicable trigger, restrictions, and retrieval requirements | Treatments in Chapter 24; system documentation in Appendix H |

For each material approval condition, identify the decision, control, test, or monitoring result that shows how it was addressed. A collection of documents does not establish that every requirement was met. The per-system record can be a maintained index of controlled source records rather than a duplicate repository, provided the references remain usable.

For classifier-specific preparation, use Chapter 23 and the questions in Appendix D.

## Keep Four Things Distinct

Use four distinctions to identify what each record establishes.

- **Verified configuration.** Available technical settings and documented procedures match the applicable approval conditions. This establishes the implemented design, not whether every use stayed within those conditions or every required step occurred.
- **Tested scope.** The record identifies the use, system version, information population, conditions, expected result, method, observed result, and evaluation limits. The conclusion applies to what was examined. A provider's evaluation is context, not proof of performance on the organization's information.
- **Observed results.** Monitoring describes performance for a stated use, population, and period. A dashboard supports conclusions about what it measures and covers, not the whole system by default.
- **Unresolved limitations.** Known issues, incidents under investigation, and accepted exposure remain visible with owners and action or review dates. Acceptance needs a rationale, authorized approver, expiry, any earlier review date, and follow-through; the decision does not resolve the underlying condition.

Human review illustrates the distinction. A configured review step shows the design. The approved route identifies whether review is individual or sampled, and linked records show the actions and timing recorded. To assess adequacy, examine a defined sample against the review criteria and supporting information. Changes and rejections can reveal patterns, but their frequency alone does not establish review quality. An unchanged output does not establish that review was absent.

## Assign Ownership of the Evidence Trail

Evidence comes from several functions. Legal interprets requirements, technology manages configuration and logs, privacy assesses personal information, records management defines retention, and the business supervises use. Each may hold valid source records without anyone maintaining the relationship among them.

Assign one accountable evidence owner per system. The role coordinates required content, currency, retrieval, and escalation. It does not replace those authorized to approve a use, operate a control, or reach an independent testing conclusion. One person may cover a small portfolio; a larger portfolio may need business-area roles coordinated centrally, as Chapter 27 develops.

The evidence map should identify the producing function, responsible person, source location, and delivery or update requirement for each element. Record how gaps reach the evidence owner and are escalated. These handoffs should survive a change of staff rather than depend on individual memory.

## Build It Into Intake and Keep It Current

Starting the record during adoption can reduce later reconstruction while decisions and their participants are available. The onboarding gate should require the initial evidence and approval conditions appropriate to the use before reliance. Distinguish authorization for controlled validation from approval for operational use, as Chapter 26 describes. Identify later assessments by their actual dates rather than present them as original intake.

Review the record on its risk-based cadence and assess relevant changes at every tier: provider updates, new uses, information sources, or degrees of automation. Apply Chapter 19's process. Provider arrangements should address change notice and the information needed for reassessment, using the approach in Chapters 17 and 23.

Preserve prior approvals, risk classifications, configurations, and review requirements with the system and use versions they governed. Record effective dates and the periods covered by testing and monitoring. Updating the current record should not overwrite history needed for Chapter 22's reconstruction test. Record unavailable provider versions or historical settings as limitations.

Assess whether a change affects only future use or requires review of earlier outputs, decisions, or retained information. Record any restrictions and follow-through needed before reliance on the revised system. Retain each evidence type under its applicable approved treatment; a shared index does not create one retention period for everything it references.

When a finding changes the basis for an approval or reliance decision, identify the affected uses and connections and reassess the relevant conditions. Keep one traceable source issue with the necessary links, owners, and authorized interim actions. Assess the response rather than duplicate the issue or assume that every use must be suspended.

## Why AI Evidence Stays Fragmented

Evidence can remain fragmented even where individual functions perform their work. Several structural conditions contribute.

- **Policy development is a bounded task.** Maintaining a policy requires review, but system-level evidence also depends on continuing contributions from several functions.

- **AI tasks are added to existing functions.** Each group takes work closest to its mandate without a shared record connecting the results.
- **Evidence is assembled after the question arrives.** The first attempt to connect records happens during an audit, inquiry, or dispute.
- **Review happens once.** An initial assessment is treated as current after uses or provider versions change.
- **Embedded features escape intake.** Capabilities added to existing platforms may not reach a process designed for new systems.
- **Provider documentation substitutes for local evidence.** Published evaluations are treated as though they described controls and performance in the organization's environment.

A program may therefore have completed governance work but lack a supported account of the system, use, and period being examined.

## What Works in Practice

A professional certification body with roughly nine hundred staff administered examinations and credentials for several professions. It had adopted an AI policy, published responsible AI principles, and established a committee that reviewed AI proposals.

An accrediting body that oversees its certification programs asked for evidence that AI-assisted steps in certification decisions had been approved, tested, and monitored, and that staff had reviewed AI outputs before relying on them. The request followed a candidate's challenge to a decision in which a remote proctoring flag had played a part.

The certification body could produce policies, principles, and committee minutes, but its system-level record was incomplete. Some approvals existed in minutes; others were missing. A few systems had risk ratings. The scoring support tool had been tested only by its provider on the provider's data. Staff reviewed proctoring flags before action, recording decisions in individual notes. Those notes remained source evidence, but were not consistently connected to the overall record. A known concentration of flags from one testing location, attributed to a camera configuration, was tracked in email.

Seven systems were known. Reconciliation against procurement records and release notes found four more, all features enabled in existing platforms. The eleven were classified across three tiers using their defined uses and criteria incorporating the retention baseline. Three were highest tier: remote proctoring flags, eligibility screening support, and scoring support for written responses.

An assigned owner assembled each high-tier record. Existing approvals in committee minutes were linked with their original dates. Where approval was missing, the relevant use was assessed and a decision made during the remediation review, using the actual decision date. The later decisions were not presented as approval before deployment. Risk classification decisions recorded the criteria and uses examined.

The scoring support tool was evaluated in the organization's environment by comparing suggested scores with independent scoring by qualified examiners on a documented sample. Results were reported by item type.

Reviewer decisions on proctoring flags were captured with candidate files, including whether the flag was upheld or dismissed and why. Monitoring measures received written methods. The camera issue became an owned open item, was corrected, and was closed with evidence from subsequent sessions, with the examined scope identified. That evidence supported the correction within scope; it did not, by itself, resolve the original challenge or establish the appropriateness of every earlier decision.

Lower-tier systems received brief records identifying purpose, owner, approved uses, information used, approval and classification basis, retention treatment, and review date. Provider updates became change triggers at every tier. The committee approved an evidence map identifying the function responsible for each element.

The accreditor's follow-up request was answered with the assembled records for the three high-tier systems, along with the limitations still open.

The figures and details in this example are illustrative, not benchmarks. Tiers, tests, and review requirements should reflect the organization's systems, supported decisions, and applicable requirements.

## Implementation Steps

Inventory systems and uses. Define the system entry and its linked uses, including embedded features and incidental uses within scope. Reconcile against procurement, platform release, and repository records, check intake status, and assign any resulting gaps.

Classify uses by risk and define system oversight. Incorporate the retention baseline, record the basis and approval, and explain how use-level classifications determine shared controls and the reporting tier.

Set evidence requirements by tier. Scale depth, frequency, and independence without removing use-specific requirements. State the criteria for reliance and keep lower-tier records proportionate.

Connect requirements to source evidence. Use existing decision, change, testing, monitoring, interaction, and open-item records. Link material approval conditions to supporting records, with approved retention and retrieval arrangements.

Keep the four evidence distinctions clear. Separate verified configuration, tested scope, observed results, and unresolved limitations. Assess required review against its criteria, not editing frequency alone.

Assign the evidence owner and handoffs. Identify each producing function, responsible person, source location, and delivery or update requirement. Keep approval and independent testing

responsibilities distinct.

Maintain currency and history. Review on the defined cadence and assess relevant changes at every tier. Preserve versions, effective dates, and testing periods, and decide whether prior outputs or decisions need review.

## What Good Looks Like

Each known in-scope system has a record connecting its defined uses, approval conditions, risk classifications, owners, and information sources. Shared evidence is referenced once; differences among uses remain visible. Records can be retrieved for the relevant version and period, not only the current state.

Approvals retain their actual dates and authority. Changes receive proportionate assessment and follow-through. Testing records its criteria and limits, monitoring uses defined measures, and required review has supporting evidence. Open conditions remain visible even when a response or risk acceptance is recorded.

When asked about a system on a particular date, the program can retrieve the relevant records, explain what they establish, and identify missing history or unresolved limitations. It does not substitute the current record for evidence of the past.

## What Proves It

The evidence is the per-system record and the retained history of how it was maintained.

- An inventory with stable system references, linked uses, stated scope, and the date, sources, and results of reconciliation.
- Use-level classifications and the basis for system oversight, with the consequence baseline, approver's authority, actual decision dates, and review points.
- Per-system records linked to material approval conditions and controlled sources, with owners, required-element criteria, version history, and last and next review dates.
- Approval and change records retaining prior versions and effective dates, including provider updates and decisions about affected prior outputs or uses.
- Test records meeting the use's requirements, with system version, population, criteria, method, results, and limits. Identify the assurance tier separately from risk.
- Monitoring methods and period-specific results, with recorded review and any adequacy testing linked to business records. Apply each source's approved retention, restrictions, and retrieval requirements.
- Open items for issues, incidents, missing evidence, and accepted exposure, with owners, action or review dates, and evidence appropriate to closure.
- An evidence map identifying each producing function, responsible person, source location, and delivery or update requirement.

The chapter's priority measure is the count of systems whose record lacks a required element or is overdue for review, reported by risk tier. State the known in-scope inventory, cutoff, required-element criteria, and total systems in each tier. Count each system once, identifying missing elements, overdue review, or both. Show systems awaiting risk classification separately rather than omit them.

Record completeness is not control performance. A retained failed-test result can meet a requirement to document the test while the failure remains an open control issue. A missing required result remains an evidence gap even when the reason is recorded. An unfavorable result and an empty required field are not the same condition.

Read the first count alongside systems newly identified through reconciliation during the period. Check intake status and record the discrepancy: missed intake, an incomplete inventory link, or a scope change. Discovery alone does not establish that intake was bypassed. Explain material inventory or classification changes, do not add overlapping counts, and state that unidentified uses remain outside both measures. These are chapter-level measures, not additions to the standing library.

The assignment, use, system, and connection measures in this part answer different questions. Keep their units and denominators explicit rather than combine them into a single AI-governance percentage.

## Business Value

It reduces avoidable reconstruction. Maintained records and usable history help the organization answer questions about a system or past decision without rebuilding the record from scattered sources.

It supports proportionate scrutiny. Use-level classifications direct attention to consequential decisions while keeping lower-risk records brief and their requirements explicit.

It reduces duplicated evidence work. A maintained index can draw on records produced for existing approvals, controls, and oversight rather than require functions to recreate them for another report.

It can support timely adoption. Defined evidence and reliance criteria make the assessment process clearer while preserving the need to examine the particular use.

## Principles Set the Standard

Principles establish expectations. The per-system record connects those expectations to approved uses, decisions, implemented controls, and evidence of what happened.

Its value depends on more than completeness. The organization needs to know which version and period each record describes, what the evidence supports, and what remains unresolved.

The work is to maintain those connections through existing processes, with accountable owners and proportionate review. That gives the organization a basis for explaining its AI decisions without claiming more than the record establishes.

*Next: Chapter 26 turns to the information AI systems depend on: how data readiness, access, provenance, classification quality, privacy, security, and lifecycle controls make responsible AI adoption possible.*

# Chapter 26: Information Governance as the Precondition for AI

In the first week of a pilot, an employee asked a newly connected AI assistant for the current expense policy. The assistant returned a clear, well-written answer based on a version of the policy that had been superseded two years earlier. The current version existed. So did four older ones, stored in shared folders the assistant could reach, with nothing to indicate which was authoritative.

The assistant accurately summarized the version it retrieved, but the answer did not meet the employee's request for the current policy. The source environment gave it no clear indication of which version the organization regarded as authoritative.

Chapters 23 through 25 addressed AI used to classify information, the information AI use creates, and system-level governance evidence. This chapter addresses what connected AI systems depend on: the condition of the information they can reach. Its focus is information readiness for retrieval and summarization. That assessment complements, rather than replaces, evaluation of the system and its approved uses.

## AI Makes Existing Conditions Visible

The conditions below can exist before an AI connection is proposed. Search, retrieval, and summarization can make their effects more visible by making information easier to find and combine. The connection itself also needs assessment.

- **Access.** Permissions broader than necessary may have received little attention while information was difficult to locate. An assistant can surface sensitive information to people who did not know it was available. In this situation, the connection exposes an existing access problem; its own access arrangements still need assessment.
- **Currency.** A superseded policy may remain for a valid historical purpose but be unsuitable as current guidance. Retained versions need a clear status and applicable dates so that the system does not present an old instruction as the current one.
- **Authority.** Where the authoritative instance is not identified, as Chapter 12 describes, the information provides no clear basis for distinguishing the organization's approved record from drafts or copies. Source identification and system evaluation both matter.
- **Classification.** Where exclusion or restriction controls depend on classification, unidentified sensitive information may not receive the intended treatment. Classification quality and the operation of those controls need examination within the proposed scope.
- **Provenance and permitted use.** Information obtained under contractual terms, consent, or other conditions may carry limits on processing and use. Assess whether the proposed use falls within those conditions, rather than treating access as permission to use it.
- **Lifecycle.** Retention, disposition, and holds operating in a source repository may not reach the indexes, extracts, and copies created by a connection. Chapter 24 distinguishes source-linked convenience derivatives from information with a separately approved purpose.

These conditions draw on capabilities developed earlier in the book. Connecting AI makes their application to a defined source, use, and group of users part of the adoption decision.

Record-category assignment, sensitive-information identification, and AI-use risk classification answer different questions. Retention-category agreement does not, by itself, validate sensitive-information identification or the restrictions that depend on it. Assess those functions within their own scope and criteria.

## Assess Information Readiness for the Intended Use

Technical integration, system performance, and security matter. They do not, by themselves, establish whether the information is appropriate for the proposed use. A repository may be suitable for one use and unsuitable for another.

Assess each system-to-source connection before authorizing operational use. Identify the repository or collection from Chapter 8's register, the information population, intended uses, and users. A system reaching five repositories needs assessment of all five connections. Reuse evidence where conditions are shared, while recording differences that affect the decision.

Authorize the controlled access needed for validation under a defined test scope and applicable restrictions. Identify the test population, users, processing restrictions, and interim protections before validation. Approval for operational connection and reliance follows the required assessment and verification. Testing permission is not permission for unrestricted use, and a pilot is not an exemption from applicable requirements.

*Figure 26.1. Information readiness questions for an AI connection.*

| Readiness area | Question before operational use | Where it is developed |
|---|---|---|
| **Scope** | Which registered source population, approved uses, and intended users are in scope, and who owns the lifecycle decision? | Chapters 8, 14, and 25 |
| **Access** | Do source permissions and connection controls support that scope, with permitted and restricted retrieval verified under the authorized test scope? | Chapters 4, 8, and 22 |
| **Authority and currency** | Are authoritative versions and applicable dates identified, with retained history distinguished from current guidance? | Chapters 12 and 13 |

| | | |
|---|---|---|
| **Classification** | Is sensitive information identified sufficiently for the required exclusions and restrictions, with identification and control operation verified within their own scope and criteria? | Chapter 4; validation discipline in Chapters 9 and 23 |
| **Provenance and permitted use** | Are origin, source version or population, and use conditions known, and does the proposed processing meet them? | Chapters 17, 18, and 23 |
| **Lifecycle** | Are derived stores and treatments identified, with required source-change, disposition, and preservation responses verified? | Chapters 13, 15, and 24 |
| **Evidence** | Does the system record link the dated assessment, decision, verification, review point, and open conditions? | Chapters 19, 20, and 25 |

Appendix D carries these questions with the questions for classification projects from Chapter 23.

Record one of four outcomes for operational connection: connect, connect after remediation, connect with a documented limitation, or exclude. Connect after remediation requires the applicable completion criteria to be met and verified before operational connection. Excluding information from an AI connection does not authorize its destruction; its approved treatment continues unless separately changed.

Connecting with a documented limitation requires an authorized decision stating the affected scope, rationale, interim restrictions, review point, any time limit, and follow-through. Where the decision includes risk acceptance, record its expiry and any earlier review date. A scheduled review does not extend an expired acceptance without a new authorized decision. Acceptance does not authorize use outside applicable requirements or resolve the condition. Where necessary conditions cannot be met, defer or exclude the affected connection.

Readiness applies to the conditions assessed. Reassess relevant changes in use, source content, permissions, classification, authoritative versions, provider arrangements, and lifecycle behavior through Chapter 19's process. Set a proportionate review cadence and retain prior assessments with their dates and scopes. Later reviews of existing connections need their actual dates and an interim decision, not an implied history of preconnection approval. Apply Chapter 25's shared-finding approach when a source or control issue changes the basis for other uses or connections.

Start with governed repositories. An owned policy library with identified authoritative versions and reviewed permissions is a more practical starting point than a legacy share with no lifecycle decision. Apply Chapter 14's legacy process where needed, rather than treat technical availability as readiness.

The source owner supplies the repository information; the authorized decision-maker approves the connection; the evidence owner maintains the link to Chapter 25's record. These roles need not be the same person. Unapproved gaps, incomplete remediation, and accepted limitations remain open work under Chapter 20.

Access assessment includes both source permissions and the connection's arrangements. Use Chapter 22's approach to verify permitted and restricted retrieval for the intended users and scope. A permissions list records the design, not the retrieval result.

A superseded version may still have a valid retention purpose. Identify its status and applicable dates, and prevent it from being presented as current guidance when that is not the approved use. Dispose of it only when its treatment and the applicable checks permit.

For provenance, record the source, origin, relevant version or population, and conditions governing the proposed processing and reliance. Apply Chapter 23's processing-constraint assessment, including where information may be processed. Unresolved permission questions need a decision; technical access does not settle them.

Identify derived stores, their source relationships, and approved treatments. Following Chapter 24, distinguish convenience derivatives from separately governed information and account for multiple sources. Verify the required response to source changes, disposition, and preservation. Use Chapter 13's authorization and applicable eligibility, encumbrance, and dependency checks.

Compare observed delays with approved timing requirements and record any gap. State whether a test established removal from retrieval, removal from an index, or the defined destruction state. Those outcomes are not interchangeable, and recording a delay does not approve it.

## Durable Patterns, Not a Survey of Laws

This section connects applicable requirements to capabilities and evidence rather than providing a survey of current laws. The six areas below help connect applicable AI requirements to existing records and controls. They are not a complete or universal statement of obligations. Determine what the particular system and use require, then identify the evidence needed.

- **Transparency.** A connection inventory, repository records, and provenance information help explain which sources a system uses. Use these records to support the explanation or disclosure required for the particular system and use.
- **Accountability.** Named owners and accurately dated decisions identify who approved a use, on what basis, and who is responsible for the resulting controls. Chapters 18 and 25 provide that structure.
- **Documentation.** The per-system record connects purpose, approved uses, controls, and evidence. Chapter 24 provides the retention approach for the supporting system documentation and interaction records.

- **Testing.** Defined criteria, scope, methods, and results support conclusions about performance and control operation. Use Chapter 22's testing discipline and Chapter 23's classification-specific validation where applicable.
- **Human oversight.** Define the review, intervention, or stopping authority required for the approved use. Chapters 23 through 25 connect those requirements to review routes, recorded actions, and criteria-based testing.
- **Minimization and purpose limitation.** Defined scope, recorded use conditions, and lifecycle controls support decisions about what information is needed and how it may be used. Disposal still requires the approved authorization and applicable checks.

Existing capabilities provide a starting point. The organization still needs to assess what applies, identify gaps, and determine whether additional controls or evidence are needed.

## Why Readiness Is Skipped

Information readiness can be deferred even when the organization recognizes its importance. Several conditions contribute.

- **Adoption is driven by urgency.** A business case focused on speed and productivity can make readiness work appear to delay the benefit.
- **Readiness is framed as integration.** Teams assess whether the system connects and performs without assigning the question of whether the information is appropriate for the use.
- **More sources seem to mean better answers.** Adding sources expands the information population and the conditions the program needs to assess. That work may be missing from the connection plan.
- **Permissions have never been reviewed.** Access accumulated over years may be assumed correct because it has not produced a visible problem.
- **Ownership falls between functions.** Data, technology, security, privacy, and records functions hold different parts of the assessment without a defined connection decision or accountable coordinator.
- **Governance is consulted late.** Findings raised just before launch can be treated as obstacles rather than information that should have shaped the scope and sequence.

The result can be a technically successful connection without a supported decision that its information is appropriate for the intended use.

## What Works in Practice

A live entertainment company with roughly five thousand employees, including a large seasonal workforce, operated arenas and outdoor venues in several regions. It planned to give staff an AI assistant connected to its document repositories so that supervisors could find procedures, event details, and policies without calling the office.

The pilot was paused in its second week. A seasonal supervisor asked about crowd management procedures for a venue and received an answer drawn from a procedure that had been replaced after a safety review. In a separate query, an employee asking about staffing levels received a summary that included compensation figures from a human resources spreadsheet stored in a shared folder with broad permissions.

After the pause, the team assessed all forty-one repositories the assistant could reach for its intended uses and users. Twelve met the readiness criteria. Seventeen required remediation before connection. Twelve were excluded, mostly legacy shares with no current owner or lifecycle decision, and were referred to the legacy process.

Remediation included reviewing and narrowing permissions, identifying authoritative policy and procedure versions, and disposing of or excluding superseded material under the approved process. A decision that a source could connect after that work was not evidence that the work had been completed.

Categories containing payment information, employee records, and venue security plans remained excluded from the assistant’s index unless the required classification and access restrictions were in place. The team also tested whether source disposition removed content from the assistant’s index. It observed a delay of several days and recorded the result and its scope. The observation did not establish that the delay met an approved timing requirement or that the information had reached its destruction state.

The assistant was relaunched in phases, beginning with the twelve ready repositories. Each connection and assessment was linked to the per-system record. Repositories with unresolved access issues remained outside the relaunch scope, with those issues tracked as open items with owners and dates. The seventeen requiring remediation could connect only after the applicable completion criteria were met and verified.

The figures and details in this example are illustrative, not benchmarks. Readiness criteria and remediation priorities should reflect the information, intended uses, users, and consequences involved.

## Implementation Steps

Assess each connection for its intended uses. Identify the system, source population, uses, and users, and authorize the restricted scope needed for validation separately from operational use. Reuse shared evidence without extending one use’s approval to another. Record later assessments of existing connections by their actual dates.

Start with governed sources and sequence the work. Record one of the four outcomes. Complete and verify required remediation before operational connection; exclusions leave existing retention treatment in place. A documented limitation needs authorized conditions, not only a target date.

Review and verify access. Examine source permissions and the connection's access arrangements. Verify permitted and restricted retrieval within the intended scope, and record the method, result, and limitations using Chapter 22's structure.

Resolve authority and currency for the use. Identify authoritative instances and applicable versions. Distinguish current guidance from retained historical material, exclude unsuitable content from the use, and carry out authorized disposition where appropriate.

Record provenance and permitted use. Identify the source, origin, relevant population or version, and applicable conditions. Record the basis for the processing and use decision, including where processing may occur and any unresolved permission questions.

Extend and test lifecycle controls. Identify derived stores, source relationships, and approved treatments. Verify required responses to source changes, disposition, and preservation. Compare observed delays with approved timing, distinguishing retrieval, index removal, and destruction.

Maintain the connection decision and its evidence. Link the assessment, approval, verification, and review point to Chapter 25's record. Reassess on relevant changes and the defined cadence, preserve prior versions, and track gaps and accepted limitations through Chapter 20.

## What Good Looks Like

Each known in-scope connection has a dated assessment covering the relevant source population, uses, and users. The record distinguishes the assessment outcome from permission to connect and from completion of required work. Shared evidence is referenced without concealing differences among uses.

Source permissions and connection restrictions have supporting verification. Current guidance is distinguished from retained historical material. Classification and use conditions support the approved scope. Derived stores have defined treatment, with lifecycle-test results and timing limitations recorded. Relevant changes prompt reassessment, and prior decisions remain available.

The clearest sign is that readiness findings shape the sequence of adoption. Sources can proceed when their conditions are met, while unsuitable connections are deferred or excluded. Open work remains visible rather than disappearing when an assessment is completed.

## What Proves It

The evidence should show when the assessment occurred, what it covered, who decided, and what was verified. Distinguish controlled-validation authorization, operational approval, and later remediation, and a completed record from a favorable readiness result.

- A connection register with stable system and source references, scoped populations, linked uses and users, validation and operational status, and inventory scope and cutoff.
- Dated readiness assessments with criteria, supporting evidence, one of the four outcomes, decision authority, conditions, review points, and the scope of controlled validation versus

operational approval. Preserve prior versions and identify later assessments accurately.

- Source-access reviews and connection tests showing permitted and restricted retrieval within scope, the method, changes made, observed results, and any limitations.
- Records identifying authoritative instances, versions, and applicable dates, with the approved disposition or use-specific exclusion of unsuitable content and the treatment of retained history.
- Provenance and permitted-use determinations identifying origin, source populations or versions, applicable conditions, processing constraints, and unresolved permission questions.
- Lifecycle tests linked to derived stores and their treatments, showing responses to source changes, disposition, and preservation. Distinguish observed delays, approved timing, retrieval or index removal, and destruction evidence.
- Open items for unapproved gaps, incomplete remediation, and accepted limitations, linked to the per-system record. Retain owners, action and review dates, any acceptance expiry, authorized conditions, and verification appropriate to closure.

The chapter's priority measure is the count of known in-scope AI connections without a recorded readiness assessment covering their current scope and use. Include absent, incomplete, and no-longer-current assessments, identifying the reason. Read it alongside active connections operating under approved limitations that remain unresolved.

State the inventory scope, reporting cutoff, and relevant totals, showing proposed and active connections separately. Count each defined system-to-source connection once within each measure, with use-specific gaps underneath it. A source reached by two systems represents two connections, not one repository count. These counts describe assessment and open-condition status, not information volume or the size of exposure.

A complete assessment may conclude that a source must remain excluded. Recording that assessment closes the assessment gap, not any underlying readiness gap. The two measures may overlap and should not be added. Newly identified connections can increase the counts without deterioration in existing controls; unidentified connections remain outside them. These are chapter-level measures, not additions to the standing library.

## Business Value

It can reduce errors caused by unsuitable source information. Identifying authoritative versions and controlling their use gives the system a better-defined information base. Testing still needs to establish the resulting performance.

It can reduce inappropriate retrieval. Reviewed source permissions and verified connection controls help limit exposure within the scope examined.

It supports the system's evidence record. Readiness assessments, provenance decisions, and lifecycle tests connect source conditions to the approvals and results described in Chapter 25.

It can create benefits beyond AI adoption. Permission changes, authorized disposition, and completed legacy decisions may improve the wider information environment. Report the benefits supported by those completed actions rather than assume that assessment alone reduced cost or exposure.

## Governed Information Makes Responsible AI Possible

An AI system's performance depends in part on the information available for its approved uses. Currency, authority, access, classification, provenance, and lifecycle treatment belong in that assessment alongside the evaluation of the system itself.

Information governance provides a basis for deciding which information a system may use, under what conditions, and with what evidence supporting the decision. That basis needs to remain current as sources, users, and uses change.

Organizations can build on the controls described in this book without assuming they already meet every AI-specific requirement. The work is to apply them to the connection, verify the relevant outcomes, and keep unresolved conditions visible.

## Closing Part V

Part V examined four relationships: AI used for classification, information created through AI use, evidence about AI systems, and the information those systems depend on. Each applies the book's operating model to a different governance decision.

Existing categories, decision records, change control, open items, measurement, and testing provide a foundation. Their application still needs to reflect the system, the approved use, and the requirements involved. Reuse that structure without assuming every AI-specific assessment is already complete.

AI can extend what an organization can classify, retrieve, and use. The value depends on defined scope, appropriate information, proportionate review, and evidence supporting reliance. Gaps remain work to address, not conditions that adoption resolves.

Part VI, Scaling and Sustaining the Program, turns to sustaining that work: assigning ownership and decision rights, funding capabilities, moving beyond pilots, maintaining pace with the business, developing skills, and treating governance as continuing infrastructure rather than an initiative.

*Next: Chapter 27 defines the operating model: ownership layers, decision rights, control owners, and the forums that decide who decides, who executes, and who produces evidence.*

# Part VI: Scaling and Sustaining the Program

# Chapter 27: Building the Operating Model

Many of the gaps described in this book have an ownership question inside them. A rule is approved but no one is assigned to confirm it operates. A repository appears on the register with an owner who left two years ago. A change is approved by a committee whose authority to approve it was never defined. An open item waits because it belongs to no one who can close it.

Earlier chapters named the roles each capability needs: policy owners and control owners in Chapter 1, information owners in Chapter 8, control owners and configuring parties in Chapter 11, authorized decision-makers in Chapter 18, owners of open items in Chapter 20, evidence owners in Chapter 25. Each role was defined where it was needed. This chapter assembles them into an operating model: who decides, who executes, who produces evidence, and how those assignments stay current as the organization changes.

Parts I through V described capabilities. Part VI addresses what keeps them working over time. The operating model comes first because funding, scaling, and speed depend in part on knowing who holds which decision and who answers for which result.

The chapters that follow develop connected responsibilities, not a sequence that postpones capability or evidence until later. Use Chapter 31's task assessment during planning, link it to the roles defined here, and identify the recurring support the funding decision in Chapter 28 needs to provide.

## The Orphaned Schedule

A common pattern starts with a well-built schedule owned by a records function. The records function researched it, the committee approved it, and it is maintained carefully. Ownership of the schedule's content is clear.

Ownership of what happens next is often less clear. The records function may not administer the systems where rules operate, may lack authority over the business processes that create information, and may not control the budgets that fund configuration. The business owns its information but may regard retention as a records matter. Technology teams may receive configuration requests without a clear route for confirming the underlying authority. Each function can reasonably believe another is responsible for operation.

The result can be a schedule with an owner and a set of controls without one. Chapter 1 described the gap between policy and control. An ownership gap can contribute: the policy layer has a clear owner, while accountability for the distributed control work has not been defined.

## Three Layers of Ownership

The roles developed across this book fall into three layers. These layers group responsibilities; they are not prescribed departments or levels of independence. The same person may hold roles in more than one layer, and the combination should be deliberate rather than accidental.

*Figure 27.1. Ownership layers and the roles within them.*

| Layer | Role | Accountable for | Developed in |
|---|---|---|---|
| Decide | Policy owner | The content of rules and standards, and proposing changes to them | Chapters 1 and 19 |
| Decide | Decision authority | Approving treatments, changes, exceptions, acceptances, and AI uses within a defined delegation | Chapters 18 and 25 |
| Operate | Information owner | The information's business use and lifecycle requirements, with decisions made or referred according to the applicable delegation | Chapters 8 and 14 |
| Operate | Control owner | A control operating as approved in a specific pairing, connection, or process | Chapters 1 and 11 |
| Operate | Configuring party | Implementing and maintaining the approved configuration or procedure, with departures referred for a decision | Chapter 11 |
| Operate | Relationship owner | Governance terms and evidence obligations with an external party | Chapter 17 |
| Assure | Evidence owner | Maintaining the evidence record for a defined scope and identifying missing or outdated elements | Chapter 25 |
| Assure | Tester | Examining control existence, alignment with approved requirements, and operation, at a stated assurance tier and level of independence | Chapter 22 |
| Assure | Program lead or designated role | Maintaining the open-items view, measures, and escalation, and coordinating across roles | Chapters 20 and 21 |

The decide layer groups responsibility for rules and authorized decisions. The operate layer groups execution responsibilities. Decisions and operations produce evidence as well as testing. The assure layer coordinates and examines that evidence, with the independence of each assessment stated separately. Maintaining an evidence record or an open-items view does not, by itself, constitute

independent assurance. Oversight bodies exercise responsibilities under their charters and reserved decisions; their oversight purpose is distinct from routine operation.

Combining roles requires attention to both people and activities. A tester's independence concerns their relationship to the control, as Chapter 22 describes; a control owner should not supply an independent assurance conclusion on their own control. Approval and implementation remain different activities with different evidence, even where an authorized individual performs both. An evidence owner coordinates the record but does not replace those authorized to approve a use, operate a control, or reach a testing conclusion. Combining roles can be practical, particularly in smaller organizations, provided applicable independence requirements are met and the combination and any limitations are recorded.

For each control, identify who performs the recurring work and produces its evidence. For each open item, identify who is responsible for the action or coordination needed. Program-level reporting does not absorb those responsibilities.

Connect each scoped assignment to its applicable authority, assessed capability, required reviewer, and available time. Chapters 28 and 31 address funding and capability; neither a confirmed name nor an approved budget, on its own, establishes that the recurring work can be performed. Use links to the existing records rather than create another role inventory.

## Decision Rights Belong in Writing

Authority is often assumed from seniority or committee membership rather than defined. Chapter 18 requires the approval record to identify the authority under which the decision was made. Naming the approver is not enough to establish the scope of that authority. The operating model makes the source and limits explicit.

A decision rights statement identifies, for each type of decision, who may make it, within what limits, with whose input, and who must be informed. Useful decision types include approving a new category or retention treatment, approving a jurisdictional variation, approving a substantive or urgent change, granting a rule exception, accepting a risk or limitation, authorizing disposition at the category level, approving an AI use and its risk classification, and approving a connection between an AI system and a source. Appendix B carries role definitions and a template for recording these rights.

Limits matter as much as holders. A business unit may be authorized to approve local administrative changes within the approved treatment, while a change that shortens a period requires the central decision authority and counsel. Such administrative changes are distinct from jurisdictional variations in applicable treatment and from rule exceptions. As an illustration only, an organization might allow a program lead to accept a limitation for a short, defined period, such as ninety days, and require the accountable risk owner to approve anything longer. Duration is one limit; the decision type, scope, consequence, and applicable requirements also determine delegated acceptance. Each acceptance remains subject to Chapter 20's requirements for rationale, authority,

expiry, and review. The appropriate holders, periods, and limits are for each organization to set. Stating limits can help routine decisions move without escalation and direct consequential ones to the appropriate authority. Chapter 30 returns to how clear decision boundaries reduce unnecessary central approval.

Record the source of each authority: the charter, policy provision, or delegation that grants it, and its effective date. When a decision is later examined, the question is whether the person held that authority on the date of the decision. A current organization chart does not answer that question.

## Assign Roles, Name Holders, and Keep Both Current

Ownership assigned to a named individual tends to decay when that person moves. Ownership assigned only to a role can leave no one answerable in practice. The operating model needs both: a role defined by position or function, and a named current holder of that role.

Ownership decays in predictable ways. People change roles, leave, or are reorganized into new structures. Acquisitions add repositories, systems, and staff whose roles map imperfectly to the existing model. Divestitures remove owners while leaving obligations. Each of these events can leave register entries, controls, open items, and decision rights pointing at people who no longer hold them.

Treat ownership as a control that can fail. Define how ownership is confirmed, how often, and what happens when a holder leaves. Reconcile the ownership register on a defined cadence and after acquisitions, divestitures, reorganizations, or other changes affecting assignments. Route the resulting work through the change process described in Chapter 19. Confirmation establishes that the holder acknowledges the assignment and its scope. It does not, by itself, establish authority, access, or performance of the required work. Evidence of decisions, control operation, and follow-through remains with the relevant source records.

The ownership register can be a maintained view linked to existing records. Retain prior holders, applicable delegations, and their effective periods rather than overwriting them. Record the handover of controls, open items, and evidence responsibilities, including anything unresolved at the transition.

Define a vacancy rule. When a role becomes vacant, assign interim accountability to a designated role, such as the unit's governance lead or the next level of management. Identify the interim holder, scope, period, and authority under the applicable delegation. Assigning responsibility does not, by itself, transfer the previous holder's approval rights. Decisions outside the interim holder's authority follow the defined escalation route.

An approved interim holder provides accountability for the stated scope and period. Keep the underlying vacancy or temporary arrangement visible until an enduring assignment is confirmed, whether through replacement or permanent reassignment. Extending an interim arrangement requires a recorded review and decision rather than an automatic reset of the due date.

Manage the ownership gap under Chapter 20 with an owner for the enduring assignment and a date. Where one vacancy affects several controls or existing findings, retain one traceable ownership issue linked to the affected work, or attach the deficiency to the existing item, rather than duplicating findings for each owner field. Resolving the ownership deficiency does not close the control failure or other work inherited by the new holder.

Acquisitions deserve a defined onboarding path. Chapter 7 noted that a structured schedule can absorb growth; the operating model helps put that capability into practice. When an organization is acquired, identify its repositories and systems through the register process in Chapter 8, confirm or assign information and control owners, determine which decision rights apply during transition, and record the transition period and its end. Record unresolved ownership and decision rights as specific transition gaps, with interim responsibility and due dates. Assess existing controls and approved treatments on their own evidence rather than treating either the acquisition or the assignment of names as proof of governance.

## Forums With Defined Purposes

Governance programs may ask one committee to approve the schedule, resolve disputes, review progress, and oversee risk. Without clear purposes and authority, status reporting can crowd out decisions.

Separate the purposes of decision-making, operational coordination, and oversight. Depending on scale, these may use existing forums or clearly separated agenda sections, provided authority, records, and required independence remain clear.

- **A decision forum.** Makes decisions reserved to it under the decision rights statement and resolves escalations. Its record is a set of decisions with their basis and authority, not only minutes of discussion.
- **An operating forum.** Brings together control owners, configuring parties, information owners, and the program lead to work through open items, changes, and blockages. Its cadence follows the work, and it decides within its delegated limits.
- **Oversight.** An audit or risk committee receives reporting in the format described in Chapter 20 and holds the program to account. Its charter defines its oversight responsibilities and any reserved decisions.

Define which matters follow the normal forum sequence and which require direct escalation. Chapter 20's criteria address exposure, urgency, and age. Urgent concerns and required independent reporting should not wait for an intermediate meeting. Record decisions with their basis and authority, and make the reporting routes to oversight explicit.

## Central Standards, Distributed Execution

A common arrangement combines central standards with distributed execution. The balance depends on the organization's structure, capabilities, and decision rights.

The delivery arrangement may use local staff, central teams, shared services, or defined partner support. Local accountability does not require every task to be performed locally. Identify who performs the work, who reviews it where required, and how the responsible owner receives evidence and raises a gap.

The central program typically holds the schedule, the category model, the decision rights statement, the measurement methods, and the open-items view. Business units hold information ownership and many control-owner roles. Technology functions hold configuring-party roles and many system-level controls. Specialist functions, such as legal, privacy, and security, hold decision authority or advisory roles for defined decision types.

The arrangement works when the connections are defined: which decisions are central and which are local, how local roles report into the program, and how the program learns when local ownership changes. Many organizations designate a governance liaison in each business unit to hold the local connection. The liaison role should be defined, with time allocated to it, rather than added to someone's existing responsibilities without recognition.

## Why Operating Models Drift

Operating models are often designed carefully and then left to drift. Several forces contribute.

- **Ownership is assigned once.** Assignments made during a program launch are not revisited, and the register gradually describes an organization that no longer exists.
- **Authority is assumed rather than delegated.** Committees and senior staff approve decisions without a recorded source of authority, which becomes visible only when a decision is questioned.
- **Roles are added without time.** Control ownership or liaison duties are assigned without adjusting existing responsibilities, so the role exists in name and receives little attention.
- **Organizational change is not assessed.** Reorganizations, acquisitions, and divestitures proceed without a step that checks affected governance assignments and decision rights.
- **Forum purposes are unclear.** Decisions, operations, and oversight compete for attention without clear agenda boundaries, authority, or escalation routes.

The result can be a program whose design is sound and whose assignments no longer match the people doing the work.

## What Works in Practice

A diversified holding company with roughly fourteen thousand employees had grown over fifteen years by acquiring eleven operating companies in distribution, industrial services, and specialty manufacturing. A central records function owned the schedule. Each operating company ran its own systems, and several had their own records staff.

The divestiture of one operating company exposed the gaps. The buyer needed certain records transferred, the seller needed to retain others, and some information needed to be destroyed at

closing. No one could say with confidence who was authorized to decide which records went where. The central committee believed the operating company held that authority; the operating company believed it sat with the holding company. The decisions were eventually made by the transaction team with counsel, under time pressure and without reference to the schedule's decision records.

A review after the divestiture found that the register listed two hundred twelve repository-owner assignments across the group. Seventy-one assignments named people who had left or changed roles. Forty-three pairings in the implementation matrix had no named control owner. Schedule changes for two operating companies had been approved by local committees whose authority to approve them had never been defined.

The team defined the three ownership layers and wrote a decision rights statement. Shared categories and changes that shortened a period were reserved to the holding company's decision forum. Local administrative changes within the approved treatment could be approved by each operating company's governance lead. Transfers and destruction in acquisitions and divestitures required a named decision authority from each side and a decision record. The new statement defined rights for subsequent decisions; it did not establish the authority behind the earlier local approvals.

Roles were defined by position, and each had a named holder who confirmed the assignment. The register was reconciled quarterly and after every reorganization, acquisition, or divestiture, which were added to the company's change process as triggers. A vacancy rule assigned interim accountability to the operating company's governance lead and kept the vacancy recorded as an open item pending an enduring assignment. Each operating company designated a governance liaison with time allocated to the role.

A year later, nine repository-owner assignments were unconfirmed, each recorded as an open item with an owner and date. That count described unconfirmed assignments; it did not establish that all nine had departed holders or that no one was performing the work. All forty-three pairings had named control owners, an ownership result rather than evidence that every underlying control operated successfully. When the group acquired another operating company, owners were assigned and confirmed for its repositories during the transition period recorded in the acquisition plan, and the transition decisions were recorded against the decision rights statement.

The figures and timeframes in this example are illustrative, not benchmarks. Decision limits, review cadences, and vacancy arrangements should reflect the organization's structure, risk, and the decisions involved.

## Implementation Steps

Inventory the roles the program already uses. Collect the roles named across the register, the implementation matrix, decision records, open items, and AI records, and map them to the three layers.

Write the decision rights statement. For each decision type, record who decides, the limits, who is consulted and informed, and the source and effective date of the authority. Use the Appendix B template.

Define roles by position and name current holders. Record each assignment's responsibility and scope, holder, effective period, and last confirmation. Retain prior assignments and delegations, and record handovers. Record deliberate role combinations, required independence, and separate evidence of approval and implementation.

Treat organizational change as a governance trigger. Assess acquisitions, divestitures, reorganizations, and other events that affect assignments or decision rights through the existing change process. Record what was checked, what was confirmed or changed, and any unresolved discrepancy.

Set a vacancy rule. Name interim holders and their delegated authority, link ownership gaps to affected work, and assign an owner and date for an enduring assignment. Keep temporary arrangements visible and require a recorded review and decision for extensions.

Define forum purposes and connect them. State membership, authority, records, and reporting for decision-making, operational coordination, and oversight, using separate forums or agenda sections as appropriate. Preserve direct escalation for urgent concerns and required independent reporting.

Allocate time to the roles in the delivery model. Recognize control ownership, review, and liaison duties in role expectations and capacity. Link the supported capability assessed in Chapter 31 to the recurring provision described in Chapter 28, rather than adding responsibilities without adjustment.

## What Good Looks Like

In a program with a working operating model, every role the program depends on has a definition and a confirmed current holder. Decision rights are written, with limits and a recorded source of authority, so that the authority behind a decision can be established for the date it was made.

Organizational change prompts an assessment of assignments and decision rights. Vacancies and temporary arrangements remain visible and owned. Decision-making, operational coordination, and oversight have clear purposes, with decisions recorded under stated authority and connected through defined escalation and reporting. Central standards and distributed execution are linked by named liaisons with time to do the work.

An unexpected decision, such as a divestiture or an urgent change, provides a practical check. The record identifies who may decide, or the route for resolving an authority gap, and supports the decision made within that scope.

## What Proves It

The evidence for this chapter supports the definition and currency of assignments and authority, and shows how ownership maintenance was checked. Confirmation of an assignment remains

distinct from evidence that the underlying work was performed.

- A role inventory mapped to the three layers, with definitions, deliberate combinations, and required separations.
- A decision rights statement in the Appendix B form, with limits and the source and effective date of each authority.
- An ownership register showing each scoped assignment, its current holder, effective period, and last confirmation, with links to prior holders, delegations, and handover records.
- Reconciliation and scoped testing records showing what was examined after scheduled reviews or organizational changes, what was confirmed or changed, and which discrepancies remain. Trace selected changes through reassignment, confirmation, authority, and updates to affected records using Chapter 22's testing discipline. Where no reassignment was needed, retain the basis for that conclusion.
- Vacancy and temporary-assignment records linked to affected work, with interim scope and authority, an owner for the enduring assignment, due dates, and recorded reviews of extensions.
- Forum charters or defined agenda purposes, decision records identifying applicable authority, and escalation records covering both normal routes and direct reporting.

The chapter's priority measure is the count of in-scope role assignments without a confirmed current holder, reported with total assignments by layer. Identify the responsibility and scope of each assignment; count neither the nine role types nor unique people. One person may hold several separate assignments. State the register scope, reporting cutoff, and confirmation period. Distinguish assignments with no holder, those naming an invalid or departed holder, and those with overdue confirmation. An overdue confirmation is a gap in evidence of currency, not necessarily proof that nobody is performing the role.

Read the assignment count alongside vacancies or temporary arrangements that remain open beyond the period allowed by the vacancy rule. A confirmed, authorized interim holder can satisfy the current assignment for its stated scope and period while the underlying vacancy remains visible in vacancy aging. The measures may overlap and should not be added together. This chapter-level measure complements the unowned repository count in Appendix C; it does not expand the five-measure library or Figure 21.1. A rising count after a reorganization may reflect better detection rather than deterioration.

## Business Value

It supports examination of decisions by making authority traceable. Records of the source, limits, and effective period of each decision right help establish whether a person was entitled to make the decision on that date.

It can reduce delay. Clear decision limits let routine decisions proceed locally and direct consequential ones to the appropriate authority, reducing unnecessary escalation and decisions

deferred for lack of an owner.

It supports continuity through organizational change. Reviewing assignments and decision rights after reorganizations, acquisitions, and divestitures helps keep ownership aligned with the organization.

It gives the program a maintained record of accountability. Controls, open items, evidence records, and AI uses each depend on someone responsible for defined work. The operating model identifies those responsibilities and how they are kept current.

## Accountability Has to Be Maintained

Assigning ownership is a starting point. Its currency matters when a question arrives about a decision made years earlier, or a control in a system that has changed hands.

Maintain roles, authorities, and forum purposes alongside rules and controls. Confirm assignments, retain their history, and test whether organizational changes reach the records and responsibilities they affect.

That work keeps current accountability visible while preserving the basis for examining earlier decisions. A name on a chart starts the inquiry; the assignment, applicable authority, and evidence of the work support the answer.

Next: Chapter 28 turns to making the case and funding it: how to frame governance in the terms executives use, and how to sequence a roadmap leadership can understand and approve.

# Chapter 28: Making the Case and Funding It

The governance lead presented a detailed proposal: a maturity assessment, a target operating model, a technology evaluation, and a three-year plan. The executive committee listened, asked thoughtful questions, and deferred the request to the next budget cycle. One question received no clear answer: what would happen if the organization did nothing for another year?

The proposal described what the program wanted to build. It did not describe what the current state was costing, what decision leadership was being asked to make, or how leadership would know whether the investment was working. Without those, deferral was a reasonable response.

Chapter 3 described how to assemble the cost of the current state, and Chapter 21 described measures that can support a funding conversation. This chapter explains how to combine them into a case leadership can evaluate: framed in the terms executives already use, sequenced so each step can be approved on its own merits, and structured so subsequent funding decisions are informed by agreed evidence.

## Why Governance Proposals Are Deferred

Governance proposals often compete poorly for funding, even when the underlying need is sound. Several patterns contribute.

- **The request is measured against zero.** Without a current-state cost, the investment appears as new spending against an apparent baseline of nothing, as Chapter 3 described.
- **The proposal leads with the program's vocabulary.** Maturity levels, schedule structures, and control frameworks describe the program's work. They do not answer the questions executives are accountable for.
- **The request is framed as a technology purchase.** A platform is easier to describe than an operating capability, but a technology request invites a technology evaluation and a one-time decision.
- **Everything is requested at once.** A multiyear plan presented as a single commitment asks leadership to accept risk it cannot yet assess.
- **Benefits are stated as certain.** Projected savings presented as though already realized can undermine the credibility of the entire case when they are questioned.
- **The run cost is missing.** A proposal that funds the build without funding the ongoing roles and activities leaves the capability to decay after the project closes.
- **The case belongs to one function.** A proposal from the records function alone can read as a departmental request rather than an enterprise decision.

Each pattern can be addressed. Together they explain why a sound proposal can receive a sympathetic hearing and no funding.

## Answer the Questions Leadership Is Accountable For

Executives are generally accountable for cost, risk, resilience, and the organization's ability to pursue its priorities. A governance case is more likely to be evaluated on its merits when it answers questions in those terms, using figures the program can support.

*Figure 28.1. Executive questions and what the program can show.*

| Executive question | What the program can show | Where it is developed |
|---|---|---|
| What is the current state costing us? | An annual run rate and the cost of a further year of delay, each as a range, with contingent risk shown separately | Chapter 3 |
| What is our exposure? | Indicators of exposure, including sensitive record count within a stated scope and unreconciled volume the organization cannot yet attribute | Chapters 4 and 21 |
| Can we answer when we are asked? | Measured time to produce evidence for a traced rule or answer a baseline question | Chapters 1, 6, and 22 |
| How much of our in-scope information has verified retention treatment? | Coverage in four bands, with exclusions stated | Chapter 21 |
| What is unresolved, and who owns it? | The committee view of open items, aging, and accepted limitations | Chapter 20 |
| What else depends on this? | Funded initiatives slowed or enlarged by the current state, such as migrations, acquisitions, and AI adoption | Chapters 3, 14, 26, and 27 |
| What will change if we fund this, and how will we know? | Phase outcomes, the measures that will show them, and the criteria for releasing further funding | This chapter |

Each answer should be traceable to a method and a source. Figures supplied by the function that owns them, as Chapter 3 recommends, are more likely to be accepted than figures estimated on that function's behalf. Where a figure is an estimate, say so and show the range. Where a measure covers only part of the information environment, state the scope. Identify overlaps so the same cost is not counted twice. Keep gross current-state cost, avoidable cost, released staff capacity, and cash savings distinct, and do not add contingent exposure to realized operating cost.

The last question matters most. Leadership can approve a request with uncertain benefits if it can see how uncertainty will be reduced and when it will be able to decide again.

## Infrastructure Has a Run Cost

Governance work is often funded through projects: a defined scope, a budget, and an end date. Projects are well suited to building capability. They do not, by themselves, fund the continuing work needed to sustain it.

The capabilities described in this book require continuing work: maintaining the schedule and decision records, reconciling the register and ownership, operating change control, testing controls, maintaining measures, and managing open items. Chapter 27 described the roles that perform that work. When a program is funded only as a project, those roles may disappear when the project closes, and the controls it built begin to decay.

Present the case in two parts. The build cost covers the work needed to establish or remediate a capability. The run cost covers the roles, time, and activities needed to keep it operating, based on the operating model. Leadership may fund the two differently, often from different budgets, but both should be visible in the same decision. If the continuing work is not funded, the capability can begin to degrade after the implementation project ends.

Base recurring provision on the work and delivery arrangement, not only on new positions. Explicit allocations from existing functions or shared services can provide capacity where the responsibility, time, and support are agreed. Assess the capability and reviewer support described in Chapter 31 before making the commitment. Annual budget authorization can support a deliberately recurring operating model; temporary support and its end date should remain visible.

This is the practical meaning of treating governance as infrastructure rather than an initiative. Infrastructure is expected to have a continuing operating cost, and its value is judged by whether it keeps working, not by whether a project finished.

## Stage the Funding Against Evidence

A multiyear plan presented as one commitment asks leadership to accept a large, uncertain decision. A staged request asks for a smaller decision now, with defined criteria for the next one.

Divide the roadmap into phases, each with a stated outcome, a cost, and the measures that will show whether the outcome was achieved. Define the criteria for the next funding decision before the first phase begins. Criteria might include verification of the applicable treatment within a defined scope, recorded exit decisions for a set of legacy systems, a measured reduction in evidence response time, or a reduction in open items above a stated age. Verification is established for scoped treatments within pairings; coverage is reported for the corresponding in-scope volume. Draw on evidence the program already produces where it fits the decision, rather than build a separate reporting exercise solely for the funding request.

Figure 28.2 shows how a staged roadmap might be expressed. The phases, measures, and criteria are illustrative; each organization sets its own based on its gaps, exposure, and dependencies. The phases identify the main focus of investment, not when evidence or maintenance first becomes

necessary. Begin the required records, ownership, and checks when the corresponding work begins; later phases deepen or extend them.

*Figure 28.2. An illustrative staged roadmap with release criteria.*

| Phase | Outcome | Measures reported | Criteria for the next funding or continuation decision |
|---|---|---|---|
| 1. Foundation | Register and implementation matrix established for high-exposure categories; exit decisions recorded for legacy systems within a funded migration | Unreconciled volume; exit decisions recorded; unowned repository count | Register reconciled for the phase scope; exit decisions recorded for every legacy system in scope; owners confirmed for in-scope repositories |
| 2. Execution | Applicable configuration or procedure verified for phase 1 treatments; category-level disposition authorization and first disposition cycle | Coverage bands for phase scope; disposition backlog and overdue age; configuration lag | Applicable treatment verified within each pairing's defined scope; first cycle complete with records of the approved information outcome, including the terminal state for destruction; backlog trend for two periods |
| 3. Evidence | Independent testing and committee reporting deepened; scope extended to the next set of categories | Control test coverage; test exception aging; evidence response time | Tests completed and results, limitations, and responses considered; exceptions tracked against action or review dates, with overdue items escalated; response time compared with baseline |

Each row states what leadership is funding, how it will see the result, and what evidence will support a next-phase, continuation, or renewal decision. Criteria may include quantitative thresholds, completed decisions, verified implementation, or other evidence appropriate to the phase. Completed testing must be read with its findings, limitations, unresolved conditions, and authorized response; it does not predetermine a favorable decision. The point is to agree the basis for the decision in advance, not to force every phase into a numeric threshold.

Keep approval in principle, authorization, and release of funding distinct. Leadership may endorse the full plan while releasing each phase only after considering its criteria, resources, and dependencies. Where criteria are not met, report why and what is proposed: continue, adjust,

restrict, defer, or stop. Meeting criteria supports the decision; it does not itself release funds. Urgent controls or required work need their defined escalation and interim response rather than waiting solely for a routine funding gate.

Report realized results separately from projections. A cost avoided because a legacy system was retired can be stated once the retirement is complete and its costs have stopped. A projected reduction in discovery scope remains a projection until a matter demonstrates it. Programs that distinguish the two tend to retain credibility when later requests are reviewed.

## Sequence the Roadmap by Gap, Exposure, and Dependency

A roadmap leadership can understand explains why the phases come in the order they do. Three considerations usually determine the sequence.

- **The widest material gap in the maturity profile.** Chapter 5 argued that the widest material gap between dimensions is often the most informative finding. A program with strong structure and little evidence has a different first phase from one with strong execution and no inventory.
- **Exposure.** Categories and repositories with regulatory weight, litigation history, or sensitive content generally come before lower-exposure work, so that a slowed program still leaves the organization better protected.
- **Dependencies.** Some capabilities depend on others. Coverage cannot be measured reliably without an inventory, testing cannot enumerate populations without a register and matrix, and AI connections depend on information readiness. The sequence should respect these dependencies rather than promise results that depend on work not yet funded.

Attach governance steps to initiatives the organization has already funded wherever possible. A data center exit, a platform migration, an acquisition integration, or an AI rollout each create a moment when governance decisions must be made anyway. Chapter 3's vignette showed a migration that could not proceed without disposition decisions. Funding the governance step within the initiative that needs it can be easier to approve than a separate request, and it demonstrates value in the terms of a project leadership already supports.

## Build the Case Across Functions

A case presented by the records function alone can be read as a departmental request. A case supported by legal, technology, security, privacy, and finance is more likely to be read as an enterprise decision.

Agree the figures, measures, and phase outcomes with those functions before the proposal reaches leadership. Finance should own or validate the cost model, as Chapter 3 recommends. Security can present exposure measures through its own reporting channel, as Chapter 4 recommends. Technology leaders can confirm the dependencies with platform and migration plans. The sponsor

should be someone with authority over the decision being requested, and the decision rights described in Chapter 27 should identify who that is.

Where a proposal is declined, record the decision and its known financial and operating consequences. Chapter 3's current-state cost remains part of that record and informs the next request. Where the resulting gap requires risk acceptance or interim restrictions, obtain the decision through the applicable authority in Chapters 20 and 27. Budget refusal alone does not supply that acceptance or waive a requirement. Keep the underlying work visible with its responsible owner and response.

## What Works in Practice

A city government with roughly nine thousand employees had asked three times, in three consecutive budget cycles, for funding to replace its records management system. Each request was deferred. The requests described the system's age and the program's needs. None described what the current state was costing the city.

For the fourth request, the records manager worked with the budget office to assemble a cost model from existing figures. It included the annual cost of maintaining eighteen legacy systems retained because no one could authorize their retirement, staff time spent responding to records requests from the public, and outside counsel costs for collection in recent litigation. Each figure was supplied by the department that owned it and presented as a range, with the cost of a further year of delay shown alongside. Contingent exposure from sensitive information in unowned shared drives was described separately, without a speculative value.

The request was also restructured. Instead of a system replacement, it asked for three things: governance work attached to the city's already funded enterprise resource planning replacement, including exit decisions for the eighteen legacy systems; a smaller first phase of schedule restructuring; and three ongoing positions in the operating budget to run the program. The request proposed that release of the second phase depend on criteria agreed in advance: exit decisions recorded for all eighteen systems, a measured reduction in response time for records requests, and verification of the applicable retention treatment within the first-phase scope.

The council approved the first phase and two of the three positions. The declined position was recorded in the budget documents with the operating gap it left. Fourteen months later, the program reported against its release criteria. Exit decisions had been recorded for all eighteen systems and eleven had been retired, with their maintenance costs ended. Median response time for records requests had fallen. The applicable treatment within the first-phase scope had been verified. The second phase was released in the following budget cycle, along with the third position.

The figures and timeframes in this example are illustrative, not benchmarks. Funding structures, approval processes, and release criteria should reflect the organization's budget practices and the capabilities involved.

## Implementation Steps

Assemble the current-state cost with finance. Use the Chapter 3 method, with figures supplied by the functions that own them, presented as ranges, and a cost of delay alongside. Show contingent exposure separately.

Answer the executive questions. Prepare a concise answer to each question in Figure 28.1, with its method, source, and scope stated.

Separate build cost from run cost. Base recurring provision on the roles, activities, available capacity, and support arrangements in the operating model. Present build and run costs together even where different budgets fund them, and identify temporary provision and its end date.

Stage the roadmap and define decision criteria in advance. Give each phase an outcome, a cost, and supporting evidence, and agree what supports the next funding or continuation decision. Begin required evidence and maintenance with the activity, not only in a later phase.

Sequence by gap, exposure, and dependency, and attach to funded initiatives. Explain the order of phases and identify governance steps that belong within migrations, acquisitions, decommissioning, or AI rollouts already approved.

Build agreement across functions before the proposal is presented. Confirm figures, measures, and dependencies with finance, legal, technology, security, and privacy, and identify a sponsor with authority over the decision.

Report realized results separately from projections, and record declined decisions with their known consequences. Obtain any required acceptance or interim response separately through the applicable authority.

## What Good Looks Like

In an organization where governance is funded well, proposals are evaluated against a documented current-state cost rather than against zero. The case answers the questions leadership is accountable for, in figures that functions outside the program have supplied or validated.

Funding decisions address both building and running the capability, including agreed allocations from existing functions. The roadmap is staged, each phase has outcomes and evidence requirements, and further release follows an authorized decision against the agreed criteria. Governance work is included in initiatives already funded. Reports distinguish achieved results from projections, and declined requests retain their consequences and any separately authorized response.

The clearest sign is that funding decisions become routine. The program reports against criteria leadership has already accepted, and the next decision draws on evidence produced by operating the program rather than on a new argument.

## What Proves It

The evidence for this chapter shows how the case was built and how the funding it secured was used.

- A current-state cost model maintained with finance, with each figure attributed to its source function and presented as a range, and a cost of delay.
- A case document answering the executive questions, with the method, source, and scope of each figure.
- A statement of build and run costs for a defined planning period, with required recurring work, capacity, approved provision, and temporary support identified.
- A staged roadmap with outcomes, measures, and evidence criteria for each funding or continuation decision, agreed before the corresponding phase starts.
- Phase reports against the agreed criteria, including findings and limitations, realized results versus projections, and the decisions and releases recorded separately.
- Records of declined or partially approved requests and their financial or operating consequences, linked to separately required acceptance, interim action, or escalation.

The chapter's priority measure is unfunded run cost: the required recurring work without approved operating provision for the stated planning period, valued as a range with its basis. State the work in scope, currency, budget horizon, reporting date, total required run cost, and approved provision counted against it. Include explicit allocations from existing functions where they support that work. Show temporary support and its end date separately from continuing provision and work never funded; do not assume the entire gap is temporarily supported.

Read the funding diagnostic with the number of phases whose decision criteria have been met, evidenced, and reported, against the population due for a funding or continuation decision by the cutoff. Distinguish evidence completed, criteria met, decision recorded, and funds released. Show phases not yet due separately and explain changes in scope or method. Annual authorization can support recurring provision; neither diagnostic establishes that funded work occurred or that controls operated successfully. Both remain chapter-level measures, not additions to the standing library.

## Business Value

It supports better-informed funding decisions. Leadership can compare an investment with the documented cost of the current state and the cost of delay, rather than with zero.

It reduces the risk of large, uncertain commitments. Staged funding with release criteria allows leadership to invest progressively and to stop or adjust based on evidence.

It can help sustain capability after implementation. Making recurring provision explicit alongside the build decision reduces dependence on whatever capacity remains when a project ends.

It builds credibility over time. A program that distinguishes realized results from projections, reports against agreed criteria, and records declined decisions is easier to fund the next time it asks.

## Ask for a Decision Leadership Can Make

A governance proposal often fails not because the need is unclear but because the decision is too large, too abstract, or framed in terms leadership does not use.

A stronger case states what the current state costs, what the first step will change, how leadership will know, and what it will cost to keep the result working. It asks for a decision that can be made now and sets the terms for the next one.

A program that reports against evidence leadership already agreed to is easier to fund again than one that has to rebuild the case from the beginning each year.

*Next: Chapter 29 turns from securing funding to using it well: a phased implementation model, a first ninety days, and the mechanisms that move a governance effort from pilot to program.*

# Chapter 29: From Pilot to Program

Governance pilots can succeed on their own terms and still fail to become programs.

A typical pattern runs as follows. A bounded scope is chosen, a capable team is assigned, and the pilot delivers what it promised: a register for the pilot units, verified controls, a first disposition cycle, and a positive report to the sponsor. The next wave is planned but not started. The pilot team returns to other work. A year later, some of the pilot's controls have drifted, and the program is proposing another pilot.

Chapter 28 described how to secure funding in stages. This chapter addresses how to use it: a phased model for moving from a first implementation to a program that operates across the organization, a practical plan for the first ninety days, and the mechanisms that reduce dependence on the particular people who built it.

## A Pilot Tests the Operating Model

Pilots are often designed to prove that something works: that a platform can apply retention, that disposition can run, that a classifier can reach acceptable agreement. Those questions matter, but they do not establish whether the operating model can work outside the pilot.

Scaling depends on whether the operating model works beyond the pilot's conditions. Can accountable owners be assigned and confirmed? Are authority, capability, required review, and time available under the intended delivery arrangement? Which decisions can be made locally and which belong elsewhere? How much effort does each unit require, and what support must come from central teams or partners? A pilot designed only to prove technology may leave these questions unanswered.

Design the pilot to answer those questions. Record who performed each role, how long each step took, what was escalated and why, and where temporary support differed from the planned continuing arrangement. Use Chapter 31's capability assessment during planning. Those records allow later waves to be planned from evidence rather than from the pilot's success alone.

## Choose a Representative Pilot

A well-organized unit with a supportive leader can be a useful pilot for a bounded question. Its results show what is possible under those conditions, but may not establish the effort or support other units will need. Before generalizing the results, assess the differences in the units that will follow.

Choose a scope that tests the systems, conditions, and delivery arrangements expected in later units. Where the organization varies significantly, a pilot in one typical unit and a smaller check in a

more difficult unit may reveal useful differences. State the selection basis and limits rather than treat one chosen unit as evidence of readiness everywhere.

Staff the pilot to test the intended continuing arrangement, whether local, central, shared-service, or partner-supported. Local accountability does not require every task to be performed locally. Where temporary specialists perform work intended for others, record that dependency and assess the planned handover before temporary support ends.

Before handing over a wave, confirm who is accountable, what authority applies, whether the required capability and reviewer support are available, and how the recurring work is resourced. Link the Chapter 27 assignments, Chapter 28 provision, and Chapter 31 task assessment rather than duplicate them in a new register.

## A Phased Model

Moving from pilot to program is usually more reliable in phases, each with a purpose and criteria for moving on. The phases below describe implementation. They can align with the funding stages Chapter 28 described, but the two are not the same: a funding stage may cover several implementation phases, and the criteria for releasing funding are decided by leadership rather than by the implementation team.

*Figure 29.1. A phased model from pilot to program.*

| Phase | Purpose | Criteria for moving on |
|---|---|---|
| Define and assign | Set scope, decision rights, owners, the intended delivery arrangement, and baseline measures | Authority and owners confirmed; required capability, review, and capacity identified; baselines and methods recorded |
| Pilot | Implement in one representative scope and test the operating model as well as the controls | Pilot outcomes verified; effort and escalations recorded; continuing delivery and handover requirements assessed |
| Expand in waves | Repeat unit by unit using a versioned wave template reviewed after each wave | Outcomes verified; template review recorded and revisions made where needed; effort measured and material variation explained |
| Institutionalize | Embed work in standing roles and processes without depending on particular founders or a temporary team | Required mechanisms embedded, with evidence that the applicable onboarding, budgeting, change, role, and reporting processes are being used |

| | | |
|---|---|---|
| Sustain | Operate, test, and improve on a defined cadence | Post-wave checks completed on the defined cadence; findings and limitations recorded; measures reported through standing channels |

The criteria describe the evidence supporting progression, not a timetable or automatic permission to advance. Meeting them supports the next decision; applicable authority, funding release, resources, and dependencies must also be in place. Where criteria are not met, report why and the proposed response. Urgent or required work follows its defined escalation and interim arrangements rather than waiting solely for a routine phase gate.

The phases identify the main emphasis, not when ownership, evidence, or maintenance begins. Establish the records and checks needed for each activity when it starts. Later phases extend those arrangements, and ongoing work must be handed over before temporary support ends.

## Build a Wave Template

The most useful product of a pilot is often not its results but a template for repeating them. A wave template describes what each subsequent unit needs to do, in what order, with which roles, and with what evidence of completion.

A practical template includes the unit scope, assignments to confirm, applicable authority, required capability and reviewer support, register and implementation-matrix entries, and controls to verify. It identifies local and escalated decisions, measures, role training, and closing criteria. State the central or partner support provided, its duration, and the continuing arrangement. Give the template an identifiable version, effective period, and approval reference, and link each wave to the version it used.

Review the template after each wave and revise it where findings warrant a change. Record unexpected effort, temporary support, recurring questions, and the review outcome, including why no change was needed. Assess whether a revision affects completed or active waves or only future work, and record any required transition, restriction, or correction through the existing change and open-item processes.

## The First Ninety Days

A defined plan for the first ninety days from the authorized implementation start helps convert the funding decision into visible progress. Approval in principle does not establish that funds, people, and dependencies are ready. Confirm the starting conditions and use the plan to build traction rather than promise immediate scale.

*Figure 29.2. An illustrative plan for the first ninety days.*

| Period | Focus | Examples of completed work |
|---|---|---|

| | | |
|---|---|---|
| Days 1 to 30 | Establish authority, ownership, and baselines | Authorized start and sponsor confirmed; pilot decision rights and owners confirmed; delivery, capability, and support needs identified; pilot scope selected; baseline trace, response-time, and first-pass coverage evidence recorded |
| Days 31 to 60 | Implement in the pilot scope | Register and matrix entries completed for the pilot scope; selected configurations verified; exit decisions recorded for pilot-scope legacy systems; role-based training completed, with initial task performance checked where the role begins operating during the pilot; open-items view started |
| Days 61 to 90 | Report, learn, and prepare the next wave | First report to the sponsor in the committee format; progress against the pilot's moving-on criteria reported, with any applicable funding release criteria reported separately; wave template drafted from pilot records; decision requested on the next wave |

The plan is illustrative. What can be completed in ninety days depends on the scope, the environment, and the people available. What matters is that each period ends with work that can be shown to have been completed, and that the third period ends with a decision about what comes next rather than a general sense of progress. The moving-on criteria are implementation criteria. Any funding release criteria established under Chapter 28 remain separate and are reported alongside them where applicable.

## Adoption: Communication, Training, and Accountability

A program scales through the people who carry its roles in each unit. Three mechanisms usually determine whether they do.

- **Communication in business terms.** Explain what changes for each audience and why, in the terms of their work: where to save a record, who approves a rule exception, what happens to a legacy system. Communicate through managers as well as central channels, and repeat the message as waves proceed rather than announcing the program once.
- **Role-based training.** Train each role for the tasks it performs. Control owners need to know how to confirm a configuration and record evidence. Information owners need to know how to make and record lifecycle decisions. Employees need to know the few actions expected of them. Attendance shows participation. Defined task checks show whether people can perform the work expected of their role. Keep the two separate.
- **Accountability in standing expectations.** Include governance responsibilities in role descriptions, workload expectations, and, where appropriate, performance objectives, as Chapter 27 described. A role treated only as an informal extra is more likely to receive attention only when a problem surfaces.

Provide a route for questions and feedback from the units. Resolve routine clarification through the appropriate support path. Where feedback identifies a confirmed gap, unresolved condition, or required change, record it through the applicable source process and link it to the Chapter 20 open-items view.

## Institutionalize the Program

Institutionalization reduces dependence on particular founders, sponsors, or a temporary implementation team. It does not remove the need for continuing program coordination, sponsorship, or oversight. The responsibilities described in Chapter 27 need standing holders and support as people change.

- Governance roles and decision rights are written into charters and role descriptions, as Chapter 27 described.
- The onboarding gate from Chapter 2 is part of procurement and architecture review, so new systems enter through a governed intake with the required decisions and implementation work identified.
- The run cost is part of the regular budget cycle, as Chapter 28 described, rather than renewed as a special request.
- Organizational change, platform upgrades, and new AI uses are recognized change triggers under Chapter 19.
- Measures are reported through appropriate standing enterprise channels, which may include risk, audit, or security reporting, rather than only through program updates.
- Testing runs on a defined cadence under Chapter 22, including scoped post-wave checks that distinguish alignment with the approved treatment from evidence of operation.

A practical test is to ask what would happen if the program lead left. If roles, decisions, reporting, testing, and funding continue through the people and processes already assigned to them, the program has moved beyond dependence on its founders.

## Why Pilots Stall

Pilots that succeed and then stop tend to share several conditions.

- **Favorable pilot conditions were generalized without assessment.** The selected unit answered a narrow question, but differences affecting later-wave effort were not examined.
- **Temporary support replaced the intended delivery arrangement.** The pilot did not establish whether the people or partners expected to carry the work later could do so.
- **The pilot tested technology rather than the operating model.** Questions about ownership, decision rights, and effort per unit remained unanswered.
- **No template was produced.** Each later wave had to be designed from the beginning, and none was.
- **Success was reported as activity.** The sponsor heard what the pilot did rather than what it established about scaling.

- **The pilot team dispersed.** Staff returned to other work at the pilot's close, and no run budget sustained the roles.
- **Controls were not checked after the pilot.** Upgrades, staff changes, and new systems allowed pilot controls to drift unnoticed.

The result can be an organization with a successful pilot on record and little lasting change in its information environment.

## What Works in Practice

A veterinary hospital group with roughly seven thousand five hundred employees operated about four hundred clinics in nine regions, most of them acquired over the previous decade. Clinics used several practice management systems and kept client and medical records under local practices that varied widely.

The group's first pilot covered one region of thirty-five clinics. It was chosen because its regional leader was supportive and its clinics were well organized. A central team of three, supported by outside help, built the register, configured retention in the region's practice management system, and ran a first disposition cycle. The sponsor received a positive report.

Nine months later, no second region had started. The central team had been reassigned. An upgrade to the practice management system in two pilot clinics had reset their retention settings, and no one had noticed. A review found that the central team had performed nearly every local role during the pilot, so no clinic staff had experience carrying them.

The group restarted with a different design. The second pilot was chosen from a region of average organization and a mix of systems. Practice managers were named as information owners and regional operations leads as control owners, with the central team coaching rather than performing the work. The team recorded who did each task, how long it took, and what was escalated. At the end of the second pilot, it produced a wave template covering scope, roles, register entries, controls to verify, local and escalated decisions, training by role, and closing criteria.

The restarted rollout proceeded at a pace of one region per quarter. Training for practice managers focused on the lifecycle decisions they would make and took under an hour. Governance duties were added to the role descriptions of practice managers and regional operations leads. Practice management system upgrades were added to the change triggers, with a check that retention settings survived each upgrade. Six months after each wave, a post-wave check examined a defined sample of controls to determine whether they still matched the approved treatment and whether the available evidence supported operation within the period examined.

Eighteen months after the restart, six of the nine regions had been onboarded: the redesigned pilot region, the original thirty-five-clinic region after reassessment against the revised criteria, and four additional regions. Later post-wave checks found drift in two clinics, traced to changes to existing

platforms that had bypassed the change-control process. Both corrections were recorded. Central effort per wave had fallen after the first two waves as the template and local experience matured.

The figures and timeframes in this example are illustrative, not benchmarks. The pace of waves and the effort each requires depend on the organization's structure, systems, and the people available to hold local roles.

## Implementation Steps

Design the pilot to test the operating model. Define the questions about ownership, decision rights, effort, and escalation the pilot must answer, and record the information needed to answer them.

Choose a scope that tests later-wave conditions and the intended continuing delivery arrangement. Identify differences that limit generalization, and assess capability, reviewer support, authority, and capacity before handover.

Adopt a phased model with evidence-based criteria. Define each phase's purpose and moving-on criteria, distinct from funding-release criteria. Proceed only with applicable authorization, resources, and dependencies in place; required evidence and maintenance begin with the work.

Produce and maintain a versioned wave template. Review it after each wave, revise where warranted, and record no-change conclusions. Link waves to the version used and assess the effect of revisions on active or completed work.

Plan the first ninety days from the authorized implementation start. Set work to complete in each thirty-day period and end with a report and a decision on what comes next, showing implementation and funding criteria separately.

Support adoption through communication, role-based training, and standing accountability. Communicate in business terms through managers, train roles for their tasks, and include governance responsibilities in role descriptions, workload expectations, and, where appropriate, performance objectives.

Institutionalize and check. Embed governance in charters, onboarding, budgeting, change triggers, and standing reporting, and run scoped post-wave checks on a defined cadence to assess whether approved treatments remain aligned and what the available evidence supports about operation.

## What Good Looks Like

In an organization that has moved from pilot to program, waves use a versioned template reviewed against experience. Accountable owners are supported by the planned local, central, shared-service, or partner arrangement. Required capability, review, authority, and resources are confirmed before handover. Progression follows the applicable decision, informed by criteria and dependencies rather than by a calendar alone.

Communication reaches each audience in the terms of its work. Training participation and defined task checks are reported separately. Governance responsibilities appear in role descriptions and workload expectations and, where appropriate, performance objectives. Charters, onboarding, budgeting, change control, reporting, and testing each carry part of the program, so that it continues when individual people change.

The clearest sign is that onboarding is followed by continuing maintenance and checking. Post-wave results describe alignment and operation within the scope examined, while unresolved findings remain visible until their appropriate resolution is supported.

## What Proves It

The evidence for this chapter shows how the program is repeated and what supports the conclusion that its results are being maintained.

- A pilot design stating the operating-model questions to be answered, and pilot records of roles performed, effort, and escalations.
- A phased plan with moving-on criteria, the evidence considered, and recorded progression decisions distinguished from funding authorization and release.
- A wave template with version, effective period, and approval references; review outcomes, including no-change decisions; links from waves to the versions used; and assessed effects of revisions.
- A first-ninety-days plan from the authorized start and its closing report, including the next-wave decision and any conditions still unmet.
- Role descriptions and workload expectations that include governance responsibilities, performance objectives where appropriate, and training records that distinguish participation from defined task checks.
- Evidence of continuing delivery: charters, governed onboarding, recurring provision, change triggers, standing reports, and handover records linking assignments to authority, capability, review support, and capacity.
- Post-wave check records by onboarded unit, with due dates, scope, methods, findings, limitations, and links to all unresolved drift and supported corrections, including findings from earlier checks.

The chapter’s priority measure is the number of onboarded units with a current post-wave check completed within the defined interval. State the total in-scope unit population and the number onboarded at the reporting cutoff. For onboarded units without a current check, distinguish a first check not yet due, an overdue or incomplete check, and status not yet established. Show units awaiting onboarding separately. Define the unit, check scope, method, interval, and counting rules; keep regions, clinics, repositories, and controls distinct.

Read the checking count with the number of units affected by unresolved drift from any check, not only the latest sample. Retain earlier findings until their resolution is supported; a later check of

different controls does not close them. Checking states and affected-unit counts can overlap and should not be added together. Explain population or method changes. A completed check establishes that the defined review occurred; its findings support only the scope examined, not every control in the unit. These remain chapter-level measures.

## Business Value

It can extend value beyond the pilot scope. A program that scales through waves and maintains its results can produce benefits beyond a single successful demonstration.

It can reduce repeated implementation effort. A maintained template and growing local experience can reduce the central effort required for similar later waves.

It can help protect results after implementation. Post-wave checks and standing mechanisms can reduce the likelihood that control drift goes unnoticed after the project team moves on.

It reduces dependence on particular individuals. Embedding governance in charters, budgets, roles, and reporting gives the work a better basis for continuing through changes in leadership and staff.

## A Program Outlasts Its Pilot

A successful pilot shows that something can be done once, within a defined scope and set of conditions. A program shows that the work can be repeated by the people who own it and that the resulting controls can be maintained and checked over time.

The difference is not only technical. It depends on whether the pilot tested the operating model, whether its lessons became a repeatable implementation method, and whether standing roles and processes took over the work.

That is the transition from pilot to program: not another successful implementation, but a way to repeat the work without rebuilding the approach each time.

*Next: Chapter 30 turns to governance at speed: pre-positioned controls, modular policy, and clear decision boundaries that let the program keep pace with the business without weakening oversight.*

# Chapter 30: Governance at Speed

A product team planned to launch a new service in six weeks. The governance review queue for new systems was running at nine. The team launched on schedule, registered nothing, and accepted the platform's default retention for the information the service created. No one had set out to bypass governance. The approval path was longer than the business timeline, and the business did not wait.

Situations like this are often described as a conflict between governance and agility. Often, the conflict is between the way governance has been designed and the pace of the organization it serves. Where every decision is treated as new, routed to a central group, and approved before anything proceeds, governance becomes a queue. When the approval path is longer than the business timeline, teams have a stronger incentive to proceed without it.

Chapter 27 defined roles and decision rights, and Chapter 29 described how a program scales. This chapter addresses how a program keeps pace: by approving recurring decisions within explicit conditions, maintaining shared policy elements, and defining usable decision boundaries. Patterns can replace repeated case-by-case approval without removing required pre-use checks or subsequent oversight. Decisions and controls outside their scope retain their established review requirements.

## Where Time Is Lost

Governance delay tends to accumulate in a few predictable places.

- **Repeated decisions.** Many requests reaching a governance group are variations of decisions already made: another team workspace, another provider of a familiar type, another low-risk AI use. Each is reviewed as though it were new.
- **Unclear authority.** When people cannot tell who may decide, the request moves upward until it reaches someone who can, or waits.
- **Policies that require interpretation.** Guidance written for review rather than for use leaves teams needing an interpreter before they can act.
- **Late involvement.** Governance hears about an initiative when it is nearly ready to launch, when any finding can only delay it.
- **No expected response time.** Requests enter a queue with no stated time for a decision, so teams cannot plan around it.

Each of these can be addressed without reducing oversight of the decisions that matter. The aim is to spend review capacity on decisions that need judgment, rather than on decisions that have already been made.

## Decide Once, Apply Many Times

A pre-approved pattern is a decision made for a class of situations rather than a single case. It states the conditions for use, required treatment or controls, records, and conformity checks. A team can act without a separate approval when its situation meets those conditions and the required pre-use checks are complete. Its use record identifies the basis for that determination; the record alone is not independent verification of conformity or proof of successful operation.

Pre-approval removes the need to repeat the underlying decision. It does not remove checks required for the particular use. Specify which applicability and implementation checks precede action and which are subsequent oversight. Previously approved configuration is reusable design evidence, not automatic evidence that every deployment matches it.

Patterns are decisions in the sense Chapter 18 described. Use the Appendix F decision record for the basis, rationale, authority, conditions, required checks, and review point. Give each approved version an effective period and reference, and record which version governed each use. A pattern without conditions does not establish a bounded pre-approval.

*Figure 30.1. Illustrative pre-approved patterns.*

| Pattern | Conditions for use | What the team records | Checks and subsequent oversight |
|---|---|---|---|
| New repository of a known type | The repository type, categories, and platform match an approved configuration | Register entry, owner, pattern version, applied configuration, responsible party, and required pre-use verification reference | Required implementation verification before reliance; subsequent sampling of conformity with the pattern |
| Standard provider agreement | The relationship type and information profile match the approved pattern, with no material departure from the required governance terms or evidence obligations | Relationship type, pattern version, executed-contract reference, and basis for matching required terms | Required terms checked before reliance on the agreement; later agreement sampling. Provider performance requires separate evidence |

| | | | |
|---|---|---|---|
| Lowest-tier AI use | The use meets the specifically approved lowest-tier criteria; each required source connection is approved for that use | Use, owner, pattern version, risk-classification basis, sources, and interaction-record treatment | Required use and connection assessments before operation; later sampling of eligibility and supporting records |
| Administrative rule change | The change meets the administrative class in Chapter 19 and alters no lifecycle outcome | Change record, pattern version, reason, authority, and approving person or authorized role | Classification against Chapter 19 before applying the pattern; periodic review for misclassification |

Define the criterion, timing, and evidence for each check. Configuration verification establishes alignment with the approved treatment, not successful operation. Agreement review establishes whether required terms were included, not whether the provider performed them. A review of use records and supporting evidence can establish eligibility within the scope examined, not acceptable system performance. Apply the relevant operating evidence separately. For AI, Chapter 24 supplies interaction-record treatment, Chapter 25 risk classification and system evidence, and Chapter 26 use-specific source readiness; classifier-specific requirements remain in Chapter 23.

The patterns in the figure are examples. Each organization identifies its own from the requests it receives most often. A useful starting point is to sort recent requests by the decision actually made. Requests that repeatedly received the same answer are candidates for a pattern.

Patterns need maintenance. Review them on a defined cadence and when relevant conditions change. Record the result, including a supported no-change decision, and retain prior versions. Distinguish a use outside a sound pattern from a defect in the pattern itself. Assess whether revision or withdrawal affects active deployments, earlier uses, or only future requests; record the necessary transition, restriction, or correction through Chapters 19 and 20. A new version does not, by itself, close earlier nonconformance.

## Structure Policy So Change Lands Once

Governance is slowed when a small change requires reopening a large document. A modular structure separates material that changes at different rates and has different owners.

Principles and policy statements typically change less frequently and may require senior approval. Standards and the retention schedule can change more often through the process in Chapter 19. Procedures and patterns can often be maintained more frequently by those authorized to operate them within defined limits. Keeping these layers distinct allows a procedure or pattern to be updated without re-approving the policy it implements.

The same principle applies within the schedule. Chapter 7 describes shared rule elements held once, and Chapter 10 describes global defaults with modeled jurisdictional variations. Structure can reduce duplicated maintenance when an applicable requirement changes. It does not establish that every affected configuration or procedure has been updated. Assess the effects and carry the approved change through the applicable transition and verification process in Chapters 11 and 19.

## Draw Decision Boundaries People Can Use

Chapter 27 described decision rights with limits. For speed, the limits must be clear enough that the person facing a decision can tell, without asking, whether it falls within their authority.

Boundaries are clearest when they are stated in terms of effect. Chapter 19's change classes illustrate the approach: a change that alters no lifecycle outcome can be approved within the program, while a change to period, trigger, or disposition action receives full review. Similar boundaries can be drawn for other decisions, such as whether a provider relationship involves information of a sensitivity the standard terms were not designed for, or whether an AI use falls within the lowest risk tier.

Every boundary needs a defined path for what falls outside it. State who receives the escalation and the expected time for a decision. A stated response time allows teams to plan and gives the program a measure of its own responsiveness. A boundary without a path only moves the queue.

Where the boundary is uncertain, route the decision to the authority designated for that uncertainty rather than stretching the delegation to fit. Clear boundaries reduce uncertainty; they do not remove the need for judgment about cases that sit near the line.

## Verify After, Not Only Approve Before

For decisions within a pattern, the organization need not repeat the approval already made. The team still records applicability and completes required checks before action. Subsequent sampling examines whether conditions, procedures, and record requirements were followed; it does not substitute for a prerequisite to reliance.

Subsequent checks use Chapter 22's discipline: a defined population, selection method, criteria, evidence, results, and limitations. Record nonconforming uses and evidence gaps through Chapter 20 with owners and dates. A missing required record can itself be a finding, but does not establish every aspect of the action. Determine whether the response concerns the use, guidance, delegation, or pattern, and verify the corrective work appropriate to that finding.

The practical change is fewer repeated approvals, not fewer required controls. Retained applicability evidence, any required pre-use verification, and proportionate subsequent checking support the delegated path. Keep conclusions limited to what those records and tests establish.

## What Does Not Change

Patterns and delegated paths do not replace decision and control requirements that already have their own defined process.

- Substantive and urgent rule changes continue through Chapter 19's required review and transition process.
- Category-level disposition authorization must be in place, and the applicable eligibility, encumbrance, and dependency checks still apply before execution, as Chapter 13 describes.
- Legal holds and releases continue through the scope, implementation, and verification process in Chapter 15.
- AI uses outside the approved pattern, including higher-tier uses or source connections that have not met readiness requirements, follow Chapters 25 and 26.
- Risk acceptances outside delegated limits go to the authority identified in the decision-rights statement.

Moving routine decisions onto patterns and delegated paths can preserve review capacity for the decisions and controls that still require judgment or separate verification.

## Engage Early

The simplest way to reduce governance delay is to be involved before a timeline is fixed. The onboarding gate from Chapter 2 serves this purpose for new systems. The same principle applies to new products, acquisitions, AI rollouts, and platform changes. Chapter 28 recommended attaching governance steps to initiatives the organization has already funded; engaging at the planning stage allows those steps to be scheduled rather than discovered.

Early engagement also allows a team to see whether its plans fit an existing pattern. A team that knows a pattern exists can design within it and avoid a separate case-by-case review.

## Why Governance Slows the Business

When governance slows the business, the cause is often structural rather than individual.

- **Every decision is treated as new.** Without patterns, the same decisions are reviewed repeatedly.
- **Authority is centralized by default.** Decisions go to a central group because no one has defined which can be made elsewhere.
- Policies and standards are sometimes written primarily for review rather than operational use. Guidance that requires repeated interpretation can create dependency on a small number of specialists.
- **Governance is engaged late.** Findings arrive when they can only delay a launch.
- **Response times are unstated.** Teams cannot plan around a queue whose length they do not know.

- **Delegation feels risky.** Without verification afterward, delegating a decision can feel like giving up control, so it is not delegated.

The result can be a program whose rigor is concentrated on the decisions least in need of it, while decisions made outside the program receive none.

## What Works in Practice

An urban mobility company with roughly three thousand employees operated shared scooters and bicycles in about sixty cities and launched in new cities or with new features almost every month. Each launch created new repositories, new provider relationships, and new flows of trip and customer information.

The governance review queue had grown to seven weeks. Launch teams increasingly proceeded without waiting. A review found that repositories created for recent city launches had been registered in fewer than half of cases, and that trip information in several of them was being retained indefinitely under platform defaults.

The team sorted a year of review requests by the decision actually made. Most fell into five recurring situations: a new city launch using the standard platform configuration, a new local provider of an established type, a new team workspace, an administrative change to a rule, and an AI use within its lowest risk tier for internal productivity. The team wrote each as a pre-approved pattern with conditions, required records, and a conformity check, and approved each through the decision forum with a review date.

Launch teams recorded their use of a pattern through the intake process. The program sampled a portion of pattern uses each month and verified them against the conditions. Requests outside a pattern went to full review with a stated response time of ten business days. New categories of personal information, changes to retention periods, and AI uses above the lowest tier continued to receive full review regardless of timing.

Within six months, most requests were handled through patterns, and teams using a pattern could proceed on the day they recorded it. Within the sampled pattern uses, the monthly review identified nonconformance in a small number of launches, including two where a local provider’s terms differed from the standard profile. Those were recorded as open items and corrected, and the provider pattern’s conditions were clarified. Review capacity freed from routine requests was redirected to the full-review path. For the full-review requests measured during the period, the time-to-decision distribution moved within the stated target.

The figures and timeframes in this example are illustrative, not benchmarks. The patterns, conditions, sampling rates, and response times an organization adopts should reflect its own requests, risks, and capacity.

## Implementation Steps

Sort recent requests by the decision made. Identify decisions that recur with the same answer, and treat them as pattern candidates.

Write and approve versioned patterns as decisions. Record conditions, treatment, pre-use checks, subsequent conformity checks, authority, effective period, and review point. Link each use to the version and evidence supporting its applicability.

Separate policy layers by rate of change and ownership. Keep principles, standards and the schedule, and procedures and patterns distinct, with approval appropriate to each.

State decision boundaries by effect, with an escalation path and response time. Use Chapter 19's change classes as the model, and define what happens to decisions outside each boundary.

Check pattern use and maintain its history. Complete required pre-use verification, then sample subsequent use on a defined cadence. State population, selection, findings, and limitations; distinguish use errors from pattern defects and assess existing uses when a pattern changes or is withdrawn.

Preserve established review and control requirements. Name the decisions and controls that continue through their defined processes regardless of timing or pattern availability.

Engage at planning. Build governance into intake for systems, products, acquisitions, AI rollouts, and platform changes, and make available patterns visible to planning teams.

## What Good Looks Like

In a program that keeps pace with the business, routine decisions are approved within explicit conditions and reused. Teams know the current pattern, record the basis for applicability, and complete any required pre-use checks without waiting for a repeated approval. Defined authorities, escalation paths, and response expectations address cases outside the pattern.

Recorded use, required verification, and subsequent sampling support oversight of the pattern path. Findings are assigned and followed through; decisions and controls outside the pattern retain their established processes. Shared rule structure reduces duplicated maintenance while changes still reach and are verified in their affected scope.

The clearest sign is that teams bring decisions to governance early because doing so is faster than working around it.

## What Proves It

The evidence for this chapter shows that speed was achieved through recorded decisions and scoped conformity checks, not through reduced oversight. Operating or performance claims still require the evidence appropriate to those questions.

- An analysis of recent requests by decision made, identifying pattern candidates.
- Pattern decision records in the Appendix F form, with version, effective period, authority, conditions, required records, pre-use checks, subsequent conformity checks, and review points.
- Pattern-use records identifying the governing version, applicability basis, and required pre-use evidence, linked to the resulting register, contract, AI-use, or change entries.
- Sampling records with population, selection method, sample size, cutoff, and separate nonconformance and evidence-limited findings, linked to open items and any assessed revision or withdrawal effects.
- Decision boundaries with escalation paths and response times, and request records retaining receipt, completeness, decision, and any pause dates and reasons.
- A list of decisions and controls that remain outside the pattern path, with records showing they followed their required process.

The chapter's priority measure is the number and share of sampled pattern uses found outside their approved conditions, against all uses sampled in the period. State the population, sample size, selection method, pattern versions, and reporting cutoff. Count each use once in that result and retain links to all associated findings. Report evidence insufficient to establish conformity separately. Missing required records are findings, but do not establish every aspect of the underlying action; identify overlap between evidence limitations and demonstrated nonconformance. Explain material changes in pattern mix or sampling before comparing periods.

Read conformity results with time to decision on the full-review path against the stated service expectation. Preserve request receipt, the date it was accepted as complete, the recorded decision, and pause dates and reasons. Report completed-request time from completeness to decision alongside the age of requests still open and material time awaiting completeness. State elapsed time and any adjusted time distinctly; show requester or dependency delays rather than letting them disappear from the clock. The measures describe sampled conformity and responsiveness, not the effectiveness of every resulting control, and remain chapter-level.

## Business Value

It can reduce avoidable delay while preserving required oversight. Teams can use approved decisions without repeating the approval, while retaining the prerequisites and checks appropriate to the use.

It gives teams a faster governed route for recurring decisions, reducing the incentive to proceed outside the process.

It can concentrate specialist judgment on decisions that actually require it. Capacity freed from repeated decisions can support more timely review of consequential ones.

Modular policy and defined decision boundaries can reduce the work required when requirements or business conditions change.

## Speed Comes From Decisions Already Made

Governance is sometimes asked to move faster by removing steps. Simply removing steps can remove oversight along with delay.

A more durable approach is to make recurring decisions in advance, define the conditions under which others may rely on them, and verify that those conditions were followed. Decisions outside those boundaries still receive the review they require.

The speed comes from decisions already made and conditions already defined, not from weaker control.

*Next: Chapter 31 turns to the people who operate these programs: how the information governance profession is changing, the skills teams need, and practical options for closing the gap.*

# Chapter 31: The Evolving Information Governance Professional

Consider a role centered on retention schedules, policy, and employee training alongside one responsible for supporting an operating governance program. The traditional work remains. The second role may also require platform knowledge, measurement, AI assessment, and change management. The title may be the same. The work has expanded.

The preceding chapters connect governance as written to governance as operated. Rules are implemented and verified, changes close against appropriate evidence, controls are tested, AI uses are recorded and reviewed, and patterns combine required checks with subsequent oversight. Each activity requires people able to perform it. Teams built primarily around policy, schedules, and traditional records operations may not yet cover all of that work.

This chapter describes the shift from authoring policy to operating programs, the skill areas an operating program draws on, and practical options for closing the gap. It focuses on the governance team itself. Chapter 29 addressed training for the business roles that carry out governance tasks.

## From Author to Operator

The traditional core of the profession remains necessary. Someone must still interpret requirements, design categories, and write policy that people can follow. What has changed is how much of the work follows publication.

- **From writing rules to confirming implementation.** Approval establishes the rule. Implementation is complete when the applicable configuration or documented procedure has been verified in the affected scope.
- **From answering questions to producing evidence.** The team needs records that show what was implemented and testing that supports conclusions about operation within a defined scope. Chapters 21 and 22 distinguish those questions.
- **From periodic review to continuous change.** Systems, requirements, and uses change throughout the year, and the change process in Chapter 19 runs continuously rather than at the annual review.
- **From a function to a service.** Business teams, technology teams, and AI projects depend on governance decisions to proceed. The team is judged by how reliably and how quickly it supports that work, as Chapter 30 described.

None of this makes the professional less important. It changes where their value is visible. In an operating program, value shows in controls that work, decisions that can be explained, and a business that brings its questions to governance early.

## Five Skill Areas

An operating program draws on five skill areas. The figure describes what each involves in practice and where the book relies on it.

*Figure 31.1. Skill areas an operating program draws on.*

| Skill area | What it involves in an operating program | Where the book relies on it |
|---|---|---|
| Legal and regulatory | Interpreting requirements into retention periods, triggers, and limits; recording the basis and rationale; working with counsel on privileged analysis and holds | Chapters 7, 10, 15, and 18 |
| Technical | Understanding how platforms apply retention, holds, and deletion; specifying and verifying configuration; assessing AI systems and their connections to information | Chapters 8, 11 to 13, 23, and 26 |
| Analytical | Designing measures and counting rules; sampling and testing; reading results with their scope, cutoffs, and limitations | Chapters 2, 21, and 22 |
| Change management | Engaging owners and business teams; communicating in business terms; sustaining adoption through turnover and reorganization | Chapters 27 and 29 |
| Product and service management | Treating governance services as products or services with users; managing intake and backlog; designing patterns; measuring response times and service performance | Chapters 28 and 30 |

The starting point will differ by team. A records-focused group may already hold substantial legal and regulatory knowledge while depending on technology or assurance partners for other work. Another team may begin with stronger technical or analytical capability. The assessment below is intended to show the actual capability coverage rather than assume where the gaps will be. Product and service-management skills become relevant where the program operates an intake, manages recurring requests, uses patterns, and measures responsiveness.

The list is not a set of job titles. A single role may draw on several areas, and a single area may be spread across several roles.

## Capability Coverage Belongs to the Team

No one person needs to hold every skill area in depth. The team, together with the partners it relies on, needs to cover the work the program performs.

A useful way to describe capability coverage is by level. At awareness level, a person understands what the skill involves and when to call on it. At working level, a person can perform the related tasks with review. At lead level, a person can perform the tasks independently, review the work of others, and improve the method. The required level belongs to the task, not permanently to the person. The same individual may lead one type of work, perform another with review, and need only awareness of a third. Appendix E provides a skills matrix built on these levels.

A skill level describes supported capability, not delegated decision authority. Lead-level capability does not itself authorize approval of a treatment, acceptance of exposure, or removal of a required review. Apply the decision rights and independence requirements in Chapters 27 and 22, and identify available time and provision for the work through the operating model and Chapter 28.

Where a task requires review, identify who provides it and whether that support is available when the work is due. A task should not be treated as fully covered if the necessary reviewer is unavailable.

A practical model is depth in selected areas with enough awareness across the others to recognize dependencies and engage the right expertise. A records specialist who understands how a platform applies retention asks better questions of the administrator. A technical specialist who understands why a trigger matters configures it with more care. Awareness across areas helps a small team work as one rather than as a set of handoffs.

Capability coverage should also be assessed for dependence on individuals. A task only one person can perform creates a continuity risk, much as an unowned repository creates an ownership risk. Where the task matters to continuing operation, identify how the work will proceed if that person is unavailable or leaves.

## Assess the Team Against the Work

Skills assessments often start from a list of competencies and ask people to rate themselves. A more useful assessment starts from the work.

List the recurring tasks the program performs or has committed to perform. Chapters 27 through 30 supply many of them: verifying configuration, assessing changes, running sampling and control tests, maintaining the register and matrix, assessing AI uses and connections, maintaining patterns, and reporting measures. For each task, record the skill areas it draws on, the level required, and who on the team or among its partners can perform it at that level.

Record the basis for the assessed level, such as relevant work reviewed or a task demonstration, rather than relying only on a title or self-rating. Link any required reviewer and availability arrangement. Where the assessment basis or support cannot be established, record the limitation instead of treating the task as fully supported.

The result shows three kinds of conditions. Tasks with no identified person or partner at the required level are capability gaps. Determine separately whether the task is not being performed, is

being performed through an unrecorded dependency, or lacks the required support. Tasks with one qualified holder are continuity risks. Tasks dependent on partners outside the team are valid capability coverage where the arrangement is defined, available, and reflected in the role assignments described in Chapter 27. Preserve any independence requirements that apply to the partner's role.

Repeat the assessment when the program's work changes, such as when a new phase begins, a new class of system is brought into scope, or a person with a key skill leaves.

## Options for Closing the Gap

Closing a gap does not always mean hiring. Five options are available, and most programs use several.

- **Develop.** Build skills in current team members through assigned work, paired with someone who holds the skill, and supported by targeted learning. Development tied to real work allows the program to observe whether learning transfers into task performance.
- **Borrow.** Partner with functions that already hold the skill. Platform administrators, data teams, security, internal audit, and analytics groups may perform related work. Record responsibility, availability, required review, and handoffs rather than rely on informal goodwill. Preserve the partner's independent responsibilities and judgment; participation by internal audit is not interchangeable with program operating support.
- **Hire.** When a position opens, hire for the skills the program lacks rather than automatically recreating the prior role. A vacancy creates an opportunity to reconsider which capabilities the team needs.
- **Buy.** Engage external support for bounded work, such as designing a sampling method or assessing a class of AI systems. Where the capability will be needed internally after the engagement, include knowledge transfer as a deliverable. For genuinely occasional specialist work, continuing external support may be the deliberate model.
- **Redesign the work.** Reduce unnecessary dependence on specialist judgment. Patterns, templates, and structured tools can move repeatable tasks to working level while reserving lead-level judgment for decisions that require it. Preserve the required standard and any review conditions.

The right mix depends on how often the task recurs, how central it is to the program, and how quickly the gap must close. Recurring tasks at the center of the program often warrant development or hiring. Occasional or specialized tasks may be better borrowed or bought, provided the dependency is deliberate and reliable.

## Credentials, Paths, and Judgment

Professional credentials establish a shared baseline of knowledge and signal commitment to the field. They do not, on their own, show that a person can perform a program's tasks. Assess

performance on the work alongside credentials, and support learning that builds skills the program uses.

Career paths matter for retention of the team as well as its information. Where meaningful advancement requires leaving governance for another function, retaining experienced people can become more difficult. Paths that recognize depth in technical, analytical, or product and service-management skills within governance can help the team retain what it has built.

AI changes the mix of work but not the need for human judgment. Classification at scale, retrieval, and analysis increasingly involve automated tools. The professional's work increasingly includes defining the conditions for use, validating results, handling exceptions, and maintaining evidence supporting reliance, as Chapters 23 and 25 describe. Accountability for governance decisions remains assigned to people through the operating model.

## Why the Skills Gap Persists

Skills gaps can persist for structural reasons that are not primarily about individual capability.

- **Roles are defined by history.** Position descriptions reflect the work the team did when the roles were created.
- **Vacancies are refilled as they were.** Replacing a role with the same description is faster and feels safer than redesigning it.
- **Technical or analytical work may depend on other functions without a defined arrangement.** When the responsibility, availability, or handoff is unclear, work can fall between functions.
- **Learning is disconnected from the work.** Development budgets support general knowledge rather than the tasks the program has committed to perform.
- **Limited capacity can leave little time for development while the team is meeting current obligations.** As the program grows, capability building then competes directly with existing work.
- **Where success is measured primarily through policies and schedules produced, capabilities needed to implement, measure, and test the program may receive less attention.**

The result can be a team well suited to the work it has historically performed and short of capability needed for newer program commitments.

## What Works in Practice

A public school district with roughly seven thousand employees and sixty schools had a records team of four: a records manager and three records staff. Their work centered on maintaining the retention schedule, managing off-site storage, and responding to requests for records. Over three years, the district's program expanded to include an onboarding gate for new systems, verification of

retention in the student information and collaboration platforms, and review of AI tools proposed for classroom and administrative use.

The expansion was approved, but much of the new work was not being done. Configuration verification waited on IT staff who had other priorities. Sampling for the district's measures had been attempted once and not repeated. AI tool reviews were completed by the records manager alone, and stopped whenever she was away.

The records manager listed twenty-two recurring tasks the program had committed to perform and recorded who could perform each. Five tasks had no one able to perform them at the required level. Nine depended on a single person. The team then chose an option for each gap. Two platform administrators were explicitly assigned control-owner responsibilities and paired with records staff for configuration verification. One records staff member took on sampling, working with the district's assessment office, which already sampled student data for other purposes. When a records position opened, it was filled by an analyst with platform experience rather than recreated as it had been. An external advisor helped design the AI review method, with a requirement to train two team members to perform it.

After a year, every recurring task had at least one identified person or partner able to perform it at the required level, and the number depending on a single person had fallen from nine to four. The remaining four were recorded with development plans and dates. Sampling was repeated on schedule for the first time, and AI tool reviews continued during the records manager's absence. Those activities provided separate operating evidence; the capability assessment did not establish that every task was performed on schedule.

The figures and timeframes in this example are illustrative, not benchmarks. The tasks, levels, and options an organization selects should reflect its own program, team, and partners.

## Implementation Steps

**List the program's recurring tasks.** Draw on the work committed in the operating model, the change process, testing, AI governance, and pattern maintenance.

**Record the skill areas and level each task requires.** Use the five areas and three levels, and adjust them to the program's needs.

**Record who can perform each task at the required level.** Retain the assessment basis and date, identify any required reviewer and their availability, and record the support arrangement for partners. Reflect assignments in Chapter 27's operating model.

**Identify capability gaps and concentrated coverage.** Record tasks without supported coverage, including missing required review, as open work with owners and dates through Chapter 20. Distinguish assessed capability gaps from missing assessment evidence, and identify concentrated dependencies as continuity risks.

**Choose an option for each gap.** Develop, borrow, hire, buy, or redesign the work based on how often the task recurs, how central it is, and whether the capability should reside inside or outside the team.

**Revise position descriptions and paths.** Update descriptions to reflect the work the program performs, and use vacancies to change the team's composition.

**Reassess on change and on a defined cadence.** Set a review cadence appropriate to the rate of change and repeat the assessment when the program's scope changes, a new phase begins, organizational responsibilities change, or a key capability is lost.

## What Good Looks Like

In a program with the capability it needs, each recurring task has supported coverage at the required level, with any necessary reviewer available. Continuity-critical tasks have defined backup coverage. The team holds depth in the areas at the center of the program and enough awareness across the rest to recognize dependencies and engage the right expertise. Partner coverage is recorded and supported by explicit arrangements rather than assumption.

Position descriptions reflect the work performed. Vacancies are used to reconsider needed capabilities. Learning is tied to tasks, and career paths can recognize technical, analytical, and product or service-management depth within governance.

The clearest sign is that the absence of one person does not leave critical work without a defined way to continue or obtain the required expertise.

## What Proves It

The evidence for this chapter shows where the program has identified capability for its recurring work and where gaps or concentrated dependencies remain.

- A list of recurring program tasks with the skill areas and levels each requires.
- A task-capability record identifying the holder or provider, assessed level, assessment basis and date, any required reviewer and their availability, and the nature of partner support.
- Role assignments that record partner coverage, in the Appendix B form.
- Open items for uncovered capability, with the option chosen, owner, and target date, and continuity records for single-holder or single-provider dependencies.
- Current position descriptions and development plans linked to identified gaps.
- Knowledge transfer records for external engagements where internal capability is expected after the engagement.

The chapter's priority measure is the number of recurring tasks in the defined assessment population with no supported coverage at the required level, read with the number that depend on only one qualified holder or one defined provider. Establish coverage from the recorded assessment basis and availability of required review. Distinguish a capability gap from an unverified assessment

or missing reviewer support. Report the two counts separately against the total recurring-task population, with the assessment date and task-definition method stated.

The first count identifies gaps in supported capability coverage. The second identifies concentration and continuity risk. Neither establishes whether covered tasks were performed; use the relevant operating and testing evidence for that conclusion. A provider with demonstrated backup capacity should not automatically be treated as one individual, nor should its name conceal dependence on one specialist. The counts can overlap where unsupported performance or review also depends on one person or provider. Assess continuity across both the performer and required reviewer, disclose the overlap, and do not add the counts into a total of distinct affected tasks. Explain changes to the task population or assessment method. This remains a chapter-level measure, not an addition to the standing library.

## Business Value

It makes the program's commitments more executable by identifying where required capability exists and where it still needs to be built or sourced.

Broader and deliberately arranged coverage can reduce dependence on particular individuals and improve continuity through absence, turnover, and reorganization.

It gives the program a clearer basis for directing learning, hiring, and external support toward gaps that affect committed work.

Broader operating capability can strengthen the team's usefulness to business and technology partners by allowing it to participate in implementation and evidence, not only review.

## The Work Defines the Role

Information governance continues to depend on judgment about requirements, categories, risk, and the decisions an organization needs to make. That remains core work.

The operating model developed in this book adds work around implementation, verification, measurement, change, and evidence. No single professional needs to lead every discipline, but the program needs defined access to the capabilities its commitments require.

Start with the work. Identify the capability it requires, decide where that capability will come from, and keep capability coverage current as the program changes. The role should follow the work, not the job description the organization inherited.

*Next: Chapter 32 closes the book by treating governance as infrastructure: a capability that is built, tested, maintained, and improved, and judged by whether the organization can rely on it.*

# Chapter 32: Governance as Infrastructure

Ask an organization whether it has a retention policy. Then ask whether it can show how that policy operates today for a specific category in a specific repository. Producing the policy does not answer the second question.

Chapter 1 began with that difference. A program can have a current policy library, an approved schedule, and a committee that meets on time, and still be unable to show quickly what happened to a set of records. The chapters since have described the work that closes the distance: structure that people and systems can apply, execution across the information environment, decisions and controls carried into operation, evidence showing what was implemented and what testing found, extension to AI, and an operating model that sustains the work over time.

This chapter draws that work together under one idea. Information governance is most useful when it is treated as infrastructure: a capability that is implemented, maintained, inspected, and renewed, and evaluated by whether the organization can rely on it within known scope and limitations. The closing question of the book is not whether a policy exists. It is whether the organization can rely on the system behind it.

## From Written to Operated

Governance as written consists of what the organization has approved: policies, standards, the schedule, and assigned responsibilities. Governance as operated connects those decisions to implemented controls, maintained records, assigned ownership, and evidence showing what happened within a defined scope.

The written layer is necessary. The gap appears when approval is treated as completion and the implementation and maintenance work that follows is not carried through. Over time, the distance between the approved requirement and current operation can grow without a deliberate decision to change either one.

Infrastructure is a useful frame because a water system, power grid, or data network is not treated as finished when its design is approved. It requires implementation before reliance, continuing maintenance, inspection, reporting on its condition, and funding for renewal. Those expectations provide a useful way to view information governance.

## What Infrastructure Requires

Several disciplines used to maintain physical and technical infrastructure have useful counterparts in the governance work described here.

*Figure 32.1. Infrastructure disciplines and their governance counterparts.*

| Discipline | Governance counterpart | Where the book develops it |
|---|---|---|
| Asset register | Repository register and systems map, with owners and lifecycle decisions | Chapter 8 |
| Design standards | Retention schedule with structured rule elements, categories, and variations | Chapters 6, 7, 9, and 10 |
| Commissioning | Onboarding gate and verification that the applicable configuration or procedure matches the approved treatment before reliance | Chapters 2 and 11 |
| Operation | Consistent application, disposition, holds, and control of information held by others | Chapters 12 to 17 |
| Change control | Assessed, approved, and verified changes with recorded transitions | Chapters 18 and 19 |
| Inspection | Control testing, reconstruction, and visible open work | Chapters 20 and 22 |
| Condition reporting | A small set of measures reported with scope, cutoffs, and limitations | Chapter 21 |
| Extension to new uses and technologies | Classification, retention, evidence, and readiness for AI | Chapters 23 to 26 |
| Operators and funding | Owners, decision rights, funding, scaling, pace, and skills | Chapters 27 to 31 |

The mapping is not a new framework. It is a way of seeing the book's parts as one system. Each discipline depends on the others. A register without implementation records shows what exists but not whether approved treatment is in place. Verification without maintenance can leave a control correct only at one point in time. Testing without reporting can identify conditions that never reach the people who need to act.

## Reliance Is the Test

Infrastructure is useful when people can rely on it within defined conditions. Governance is tested in similar moments: a request for records, an audit, a dispute, an incident, a regulatory inquiry, a system migration, or a proposal to connect an AI system to organizational information.

In each of those moments, the organization needs to answer a small number of questions.

- **What do we have, and where?** The register, systems map, and reconciliation records support the answer, with known limitations made visible.
- **What rule applies, and why?** The schedule, decision record, implementation matrix, and retained history support the answer for the category, scope, and date in question.
- **What was implemented, and did it operate?** Verification records show whether the applicable configuration or procedure matched the approved treatment. Test results and execution records support conclusions about operation within the period and scope examined.
- **Who was accountable and authorized?** Current and historical role assignments, delegations, and decision records identify responsibility and authority for the relevant period.
- **What remains unresolved or limited?** Open items, accepted exposure, test exceptions, and recorded limitations identify the conditions still requiring action or review.

These questions draw on the five dimensions introduced in Chapter 5: structure, execution, ownership, visibility, and evidence. A program that can answer them from retained records has a stronger basis for evaluating reliance, including any failure or limitation. The aim is to retrieve and assemble retained evidence, including the history needed for Chapter 22's reconstruction test, rather than recreate missing reasoning or operating records after a question arrives.

## Maintenance Determines Whether the Program Holds

Even a sound design can degrade through deferred maintenance. Owners change. Systems enter outside the expected process. Configurations drift after platform changes. Testing falls behind. Open items age. None requires a dramatic failure to weaken the program over time.

Infrastructure disciplines treat maintenance as continuing work with defined ownership and funding, not as something done only when time allows. For governance, that means defined cadences for review, reconciliation, and testing; a change process that runs when triggers occur; reporting that makes drift visible; and run funding that does not depend on the next initiative. Chapters 19 through 22 and 27 through 29 describe these mechanisms. The infrastructure frame helps explain why they are continuing operating responsibilities rather than project-close activities.

These responsibilities begin with the activity they support, not after the final rollout phase. A temporary project team can hand over the work, but continuing coordination, oversight, and support still need current holders. Chapters 27 to 31 connect those assignments to authority, capability, capacity, and provision.

A defined cadence of verification, testing, reporting, and follow-through is more useful than periodic attempts to rebuild everything at once. The scope and frequency should reflect the work, exposure, and rate of change, with limitations stated rather than hidden.

## Why Operating Governance Degrades

Even organizations that have implemented substantial governance capability can find it difficult to sustain.

- **Success may be measured at publication.** Policies approved and schedules updated are easy to count. Operation and test results require more evidence, so they may receive less attention.
- **Projects end.** Governance established through a project can lose dedicated ownership or funding when the project closes unless the operating model and run cost were planned from the start.
- **Successful maintenance can be less visible than remediation.** When controls operate without incident, attention and funding can move toward more visible problems until maintenance falls behind.
- **New technology can reset attention.** A new platform or AI capability may be treated as a separate governance project rather than as another use that needs to enter the existing operating model.
- **Ownership changes.** Responsibilities assigned at launch can become stale through reorganizations, turnover, and acquisitions unless reassignment is part of the operating model.

Each of these conditions is addressed elsewhere in the book. The infrastructure frame connects them because both physical infrastructure and governance depend on defined ownership, maintenance, inspection, reporting, and renewal.

## What Works in Practice

A regional water and wastewater authority serving roughly a million people managed its physical assets through a mature asset management program. Pipes, pumps, and treatment facilities each had a register entry, an owner, a condition rating, a maintenance schedule, and a renewal forecast reported to the board every year.

Its information governance program was younger and managed differently. The schedule was current and well regarded, but the authority could not say with confidence which of its roughly 140 repositories applied it. Testing had occurred once, for an audit. Governance reporting to the board consisted of policies approved and training completed.

The records manager proposed managing governance with the same operating discipline used for physical assets. Each repository received a register entry with an owner and separate condition indicators: whether the applicable treatment had been verified, whether required testing was current, and whether material open items were overdue. The indicators were not combined into a single score. In this example, high-exposure repositories were placed on an annual testing cycle published in advance. Changes followed a documented process with verification before closure. Governance reporting used the asset-management format the board already knew, but kept the

measures separate: coverage bands, repositories overdue for scheduled testing, aging open items, and planned remediation or renewal work.

The board already understood the reporting format and could read governance condition alongside asset condition without collapsing the measures into one score. After two years, most high-exposure repositories had current verification records and were on their testing cycle, and the remaining gaps appeared as renewal items with owners and dates rather than as unknowns. When a regulator asked how a category of operational records had been disposed of, the authority answered from its records within days.

The figures and timeframes in this example are illustrative, not benchmarks. The indicators, testing cycles, and reporting an organization adopts should reflect its own information environment, exposure, and governance structure.

## Implementation Steps

Name the components. Identify the register, schedule, implementation matrix, controls, decision records, and evidence the program depends on, and treat them as one system.

Confirm an owner for each. Use the ownership layers and role assignments in Chapter 27. Confirm ownership on a defined cadence appropriate to the role and when organizational change affects the assignment, retaining the history needed to establish responsibility for prior periods.

Verify implementation before reliance. Confirm that the applicable configuration or documented procedure matches the approved treatment before the control is counted as operative, and retain the verification record. A reusable approval under Chapter 30 does not remove verification required for the particular use; subsequent sampling examines whether the pattern and required checks were followed.

Schedule recurring maintenance and define event-driven triggers. Set cadences for reconciliation, review, testing, and pattern maintenance, while routing relevant changes through Chapter 19 when they occur.

Inspect on a cycle. Test controls on a risk-based schedule using the approach in Chapter 22, state the scope and limitations, and route findings through the applicable open-item process.

Report condition through the existing measures and open-item views. Keep coverage, testing, open work, and other relevant measures distinct, with scope, cutoffs, and limitations, rather than collapsing them into one condition score. Lead with results and supporting evidence; include activity information where it helps the decision, while retaining routine operating detail beneath the leadership view.

Fund recurring operation and plan renewal. Following Chapter 28, make build and run costs visible in the appropriate budget decisions. Renewal work may be delivered through projects or stages; continuing operation should not depend solely on the next initiative.

## What Good Looks Like

In a program operated as infrastructure, the components within scope have defined ownership, current records, and visible status. Applicable configurations and procedures are verified before reliance, operation is tested on a defined cycle, and changes close against the evidence appropriate to their scope. Leadership sees condition through measures and open work that retain their definitions and limitations.

New systems, changed requirements, and new AI uses enter the established intake, decision, change, readiness, and testing processes appropriate to them. The program extends those structures rather than designing a separate governance response each time.

The clearest sign is that a question about reliance can be answered from maintained records without an ad hoc effort to recreate missing evidence.

## What Proves It

The evidence for this chapter is the evidence developed throughout the book, kept current, linked where necessary, and retrievable when the organization needs it.

- A register, schedule, and implementation matrix that are reconciled with one another and with the environment.
- Implementation verification records showing that the applicable configuration or documented procedure matched the approved treatment before reliance.
- Maintenance and testing schedules with completion records, together with event-triggered change records where applicable.
- Decision records, change records, and open items with owners and dates.
- Condition reporting using the existing Appendix C measures and open-item views, with scope, cutoffs, and limitations stated.
- Approved provision for recurring operation, together with a renewal plan identifying its funding decisions and any unresolved funding gaps.

This chapter adds no new priority measure. Read the existing Appendix C measures together with control test coverage and test exception aging from Appendix G, for a stated scope and reporting date. Together they provide a view of implementation, open work, measurement, and testing without creating a new aggregate score. The five-measure standing library remains unchanged.

## Business Value

It can make governance more dependable. Maintained rules, decisions, controls, and evidence give the organization a stronger basis for relying on the program when questions arise.

It can reduce the effort required to answer future questions. Requests, audits, disputes, and incidents can draw on maintained records, including the history needed for point-in-time reconstruction.

It supports more informed adoption of new technology. New platforms and AI uses can be assessed against known information, existing controls, and visible unresolved work.

It treats governance as a continuing operating capability, with ownership, maintenance, testing, and funding that continue beyond individual initiatives.

## The System Behind the Policy

Part VI addressed what sustains a program: an operating model that assigns ownership and decision rights, funding that treats governance as infrastructure, a path from pilot to program, governance that keeps pace with the business, and people with the skills to operate it. Those are the conditions under which the work of the earlier parts continues after the people who started it move on.

The book began with the observation that many governance programs are better than they look from the inside. Many organizations already have a foundation of policy, structure, and expertise. The recurring gap described in this book is what connects that foundation to operation: rules implemented where information lives, controls verified and tested, decisions and authority recorded, unresolved work visible, and evidence retained with its scope and limitations.

Closing that gap does not require starting over. It requires treating governance as continuing work: implemented, maintained, tested, measured, and renewed as the organization changes. The question at the end is the same one that began the book. A policy may say what should happen. Can the organization rely on the system behind it, and can it show what that reliance is based on?

# Appendices

# Appendix A: Baseline Diagnostics and Maturity Self-Assessment

This appendix supports Part I. It contains a baseline worksheet that sequences the Part I diagnostics, working forms for each diagnostic, and the five-dimension self-assessment from Chapter 5. The chapters explain why each diagnostic matters. This appendix is for running them.

The instruments are self-assessments. They record the organization's own view of its program, based on the records and people available to it, and they are meant to establish a baseline and guide priorities. They are not an audit, a certification, or a substitute for the verified measures and control tests in Chapters 21 and 22. Each result should state its scope, its date, and whether it rests on system records, owner reports, or direct verification, so that later, more rigorous work can explain any change in basis.

## 1. Baseline Worksheet

Run the diagnostics in the order below. Each step reuses what the earlier steps produced, and the full sequence can usually begin with records the organization already holds.

Figure A.1. Baseline sequence.

| Step | Diagnostic | What it produces | Dimension informed |
|---|---|---|---|
| **1** | Trace test (Chapter 1) | For selected categories, the systems, mechanisms, owners, and evidence behind each rule, and a gap list | Execution and ownership |
| **2** | Coverage, first pass (Chapter 2) | An estimate of the share of in-scope volume subject to an operative rule, with its basis and limitations | Execution |
| **3** | Sensitive record count (Chapter 4) | A ranked list of repositories by sensitive content, age, and lifecycle ownership | Visibility |
| **4** | Cost model (Chapter 3) | An annual run rate and cost of delay, as ranges, with risk exposure shown separately | None. Measures consequence |
| **5** | Structure and evidence reviews (Chapter 5) | A review of sampled rules and of what the program can produce to show that controls operated | Structure and evidence |
| **6** | Dimensional assessment (Chapter 5) | Five separate ratings, the widest material gap, and priorities | All five, rated separately |

The cost model is placed after the other diagnostics, although Chapter 3 precedes Chapter 4, because it expresses the consequences of what the trace test, coverage estimate, and sensitive record count find. It can be started earlier, since finance and other functions may need time to supply figures, and finalized once the other findings are in hand.

For every step, record the date, scope, who performed the work, sources used, and what could not be confirmed. Keep the completed worksheet with the program's records. A second baseline, run a year later on a comparable basis, shows changes in the assessed position. Use supporting work records to identify which changes can be attributed to funded work.

## 2. The Trace Test

The trace test follows one rule from the retention schedule to the systems where it should apply. An initial trace of a single rule may take about an hour when the necessary people and records are available. A baseline across several categories needs a defined scope.

- **Select deliberately.** Ten to fifteen categories usually provide a useful first diagnostic. Choose by the strength of the underlying requirement, recent litigation or investigation history, and the sensitivity of the information, rather than by ease. Include one or two categories expected to pass, and make the selection jointly with legal and IT.
- **Ask the five questions for each category.** Which systems hold information in this category? What mechanism enforces the rule in each of those systems? Who owns that mechanism, by name? When did it last operate, and how was that confirmed? If someone asked for evidence that it operated, what would you provide?
- **Describe the selection, not the program.** Findings describe the categories traced. They indicate where to look further but do not estimate the condition of the whole program.

Record each category and system pairing on one row.

Figure A.2. Trace test record.

| Field | What to record |
|---|---|
| **Category and rule version** | The category traced, the current rule version and trace date, and the applicable treatment and version for any earlier operating period examined |
| **System** | Each system that holds information in the category, one row per system |
| **Enforcement mechanism** | Native, integrated, manual, or not enforced, as defined in Chapter 11, with a short description |
| **Control owner** | The named person accountable for the mechanism, distinct from the policy owner, or none |

| | |
|---|---|
| **Last operation and confirmation** | When the mechanism last operated, the period examined, and how that was confirmed. Distinguish unknown operation from a control that has not yet had an eligible population or an operation due |
| **Evidence available** | What could be produced today and how long producing it would take |
| **Source basis** | System record, owner report, or direct verification |
| Observed condition and findings | The condition observed: capability available but unconfigured; capability absent; unreliable category identification; configuration or procedure mismatch; missing operating evidence; no issue observed within scope; or operation not yet due. Link multiple findings where needed; missing evidence does not itself establish control failure |

Record each gap as an open item with an owner, priority, target date, and the evidence needed to verify completion. Chapter 20 describes how open items are managed.

## 3. Coverage, First Pass

Coverage uses the single verified-implementation definition in Chapter 21 and Appendix C. The first pass estimates the same population using available information, with provisional attribution and verification limitations stated. It does not create a second definition of operative or present unverified estimates as verified coverage.

- **Denominator.** All in-scope stored volume, including volume not yet reconciled to the register. State the scope, reporting cutoff, unit, approved exclusions, and limitations.
- **Numerator.** Identify volume attributed to applicable approved treatments and distinguish supported verification from provisional attribution. Verified coverage requires the configuration or documented manual procedure to have been verified against the treatment. Owner reports and unverified configurations may inform a labeled preliminary estimate, but do not establish verified coverage or successful operation.
- **Proxies.** Where only system-level figures are available, label the result a preliminary proxy and document the assumption about coverage within each system. Do not count a whole system as covered because one category in it has a configured rule.
- **Basis.** Identify which figures come from system records, owner reports, or direct verification, and what remains unconfirmed. Do not require demonstrated manual operation to define implementation while allowing technical configuration alone to qualify. Record evidence of operation separately.
- **Repeat.** Calculate coverage at least twice on a comparable basis, so that a trend exists. Keep the method consistent between periods, and explain any change in basis.

Record new systems separately. A falling share can reflect new volume entering repositories without controls rather than failure of existing configurations. The onboarding gate in Chapter 2 and

Appendix F addresses that inflow.

## 4. Sensitive Record Count

The sensitive record count identifies where potential exposure is concentrated. It is a prioritization measure, not a notification population.

- **Count by repository.** Count sensitive records by repository rather than relying on storage volume alone.
- **State the population.** Record the repositories in scope, the sensitive data types counted, the discovery method, and whether the count covers all sensitive records or only aged or unmanaged information.
- **Start with the likeliest locations.** A first pass might cover ten to fifteen high-risk repositories, such as decommissioned application data, shares belonging to reorganized units, migration staging areas, and long-lived departmental storage. Adjust the selection to the environment and describe it as a starting sample, not a representative one.
- **Rank and record ownership.** Rank repositories by aged sensitive content and record whether each has a lifecycle owner. Feed new repositories into the register described in Chapter 8.
- **Report through security.** Report the count through the organization's existing security reporting, alongside measures that audience already reviews.

## 5. Cost Model

The cost model assembles a conservative estimate of what the current state costs from figures that already exist across several budgets. It measures consequence, not capability, and is not a maturity dimension.

Figure A.3. Cost model worksheet.

Record the currency, reporting period, source date, budget line, calculation basis, assumptions, overlaps, and omitted amounts for each component. Distinguish gross current-state cost, the portion avoidable through the proposed action, released staff capacity, and cash savings. Do not add overlapping cost categories.

| Category | Typical source | What to record |
|---|---|---|
| **Storage and infrastructure** | Infrastructure | Storage, backup, replication, capacity-based licensing, and migration costs priced by volume |
| **Discovery and response** | Legal | Collection, processing, and review spending, with a conservative estimate of the share attributable to scope that disposition might have reduced |

| | | |
|---|---|---|
| **Manual effort** | Business operations and records | Time spent searching, answering questions systems should answer, and reconciling duplicates, from existing surveys or time records |
| **Decision friction** | IT and project offices | Costs of delayed system retirement, expanded migration scope, and deferred consolidation attributable to unknown content or ownership |
| **Risk exposure (shown separately)** | Legal, privacy, and security | Contingent exposure described without a speculative value, and excluded from the run rate |

- **Use figures the organization already reports.** Have each function supply its own figures, and attribute each figure to that function and its budget line. Leave out anything that would require a separate study, and document the omission.
- **Show ranges.** Use the conservative end as the working figure and the upper end as context.
- **Separate legitimate retention.** State what share of retained information has a defensible basis and is not affected.
- **Express two results.** Show the annual run rate and the cost of delay as ranges, with contingent exposure separately. State the delay period and the calculation: for a stable annual amount, multiply by the stated fraction of a year, adjusting for one-time costs, timing, and dependencies. Continuing cost is not automatically a recoverable saving. Chapter 28 uses these figures alongside requirements, exposure, resilience, and dependencies; a funding refusal does not supply any separately required risk acceptance.

## 6. Structure and Evidence Reviews

The Part I diagnostics inform execution, ownership, and visibility. Two short reviews complete the inputs for the remaining dimensions.

- **Structure review.** For a sample of rules, preferably the categories traced in step 1, confirm that each has a clear category definition, an authority, an approved retention treatment, any required trigger, and a maintained history of changes. Chapters 6 and 7 develop these requirements.
- **Evidence review.** Using the trace results, record what can support conclusions about operation in the period examined, whether the relevant records were captured and retained, and how long production would take. Distinguish assembling retained history from recreating missing reasoning or operating evidence. Chapters 18 and 22 develop this dimension.

## 7. Five-Dimension Self-Assessment

Rate each dimension separately on a consistent scale. Strong, adequate, and weak are sufficient. Where the available material cannot support a rating, record insufficient basis rather than

estimating. Do not combine the ratings into a single score.

- **Structure.** Are rules expressed in a form people and systems can apply? Do sampled rules have clear categories, an authority, an approved treatment, any required trigger, and a maintained history? Can people apply the categories consistently? Inputs: the structure review.
- **Execution.** Do rules operate in the systems where information lives? How many traced pairings have an operating mechanism? What share of in-scope volume is subject to an operative rule, and on what basis? Inputs: the trace test and coverage first pass.
- **Ownership.** Does a named person own each control in each system? Are control owners distinct from policy owners where they should be? Did assignments survive the last reorganization? Inputs: the trace test, supplemented by the repository register where one exists.
- **Visibility.** Can the program say what systems exist, what they hold, and where sensitive information sits, without launching a project? Does each repository with sensitive content have a lifecycle owner? Inputs: the sensitive record count and the register.
- **Evidence.** Can the program show that controls operated, from records captured at the time? How long would production take? Inputs: the evidence review.

Record each rating on one row.

Figure A.4. Dimensional assessment record.

| Field | What to record |
|---|---|
| **Dimension** | Structure, execution, ownership, visibility, or evidence |
| **Rating** | Strong, adequate, weak, or insufficient basis |
| **Basis** | The diagnostic results and records the rating rests on |
| **Source basis** | System record, owner report, or direct verification |
| **Limitations** | What could not be confirmed and what would be needed to confirm it |
| **Prior rating** | The rating from the previous assessment, where one exists |

## 8. Reading the Profile

- **Identify the widest material gap.** Compare the ratings and state where the distance between dimensions is largest, with a short explanation of what produced it. Strong structure with weak execution may indicate an implementation gap. Automation in place with weak evidence may concern evidence generation, control performance, or both; the assessment should identify the work needed without assuming the answer.
- **Prioritize the weakest dimension that creates a material constraint or risk.** Do not default to the most familiar or easiest work. If the team lacks the capability to address the priority

gap, treat that as a staffing or sourcing question. Chapter 31 describes options.

- **Use the cost model to explain consequence.** Pair the priority gap with the costs it drives, without treating cost as a rating.
- **Report dimensions separately.** Show leadership the five ratings rather than a level, so that improvement in one dimension does not hide weakness in another.
- **Reassess annually.** Retain each profile and compare it with the prior one on a stated, comparable basis. Explain scope or method changes and use supporting work records before attributing an improvement to a particular investment.

Findings from any step that identify work should become open items with owners and dates. The baseline is useful to the extent that it changes what the program does next.

# Appendix B: Operating Model Roles and Decision Rights

This appendix supports Chapter 27. It contains definitions for the three responsibility groupings, the separations to maintain, a role-assignment field guide, a decision-rights field guide with an abbreviated illustrative extract, a forum outline, and maintenance rules. Chapter 30's decision boundaries and Chapter 31's capability coverage draw on the same roles.

The roles describe accountability, not job titles. One person may hold several roles, and one role may be held by different people for different repositories, pairings, or relationships. What matters is that each role in scope has a named current holder and that deliberate combinations are recorded.

Decide, operate, and assure group responsibilities; they are not prescribed departments or levels of independence. Evidence arises from decisions and operations as well as testing. State the assurance and independence of each assessment separately.

## 1. Role Definitions

Figure B.1. Role definitions.

| Role | Layer | Accountable for | Commonly held by |
|---|---|---|---|
| **Policy owner** | Decide | The content of rules and standards, and proposing changes to them | Records or information governance lead, with counsel for requirement interpretation |
| **Decision authority** | Decide | Approving treatments, changes, exceptions, acceptances, patterns, and AI uses within a defined delegation | Named executives, a decision forum, or delegates, as the decision rights statement records |
| **Information owner** | Operate | The information's business use and lifecycle requirements, with decisions made or referred according to the applicable delegation | Business unit or function leaders, or their designees |
| **Control owner** | Operate | A control operating as approved in a specific pairing, connection, or process | Platform, application, or process owners in the business or technology function |

| | | | |
|---|---|---|---|
| **Configuring party** | Operate | Implementing and maintaining the approved configuration or procedure, with departures referred for a decision | Platform administrators, application teams, or providers under contract |
| **Relationship owner** | Operate | Governance terms and evidence obligations with an external party | The business owner of the provider or partner relationship |
| **Evidence owner** | Assure | Maintaining the evidence record for a defined scope and identifying missing or outdated elements | Program staff, or a named owner for each AI system under Chapter 25 |
| **Tester** | Assure | Examining control existence, alignment with approved requirements, and operation, at a stated assurance tier and level of independence | Program staff, internal audit, or an independent function |
| **Program** lead or designated role | Assure | Maintaining the open-items view, measures, and escalation, and coordinating across roles | The person or role assigned to coordinate the program |

A governance liaison can connect each business unit to the program, with responsibilities and time allocated. Oversight bodies exercise their responsibilities under their charters and reserved decisions; their oversight purpose is distinct from routine operation.

## 2. Required Separations

- **Tester and control owner.** Where a test result will be relied on as independent assurance, the tester does not own the control tested (Chapter 22).
- **Approval and implementation.** An approval is not recorded as evidence that a control was implemented or operated (Chapter 18).
- **Evidence owner and decision makers.** The evidence owner coordinates the record but does not replace those authorized to approve a use, operate a control, or reach a testing conclusion. One person may hold another role where permitted, with assignments and limitations recorded and required independence maintained.
- **Decision and escalation.** A person does not resolve an escalation of their own decision.

Other combinations can be practical, particularly in smaller organizations. Record each combination and any limitation it creates, such as a test that cannot be relied on as independent.

## 3. Role Assignment Record

Record one row per role and scope.

Figure B.2. Role assignment record.

| Field | What it records |
|---|---|
| **Role and scope** | The role and what it covers: a category, repository, pairing, connection, relationship, AI system, or forum |
| **Named holder** | The current holder, identified by name and position |
| **Assignment source and date** | The charter, delegation, or appointment, its actual date, and the assignment effective period; link the source and effective period of any decision authority |
| **Confirmation** | When the holder acknowledged the assignment and scope; retain checks of current validity separately. Confirmation alone does not establish authority, access, or performance |
| **Deliberate combinations** | Other roles the holder has in the same scope, and any resulting limitation |
| **Interim holder** | The named interim holder, scope, start and end or review point, delegated authority and limits, and any recorded decision extending the arrangement |
| **Vacancy open item** | The underlying vacancy or temporary-arrangement reference, original date, owner, enduring-assignment target date, and links to affected work |
| **Next reconciliation** | The scheduled date for review, and any organizational change that triggers an earlier one |
| History and handover | Prior holders and delegations with effective periods; controls, evidence responsibilities, and open work handed over, including unresolved transition items |
| Capability and provision links | The Chapter 31 / Appendix E assessment, required reviewer and availability, allocated time, and continuing funding or partner-support arrangement; link rather than duplicate source records |

Assess currency from the recorded assignment, effective period, holder validity, and required confirmation. Distinguish no holder, a known invalid or departed holder, and overdue confirmation. An overdue confirmation is a gap in evidence of currency, not proof that nobody performs the work. Chapter 27 counts scoped assignments without confirmed current holders and reads that count with vacancy aging without adding the overlapping populations.

## 4. Decision Rights Template

Record one row per decision type, following Chapter 27. A RACI mapping is activity-specific: the authority accountable for approval is not automatically accountable for implementation, independent testing, or corrective action. Map the responsible, consulted, and informed roles for each activity, preserving the relevant delegation and independence.

Figure B.3. Decision rights statement fields.

| Field | What it records |
|---|---|
| **Decision type** | The kind of decision, stated so that a person facing one can recognize it |
| **Decides** | The role that holds the authority, and any named delegate |
| **Limits** | The boundary of the authority, stated by effect, such as whether the decision changes a lifecycle outcome |
| **Consulted** | Roles whose input is required before the decision |
| **Informed** | Roles notified after the decision, including those who must act on it |
| **Outside the limits** | Where decisions beyond the limits go, and the stated response time (Chapter 30) |
| Source and effective period | The charter, policy provision, or delegation that grants authority, its effective period, and retained prior versions |
| **Record** | Where the decision is recorded, normally the Appendix F decision record |

## 5. Illustrative Decision Rights

The table below is an abbreviated illustrative extract, not a completed decision-rights statement or a grant of authority. Each organization sets its holders and limits and completes all applicable fields in Figure B.3, including authority source and effective period, out-of-limits route, response expectation, and record reference.

Figure B.4. Abbreviated illustrative decision rights statement.

| Decision type | Decides | Limits | Consulted | Informed |
|---|---|---|---|---|
| **New category or retention treatment** | Decision forum | Any new category or treatment | Counsel, information owner, policy owner | Control owners, configuring parties |

| | | | | |
|---|---|---|---|---|
| **Administrative rule change** | Program lead or designated delegate | No change to any lifecycle outcome (Chapter 19) | Policy owner | Decision forum, in periodic reporting |
| **Substantive rule change** | Designated decision authority | Changes to period, trigger, disposition action, scope, or variation | Counsel, information owner, affected control owners | Configuring parties, hold function, affected providers |
| **Urgent change** | Named expedited approvers | Substantive changes that cannot wait, within a stated time limit | Counsel | Decision forum, for ratification |
| **Jurisdictional variation** | Authority designated for variations | Approved jurisdictional scope and required review; distinct from local administrative changes within an existing treatment | Policy owner | Program, affected control owners |
| **Rule exception** | Decision authority for the category | Defined scope and expiry date | Counsel, information owner | Program, control owner |
| **Risk or limitation acceptance** | Delegate for the stated acceptance scope; accountable risk owner beyond it | Decision type, scope, consequence, applicable requirements, rationale, expiry, and any earlier review. Duration alone does not confer authority | Control owner, counsel where relevant | Decision forum, oversight |
| **Category-level disposition authorization** | Authority designated for the category | Standing authorization for defined classes; eligibility, encumbrance, and dependency checks still apply before each execution | Counsel, information owner, hold function, records | Control owner, configuring party |

| | | | | |
|---|---|---|---|---|
| **AI use and risk classification** | Decision authority for the tier | Tier criteria set by the organization (Chapter 25) | Privacy, security, information owner | Evidence owner, program |
| **AI system connection to a source** | Authority designated for system, use, and source scope | Recorded outcome and permission limits; applicable readiness conditions and required remediation satisfied and verified before operational connection | Source owner, security, privacy, program | Evidence owner |
| **Pre-approved pattern** | Decision forum | Approval, revision, and withdrawal of patterns (Chapter 30) | Counsel, affected owners | Teams permitted to use the pattern |

## 6. Forums

Figure B.5. Forum outline.

| Forum | Purpose | Typical members | Record produced |
|---|---|---|---|
| **Decision forum** | Makes decisions reserved to it and resolves escalations | Decision authorities, counsel, policy owner, program | Decisions with basis and authority, in the Appendix F form |
| **Operating forum** | Works through open items, changes, and blockages within delegated limits | Control owners, configuring parties, information owners, liaisons, program | Actions, open-item updates, and escalations with their criteria |
| **Oversight** | Receives reporting and holds the program to account | Audit or risk committee | Reporting received, questions raised, and follow-up requested |

Decision-making, operational coordination, and oversight are functional purposes, not three mandatory committees. Existing forums or separate agenda sections may serve them where authority, records, and required independence remain clear. Define normal escalation under Chapter 20 and direct routes for urgent concerns or independent reporting; those matters need not wait for an intermediate meeting.

## 7. Keeping the Model Current

- **Reconcile on a cadence.** Check assignment validity and applicable authority at a defined interval, ask holders to acknowledge their responsibilities, and record what was examined, confirmed, changed, or left unresolved.
- **Reconcile on organizational change.** Treat reorganizations, acquisitions, and divestitures as change triggers under Chapter 19, with a step to reassign affected roles and decision rights.
- **Apply the vacancy rule.** Name an interim holder with stated scope, period, and delegated authority, and retain the original vacancy date. An approved interim holder can provide current accountability while the vacancy or temporary arrangement remains visible until an enduring assignment is confirmed. Extensions require a recorded review and decision, not an automatic reset. Link one ownership issue to affected work rather than create duplicate control findings.
- **Onboard acquisitions deliberately.** Register acquired repositories and systems, assign owners, state which decision rights apply during transition, and record when the transition ends.
- **Recognize the work.** Include control ownership and liaison duties in role descriptions and capacity planning, as Chapters 29 and 31 describe.

# Appendix C: Metrics Library and Sample Governance Scorecard

This appendix holds the standing metrics library and a sample scorecard. The library has five measures: two anchor measures, coverage and sensitive record count, and three supporting measures, trigger coverage, unowned repository count, and configuration lag. Chapter 21 explains why the library is kept small. Chapter-level measures and diagnostics remain in the chapters that own them and do not become standing measures by being reported.

Each measure carries a written method, an owner, a data source, a reporting cadence, and a counterweight. Report each with its scope, reporting cutoff, unit, and method version, and label each component as measured, estimated, or proxy. When a method changes, restate prior periods where possible or mark the break in the series.

## 1. The Library

Figure C.1. The five standing measures.

| Measure | Type | Dimension | Counterweight | Developed in |
|---|---|---|---|---|
| **Coverage** | Anchor | Execution | Excluded volume and the unreconciled band | Chapters 2 and 21 |
| **Sensitive record count** | Anchor | Visibility | Repositories scanned and any change in inclusion criteria | Chapter 4 |
| **Trigger coverage** | Supporting | Structure | Permanent designations reported separately | Chapter 7 |
| **Unowned repository count** | Supporting | Ownership | Total active repositories on the register | Chapter 8 |
| **Configuration lag** | Supporting | Execution | Reported as a distribution with the longest open item | Chapters 11 and 21 |

## 2. Coverage (Anchor Measure)

Definition. The share of in-scope volume subject to an operative retention rule. Count verified implementation of the applicable approved treatment within each category/repository pairing, including any distinct scoped populations. Native, integrated, or documented manual implementation qualifies only where the configuration or procedure has been verified against that treatment and any required trigger. One verified treatment does not cover other populations in the pairing. Coverage describes verified implementation; control test coverage describes the extent of independent testing; execution records and test results support conclusions about operation within the period and scope examined.

Bands. Report all in-scope volume in four mutually exclusive bands that together account for the total. Each unit of in-scope volume belongs in one band only.

- **Covered.** Volume in pairings with a verified operative rule, through system configuration or a documented manual procedure.
- **Configured, not verified.** Volume in pairings where a configuration or procedure is in place but has not been verified against the approved rule.
- **Known, not covered.** In-scope volume in registered repositories that is neither subject to verified treatment nor in the configured-but-unverified band. Unclassified content remains here unless an applicable approved container default has been verified, or its implemented treatment awaits verification and belongs in that band. Missing ownership, dates, or risk decisions remain visible and do not remove volume from the denominator.
- **Unreconciled.** In-scope stored volume that cannot yet be traced to an entry on the repository register.

Report approved exclusions separately, with the total excluded, alongside the bands.

Method.

- **Denominator.** Reconcile the register with total in-scope infrastructure storage at each reporting cutoff. Include volume not yet traceable to a registered repository in the unreconciled band, rather than derive the denominator only from known entries. Account for each unit once and report approved exclusions separately.
- **Numerator.** Take covered volume from verification records referenced in the implementation matrix, not from recorded status or owner confirmation.
- **Attribution.** Attribute volume to categories within each repository and, where a pairing holds populations with different approved treatments, to those scoped treatments. Count a container default as covered only where it applies an approved rule and has been verified. Where attribution can only be estimated, use a documented sample or classification proportions and label the result.
- **Proxy triggers and alternatives.** A proxy retention date counts only where the difference from the approved trigger has been documented and authorized as an alternative treatment, and that treatment has been verified. An authorized alternative is not the same as an accepted risk that the required treatment has not been implemented; an acceptance does not make a pairing covered. A pairing recorded as not enforced is uncovered, whatever its target date.
- **Rule versions.** Verification applies to the rule version and configuration examined. A later change or finding that invalidates the match requires the coverage status to be reassessed.
- **Exclusions.** Record each exclusion with a reason and an approver.
- **Unit and retention of inputs.** State the unit, stored bytes or item counts, and keep it consistent. Retain the source extracts, reporting cutoff, verification references, assumptions, and any samples or proportions used for estimates, so that the figure can be reproduced.

First-pass note. Chapter 2 and Appendix A estimate this same measure from the information available, using labeled provisional attribution and verification limitations. System-level figures may be used as a preliminary proxy, but a whole system does not qualify because one category has a configured rule. Do not label unverified attribution as verified coverage. The banded method makes the status explicit as the register and matrix mature; explain the change in basis and comparability.

### 3. Sensitive Record Count (Anchor Measure)

Definition. The count of sensitive records by repository, for a stated repository population and inclusion criteria. It identifies where potential exposure is concentrated. It is not a notification population.

Counting basis. State whether totals count record instances across repositories or deduplicated records, how duplicates are treated, the discovery and reporting dates, and the applicable inclusion criteria. Neither basis is automatically a count of people or a notification population. Identify any break caused by changed scope or method.

- **Scope.** State the repositories scanned, the sensitive data types counted, the discovery method, and whether the count covers all sensitive records in scope or focuses on aged information or repositories without clear lifecycle ownership.
- **Reporting.** Report by repository and in total, alongside the number of repositories scanned. Report changes in sensitivity criteria, classification criteria, or scanning scope with the count.
- **Audience.** Report through the organization's existing security reporting, as Chapter 4 describes.

### 4. Trigger Coverage (Supporting Measure)

Definition. The percentage of in-scope categories requiring a trigger that have all required triggers explicitly defined in the controlled vocabulary. Count each category once. Where a category has several scoped treatments, it qualifies only when the required trigger is defined for each; link the underlying missing-trigger findings. Record justified not-applicable treatments and reviewed denominator exclusions.

- **Exclusions.** Document and review exclusions from the denominator.
- **Permanent designations.** Permanent designations are not automatically excluded. Include them where a lifecycle action, such as cutoff, transfer, or review, requires an event. Report permanent designations separately to distinguish approved treatment from unresolved rules.
- **Limits.** Trigger coverage is a structural measure. Reconciliation and configuration verification establish whether receiving systems can apply the triggers.

## 5. Unowned Repository Count (Supporting Measure)

Definition. The number of active register entries in scope without a current named information owner, as of the reporting date. Here, active means repositories still holding in-scope information or requiring governance, not only those labeled active in a lifecycle-status field. Include relevant legacy, orphaned, and planned-for-retirement repositories until the recorded scope and closure criteria remove them.

- **Counting.** Count a missing current owner or known invalid/departed holder as unowned, distinguishing those causes. A valid named interim holder can satisfy ownership for the stated period while the vacancy remains open. Show overdue confirmation separately unless validity is also known to have failed. Confirmation alone is not proof of authority or performance; document method changes and their effect on trends.
- **Reading it.** Report alongside the total number of active repositories. A rising count may reflect better detection as unreconciled volume is traced to the register, rather than deterioration.
- **Relationship to Chapter 27.** That chapter examines confirmed current holders across all scoped roles, including overdue confirmation and interim arrangements. It complements this repository-level count rather than replacing it or providing an additive population.

## 6. Configuration Lag (Supporting Measure)

Definition. The elapsed time from the effective date of an approved rule or rule change to verified configuration in each affected pairing. The clock starts at the effective date, not at approval, and stops at verification, not at a recorded status of complete.

- **Reporting.** State the completed-verification population and show its distribution, including the median, separately from the age and significance of items still awaiting verification at the cutoff. Do not pool unfinished ages with completed durations. The longest open item indicates delay, not exposure size on its own. Record separate scoped treatments where they have different dates or verification outcomes.
- **Related timing.** Decision lag, from the point a change trigger enters intake to approval or a documented no-action decision, is a separate chapter-level measure in Chapter 19 and is not combined with configuration lag.

Early and not-yet-due work. Retain actual effective and verification dates. Verification before effect is completed early, not overdue; show the lead time separately rather than pool it with positive lag. Keep changes whose effective dates have not arrived in a not-yet-due population, distinct from overdue work. Record any zero-floor or other display convention, retaining the original dates and marking method changes. Use the same convention in Chapter 11 reporting.

## 7. Measures Drawn From Appendix G

Control test coverage and test exception aging appear on the scorecard beside the library measures. They are defined with the testing material in Appendix G and are not sixth and seventh library measures. Classification agreement remains a sampling test in Appendix G.

Coverage is volume-based and control test coverage is control-based. Report their units and denominators explicitly. They are not interchangeable percentages, and multiplying them does not establish the share of information subject to effective controls.

## 8. Control Evidence Retention Note

The records supporting these measures, including verification records, source extracts, and test records, are control evidence. Their contents support different conclusions about decisions, implementation, operation, measurement, and testing. Retain them under an approved retention value, any required trigger, and documented rationale in the schedule's control-evidence category (Appendix H). Life of the control alone is not an operable rule; a defined retirement event may be a trigger where the full approved treatment is explicit. Linked business and documentation records retain their separately assessed purposes.

## 9. Sample Governance Scorecard

The sample below illustrates reporting by dimension without a composite, with each measure beside its counterweight. Complete the reporting date, reporting period, scope, units, method versions, and material limitations. Lead with outcomes and the information needed to interpret them. Decision-relevant activity or a chapter diagnostic may also be shown where useful without becoming a standing library measure. Values are left blank; the layout is the point.

Figure C.2. Sample governance scorecard layout.

| Dimension | Measure | Current | Prior | Counterweight | Basis and notes |
|---|---|---|---|---|---|
| **Structure** | Trigger coverage | | | Permanent designations | Method version; exclusions |
| **Execution** | Coverage (bands) | | | Excluded volume; unreconciled band | Share measured, estimated, proxy |
| **Execution** | Configuration lag | | | Longest open item | Median; pairings awaiting verification |
| **Ownership** | Unowned repository count | | | Active repositories | Unconfirmed owners shown separately |
| **Visibility** | Sensitive record count | | | Repositories scanned | Criteria changes this period |

| | | | |
|---|---|---|---|
| **Evidence** | Control test coverage (Appendix G) | Test exception aging | Inconclusive and overdue shown separately |

Accompany the scorecard with material scope or method changes, the oldest relevant open items and accepted exposures from Chapter 20, and decisions requested. Clearly label additional decision-relevant information and retain its own definition, unit, and scope rather than combine it with the five standing measures.

# Appendix D: Questions to Ask Before an AI Project

These questions support Chapters 23 and 26. Sections 1 through 8 apply when considering automated classification, before committing to scope, provider, or timeline. Section 9 applies to operational AI systems connecting to organizational information, including classifiers. The appendix addresses operational adoption, not detailed model development or training-related information.

The questions are not a scoring instrument. Questions may remain open during controlled validation where the test population, users, access, processing restrictions, interim protections, and end point have been authorized. Operational connection and reliance require the applicable assessment, decision, and verification. Recording a limitation with an owner does not authorize proceeding. Confirm provider-dependent answers through appropriate validation and record the evidence and its limits.

## 1. Scope and Constraints

- **What decision will the classification support?** Name the decision: retention assignment, identification of duplicate and transitory content, a legacy exit, a migration, or locating a category of sensitive information within a known scope. A project without a named decision is difficult to validate.
- **What is the first question the classifier will be asked?** Record whether the first pass targets duplicate and transitory content, a small number of category groups sharing a lifecycle outcome, or the full schedule. Broader first questions are harder to validate and improve.
- **Which repositories and content are in scope, and which are excluded?** Identify the repositories from the repository register and the reason for each exclusion.
- **Are there constraints on where processing can occur?** Identify information that cannot leave the organization's environment or may be processed only by approved people or systems, and record how those constraints affect the approach.
- **Is anything in scope subject to a hold or another encumbrance?** Confirm how held content will be identified and protected before any output is used to support disposition.

## 2. Category Readiness

- **Does each category in scope have a definition, exclusions, and a worked example?** Where it does not, expect the classification project to become a category redesign project.
- **Have categories that produce the same lifecycle outcome been consolidated or grouped?** The different-outcome test in Chapter 9 identifies candidates. Record distinctions that remain for legal, business, access, or evidentiary reasons.
- **Has classification agreement been measured for these categories?** A baseline from independent reclassification shows whether people can apply the distinctions consistently. Categories people cannot distinguish are unlikely to be distinguished reliably by a classifier.

• Does each category have an accountable owner and a designated reliance authority? Identify who may approve automated assignment and its validation criteria under the applicable delegation. Category ownership alone does not grant that authority.
• **Is there a documented approved treatment for each category in scope?** Classification identifies which approved treatment applies. It does not create one.

## 3. Routing and Thresholds

• **What routes will assignments follow?** Confirm that the design includes assignment with sampling, assignment after review, referral to a specialist, and an unresolved route that does not force a category.
• How are reliance criteria set for each category or group? Use validation and the consequences of both premature disposition and excessive retention. Use confidence only where it is available, meaningful, and validated for that scope. Without a usable score, use approved non-score criteria supported by validation, review, or unresolved status; do not invent confidence values.
• What content requires specialist review or preservation action regardless of confidence? Identify restricted, sensitive, or potentially held content, the applicable restrictions, and the qualified and authorized reviewers. A high confidence value does not override a required route.
• **How will unresolved items be recorded and owned?** Retain current treatment only where review confirms it is appropriate. Otherwise record the authorized interim treatment, restrictions, owner, and next-action date, linking the unresolved condition to open work. Unresolved does not automatically mean indefinite retention or blanket suspension.

## 4. Review Capacity

• **Who will review assignments, and how much time is committed?** Name the reviewers and specialists and the capacity assigned to each.
• **How was review volume estimated?** Use authorized controlled validation to estimate the volume and effort of each route. Let capacity inform scope, sequence, and resourcing, not justify weaker reliance criteria.
• What happens if review demand exceeds capacity? Narrow or stage the work or provide additional support. Change reliance criteria only where validation and the consequences of error support the change and the designated authority approves it; formally recording a decision is not enough to justify clearing a queue by lowering standards.
• **How will reviewer corrections be used?** Confirm that corrections feed the next cycle and are analyzed for whether the cause lies in the classifier, the instructions, or the category definitions.

## 5. Validation and Reliance

• **How will results be validated in the organization's own environment?** Use the classification agreement test in Appendix G: a documented sample, classified independently by a qualified reviewer, compared with the automated assignments.

• **Will agreement be reported by category or group?** Retain examined and agreeing counts, the category-allocation method, and confusion information, with corrected-assignment results separate. An overall figure or a provider test set is not sufficient alone, and group-level agreement does not validate every distinction within the group.
• **What level of agreement is required before automated assignment takes effect?** Record the level for each category or group and who approved it. Categories that do not reach it remain in a review route or unresolved.
• **How many validation cycles are expected?** Plan for refinement of definitions, exclusions, examples, and instructions between cycles, and record what changed.
• **What will each assignment record contain?** Retain the item or population reference; proposed and applied category or mapped group; system and use references; classifier, category-model, instruction, and routing-criteria versions; date; confidence where available; actual route; and any reviewer decision or correction. For an approved automatic route, link applicable sample records without implying individual review. Grouping remains a mapped classification aid, not an unapproved schedule change.

## 6. Connection to the Lifecycle

• How will populations identified for disposition reach the approved action? Confirm standing category-level authorization and the applicable eligibility, encumbrance, and dependency checks in Chapter 13. Classification output identifies candidates; it is not permission to delete.
• How will assignments reach the systems applying retention? Keep item or batch assignment records in their appropriate source and verify their application. The System-to-Policy Implementation Matrix maps scoped treatments and controls to category/repository pairings and links the assignment evidence; it is not an alternative item-level ledger.
• **How will results be reported?** Show original-assignment agreement by category or group, alongside human-review routing and unresolved shares for the same population and cutoff. Count each item once by primary route. Samples of automatic assignments are not individually routed reviews; report review completion separately. Keep unresolved conditions and accepted limitations visible.

## 7. Change and Revalidation

• How will classifier, instruction, or routing-criteria updates be handled? Record versions and what changed. A rerun of an earlier sample shows changes on those items; record whether the sample informed tuning and use independent resampling for broader reliance. Assess whether earlier assignments need review and apply the approved transition.
• How will category or source-content changes be handled? Use Chapter 19 to assess effects on assignments, treatment, scope, and reliance, revalidate the affected work, and decide whether existing assignments require review. Retain prior versions and the evidence supporting any continuing reliance.
• **How often will results be resampled when nothing has visibly changed?** Set a cadence so that gradual decline in agreement is identified before it affects a disposition cycle or a

request for records.

## 8. Provider Terms

- **Will the provider give notice before changing the classifier?** Seek notice and enough time to revalidate before a change takes effect, where the relationship allows.
- **What information will the provider make available for validation and explanation?** Confirm access to usable version identifiers, configuration and instructions, assignment records, and any meaningful confidence values actually available. Where a score is unavailable, assess the approved non-score route rather than request an invented value.
- **Where will content be processed and retained, and for how long?** Confirm processing locations against the constraints identified in scope, and the provider's retention of submitted content and outputs, as Chapter 17 describes for other externally held information.
- **What happens to configuration, instructions, and records at the end of the relationship?** Confirm export of the organization's configuration, instructions, and assignment records, and deletion of submitted content where appropriate.

## 9. Connecting an AI System to Information

Assess each system-to-source connection before operational use. Record stable system, use, source, and population references. Chapter 26 distinguishes this decision from bounded access authorized for validation; testing permission does not authorize unrestricted processing or reliance.

- **Scope.** Which registered repository or collection, source population, intended uses, and users are in scope, and who is accountable? Record known legacy and ownership gaps and the authority designated for the connection decision.
- **Access.** Do the permitted users retrieve what they should and fail to retrieve what they should not? Verify the relevant retrieval and restriction behavior within the intended scope rather than rely only on a permissions list.
- **Authority and currency.** Which instances and versions are authoritative for this use? Distinguish superseded or expired convenience content from legitimate retained history; specify appropriate retrieval or exclusion without assuming all older information should be destroyed.
- **Classification.** Is sensitivity identification validated well enough for the restrictions that depend on it? Keep record-category assignment, sensitivity identification, and AI-use risk classification distinct and examine each against its own criteria.
- **Provenance and permitted use.** What is the information's provenance, and what contractual, consent, residency, or other restrictions apply to the proposed users, purpose, and processing? Verify that the proposed use is permitted.
- **Lifecycle.** Which indexes, caches, copies, and other derived stores are created, and how will the approved retention, hold, and disposition treatments reach them? Distinguish

retrieval removal, index removal, and destruction, identify residual periods, and compare observed delay with approved timing.

- **Evidence.** Do stable references link the connection to its assessment, actual decision date, applicable version, permission conditions, verification, review point, and unresolved work in the system record? Retain historical assessments and their limits.

Record the operational outcome: connect, connect after remediation, connect with a documented limitation, or exclude. Required remediation must be completed and verified before connection. A limitation needs an authorized decision with scope, rationale, interim restrictions, review point, any time limit, and follow-through; a risk acceptance also needs expiry and any earlier review. Acceptance does not resolve the condition or authorize use outside applicable requirements. Exclusion from retrieval does not authorize destruction.

## Using the Answers

Reassess relevant changes in use, source content, permissions, authoritative versions, provider arrangements, or lifecycle behavior, and on a proportionate cadence. Retain prior assessments and actual decision dates, and link shared findings to affected uses and connections through the existing change and open-item processes.

Record answers and evidence with the scope record and keep them current. Some unanswered questions can be investigated within authorized validation; unmet prerequisites can require deferral or exclusion. Assign owners, dates, and interim restrictions to actual gaps, without treating the existence of an open item as permission to process information or rely on a result.

The classifier is also an AI system. The system-level record described in Chapter 25 should draw on these answers rather than restate them.

# Appendix E: Skills Matrix for Information Governance Teams

This appendix supports Chapter 31. It contains definitions of the three skill levels, a matrix of the five skill areas with observable indicators at each level, a task coverage worksheet, and an illustrative starting list of recurring program tasks.

The matrix describes the capability the program's work requires, not a general rating of a person or a performance appraisal. Use observable work or reviewed demonstrations to support the assessed task level. Capability coverage also depends on required review and available support; it does not establish that recurring work was performed on schedule.

## 1. Skill Levels

- **Awareness.** Understands what the skill involves, recognizes when a task calls for it, and knows whom to involve.
- **Working.** Performs the related tasks with review, using the program's methods and templates.
- **Lead.** Performs the tasks independently, reviews the work of others, and improves the method.

Levels apply to tasks, not to people in general. A person may work at lead level in one area and awareness level in another, which is expected.

A lead level describes capability, not delegated decision authority or an exemption from required independent review. Identify the assignment, authority, reviewer, and available time appropriate to each task, using Chapters 27 and 31 and Appendix B.

## 2. Skills Matrix

Figure E.1. Skill areas and observable indicators by level.

| Skill area | Awareness | Working | Lead |
|---|---|---|---|
| **Legal and regulatory** | Recognizes when a new requirement, matter, or business event may affect a period, trigger, or hold, and refers it | Drafts rule elements, basis, and rationale for authorized review; implements holds within an approved scope | Interprets requirements into treatments; records the basis and alternatives considered; works with counsel on privileged analysis; reviews others' drafts |

| | | | |
|---|---|---|---|
| **Technical** | Knows how the main platforms apply retention, holds, and deletion, and what the enforcement modes mean | Specifies configuration for a pairing and verifies it against the approved rule with a configuring party | Assesses platform capability and limits; designs controls where native capability is absent; assesses AI systems and their connections to information |
| **Analytical** | Reads a measure with its scope, cutoff, and limitations and can explain what it does not show | Runs a defined sample or test using a documented method and records results and exceptions | Designs measures, counting rules, and sampling methods; reviews test design; explains changes in basis between periods |
| **Change management** | Explains governance requirements in the terms of the audience's work | Plans and delivers role-based communication and training; checks defined task performance separately from attendance | Plans sponsor and owner engagement, tracks adoption through change, and records responses to resistance or capability gaps |
| **Product** and service management | Knows which governance services the business uses and how requests enter the program | Maintains an intake and backlog; records pattern uses; reports response times | Designs services and patterns around user needs, proposes response expectations for approval, and uses request evidence to identify standardization opportunities |

## 3. Task Coverage Worksheet

Record one row per recurring task. The worksheet follows the assessment method in Chapter 31.

Figure E.2. Task coverage worksheet fields.

| Field | What it records |
|---|---|
| **Task** | A recurring task the program performs or has committed to perform |
| **Source** | The chapter, procedure, or commitment the task comes from |
| **Frequency** | How often the task recurs, such as continuous, monthly, quarterly, or annual |
| **Skill areas and level** | The areas the task draws on and the level required to perform it |

| | |
|---|---|
| **Team holders** | Named performers, assessed task levels, assessment evidence and dates, linked role assignments, and time allocated for the work |
| **Partner holders** | Named provider or function, supported capability, service and handoff arrangements, availability, and links to Appendix B assignments; preserve independent responsibilities |
| Supported capability status | Supported at the required level, or unsupported. Distinguish absent capability, unverified assessment, unavailable reviewer, and other missing necessary support |
| **Option and owner** | For each gap or continuity response: develop, borrow, hire, buy, or redesign; responsible owner, action or review date, and linked open-item or continuity record |
| **Assessment date** | The date of the assessment and the next scheduled review |
| Required review | Named reviewer or defined reviewing function, required level and independence, assessment basis, availability when work is due, and backup path |
| Continuity status | Concentration in the performer, reviewer, or provider; alternate capability or backup arrangement; distinguish one named provider with supported backup from dependence on one specialist |
| Authority and support links | Relevant assignment and delegation, partner agreement, continuing provision, and any limits; keep authority separate from skill level |

Continuity-critical tasks need a defined way to continue or obtain the required expertise during absence or change. Two names do not establish supported capability, and one capable holder may provide current coverage while creating a concentration risk. Assess both the performer and any required reviewer, including deliberate external support with demonstrated backup capacity.

## 4. Illustrative Recurring Tasks

The list below is an illustrative starting point. Adapt the tasks and levels to the program's commitments. Preparing, approving, implementing, testing, and correcting work are separate activities: a suggested skill level grants no authority and does not waive a required review.

Figure E.3. Illustrative recurring program tasks.

| Task | Skill areas | Typical level | Source |
|---|---|---|---|

| | | | |
|---|---|---|---|
| **Interpret a new or changed requirement into a treatment** | Legal and regulatory | Lead | Chapters 7, 18, and 19 |
| **Assess and classify a proposed rule change** | Legal and regulatory, analytical | Working | Chapter 19 |
| Prepare an onboarding assessment and record the authorized decision | Technical, legal and regulatory | Working | Chapter 2, Appendix F |
| **Specify and verify configuration for a pairing** | Technical | Working | Chapter 11 |
| **Maintain the register and implementation matrix** | Technical, analytical | Working | Chapters 8 and 11 |
| Support authorized hold scoping and release decisions; implement and verify their approved scope | Legal and regulatory, technical | Working | Chapter 15 |
| Perform and evidence disposition under standing category-level authorization and recurring eligibility, encumbrance, and dependency checks | Legal and regulatory, technical | Lead | Chapter 13 |
| **Calculate and report the standing measures** | Analytical | Working | Chapter 21, Appendix C |
| **Design and perform control tests** | Analytical, technical | Lead | Chapter 22, Appendix G |
| **Manage open items and escalation** | Product and service management, change management | Working | Chapter 20 |
| **Assess an AI use and its risk classification** | Legal and regulatory, technical | Lead | Chapters 24 and 25 |
| **Assess readiness of a source for an AI connection** | Technical, analytical | Working | Chapter 26, Appendix D |
| **Validate and revalidate a classifier by category** | Analytical, technical | Lead | Chapter 23 |
| **Reconcile role assignments** | Change management | Working | Chapter 27, Appendix B |

| | | | |
|---|---|---|---|
| **Onboard a unit in a wave, including training** | Change management, technical | Working | Chapter 29 |
| **Maintain patterns and sample pattern use** | Product and service management, analytical | Working | Chapter 30 |
| **Review provider terms and evidence obligations** | Legal and regulatory, technical | Working | Chapter 17 |

## 5. Using the Results

- Report supported capability and continuity separately. State the recurring-task population, assessment date, and method. Count tasks without supported capability at the required level and those dependent on one holder or provider, including required review. Disclose overlap and do not add the counts as distinct tasks. Neither count establishes task completion.
- Treat actual gaps as managed work. Record unsupported capability, missing assessment evidence, or unavailable required review with an appropriate owner, response, and date under Chapter 20. Record concentration and continuity arrangements separately where appropriate; not every single-holder arrangement is the same deficiency.
- **Choose options by task.** Recurring core work often warrants development or hiring; occasional specialist work may be deliberately borrowed or bought. Record reliable access, required review, and handoffs. Redesign reduces unnecessary dependence on specialist judgment without lowering standards or removing review conditions.
- **Use vacancies.** When a position opens, compare the worksheet with the position description before refilling it.
- Reassess on change and on a defined cadence. Set the interval according to the rate of change. Reassess when scope, phases, organizational responsibilities, or key capability changes. Retain prior worksheets and explain changes in the task population or method before comparing results.

# Appendix F: Implementation Artifacts

This appendix holds five artifacts that connect the retention schedule to the environment: the schedule requirements checklist from Chapter 6, the onboarding gate from Chapter 2, the governance-grade repository record from Chapter 8, the decision record template developed in Chapters 18 and 19, and the System-to-Policy Implementation Matrix from Chapter 11.

Each artifact has a distinct role, and they are linked rather than merged. The register identifies repositories and ownership. The matrix identifies category/repository pairings, their scoped treatments, implementation, and verification. The onboarding gate supplies the required intake and decision records; its completion does not itself establish implementation. The decision record captures the decision, basis, authority, and conditions. Figure 12.1 remains in Chapter 12. Field tables describe what to record in the organization's existing tools; they are field guides, not completed approvals.

## 1. Schedule Requirements Checklist

Use the checklist before evaluating any solution. Add what the environment requires, and mark each item required or preferred. These are capability requirements, not product requirements; they may be met through a governance platform or a well-governed database and disciplined process.

Figure F.1. Schedule requirements checklist.

| Capability | What it must do | Required or preferred |
|---|---|---|
| **Single authoritative source** | Provide controlled access to the authoritative version and identify superseded or reference copies | |
| **Structured rules** | Hold the eight rule elements in Appendix H as discrete, queryable elements rather than text | |
| **Change control with durable history** | Record the author, approver, effective date, and reason for every change, and keep prior states retrievable | |
| **Point-in-time reconstruction** | State what rule governed a category on a given past date, produced from the system | |
| **Jurisdictional layering held once** | Define shared elements once and express variation as related rules rather than duplicated categories | |
| **Role-appropriate access** | Present what each audience needs from the same source | |

| | |
|---|---|
| **Machine-readable output** | Publish rule sets systems can consume, or support the reviewed translation in Chapter 11 |
| **Reportable evidence** | Report what the schedule holds, what changed, and who approved it without a separate project |

## 2. Onboarding Gate

Answer four questions before any new repository or system is provisioned, and record the answers in the repository register and, for each category, in the implementation matrix.

- **Which information categories will live here?** Use the schedule's own vocabulary.
- **What retention applies?** Identify the approved rule version for each category.
- **Who owns the control?** Name the control owner, distinct from the configuring party.
- **Can the platform enforce it?** Record native, integrated, manual, or not enforced. An honest no creates a known gap with an owner rather than an assumed control.

AI tools. For a tool that generates, summarizes, or processes content, identify what it retains, including prompts, outputs, logs, and derived content; where; for how long; and its capture, export, preservation, and deletion capabilities. Record use-specific treatment and evidence needs under Chapters 24 and 25. Source connections follow Appendix D Section 9, distinguishing bounded authorized validation from permission for operational connection and reliance.

Record the gate outcome using all applicable mandatory fields in both parts of the decision guide below. Distinguish authorized controlled validation, including its scope and end point, from operational approval. Record conditional, deferred, refused, or other applicable outcomes. Conditions identify the evidence and authority required before operational permission takes effect; a recorded assessment or named gap owner does not supply that permission.

## 3. Governance-Grade Repository Record

Figure F.2. The governance-grade repository record.

| Field | What it answers |
|---|---|
| Repository identifier, name, and type | A stable identifier and descriptive name; whether the repository is a system of record, archive, extract, or working store; link related source, copy, and derivative stores |
| **Business function and named owner** | Who is accountable for the information, as a person rather than a department, and the decision authority they hold or the authorized decision maker to whom they refer lifecycle decisions |
| **Business processes supported** | Why the information exists, which makes the category mapping defensible |

| | |
|---|---|
| **Record categories present** | Which rules apply, in the schedule's own vocabulary |
| **Sensitive data types present** | Sensitive types identified and the scope, method, and date of identification; an indicator of potential exposure, not the full incident or notification population |
| **Hosting location and residency constraints** | Where the information is hosted, whether it may move, and where the control must operate |
| **Third party and contract reference** | Which outside party holds or manages the information and which agreement governs the arrangement |
| **Lifecycle capability** | How retention, disposition, and holds are supported, including any manual controls; for deletion, the terminal state and the residual period before that state is reached |
| Lifecycle and ownership status | Lifecycle state, such as active, legacy, or planned retirement, separately from missing, invalid, or unconfirmed ownership. A legacy repository may also be orphaned |
| **Last verified date and by whom** | What fields or capabilities were checked, against which evidence, when, by whom, and with what result and limitations; link retained checks and prior states |

Being named as owner in the register does not by itself grant approval authority; Chapter 18 develops the authority record and Appendix B the decision rights. For deletion, the terminal state is the point at which content can no longer be restored through routine administrative action, and the residual period runs from the deletion instruction to that state. Record these in the repository record or a linked capability record, including the treatment of versions, replicas, and backups described in Chapter 13. Part V reuses repository references when recording AI systems, uses, and connections; those are related units, not interchangeable inventory entries.

## 4. Decision Record Template

One decision structure serves retention and other governance decisions. Every decision records the decision, basis, rationale, authority, information available, and review point in proportionate detail, completing applicable fields across both parts. Mark a genuinely inapplicable field with its reason; distinguish it from awaiting information or an unresolved required condition. Onboarding, pattern, acceptance, and AI decisions do not omit authority merely because it appears in Part 2.

AI-use approvals, risk classifications, source connections, and pattern decisions adapt the subject and stable references while retaining the six common elements. Retention decisions additionally use the requirement-driven, requirement-extended, or judgment-based classification from Chapter 18.

Those labels are not AI risk classifications and are not required where no retention treatment is decided.

Part 1: Scope of the Decision

Figure F.3. Decision record, Part 1.

| Field | What it records | When required |
|---|---|---|
| **Decision reference and date** | A unique reference and the date the decision was actually made | Every decision |
| **Decision type** | Onboarding; new or changed rule; rule exception; current re-decision; risk or limitation acceptance; pattern approval, revision, or withdrawal; AI use/risk classification; source connection; or another identified governance decision | Every decision |
| **Subject** | The decision subject and stable references: category, scoped treatment, repository, system, use, connection, pattern, or unresolved condition, including relevant versions | Every decision |
| **Repository or system** | Existing register and system identifiers, with their relationships distinguished; do not create a repository entry merely for an AI use or source connection | Onboarding gate and pairing-specific decisions |
| **Applicable scope** | The populations, jurisdictions, or uses to which this decision applies, where a category or system has more than one treatment | Every decision; state the whole scope or the defined subset |
| Decision and resulting treatment or conditions | What was decided and any resulting treatment, rule version, use conditions, pattern version, restrictions, or no-action outcome. For retention, include the approved value, any required trigger, and lifecycle action | Every decision |
| **Control owner** | The named owner accountable for the control that carries out the decision | Every decision establishing, changing, or relying on a control, including a future control |

| | | |
|---|---|---|
| **Enforcement mode** | Native, integrated, manual, or not enforced, as defined in Chapter 11 | Onboarding gate and pairing-specific decisions |
| Outcome, permission, and conditions | Approved operational use; controlled validation with defined scope/end; conditional; refused; deferred; withdrawn; or no action, as applicable. State unmet prerequisites, verification needed, who confirms them, and the limits of any permission | Every decision, as applicable to its subject |
| Effective period | When the decision and its conditions take effect, and any end date; distinguish these from the actual decision date, review point, expiry, and completion status | Every decision with continuing effect |

Part 2: Basis for the Decision

Figure F.4. Decision record, Part 2.

| Field | What it records | When required |
|---|---|---|
| **Basis** | The applicable requirements, criteria, and evidence supporting the decision. For retention treatment, also identify requirement-driven, requirement-extended, or judgment-based | Every decision; retention-basis classification only where relevant |
| **Requirements considered and interpretation** | Requirements and limits considered, how they were interpreted, and any trigger or permission conditions they imply | Every decision to the extent relevant; explain a not-applicable entry |
| **Rationale and alternatives considered** | Why the decision or treatment was chosen over reasonable alternatives; include requirement interpretation and any business need extending a period beyond an applicable minimum | Every decision, in proportionate detail |
| **Limits and deletion obligations weighed** | Confirmation that any applicable retention limit or deletion obligation was considered, and how it was resolved | Every decision with applicable limits or deletion obligations, not only extended retention |

| | | |
|---|---|---|
| **Decision maker and authority** | Who decided, and the delegation, charter, or policy provision that gave them the authority on the decision date | Every decision |
| **Information available** | What was known at the time, including open questions, dissenting views, and any risk accepted | Every decision, including pattern and onboarding decisions |
| **Legal advice reference** | Where any privileged analysis is held, and a statement of the basis that can be recorded without disclosing protected advice, as agreed with counsel | Decisions relying on legal advice |
| **Review point** | When, or under what conditions, the decision will be revisited | Every decision; record any justified not-applicable treatment |
| **Expiry** | For rule exceptions, risk acceptances, and time-bound authorizations, the date the decision ends. A review does not extend an expired decision; a new decision is required | Rule exceptions, risk acceptances, and time-bound authorizations |
| **Transition** | Affected existing and future populations, uses, sources, outputs, or pattern uses; effective timing; approved completion criteria; interim restrictions and their owner/end condition; required verification; and any newly eligible disposition population and checks. A changed trigger identifies the approved fallback for existing information | Changes and re-decisions affecting existing treatment, implementation, use, or reliance |
| **Pattern conditions** | Eligible situations; approved version/effective period; applicability criteria; required treatment; pre-use checks and who performs/confirms them; retained records; later conformity checks and cadence; review outcome; and effects of revision or withdrawal on earlier uses | Pre-approved pattern decisions |
| **Version links** | Superseded and superseding decision, rule, pattern, AI approval, or risk-classification references, with effective periods and retained history | Every revision or replacement with prior states relevant to reliance |

| | | |
|---|---|---|
| Unresolved work and response | Underlying condition, owner, original finding date, action/review dates, restrictions, and open-item links. Acceptance supplements rather than replaces ownership and follow-through | Decisions leaving unresolved work, including acceptances |

Drafting rules.

- **Date the decision when it is made.** A re-decision of an inherited rule carries its own date. It is never backdated to the date of the original rule, and a new dated review does not recreate missing historical reasoning.
- **Link, do not overwrite.** Retain prior decisions, applicable authority, conditions, versions, and effective periods for rules, AI uses, connections, and patterns. A new decision links to what it supersedes so reconstruction uses the actual record applicable to the date examined.
- Make the record part of the decision process. Complete the applicable mandatory fields across both parts for every authorized decision. Record actual refusals or deferrals with their reasons and outstanding conditions. An incomplete prerequisite does not become permission because a form is filled out. Changes affecting implementation require the relevant transition and verification; other decisions close against their recorded completion criteria.
- **Document each treatment.** Where a category has different treatments, document each treatment's basis and scope, referencing shared reasoning where appropriate.
- **Keep rule elements and decision contents separate.** The eight rule elements in Appendix H describe what the rule is. This template describes why. The two are linked by rule version rather than merged.
- Record pattern uses against the approved version and effective period. Retain the applicability basis, required pre-use checks, resulting register/contract/use/change references, and later conformity findings. An out-of-pattern request identified before action follows full review; it becomes an open-item finding only where a gap, unmet requirement, overdue work, or nonconforming action is identified. Distinguish misapplication from a pattern defect. Record the effect of revision or withdrawal on prior uses and any required transition or correction; a documented no-change review is legitimate.

For service timing under Chapter 30, retain receipt, completeness, decision, and pause dates with reasons. Read completed decision times beside pending-case age and material completeness delays. Pre-approval removes repeat policy approval, not required instance-level checks; reusable design evidence does not automatically verify each deployment.

## 5. System-to-Policy Implementation Matrix

The matrix keeps one parent record per category/repository pairing. Where its populations have different approved treatments, use linked scoped records for each treatment's version, effective period, enforcement mode, roles, configuration or procedure, and verification evidence. Shared

controls and evidence may be referenced without duplicating volume. It builds on the register and published rule sets; it is not an item-level assignment ledger.

Figure F.5. The System-to-Policy Implementation Matrix.

| Field | What it answers |
|---|---|
| **Repository** | Where the information sits, taken from the register rather than created here |
| **Record category** | Which rule applies, in the category model's own vocabulary |
| **Rule version and effective date** | The applicable rule version and effective date for each scoped treatment in the pairing |
| **Configured trigger** | The event or date field used, or the documented procedure, and how it matches the approved trigger or authorized alternative |
| **Enforcement mode** | Native, integrated, manual, or not enforced for each scoped treatment, with the applicable configuration or procedure version |
| **Retain, preserve, dispose, evidence** | Which lifecycle actions the repository can perform for this category and where additional controls are required |
| **Control owner** | Named person accountable for operation of each applicable scoped control; link assignment and authority history |
| **Configuring party** | Who implements and maintains the approved configuration or procedure for the scoped treatment, with departures referred for a decision |
| **Dependencies** | Downstream reliance on the information, as assessed for disposition under Chapter 13, with the date of the last review |
| **Last verified date and method** | When and how the configuration or documented procedure was verified against the approved treatment, with supporting evidence; operating-test results are linked separately |
| **Gap response** | For every gap: responsible owner, original date, appropriate action or review date, and open-item link. Any acceptance also records rationale, authorized approver, expiry, and earlier review; acceptance does not replace ownership or follow-through |
| History and evidence links | Prior scoped treatment, configuration/procedure and assignment versions with effective periods; verification references and separately linked operating tests and execution outcomes |

- **State the scope.** Record which categories and repositories are in scope and why, so that a missing row reads as an absence of analysis rather than an absence of risk.

• **Keep one controlled point of translation.** Where a platform cannot consume a published rule set, derive every configuration from the reviewed matrix row.
• **Record verification, not status.** Configuration verification keeps the implementation record current; operating tests, recorded in Appendix G, establish what the control did within their examined scope.
• **Carry changes through.** An approved change generates a pairing-level implementation assessment and a change request against each affected pairing, closed on verification. Configuration lag in Appendix C measures that route.

# Appendix G: Evidence and Testing Artifacts

This appendix holds six items that support Part IV and the testing work that later parts rely on: the defensibility evidence checklist, the control-evidence register, the control test record, the point-in-time reconstruction test, the definitions of control test coverage and test exception aging, and the classification agreement test. Chapters 18 and 22 develop the reasoning. No further structures are added here; open items are managed through the Chapter 20 open-items view rather than a separate index.

Four states are kept distinct throughout: intent (what was decided and why), implementation (what was configured or procedurally required), operation (what the control did during a period), and resolution (whether a finding has been verified as closed). Evidence answering one of these does not answer the others.

## 1. Defensibility Evidence Checklist

Use the checklist to confirm, for a category or control, that each link in the chain from decision to outcome is supported by a record the program already keeps. Where a link is missing, record it as an open item or a documented limitation.

Figure G.1. Defensibility evidence checklist.

| Link | Evidence | Source |
|---|---|---|
| **Decision and rationale** | A decision record with basis, rationale, authority on the decision date, information available, and review point | Chapter 18, Appendix F |
| Rule or approval history | Prior rule or applicable approval versions, effective periods, authority, and the recorded changes | Chapters 7 and 19 |
| **Change and transition** | Change class, impact, and transition where required, closed against recorded criteria; changes affecting implementation include verification | Chapter 19 |
| **Implementation** | An implementation matrix row with enforcement mode, owners, and the date and method of verification | Chapter 11, Appendix F |
| Authorization and pre-action checks | Category-level disposition authorization plus applicable eligibility, encumbrance, and dependency records; hold/suspension scope, implementation, and release decisions where relevant | Chapters 13 and 15 |

| | | |
|---|---|---|
| **Execution** | Records of the approved outcome for the control examined: preservation, release, continuing custody/accessibility, transfer, anonymization, archival, or destruction. For disposition, retain population, rule version, authority, checks, action/date, and verified outcome; destruction evidence identifies the terminal state | Chapters 13 and 15; applicable control process |
| **External parties** | Contractual evidence obligations and the evidence received under them | Chapter 17 |
| **Testing** | Control test records with scope, results, limitations, and responses | Chapter 22 |
| **Open items** | Known gaps, accepted exposures with expiry, and referrals, with owners and dates | Chapter 20 |
| **AI uses, where relevant** | The per-system record and its links to the structures above | Chapter 25 |

## 2. Control-Evidence Register

The register is a maintained map of expected and actual evidence, its source, production periods, location, retention treatment, and retrieval checks. Its existence does not establish that an expected artifact was generated, received, or retained. Record missing periods and limitations with the responsible source and applicable open work.

Figure G.2. Control-evidence register fields.

| Field | What it records |
|---|---|
| **Control or subject** | The pairing and implementation-matrix reference; or, for AI systems and connections, the system, use, version, and connection reference |
| **Evidence artifact** | The record produced, such as configuration exports, execution logs, disposition certificates, attestations, or test records |
| **Produced by** | Responsible system, procedure, or external source; expected production and actual periods received or retained, with known gaps |
| **Location** | Where it is held, and who can retrieve it |
| **Retention treatment** | Approved category, retention value, any required trigger, action, and supporting decision; link separately assessed business-record or documentation purposes |
| **Linked records** | Business records or other governed records the evidence refers to or reuses, with the usable reference maintained |

| | |
|---|---|
| **Retrieval check** | When retrieval was last confirmed, by whom, and the result |

Retain control evidence under the approved schedule category, with an operable retention value, any required trigger, and a documented basis for examining decisions, implementation, operation, measurement, testing, and follow-through (Appendix H). Do not leave treatment to platform defaults. Where a record also serves a business or documentation purpose, assess each and preserve usable references. The end of one purpose's period does not authorize disposal contrary to another applicable treatment.

## 3. Control Test Record

Figure G.3. The control test record.

| Field | What it answers |
|---|---|
| **Control tested** | The category and repository pairing, its implementation-matrix reference, and the scoped treatments examined; or, for AI testing, the system, use, version, connection, and approval reference |
| Rule or approval version and effective period | Applicable rule or approval conditions, version and effective period, and relevant transition or fallback; use system, use, and source references for an AI subject without inventing retention pairings |
| **Test type** | Configuration match, operation, or absence; identify or link classification-agreement and reconstruction tests |
| **Test criterion and procedure** | The expected result and the steps used to examine it |
| **Population and selection** | The population source, reporting cutoff, sample size, selection method, and scope limitations |
| **Tester and assurance tier** | Who tested, their relationship to the control, and whether the work was self-assessment, independent testing, or audit |
| **Test date and period covered** | When the test ran and which operating period it examined |
| **Result** | The observation, supporting evidence, limitations, and conclusion against the criterion; a pass or fail summary does not replace these |
| **Test exceptions and response** | The finding and its original recorded date, the chosen response, owner, action or review deadline, and closure or review evidence |

| | |
|---|---|
| **Evidence retained and where** | The retained artifacts, their location, and their entry in the control-evidence register |

- **Three control questions.** State whether the procedure examines existence, alignment with approved requirements or conditions, or operation. A missing control is a gap; a mismatch is an implementation defect; missing execution evidence limits what can be established about the period. Keep the evidence observation distinct from an inference about failure.
- **Test the approved outcome.** Permanent retention calls for evidence of continuing preservation and custody; transfer, anonymization, archival, and destruction each need their own criteria. An absence test is limited to its search scope and is read with execution records and the repository's terminal state and residual period.
- **Establish the population first.** Use the register and matrix to identify locations and controls, and source, eligibility, or execution records to establish the population for the period. An incomplete population is a scope limitation, recorded as such.
- **Three responses to a test exception.** Correct the problem; obtain an authorized risk acceptance with rationale, expiry, and any earlier review; or refer the underlying rule, use, connection, or approval conditions through the relevant change process. Record restrictions and follow-through separately from the unresolved condition. Acceptance and referral are not verified resolution, and a referral need not imply a rule change or suspension of every use.

## 4. Point-in-Time Reconstruction

The reconstruction test asks, for a stated category and past date, whether the program can show the full chain as it stood on that date.

- **Rule.** Which rule version governed the category, from the schedule's version history.
- **Reason.** Why it was approved, from the decision record in effect on that date and its authority.
- **Implementation.** What was configured or procedurally required in each relevant pairing, from the implementation history.
- **Outcome.** Whether the recorded outcome matched, from retained execution evidence and hold records.

Record the reconstruction test with the past date, sources, relevant versions, and result for each link. Where historical material is missing, identify the limitation rather than present a later explanation as the original record. Keep the limitation visible while it remains relevant and unresolved, including any authorized acceptance, expiry, and review history. Link completed current corrective work separately; later evidence generation does not recreate missing records.

## 5. Control Test Coverage and Test Exception Aging

Control test coverage. The share of in-scope operative controls independently tested in the preceding twelve months. Operative means verified configuration or a verified documented manual

procedure, consistent with Appendix C, not demonstrated effectiveness.

- **Scope and cutoff.** Define the denominator and reporting cutoff, and count each control once.
- **What counts.** Count completed tests with retained scope, methods, and results, whether or not they identified a failure. Incomplete tests remain outside the numerator.
- **What is shown separately.** Report evidence-limited or inconclusive results separately, so that completed testing is not read as confirmed operation, and show testing overdue against its risk-based cycle.

Test exception aging. The age of an unresolved test exception, from its initial recorded finding to the reporting date.

- **Dates.** Retain the original finding date and, for closed items, the verified resolution date.
- **No resets.** Acceptance, referral, or a change of owner does not reset the age of the underlying unresolved condition.
- **Lateness.** Track lateness against the relevant action or review deadline as a separate comparison.
- **Historical limitations.** Report historical evidence limitations separately from current corrective actions, with links between them.

Both measures appear on the Appendix C scorecard beside the library measures, as a pair. They are not additions to the five-measure library.

## 6. Classification Agreement Test

Classification agreement tests whether categories are being assigned consistently. A control can operate correctly on information assigned to the wrong category, so agreement is an input the other tests depend on.

- **Sample.** Draw a documented sample from the population classified, recording the population, cutoff, selection method, and size.
- **Independent review.** Have a qualified reviewer classify the sample without seeing the original assignment.
- **Compare against the original assignment.** Measure agreement with the assignment as originally made, before any later correction, so that the result describes how the method performs. For automated assignment, record the route each item took and whether review was completed, since routing an item to review is not the same as reviewing it.
- **Report by category or group.** Retain counts examined and agreeing, category-allocation method, confusion information, and corrected-assignment results separately from original-assignment results. Group-level agreement supports only the distinctions examined. State any population or method change before comparing periods.
- Record versions and sample use. Retain classifier, category model, instructions, routing criteria, and relevant content changes. A prior-sample rerun supports comparison on those

items; record whether it informed tuning and use independent resampling for broader reliance. Assess whether affected earlier assignments require review and apply the authorized transition. Chapter 23 develops this process.

Record the test in the control test record. Classification agreement remains a sampling test here, not a standing library measure. Sensitivity identification and AI-use risk classification answer different questions and are validated against their own criteria (Chapter 26).

For automated routing, retain the same population and cutoff for human-review and unresolved shares, counting each item once by primary route. Samples of automatically applied assignments are not individually routed reviews. Report completion separately from referral and work still awaiting a decision. AI-use risk tiers remain distinct from the assurance tier of the test.

# Appendix H: Schedule Structure Reference

This appendix is the structural reference for Part II. It holds the schedule data model and its eight rule elements from Chapter 7, the category model design rules from Chapter 9, and the jurisdictional variation register from Chapter 10, together with three schedule entries that later chapters depend on: permanent designations, the control evidence category, and system documentation for AI uses.

Chapter 7 defines what a rule is made of. Chapter 9 defines what the rule attaches to. Chapter 10 defines how the rule varies by jurisdiction without duplicating the schedule. The decision record in Appendix F explains why each rule is what it is, linked by rule version.

## 1. Schedule Data Model

Figure H.1. The eight elements of a retention rule.

| Element | What it holds | Why it has to be discrete |
|---|---|---|
| **Record category** | What the information is, in terms the business recognizes | Rules attach to it and systems classify against it |
| **Trigger event** | The event that starts a finite retention period or initiates a lifecycle action | Event-dependent periods and actions need a defined starting point; not applicable must be explicit |
| **Retention period** | A finite duration tied to a defined trigger, or an explicit permanent designation | An approved retention value must be distinguishable from an unassigned period |
| **Jurisdiction** | Where the rule applies, and at what level | Variation has to relate to the base rule rather than copy it |
| **Citation and authority** | The applicable requirement or source authority supporting the treatment | One citation supports many rules and changes on its own schedule |
| **Disposition action** | The authorized action when a period ends or another defined lifecycle event occurs | Destroy, transfer, anonymize, and review are different controls |
| **Effective dates** | When this version of the rule governs | Reconstructing a past rule depends on prior versions persisting |

| | | |
|---|---|---|
| **Exceptions and suspensions** | What changes or suspends normal treatment, with scope and authority; expiry and any earlier review date for time-bound exceptions; review dates and release criteria for preservation suspensions | Time-bound exceptions need expiry and explicit renewal; preservation suspensions remain until release criteria are met |

Citations, jurisdictions, and triggers are defined once and related to every rule that uses them. Triggers are drawn from a controlled vocabulary, which is what trigger coverage in Appendix C counts against. The citation-and-authority element identifies the requirements supporting a rule; it does not replace the decision record or the delegation under which the approver acted.

Retention period. The element records a finite duration tied to a defined trigger, or an explicit permanent designation. An unassigned period is an unresolved rule. Retention pending a defined event or review must identify the condition and the treatment that follows, and must not be silently converted into permanent retention.

Trigger event and disposition action. A defined trigger is required wherever a retention period or lifecycle action depends on an event. Record not applicable explicitly, with a reason, where no trigger is required. Permanent records may still require cutoff, transfer, or review events, without an artificial destruction date. The disposition action identifies the authorized treatment when a period ends or another defined lifecycle event occurs. Where a rule change alters the trigger, the rule states the fallback start date for information created before the change, approved as part of the change, rather than relying on whatever date a platform has available. Chapter 19 treats this as part of the transition statement recorded in the Appendix F decision record.

## 2. Permanent Designations

A permanent designation carries a documented basis, scope, authority, approval, accountable custodian, preservation arrangements, and review date. Review does not itself authorize destruction. Any change to the approved treatment follows the applicable authority and change process. Keep a record of permanent designations and list unsupported designations as unresolved work, reported separately from missing triggers.

## 3. Category Model Design Rules

- **Apply the different-outcome test.** Sort categories by trigger, retention period, and disposition action. Categories sharing all three are candidates for consolidation, subject to legal, business, access, and evidentiary requirements. Matching outcomes do not by themselves establish that categories should merge.
- **Assess consolidation against limits.** A shared period can retain some information longer than its own requirement demands. Assess the effect against applicable retention limits, deletion requirements, and business needs before approving it, and record the analysis. Choosing the longest period is not sufficient on its own.

- **Record why each category remains distinct.** In most cases the reason is a different lifecycle outcome. Record other reasons where they apply.
- **Define with exclusions and a worked example.** Each category carries a plain-language definition, what it excludes, and at least one example that belongs and one that does not.
- **Separate labels from rules.** Business and system vocabulary is held as aliases pointed at categories. Aliases carry no retention period. Retired category names are kept as aliases with the date of retirement.
- **Record the classification method for each category.** Container, ingest rule, user declaration, or validated inference, chosen for the operating context and supported by evidence. Record who applies each part of the model.
- **Set granularity by the applier.** Carry only the distinctions the chosen method, or a person with sufficient context, can apply reliably.
- **Provide a retirement path.** State how a category is proposed for removal, who approves it, and what evidence is required.
- **Measure agreement.** Measure classification agreement on a stated cadence using the test in Appendix G.

## 4. Jurisdictional Variation Register

One schedule holds a global default and a bounded set of modeled variations. Before analysis, write down the jurisdiction determination rules for each group of categories and the deviation criteria: the changes in trigger, period, disposition action, or other applicable requirement that require separate treatment. Materiality organizes the model; it is not a basis for disregarding a binding requirement.

Figure H.2. Variation register fields.

| Field | What it records |
|---|---|
| **Variation and base rule** | Stable variation and base-rule identifiers, their respective versions and effective periods, and retained superseded records |
| **Citation** | The requirement supporting the variation |
| **Jurisdictions covered** | Every jurisdiction the variation applies to, since one variation may serve several |
| **What differs** | The trigger, period, disposition action, or other element that differs from the default |
| **Owner** | The person accountable for keeping the variation current |
| **Approval** | The decision reference, rationale and retained local analysis, approving authority, actual decision date, and relevant effective period |

| | |
|---|---|
| Requirement confirmation | When the local requirement was confirmed, by whom and against what source; implementation verification remains separately linked through Appendix F |
| **Next review date** | When it will next be reviewed |

Record residency and hosting against repositories in the register (Appendix F), not against categories. Retain the local analysis even where the global default satisfies a local requirement. Retire superseded country schedules explicitly, with sunset dates.

## 5. Control Evidence Category

The schedule holds a control-evidence category for records supporting examination of approved treatment, implementation, operation, measurement, testing, and follow-through: configuration exports, execution and disposition records, certificates, attestations, test records, measurement inputs, and third-party evidence. Each artifact supports conclusions determined by its contents, scope, and period; not every artifact demonstrates successful operation. Assign an approved retention value, any required trigger, scope, action, and documented rationale. Life of the control alone is not an operable rule, although a defined retirement event may be a trigger within a complete approved treatment. Evidence does not automatically inherit the treatment of disposed information. Preserve separately assessed business-record and documentation purposes and usable references. Appendix G holds the evidence register and Appendix C the measurement note.

## 6. System Documentation for AI Uses

Instructions, configuration, version history, and evaluation results for an AI use are governed as system documentation. Assign an approved retention value and any required trigger, scope, action, and documented basis using the eight rule elements and Chapter 18's decision record. State justified not-applicable elements explicitly; an unspecified duration is not permanent retention. Where permanence is justified, retain its basis, custody, preservation, and review arrangements. The rationale should address the system's life and the period during which retained outputs and decisions need explanation, including availability after updates or retirement, rather than automatically inherit one output's period. Chapter 24 describes the scope; the same discipline applies to classifier documentation under Chapter 23.

- **Keep versions and uses related.** Record which uses and versions the documentation describes, so that the version in effect on a given date can be retrieved.
- **No blanket AI period.** AI-related information takes its treatment from the use and the record involved, as Figure 24.1 describes. Business records produced with AI assistance follow their own categories.
- **Derivatives are assessed, not assumed.** Assess purpose, unique content, and source populations before selecting treatment. Apply source-linked convenience treatment only where justified; a separately governed derivative needs its own approved basis. Record

multiple-source relationships and the action required when a source changes or is disposed of. Retrieval or index removal is not, by itself, destruction.

• **Preserve distinct purposes.** Where business records, control evidence, and system documentation are linked or reused, each keeps its own approved treatment.

## About the Author

Jason C. Stearns, FAI, IGP, CIPM, CIPP/US, CRM, has more than twenty years of experience in information governance, privacy, and records management. As Chief Success Officer at LexShift, he works at the intersection of governance practice and technology, helping organizations put their information requirements into operation.

His experience includes serving as Director of Information Governance at BlackRock and holding information governance and records management roles at Citadel, UBS, and New York Life Insurance Company. That work has spanned global governance standards, program strategy, project oversight, and the practical challenges of managing information in complex organizations. His subsequent work in consulting, professional services, and legal technology brings that operating experience to organizations developing and strengthening their own programs.

Jason is a Fellow of ARMA International and a former president of its Board of Directors. He has also chaired the Information Governance Professional Board of Directors and served as president of the ARMA Metro NYC Chapter. Through professional education and industry service, he helps practitioners connect governance frameworks with the decisions and responsibilities of everyday work.

He holds a master's degree in Legal Studies with a focus on Cybersecurity and Information Privacy Compliance from Drexel University's Thomas R. Kline School of Law, an MS in Applied Information Management from the University of Oregon, and a BA from Binghamton University. His professional credentials include IGP, CRM, CIPM, and CIPP/US.

Outside his professional life, Jason loves yoga and travel. He maintains a daily yoga practice and enjoys exploring new places, including through yoga adventure retreats.