

THOUGHT LEADERSHIP | ASSET MANAGEMENT CONTENT LIBRARY AND SERVICE

Running the lifecycle coverage program

Operational paper — read once you understand the layers

The reports, the workflow tools, and the maturity progression that turn a coverage framework into a sustainable program.

ABOUT THIS PAPER

From framework to program

The hub paper in this series makes the case that coverage, not accuracy, is the binding constraint. The foundation paper defines what "covered" actually means—the phase semantics, the layered data model, and the four measurement decisions every program has to make. The track papers (SAM and HAM) cover the mechanisms for closing gaps.

This paper sits at the end of the canonical reading order. It assumes the reader has internalized the framework, knows where the four tiers come from, and has worked through at least one track paper. The job here is operational: which reports the work happens on, which workflow tools wrap the activities into something repeatable, and how a program matures from baseline to steady-state.

SECTION 1

The reports where coverage is measured

Two reports anchor the operational rhythm. The Software Lifecycle Report (SAM Pro) and the Hardware Model Lifecycles report (HAM Pro) are where coverage is measured, gaps are identified, and the effects of every other capability—approximation, calculation, custom records, publisher add-ons, parent-child inheritance—become visible.

Software Lifecycle Report (SAM Pro)

The Software Lifecycle Report is generated by a scheduled job ("SAM – Generate Software Lifecycle Report") and consolidates lifecycle data from every source the platform knows about: the Content Library (Tier 1 publisher dates and Tier 2 approximations with codes), Calculated Lifecycle (Tier 3), and Custom Lifecycle (Tier 4). It contains records from normalized Software Discovery Models that have at least one Software Installation associated—regardless of whether dates exist or not, which is essential for measuring coverage honestly.

Each product is presented as a single wide row. General Availability, End of Support, End of Extended Support, and End of Life appear as columns side by side. The Current Phase and Upcoming Phase are derived from those dates, giving the report a clean "where is this product today, where does it go next" answer. The Lifecycle Code columns sit alongside the date columns; pairing them through dot-walk gives auditors and analysts a single view of every date alongside its provenance.

Hardware Model Lifecycles report (HAM Pro)

HAM Pro's equivalent is the Hardware Model Lifecycles report, sourced from the `cmdb_hardware_model_lifecycle` table. It lists every hardware model the organization owns—meaning every model with at least one asset associated—along with the lifecycle data the platform knows about. As with the software side, the report is the artifact where coverage is measured.

One structural difference matters when designing reports or coverage queries. SAM Pro presents all phases for a product in a single wide row; HAM Pro takes the opposite shape: each phase is its own record. A single hardware model produces one record per phase (GA, End of Sale, EOS, EOES, EOL), and a boolean Current Phase flag marks which row represents the model's state today.

Why the layout matters

Coverage measurements on hardware need to aggregate across phase records, not look at a single row. A query that asks "does this model have an EOL date?" looks for the EOL phase record specifically; a query asking "what phase is this model in now?" filters on Current Phase = true. The same cross-EOX priority logic from the Foundation paper applies—it just operates over rows rather than columns.

Dashboards and visibility

The Lifecycle Management Dashboard pulls signals from both reports into a single shared view that finance, security, and IT teams can reference. The Hardware Asset Workspace also surfaces the same data through the Model Lifecycle Overview card, breaking down the estate by phase at a glance and drilling into the underlying list. The reports do the measuring; the dashboards make the measurements visible to the people who don't open the report itself.

SECTION 2

The workflow tools that make it repeatable

Generating the report is the start, not the finish. Two capabilities wrap the report into a repeatable workflow.

Guided Experience for Lifecycle Reporting

Most organizations approach lifecycle work the same way: someone runs the report, finds gaps, files content requests, creates a few approximations, generates the report again, and so on. The work is real, but it is easy to lose track of where in the process the team is. Guided Experience for Lifecycle Reporting turns that workflow into a structured, step-by-step experience that walks users through every action in an optimized order. It aggregates the available tools and capabilities, creates Success Goals and tasks with assignment to the right owners, tracks progress across steps while enforcing dependencies, and makes the whole process repeatable for the next cycle.

The point is not just to make the work easier. It is to make the work consistent—so that the same coverage discipline gets applied the same way every quarter, and so that ad-hoc actions (a new acquisition, a major vendor policy change) plug into the same framework without restarting from scratch.

Asset Audit Response

For organizations subject to formal regulatory examinations, Asset Audit Response—part of ITAM for Financial Services—connects the GRC and ITAM teams through a shared workspace. It supports evidence requests tied to specific control objectives, point-in-time data with historical comparisons, gap identification with remediation rules, and approval workflows for evidence submission. Lifecycle data is one of the most frequently requested categories in audit response, and a workspace built around the audit cycle closes the loop between "we have the data" and "the data has been formally submitted and approved."

SECTION 3

Crawl, walk, run: the maturity progression

A coverage program matures in stages, each building on the work that came before. The progression below is the same one mature SAM and HAM customers tend to converge on. Each phase calls on capabilities covered in detail elsewhere in this series; this section sequences them rather than re-explaining them.

Crawl	Walk	Run
• Engage Content Service. • Confirm normalization rate. • Scope: Licensable + Not Licensable. • Generate baseline reports. • Submit prioritized content requests.	• Apply cross-EOX priority. • Count codes as coverage. • Use install-weighted measurement. • Activate Calculated Lifecycle. • Set baseline coverage targets.	• Add Custom Lifecycle for contracts. • Apply Publisher Add-Ons (SAM). • Tune Parent-Child Relationships (SAM). • Adopt Guided Experience. • Run coverage as a quarterly metric.

Crawl — establish the baseline

Engage Content Service before touching anything else (Foundation paper, Section 3). Confirm normalization rate is acceptable—if it is low, the rest of the program is capped by it. Scope the report to Licensable and Not Licensable. Generate the Software Lifecycle Report and the Hardware Model Lifecycles report; include the Lifecycle Code columns. Identify the records with no date and no code, and submit prioritized content requests through KB0790305. The crawl phase is about getting the baseline straight, not about closing every gap.

Walk — apply the measurement framework

Apply the four measurement decisions from the Foundation paper: install-weighted measurement, scope to Licensable and Not Licensable, cross-EOX priority across the three end-of phases, and count codes as coverage. Activate Calculated Lifecycle—Software Lifecycle Averages in SAM, Calculated Lifecycle Templates in HAM (track beginner papers cover the mechanics). Regenerate the reports. The output is the program's baseline coverage number, the figure against which all future improvements are measured.

Run — operationalize

Add Custom Lifecycle records for negotiated terms and internal policy (SAM and HAM intermediate papers). On the SAM side, configure Publisher Add-On Lifecycles for products with paid extensions, and tune Parent-Child Relationships to propagate dates through the publisher hierarchy (SAM advanced paper). Adopt the Guided Experience to make the entire workflow repeatable across cycles with explicit owners and dependencies. Treat coverage as a quarterly metric, not a one-time exercise—the vendor landscape shifts, the estate changes, and the program has to keep pace.

SECTION 4

What changes when coverage becomes the metric

Risk surfaces sooner

When the question shifts from "is the date correct?" to "do we have a usable signal?", more of the estate becomes visible. Approximated end-of-life flags catch products that would otherwise sit in a blank-date blind spot until an examiner found them first.

Audit posture becomes defensible

Examiners do not require perfection. They require evidence of process. A coverage program with explicit metrics, labeled approximations, custom overrides for negotiated terms, install-weighted measurement, and a documented feedback loop is materially easier to defend than a spreadsheet of dates that does not explain where they came from.

Decisions stop waiting on data

Renewal conversations, refresh planning, and consolidation analysis all reference the same lifecycle field. When that field carries confidence-tagged values across the estate, those conversations move from "we don't have the data" to "here are the tradeoffs."

The content pipeline gets smarter

Every gap detected at a customer site feeds the content acceleration backlog. Over time, the gaps with the highest install counts get closed first, and the long tail gets labeled approximation that holds up under audit. The content service stops being reactive and starts being targeted.

The discipline is what turns capabilities into a program

The capabilities to do all of this are generally available today. The reports exist. The workflow tools exist. The four tiers, the four measurement decisions, the crawl-walk-run sequence, the audit posture—all of it operates on what is in the platform now. Across all of it, accurate publisher data is always preferred; the layered capabilities exist for the cases where accurate data isn't yet available. What turns capability into program is the discipline of running the cycle consistently. That is what this series, taken together, is for.

ESSENTIAL REFERENCES

Software lifecycle reports guided playbook — <https://www.servicenow.com/docs/r/it-asset-management/software-asset-management/guidedplaybook-sw-lifecycle-reports.html>

Lifecycle Management Dashboard — <https://www.servicenow.com/docs/r/it-asset-management/software-asset-management/lifecycle-management-dashboard.html>

KB0790305 – Submitting Content Requests — https://support.servicenow.com/kb?id=kb_article_view&sysparm_article=KB0790305

