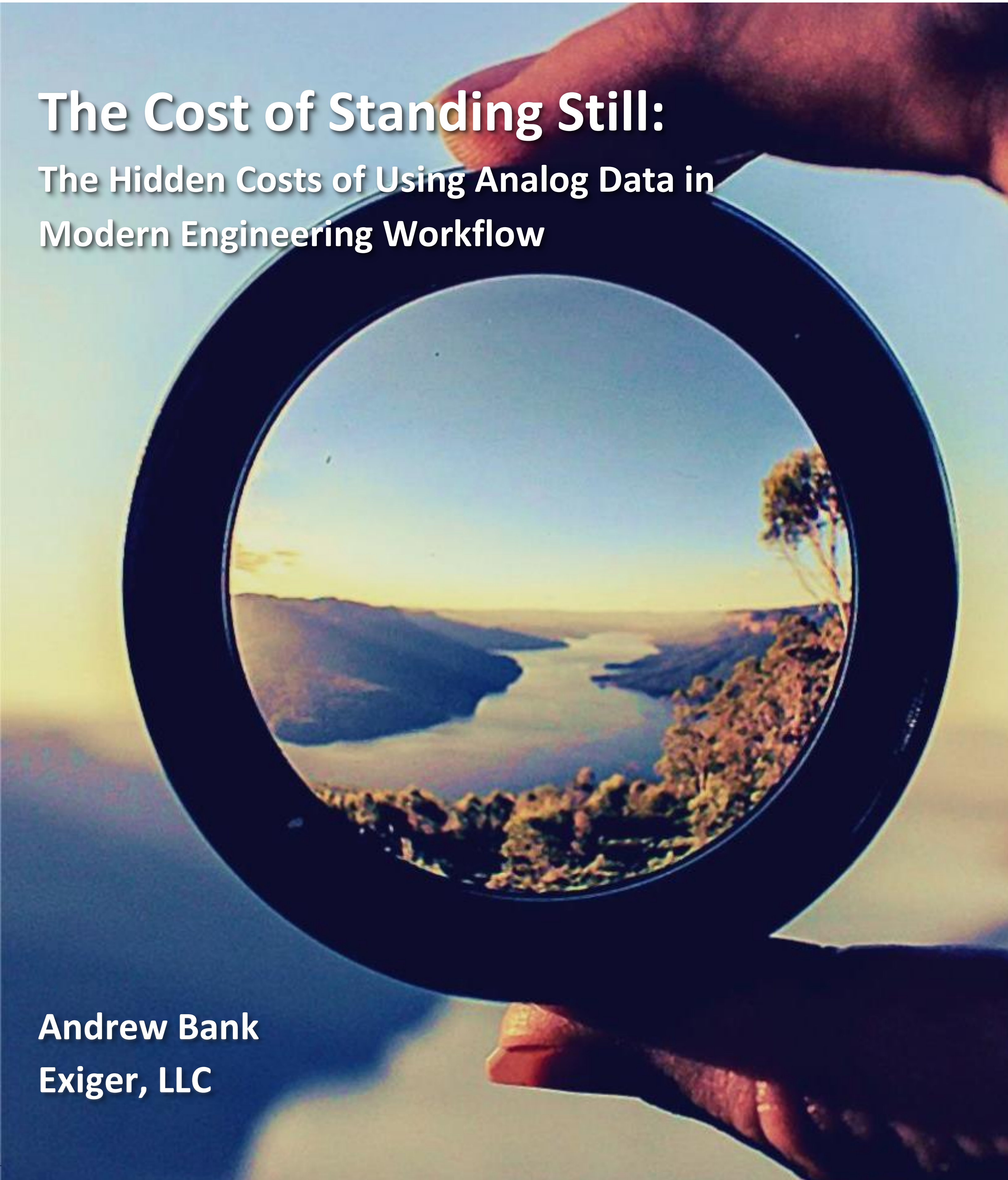
EXIGER

SWISS™

The Cost of Standing Still:

The Hidden Costs of Using Analog Data in Modern Engineering Workflow



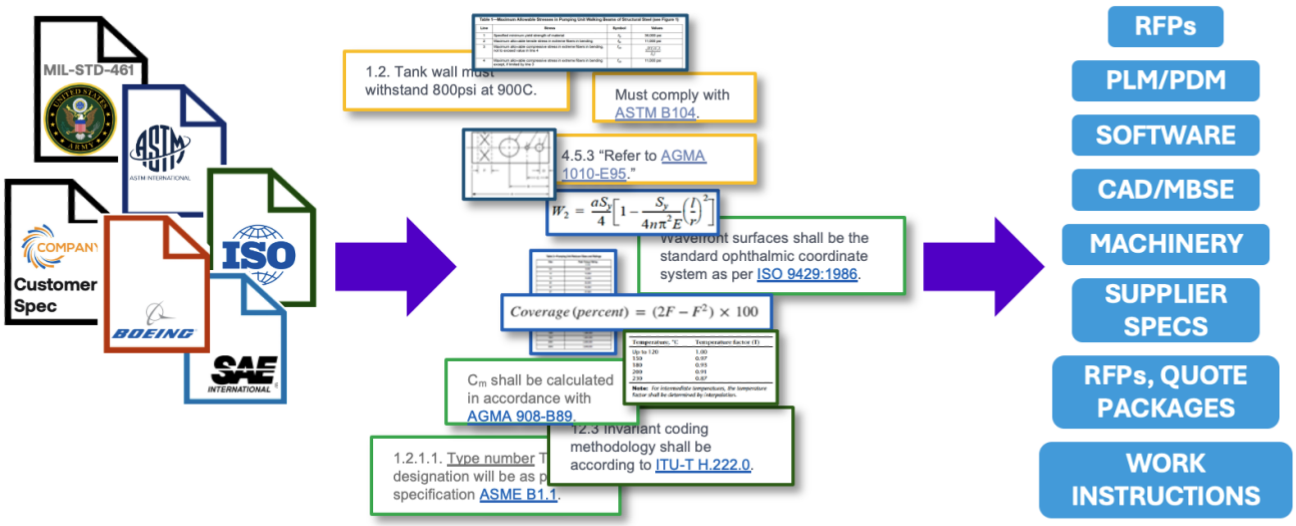
Andrew Bank

Exiger, LLC

Introduction

Most organizations pursuing Model-Based Enterprise (MBE) assume their biggest challenges are tool adoption and culture change. Those matter, but in practice, a quieter problem does the same if not more damage: analog data embedded throughout engineering workflow. “Dead text” PDFs, scanned specs, copied requirements, and manually re-keyed tables introduce friction (cost, time, and errors) at every stage of the digital thread – engineering, quality, manufacturing, and supply chain.

If you lead engineering or manufacturing at an OEM, you likely have the right building blocks: PLM for change control, CAD/MBSE for authoritative design intent, QMS for quality, and supplier portals to keep production moving. Yet when the schedule tightens, or a change notice lands late on a Friday, teams still crack open these static specs and standards, work instructions, customer requirements, and scanned drawings stored as PDF files. That habit feels harmless, but it’s not.

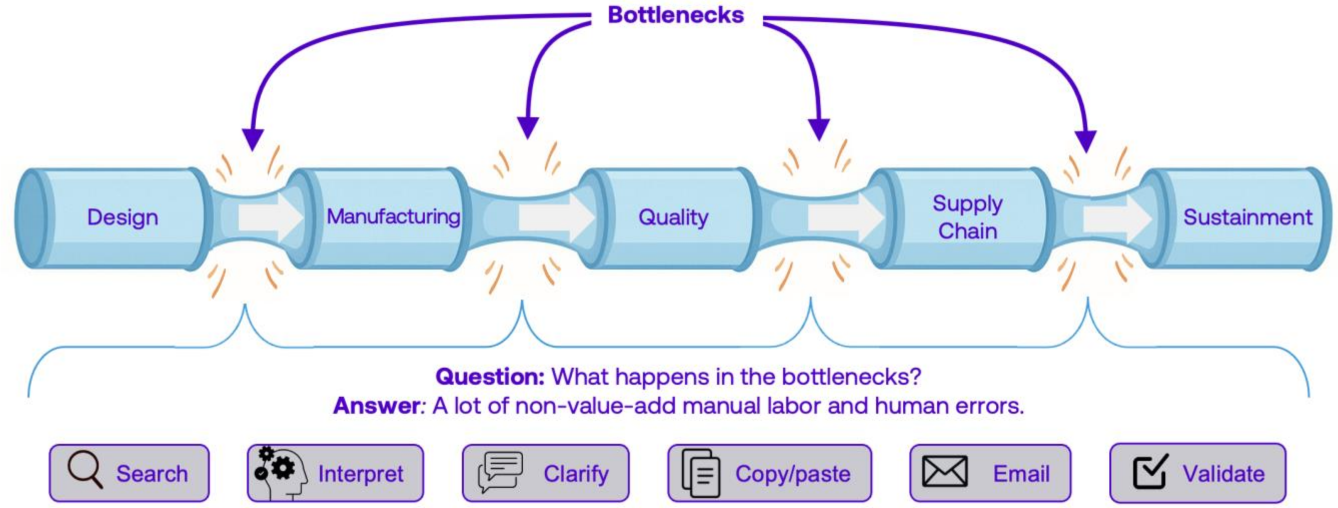


*Thousands of data points – like requirements, references, tables, equations, images, material and process specs, part numbers, suppliers, tolerances, and more – are **extracted manually** from internal and external PDF documents and used in dozens of downstream derivatives.*

The reason is simple: analog documents don’t “flow.” They force people to become the integration layer – hunting for clauses, re-typing values into downstream tools, copying and pasting requirements into work instructions, and reconciling versions across functions and suppliers. On paper, progress is apparent: drawings released, parts received, audits passed. But between those visible milestones sits the invisible effort – manual extraction, manual interpretation, manual change propagation – where time, cost, quality, and risk quietly heat to a boil.

These costs rarely appear as a single line item. Instead, they accumulate daily as lost engineering hours, extended change-to-release cycles, supplier delays, rework, procurement emergencies, military (non)readiness, and compliance exposure. What can look like minor behaviors – copy/paste, document chasing, and interpretation by tribal knowledge – often compounds into months of schedule slip and avoidable operational risk. And because the friction is spread across functions, no single team “owns” the full problem – so the full cost is not tabulated, and it persists even in organizations with modern PLM and CAD investments.

In other words, the cost of using analog data in your engineering workflow is a slow “death by a thousand cuts” leak from your bottom line.



Using analog data, like PDF, in modern engineering workflow requires too much manual labor and causes slowdowns (bottlenecks) at multiple stages in product engineering lifecycle. The daily tax adds up.

This white paper surfaces those hidden costs in plain English and shows a pragmatic path forward: convert analog definition data into true Model-Based Product Definition (MBPD) – machine-readable requirements, product attributes, constraints, and standards – so downstream teams and systems can act on it. Solve it once upstream, and the payoff compounds end-to-end: faster approvals, cleaner change propagation, fewer supplier surprises, fewer quality escapes, and a digital thread that operates the way it was always supposed to.

Most importantly, you’ll see how organizations are beginning to do this using SWISS: domain-specific AI, deep industry ontologies, and proprietary parts/materials/process data that transform analog documents and drawings into comprehensive, machine-readable models that integrate with enterprise applications (e.g., PLM and supply-chain systems) and flow cleanly from design through manufacturing, quality, supply chain, and sustainment. SWISS closes the last mile of the digital thread so you capture the full operational and financial benefit of the investments you’ve already made.

The [Defense Standardization Program Office’s 2026 Digital Standards Strategy (DSS)] makes clear where defense is headed. Standards are moving out of static documents and into machine-readable and machine-interpretable forms that can be used in digital tools and model-based workflows. For companies that want to support modern defense programs, requirements can no longer stay trapped in PDFs. SWISS provides a practical way to make that shift.

From Engineering-Ready to Procurement-Ready (Why This Matters Beyond Engineering)

Analog data doesn't just slow down engineers – it slows down everyone who depends on engineering information downstream. When requirements live in PDFs and Word files, people have to search, interpret, re-key, and reconcile. Then the same work happens again when the job moves into procurement, suppliers, quality, and sustainment – often discovering late-breaking issues like restricted materials, scarce critical minerals, long lead-time processes, obsolete parts, or supplier constraints only after release.

The practical goal is simple: make engineering *intent* usable downstream without starting over. When key requirements – materials, processes, test methods, standards references, and constraints – are captured in a structured way, downstream teams can use them immediately. Procurement can see constraints before issuing fully model-based packages; compliance can flag regulated substances earlier; and supply chain teams can assess alternates before a part becomes a bottleneck.

This is how organizations shift from reactive cleanup to proactive prevention – using upstream definition to avoid downstream schedule churn.

The DSS reinforces this point with a practical requirement: accessible repositories of interface specifications and supporting documentation in machine-readable form so third parties can integrate without OEM coordination. That means engineering intent has to be usable after release, not just readable. SWISS helps turn that handoff from another PDF package into structured requirement data.

A Defense Reality Check: Weeks to Minutes

We'll lead with a defense example – a Technical Data Package (TDP) and 339 review – because it's where the consequences are sharpest and the lessons generalize cleanly to aerospace, automotive, shipbuilding, and heavy equipment.

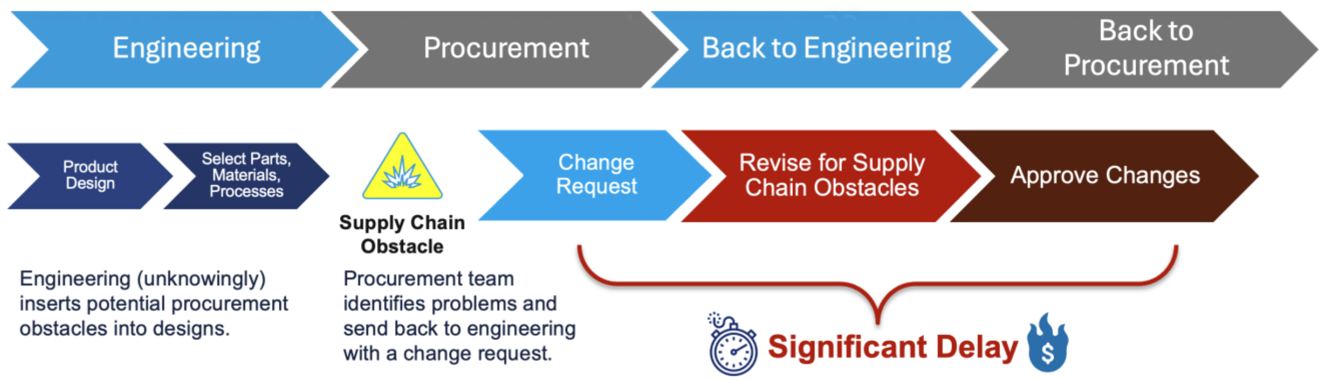
Army engineers routinely review TDPs before procurement to find out-of-date standards, regulated substances, critical mineral vulnerabilities, long-lead-time processes (e.g., castings/forgings), and other risk factors that can slow or stop production. Historically, this is brute force work: open dozens of drawings and specs, scan notes, chase references, and compile replacement standards – often up to 90 days per package.

When Army sustainment engineers were struggling with a persistent months-long delay in fielding parts for a wiring harness in the Bradley Fighting Vehicle, analysts faced 85 PDF drawings that would typically consume 4–6 weeks of engineering time. Using SWISS tools that transform drawings into digital models, those documents were parsed into structured data in ~2 minutes per drawing (under 3 hours of machine time total), automatically flagging out-of-date standards, long-lead-time castings, obsolete components, and restricted substances. *Identifying those castings is credited for pinpointing the cause of a 180-day sustainment delay for the Bradley Fighting Vehicle*. This wasn't a marginal improvement; it was a different paradigm.

The same story illustrates the larger supply chain risks: *when engineering challenges are overlooked and spill downstream into procurement and supply chain, they cost more time and money to resolve and have a much greater impact on readiness than if they were discovered earlier upstream*. Missing a casting requirement, a critical mineral shortage, a regulated substance, or an unfit supplier late in the process can easily add months, not weeks, to a delivery schedule – and that can have dire consequences to battlefield readiness.

In practice, these late discoveries often show up as urgent re-sourcing, substitute approvals, supplier re-qualification, and schedule churn – work that is avoidable when the product definition carries the full requirements context throughout the digital thread.

Keep this example in mind as we unpack the hidden cost buckets below.



When upstream engineering, compliance, and acquisition challenges are overlooked, they spill downstream and cause bigger and more costly problems in procurement and supply chain.

How to read the scores and tags in this article

For each of the "Hidden Costs" below, you'll see a pain scale (how hard it hits **Speed-to-Market**, **Cost**, and **Compliance Risk**) and **KPI tags** – the outcomes most affected if you fix it, followed by a brief solution summary.

Pain scale legend: 1 = low impact, 2 = moderate, 3 = high, 4 = very high.

KPI tags:

- **Engineering Hours Saved** (time teams get back)
- **Change-to-Release** (time from "we must change this" to official release)
- **Supplier Latency** (time from your release to suppliers producing to it)

- **Rework Rate** (time spent fixing mistakes/NCRs tied to requirement issues) (NCR – Nonconformance Report: the formal record opened when a part, process, or document fails a requirement.)

Hidden Cost #1: Finding, extracting, and re-keying requirements (is so 1990)

What it means
Finding the *right* requirement or test method still depends on people reading PDFs, finding minute details, clarifying ambiguity with experts, and taking appropriate action. Extracting requirements from complex engineering documents and drawings is difficult and time-consuming for humans; parsing multiple requirements from one clause and extracting them into subject, attribute, and value is even harder (see illustration below). So people still copy/paste or re-key critical operational data – requirements, references, test methods, parts, materials, processes – into PLM, MES, QMS, ERP, internal specs, work instructions, inspection plans, and supplier packets. It's time-consuming work that invites small but costly slips and burns down morale by forcing digital engineers to spend their days doing clerical rekeying.

Pain Scale
Speed: 3 | **Cost**: 3 | **Compliance**: 2

KPI tags
Primary: Engineering Hours Saved, Rework
Secondary: Change-to-Release

Example
Ahead of a formal change, a planner keys 200 inspection characteristics from PDF notes into the inspection system. One ± sign is missed. The plan must be rebuilt after first article, and the change slips a week as review cycles repeat.

What it costs
Hours of manual discovery for every package; large chunks of skilled time diverted to data entry; repeated effort across design, manufacturing, quality, procurement, and suppliers; bottlenecks around big releases; transcription errors that ripple into rework and corrective actions; higher odds of missing a buried reference or using an outdated clause; slower decisions because people are never fully sure they found "the right thing"; frustrated engineers who feel more like clerks than problem-solvers.

How to fix it
The fix here is not "better searching." It's eliminating the need to copy and re-key requirements in the first place. SWISS takes the requirements that are currently trapped in PDFs and drawing notes and converts them into structured, machine-readable requirement objects (for example: *what the requirement applies to, what attribute it controls, what the value and tolerance are, what test method proves it, and what revision it came from*). That turns re-keying into review: instead of a planner or engineer manually typing the same data into PLM, QMS, MES, or inspection plans, SWISS can populate those downstream systems with the same approved requirement data –

delivered by reference where appropriate – so the organization stops creating a "shadow copy" of the truth in spreadsheets and email attachments. In practice, this means engineers spend their time validating and approving extracted requirements, while PLM and downstream workflows receive consistent, up-to-date requirement data without repeated copy/paste cycles.

The eyelets shall be brass, with a coated nylon Desert Sand finish, size AA.

Three different requirements, three different attributes – a challenging requirement to parse, even for a human.

SWISS Semantic AI

Subject	Attribute	Attribute Definition	Value
eyelets	Material	Material used to make the eyelet	Brass
eyelets	Finish	Finish of the eyelet	Coated nylon Desert Sand
eyelets	Size	Size of the eyelet	AA

SWISS extracts and classifies by Subject, Attribute, and Value, and creates reusable data objects.

Extracting requirements from complex engineering documents and drawings is difficult and time-consuming for humans*.* *Parsing multiple requirements from one clause and extracting them into subject, attribute, and value is even harder.*

Hidden Cost #2: Threading requirements into work instructions is manual

What it means

Hidden Cost #1 is about *finding* the right requirement. This one is about something more painful: **deriving an entire, step-by-step manufacturing or test process when the requirements are scattered across multiple documents.** In real programs, a single shop-floor instruction often depends on a drawing note, an internal spec, one or more customer specs, and at least one external standard (ASTM/SAE/ISO). Each of those documents contains "breadcrumbs" that point to other requirements – setup conditions, acceptance criteria, sampling, reporting, retest rules, environmental conditions, and exceptions. When all of that lives as analog PDF text, the only way to build a complete process is for an expert to manually chase references, gather every relevant clause, and assemble them into a derivative: a router, a work instruction, an inspection plan, or a test procedure. It's the kind of work that *looks* like "documentation" from the outside, but inside the organization it's actually careful, high-stakes engineering interpretation, done painstakingly by a seasoned expert using manual copy/paste.

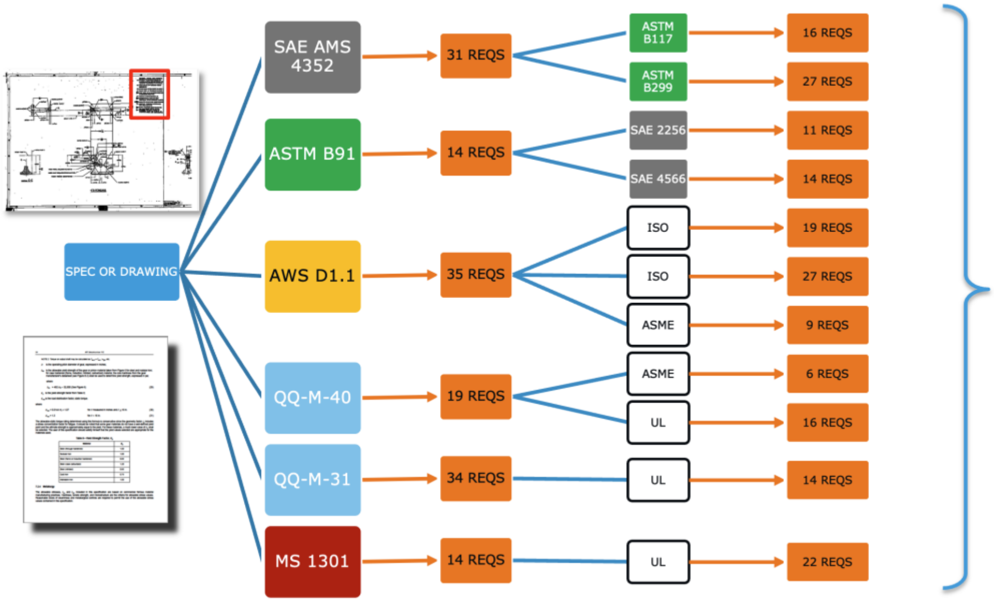
Pain Scale

Speed: 4 | **Cost:** 4 | **Compliance:** 3

KPI tags

Primary: Engineering Hours Saved

Secondary: Change-to-Release, Rework, Supplier Latency



- 17 different documents
- 892 pages
- 328 requirements across 46 different subjects

- Manual
- Painstaking
- Time-consuming
- Error-prone
- Requires deep expertise*

*Most experts are older and retiring.

Originating from just two specifications, there are references to17 different documents consisting of 892 pages and 328 requirements across 46 different subjects. Today, a human must gather all these requirements at multiple stages in the product lifecycle including manufacturing, test and quality control, and sustainment.

Example

A program needs a bend test procedure for a material/process qualification. The drawing or internal spec references a bend test requirement and points the engineer to external standards, such as ASTM E290, plus additional reporting and evaluation requirements tucked into other sections. The manufacturing or test engineer now must: locate the exact test type that applies, extract specimen preparation steps, capture tooling and setup requirements, define test parameters, identify acceptance criteria, and then pull the required reporting language. That process usually involves a dozen PDFs open at once, “Ctrl+F” searches that return dozens of partial hits, and lots of judgment calls about what is “in scope” for this specific part and process.

This isn’t hypothetical complexity. In a 2025 presentation from the Digital Standards Alliance, a team documented what it took to model a practical bend test using data from multiple standards: just one paragraph of an SAE specification plus 12 pages of an ASTM specification required an estimated 40 hours of manual effort and 13 hours of LLM assistance to create the structured deliverables. Again, that was for just a single paragraph in a single spec. Now multiply that by thousands of parts, processes, and suppliers, then do it again every time a governing standard changes.

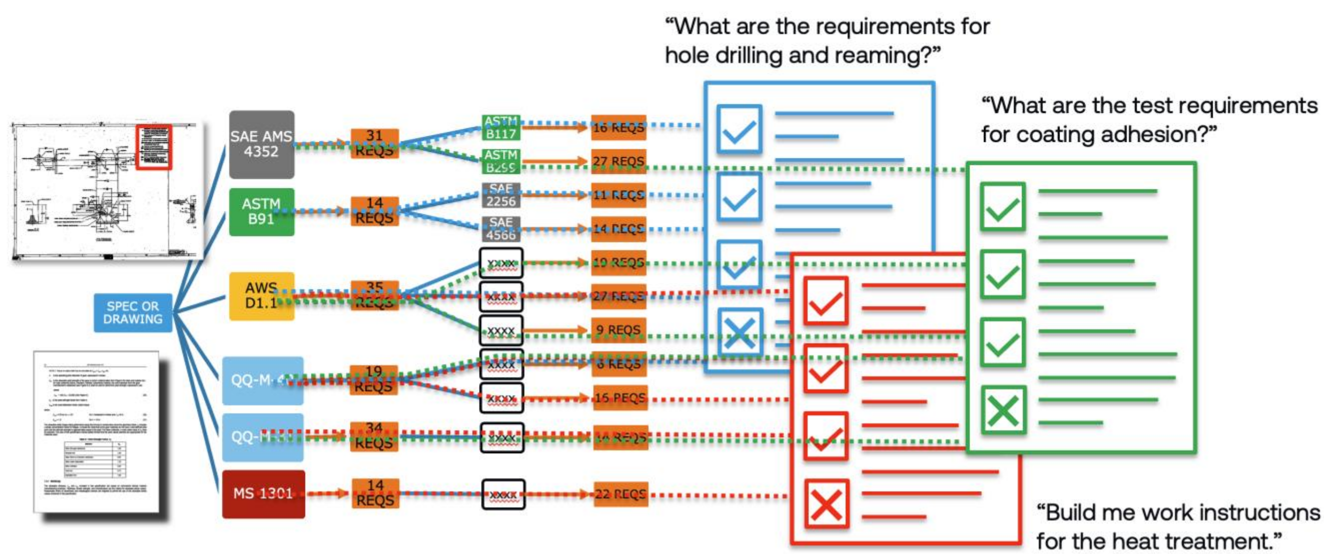
What it costs

This is where the digital thread quietly bleeds time and trust. The immediate cost is obvious: highly paid experts spending days doing clerical assembly work: hunting, copying, pasting, formatting, and reconciling. But the bigger cost is what that manual process creates downstream: inconsistent interpretations, missing clauses, wrong revision usage, and “almost compliant” instructions that only fail when parts are already built, tests are already run, or an audit is already underway. Because the work is so tedious, teams naturally try to speed it up, reusing old procedures, copying chunks from prior work, or relying on tribal memory. That’s exactly how static-data "tech debt" spreads: the derivative becomes the new source, even though it's disconnected from the authoritative standard. The result is slower releases, more rework, and more quality escapes when the instruction doesn't fully reflect the real governing requirements.

How to fix it

This is where *threaded requirements* changes the physics. Once SWISS transforms drawings, specs, and standards into digital models, it can treat requirements as connected objects instead of isolated paragraphs. SWISS can follow the same reference trail a human follows – drawing note → internal spec → external standard → related clauses for setup, evaluation, and reporting – but do it systematically, completely, and repeatably. Instead of a person hand-assembling a bend test procedure line-by-line, SWISS can derive a structured, end-to-end set of steps and parameters for the bend test, with each step carrying provenance back to the exact clause and revision it came from.

The output is not just a "summary" or a list of requirements. SWISS produces machine-readable, model-based instructions that can be delivered by reference into enterprise tools like PLM and other downstream quality/manufacturing workflows. Because those instructions are structured as data – not prose – *machines can interpret them consistently* and, where appropriate, *execute them through MES and test/inspection systems* instead of relying on humans to translate paragraphs into actions. Just as importantly, when an upstream standard changes, SWISS can re-thread the requirement network, highlight what changed, and regenerate the affected process definition – so updates become a controlled review task instead of another multi-day copy/paste project. That is the practical path from "analog documents" to truly digital manufacturing: not replacing experts but removing the manual stitching work that keeps experts trapped in PDFs.



Task-specific requirements are gathered automatically, and delivered as machine-readable, machine-interpretable, and machine-executable instructions.

Hidden Cost #3: Identifying risk factors is manual and prone to error

What it means
Even when teams know the "main" requirement, the risk factors that derail programs are often buried in analog references: regulated materials, critical mineral vulnerabilities, obsolete/withdrawn specs, obsolete parts or high-risk suppliers, and special or long lead time processes (e.g., castings/forgings). Identifying those risk factors still depends on people reading PDFs, finding minute details, clarifying ambiguity with experts, and taking appropriate action. Each team repeats the same search, and results vary depending on who did the digging and which version they opened.

Pain Scale
Speed: 3 | **Cost**: 4 | **Compliance**: 4

KPI tags
Primary: Supplier Latency
Secondary: Rework, Engineering Hours Saved

Example
A U.S. defense program sends a technical data package (TDP) for review. Analysts open dozens of drawings and standards to find a withdrawn spec related to a coating and a casting requirement that triggers a long lead-time alert. Quality repeats the hunt two weeks later and finds that several of the parts contain cadmium, and a supplier runs a third pass after release – each group starting from scratch without the benefit of previous work.

What it costs
The real cost often shows up later when procurement discovers that a selected part is obsolete, a preferred material is restricted, or a special process drives a long lead time no one accounted for. At that point, teams scramble for alternates under schedule pressure, often re-validating requirements manually and reworking documentation. The tangible impact is repeated effort across functions and suppliers, higher odds of missing a buried reference or using an outdated clause, slower decisions because people are never fully sure they found "the right thing," and preventable supply chain emergencies after release.

How to fix it
This problem isn't solved by making requirements easier to read. It's solved by making risk factors impossible to miss. SWISS doesn't just extract requirements; it automatically identifies and tags risk markers that often derail programs: regulated substances, critical minerals, long-lead processes (castings/forgings), and obsolete or withdrawn references. Instead of asking an engineer to "notice" these risks while reading hundreds of pages, SWISS uses its domain intelligence (parts, materials, and manufacturing process knowledge) to flag the risk explicitly, attach it to the relevant part/material/process callout, and carry it forward through the digital thread. Those risk flags can then flow into the systems that act on them – PLM and sourcing workflows first, then suppliers – so procurement and supply chain can respond early (alternative materials, alternate sources, long-lead mitigation) rather than discovering the problem after

release. The practical outcome is that risk moves from a late-stage surprise to an early, trackable decision point – one that's visible to engineering, quality, and supply chain in the same place (for example, in Windchill or Teamcenter) instead of being rediscovered anew in every department.

Hidden Cost #4: Analog files are not machine readable; they break interoperability, and block automation

What it means
PDF specs and analog notes on drawings don't interoperate with modern systems or with each other, so data can't flow through the digital thread, and tools can't act on it. People (and their manual labor) become the integration layer, stitching information between PLM, QMS, MES, ERP, and supplier packets with screenshots and file attachments that get old and diverge from the authoritative source.

Pain Scale
Speed: 3 | **Cost**: 4 | **Compliance**: 2

KPI tags
Primary: Engineering Hours Saved
Secondary: Rework, Change-to-Release

Example
Quality needs the precise acceptance method referenced in a drawing note. The method is buried in an external standard. Without machine readable data, they chase a subject matter expert, paste text into QMS, and hope nothing changes before release.

What it costs
Because requirements live as prose – or as cherry-picked "should/shall/may" lines that software can't interpret – the automation you've already paid for (tolerance checks, auto-built inspection plans, compliance scanning) doesn't execute. Work progresses only by people, not systems: handoffs are slow and brittle, and teams create side files and spreadsheets that diverge from the source. Without machine-readable data there's no consistent, at-scale way to run checks or regenerate plans when something changes. Interpretations differ from person to person, decisions diverge, and overall throughput stalls at the human bottleneck.

The DSS defines digital standards as machine-readable and machine-interpretable content used in digital tools and processes. It also points to models, including Systems Modeling Language (SysML), as one example of machine-interpretable output. If a requirement exists only as prose in a PDF or drawing note, it cannot move cleanly through modern systems.

That is why analog files break interoperability. SWISS turns requirements, references, tables, and notes into structured data that can flow into PLM, QMS, MES, supplier workflows, and, where needed, model-based tools. The point is simple: make the requirement usable wherever it is needed.

How to fix it
Make requirements **machine readable and machine executable**. SWISS expresses each requirement as well-labeled fields (subject, attribute, value, units/tolerance) with provenance (source clause + revision). Those fields are delivered, via API, to the tools that need them. PLM remains the system of record; QMS and MES **consume the same requirement as data**, not screenshots. The result: systems can talk to each other, and automation – rule checks, plan generation, materials/regulatory screening – can finally run.

Hidden Cost #5: Change impact and change-to-release are impaired

What it means
Determining how a change impacts products and operations is highly manual: teams must track down every place that a changed clause appears, edit multiple artifacts, and route approvals across functions. While that work crawls, other teams either stall or keep moving on old information, setting up rework, scrap, recalls, and liability risk.

Pain Scale
Speed: 4 | **Cost**: 3 | **Compliance**: 3

KPI tags
Primary: Change-to-Release
Secondary: Engineering Hours Saved, Rework

Example
A standards body revises a test method. Engineering opens an ECR (Engineering Change Request), then spends days finding every drawing, instruction, and plan that cites it. After several review loops, the ECO (Engineering Change Order) releases – by then, a supplier readiness review has slipped, and teams redo work to match the new wording.

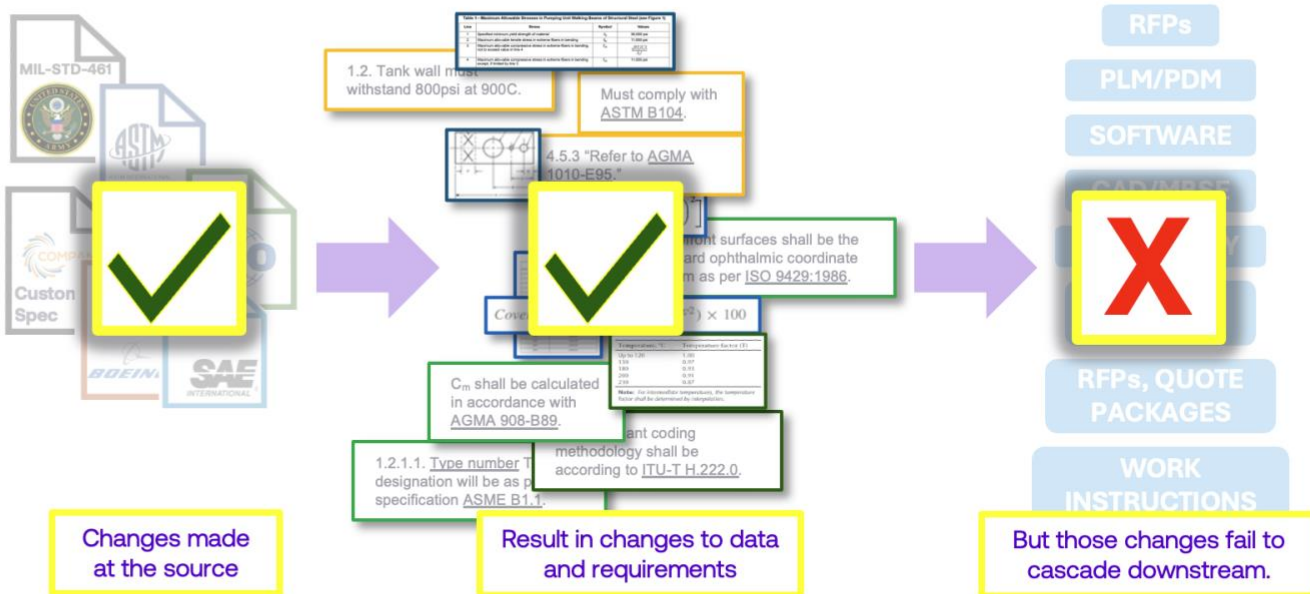
What it costs
The result is longer change-to-release (the time from deciding a change to officially releasing it), extra meetings and re-routes, and a higher chance that outdated wording leaks into production before the update is fully approved.

The DSS points to a practical Air Force example: MIL-STD-882E was modeled so teams could trace requirements and see downstream effects of change. That is the real value of moving beyond static documents. When a requirement changes, teams should not have to hunt through drawings, instructions, and plans to figure out what breaks.

SWISS supports that same end state. Because each requirement stays linked to where it is used, a change can be traced forward, reviewed once, and pushed through release in a controlled way.

How to fix it
See the impact radius and update once. SWISS maintains forward links from each requirement to where it is used. When something changes, it surfaces the affected artifacts, proposes updated

wording, and pushes the approved change into PLM for controlled release. The same updated requirement then flows to QMS/MES and supplier packets by reference, so propagation doesn't depend on email threads or manual edits.



***Today's status quo**: when the source data changes, the downstream data and derivatives do not.*

Hidden Cost #6: Copy/paste and re-key creates "tech debt": outdated specs, and even more analog data

What it means
Copying text from standards or drawings into internal documents creates **new** analog data that drifts from the source. Even when the source data changes, the downstream copies don't.

Pain Scale
Speed: 2 | **Cost**: 2 | **Compliance**: 4

KPI tags
Primary: Rework
Secondary: Change-to-Release, Supplier Latency

Example
A work instruction embeds a plating clause from an ISO standard. Months later the standard updates, but the instruction doesn't. An audit flags the mismatch, a corrective action opens, and supplier packets must be re-issued.

What it costs
Document versions diverge and multiply around the company and among suppliers. Next comes

document clean-ups and retraining; re-issues to suppliers; risk of building to outdated rules; potential licensing exposure from reproducing paid content. Over time, this "clone debt" erodes trust in what's truly current.

The DSS also notes that digital standards reduce copy/paste labor while still requiring proper licensing, rights, and security controls – another reason to manage requirements by reference instead of cloning text into disconnected documents.

How to fix it
Reference a master source, don't reproduce. With SWISS, each requirement is an addressable, versioned object with a link back to its source clause and revision. Internal documents and downstream systems **pull by reference** instead of copying text. When the source changes, the update cascades to PLM and out to every consumer automatically. One fix, everywhere it's used – no scavenger hunt.

Hidden Cost #7: Engineering-Ready to Procurement-Ready is slow and inconsistent

What it means
After you release a change, suppliers still must interpret it, assess the impact of changes, update tools and work instructions, and qualify before shipping to the new rule. Days turn into weeks and months on long-lead items like castings and forgings.

When engineering intent is trapped in static references, the real cost often shows up later – when procurement discovers that a selected part is obsolete, a preferred material is restricted, or a special process drives a long lead time no one accounted for. At that point, teams scramble for alternates under schedule pressure, often re-validating requirements manually and reworking documentation. This is how organizations become reactive: design choices that looked fine in a PDF become supply chain emergencies after release.

Pain Scale
Speed: 4 | **Cost**: 4 | **Compliance**: 3

KPI tags
Primary: Supplier Latency
Secondary: Rework, Change-to-Release

Example
A forging tolerance tightens by a small amount. The change releases internally, but the supplier needs two weeks to update and re-qualify. Parts built to the old tolerance require rework or scrap, and you pay to expedite replacements to keep the line moving.

What it costs
The tangible cost is schedule slip, scrap and rework, expediting fees, and strained relationships – plus the loss of buffer your program was counting on.

How to fix it
Send suppliers the **answer, not another PDF**. SWISS packages the updated requirement so suppliers can apply it directly, highlights what changed, and keeps the link to the source clause. Suppliers acknowledge adoption, and you can see who has implemented and who hasn't. SWISS also flags **long lead times and regulated material risks upstream**, so they're resolved before release, not after suppliers have started work, with acknowledgments you can see.

Hidden Cost #8: Rework, defect escape, and late-stage requirement churn

What it means
Ambiguous wording, stale references, or copy/paste errors lead to parts or tests that don't meet requirements, but they don't always show up immediately; they surface late during inspections, audits, and lead to scrap, rework, and repeat ECOs.

These late discoveries don't just create quality events – they often trigger procurement disruption: substitute parts, supplier changes, and additional qualification cycles when the original interpretation or source no longer holds. A single outdated standard callout or unclear requirement can quietly lock in the wrong source, then force a costly pivot once reality hits.

Pain Scale
Speed: 3 | **Cost**: 4 | **Compliance**: 3

KPI tags
Primary: Rework
Secondary: Engineering Hours Saved, Change-to-Release

Example
A note is copied without its tolerance band. Parts pass in-process checks but fail final inspection. An NCR is opened; MRB dispositions rework; an ECO clarifies the note; production pauses while the batch is corrected and paperwork clears.

What it costs
Scrap and rework; line stoppages and rescheduling; MRB and engineering time on investigations; repeat changes that consume the approval pipeline; frustrated teams and customers.

How to fix it
Remove room for interpretation. SWISS represents each requirement unambiguously – what subject it applies to, which attribute, the value and tolerance, and the linked test or process. The same wording shows up wherever it's needed (drawings, work instructions, inspection plans, supplier packets), and rule-based checks can run before release to catch inconsistencies. Fewer surprises at inspection means fewer NCRs and less churn.

Why These Costs Stay Hidden: The Business Case Blind Spot

The everyday pain of analog documents doesn't arrive as one big bill. It shows up as ten minutes to find a clause, twenty minutes to retype a table, a day waiting for an approval, a week while a supplier rewrites their work instructions, a few hours of audit prep every quarter, and a scrap heap of non-compliant parts. None of those line items by themselves looks worth a project. And because the work is spread across functions – design, manufacturing, quality, supply chain, suppliers – no single budget "owns" it.

That same diffusion is why supply chain risk stays invisible until late: obsolescence, restricted content, and supplier constraints often aren't discovered until procurement or suppliers are already executing. Leaders see the visible milestones (drawings released, parts received, audits passed) and reasonably assume the invisible effort in-between is just "how engineering works." But what they're not seeing is the huge drag on time, cost, quality, and risk – leaks in the process that together make up a tangible loss in profits.

That invisibility is exactly why the costs persist. Teams normalize the drag: search time becomes part of the job; copy/paste becomes "the quick way"; change queues stretch because impact analysis is manual; supplier delays are attributed to "supplier performance" rather than to the way requirements are delivered. Fire drills mask the pattern – people heroically close gaps, so the process looks fine on paper. When issues do surface – an NCR here, an expedite there – they read as isolated events instead of symptoms of the same root cause: requirements that live as text, not data.

The result is a strategic blind spot. Investment decisions optimize what's easy to count (licenses, deployments, headcount) and underweight what's hard to see (friction between systems, duplicate extraction, change propagation, and supplier adoption). Without a simple way to connect daily motions to business outcomes, the hidden tax keeps compounding – slowing new introductions, adding avoidable cost, and eroding confidence in the digital thread. The good news: once you name the pattern, you can start to measure it in straightforward ways and retire it deliberately.

What To Do Now

The good news: the daily leaks on your bottom line are avoidable. Turning documents into structured, machine-readable definitions and delivering them at the point of need changes the daily physics – less hunting, fewer re-types, faster, cleaner changes. It also lets teams evaluate alternates and mitigate supplier risk earlier – because key constraints (materials, processes, standards, lead-time drivers, and sourcing dependencies) are no longer buried in analog text.

What Good Looks Like – With SWISS

The fastest way to unlock your digital thread is to change how requirements travel. When requirements move as data, not paragraphs, they become findable, testable, and reusable across your stack, which translates into faster approvals, fewer supplier surprises, and fewer quality escapes. It also means procurement and supply chain teams can work from the same requirements context as engineering, so "readiness" is something you can validate before release, not something you discover after a supplier calls with a problem.

SWISS (Semantic Web for Interoperable Specs and Standards) makes this practical at scale: it uses domain-specific semantic AI, industry ontologies, and proprietary data on parts, materials, and processes to turn analog documents and drawing notes into model-based product definition (MBPD). Each modeled requirement carries clause-level provenance and is delivered by reference all the way through the digital thread from design to quality, manufacturing, and suppliers, so everyone works from the same, current source. Because the output is structured, SWISS can also identify "threaded" requirements referenced across related documents to assemble complete, step-by-step definitions for work instructions, test methods, and more, rather than forcing experts to manually stitch requirements together manually.

In practice, that looks like:

- Click-through from a drawing note to the governing clause in the source standard.
- Direct navigation from an industry spec reference to the relevant section, with provenance preserved.
- Impact analysis that shows the impact of a change before you route it.
- Supplier updates that highlight exactly what changed and capture acknowledgements.
- Early flags when a requirement implies a restricted material, obsolescent part, critical mineral shortage, or long lead-time process – before it becomes a procurement fire drill.
- Faster evaluation of alternate parts, materials, or sources because the constraints are explicit and traceable – not buried in analog text.
- Pre-release supply chain readiness checks: identify which parts, suppliers, and processes are affected by a requirement change and where lead-time or qualification risk might surface.
- Supplier adoption visibility: track who has received, acknowledged, and implemented the latest requirement set – so supply chain can manage execution proactively instead of reacting to late misses.

Next: Dive into Model-Based Product Definition

A short and sweet read (6 minutes)

The full read (15 minutes)

From PDF to Procurement: Realizing the Promise of Model-Based Product Definition (MBPD)

What you'll learn:

- **MBPD in one shot:** what it is, what it isn't, and how it coexists with your PLM/CAD tools.
- **What a "good" requirement looks like as data:** subject, attribute, value, units/tolerance, test method, and a built-in link back to the source clause and revision.
- **How MBPD moves through your stack:** delivering requirements by reference into PLM, QMS, MES, and supplier packets – no copy/paste.
- **How MBPD supports earlier procurement and supply chain decisions**: faster alternates evaluation, fewer late-stage escapes, and clearer supplier alignment to current requirements.
- **A simple pilot playbook:** how to choose a small, end-to-end scope; what to instrument with four KPIs (Engineering Hours, Change-to-Release, Supplier Latency, Rework); and how to show progress in weeks.

For More Information

Exiger's SWISS platform uses domain-specific, semantic AI trained with proprietary ontologies and data on parts, materials, and manufacturing processes to turn analog drawings, standards, and technical documents into structured, machine-readable product definition. Organized in the SWISS Knowledge Graph and available by API, that data can flow by reference into PLM/PDM, QMS, MES, and supply-chain workflows – helping teams catch outdated specs, regulated materials, critical mineral vulnerabilities, long lead-time processes, and supplier constraints before they become delays, rework, procurement emergencies, or compliance liabilities.
Learn more at: https://www.exiger.com/products/digital-thread-and-parts-intelligence/

