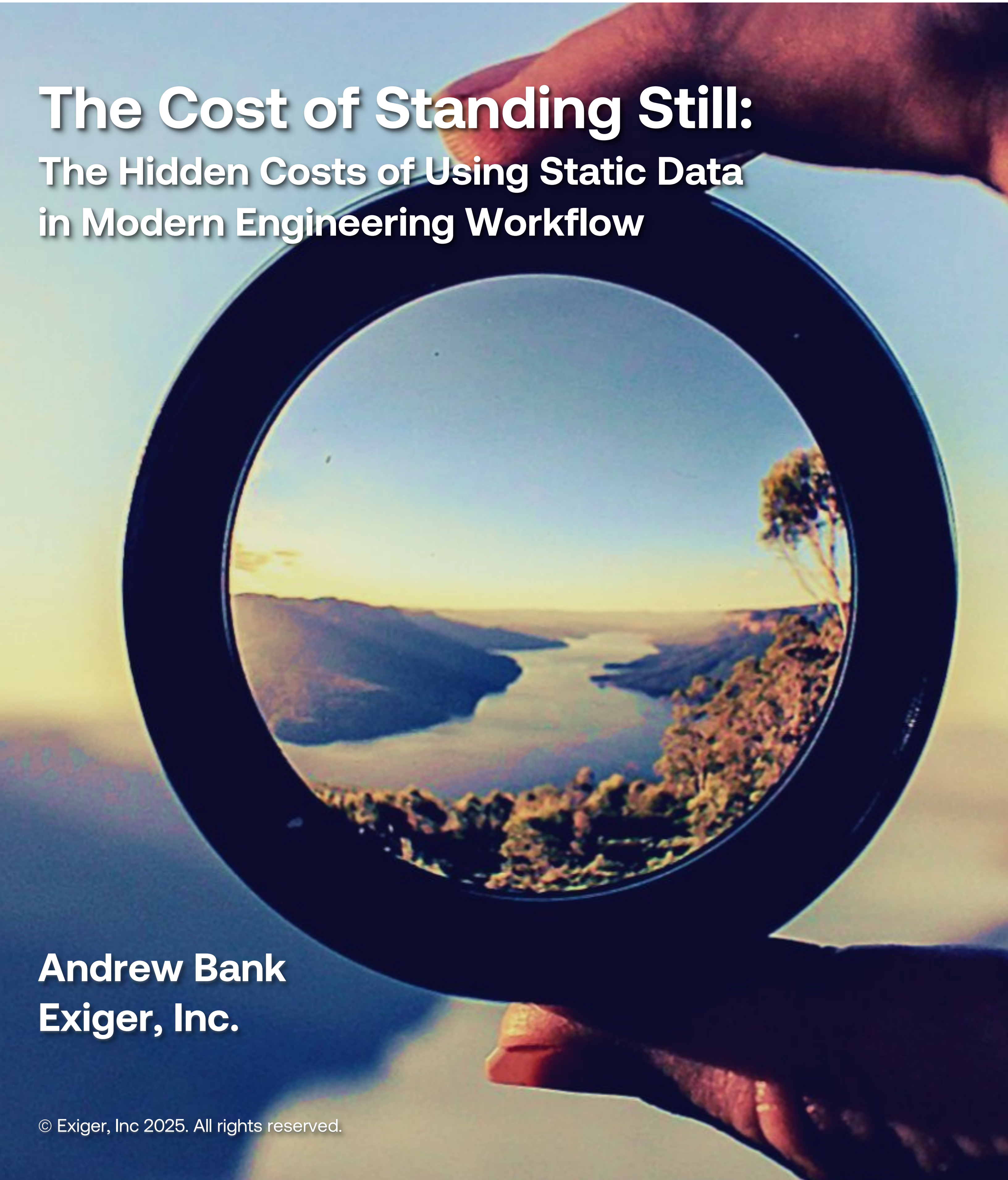


The Cost of Standing Still:

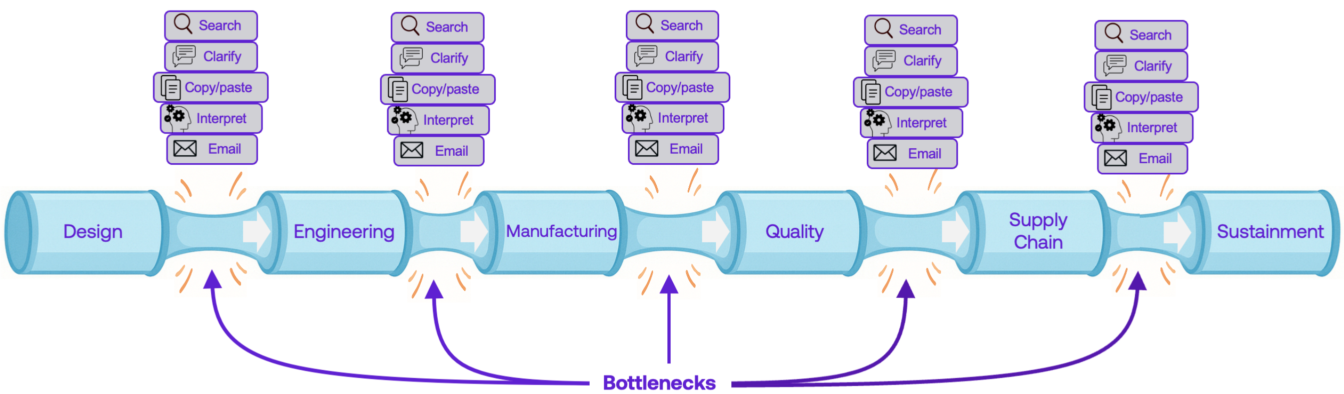
The Hidden Costs of Using Static Data in Modern Engineering Workflow

Andrew Bank
Exiger, Inc.

The Cost of Standing Still: The Hidden Costs of Using Static Data in Modern Engineering Workflow

If you lead engineering or manufacturing at an OEM, your programs likely run on a digital thread: PLM for change control, CAD/MBSE for design intent, QMS for quality, and supplier portals to keep production moving. Yet when the schedule squeezes or a change notice lands late on a Friday, teams still crack open PDFs, Word files, and scanned drawings. That habit feels harmless. It isn't.

Static documents are the slowest link in your digital thread. They force manual search, copy/paste, re-typing, and tribal interpretation – the very work your digital investments were meant to eliminate. The result is a daily tax: time lost hunting for clauses, changes that crawl through approvals, suppliers that lag while parsing emails, and quality escapes tied to ambiguous text. Because none of it shows up on a single budget line, the real costs hide in **engineering hours**, **change-to-release**, **supplier latency**, **rework, and avoidable risk**.



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

This article surfaces the hidden costs in plain English and shows a pragmatic path to turn dead text into live, **machine-readable, model-based product definition (MBPD)** so your digital thread does what you paid for it to do. The follow-up will include a simple Cost & ROI calculator to quantify the impact in your own operation and pinpoint which bottlenecks cost you most in time, money, and risk.

Most importantly, you'll see how to solve these problems with **SWISS**: using domain-expert AI, deep industry ontologies, and proprietary parts/materials/process data to transform static documents and drawings into comprehensive, machine-readable models that integrate with your enterprise applications (e.g., PLM) and flow cleanly from design to manufacturing, quality, supply chain, and sustainment without losing fidelity – **closing the last mile of the digital thread** so you capture the full financial and operational benefits of the investments you've already made.

What to Read Next: 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 (e.g., PTC Windchill, Siemens Teamcenter), QMS, MES, and supplier packets – no copy/paste.
- **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.
- **Guardrails that matter:** provenance, version control, licensing hygiene, and supplier acknowledgements – so "current" truly means current.

Read the full paper on Model-Based Product Definition here.

How to read the scores and tags

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.)

First, an example:

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.

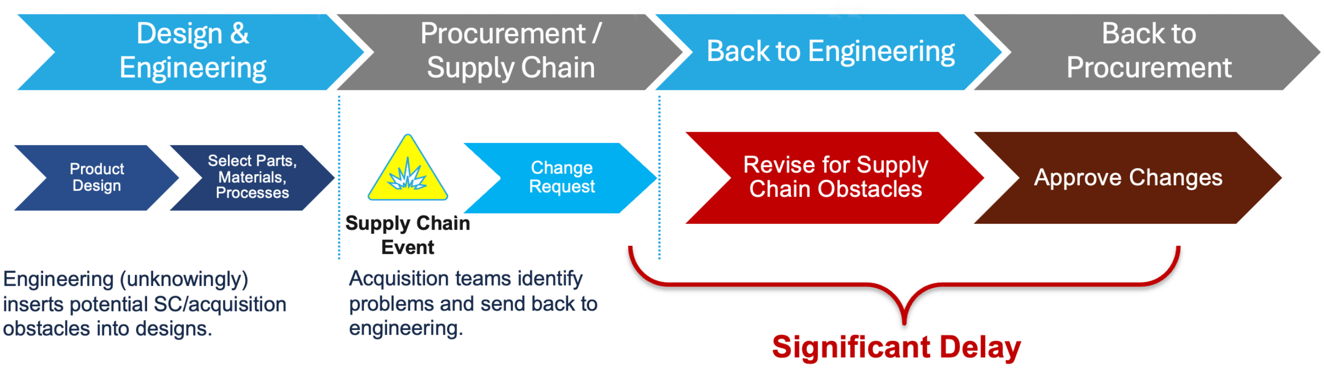
A Defense Reality Check: Weeks to Minutes

Army engineers routinely review TDPs before procurement to find out-of-date standards, regulated substances, critical mineral vulnerabilities, and long-lead-time processes (e.g., castings/forgings). Historically, this is brute-force work: open dozens of drawings and specs, scan general notes, chase references, and compile replacement standards – often over **30–90 days** per package.

Take this example when 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 a model-based approach, those documents were parsed into structured data in **~2 minutes per drawing** (under **3 hours** of machine time total), automatically flagging castings and restricted substances and generating a bill of references. ***Identifying those castings is credited for identifying and mitigating a 180-day sustainment delay with 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 dependency, a regulated substance, or discovering any of these late in the process can easily add quarters, not weeks, to a delivery schedule.

Keep that picture in mind as we unpack the following hidden cost buckets.



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

Hidden Cost #1: Finding critical requirements & risk factors is manual

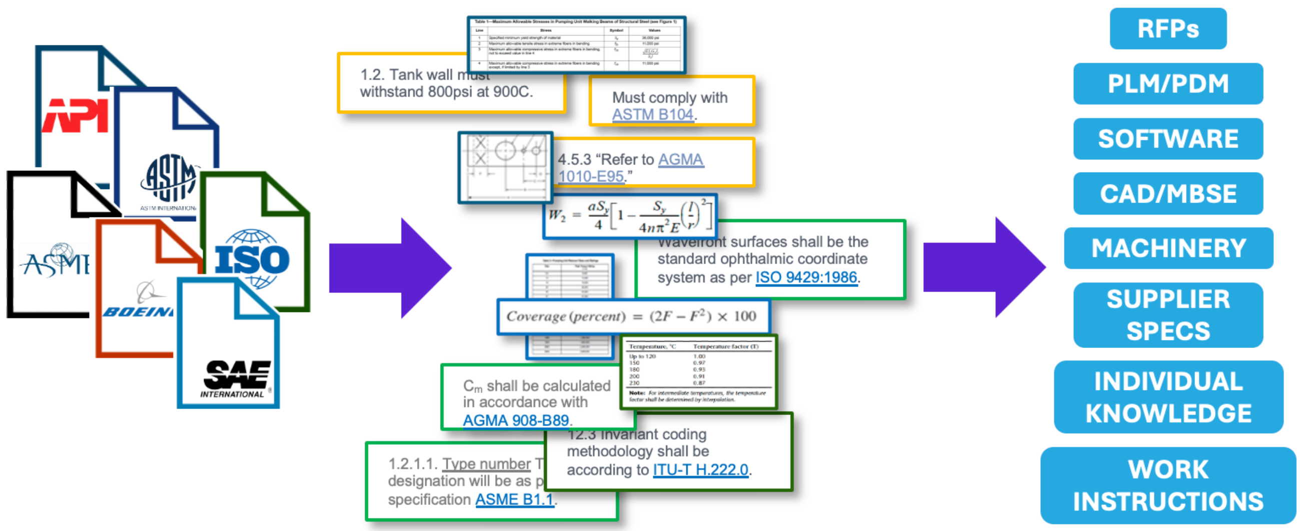
What it means

Finding the *right* requirement, test method, regulated material or critical mineral flag, long-lead process (e.g., castings/forgings), or outdated spec still depends on people reading PDFs and cross-referencing notes. 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: 3 | Compliance: 2

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

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 coating clause and a casting requirement that triggers a long-lead-time alert. Quality repeats the hunt two weeks later, and a supplier runs a third pass after release – each group starting from scratch.



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.*

What it costs
Hours of manual discovery for every package; repeated effort across engineering, quality, 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."

How to fix it
Turn documents into a **digital model you can search and trust**. SWISS ingests drawings and standards, extracts requirements and risk markers (regulated materials, critical minerals, long-lead processes), and builds a **model-based product definition (MBPD)**. It also generates a **Bill of References** – every requirement carries its source clause and revision. Inside PLM (e.g., **PTC Windchill** or **Siemens Teamcenter**), users click straight to the current clause instead of

opening a stack of PDFs. That ends the “hunt-and-hope” cycle and keeps teams aligned on one answer.

Hidden Cost #2: Manual extraction & re-keying (is so 1990)

What it means
People still **copy/paste or re-key** the same values – dimensions, tolerances, materials, test methods – into PLM/PDM, MES, QMS, ERP, inspection plans, and supplier packets. It’s time-consuming work that invites small but costly slips.

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
Large chunks of skilled time diverted to data entry; bottlenecks around big releases; transcription errors that ripple into rework and corrective actions; frustrated engineers who feel more like clerks than problem-solvers.

How to fix it
Shift from typing to reviewing. SWISS reads drawings and specifications and converts the key details into clean, structured fields that downstream tools can consume. Inspection characteristics, routing steps, and supplier packets can be **generated from the model** and reviewed by an engineer instead of re-keyed by everyone. Approved updates flow back into PLM (**Windchill**, **Teamcenter**), so one change feeds every consumer system without manual re-entry.

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. Extracting them into subject, attribute, and value is nearly impossible.

Hidden Cost #3: Static files are not machine-readable, break interoperability, and block automation

What it means
PDFs and static drawing notes don't interoperate with modern systems or each other. PLM, MES, QMS, and ERP can store them, but they can't *use* them programmatically. People become the integration layer, and the automation you've paid for – automatic checks, auto-built plans, compliance scanning – can't run. When requirements stay as paragraphs, or they are simply cherry-picked into "should", "shall", or "may" sentences, automation can't run.

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

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

Example
Quality needs the exact 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
Slow and brittle handoffs; duplicate side files and spreadsheets that drift from the source; no reliable way to run tolerance checks or compliance scans at scale; inconsistent decisions because each person interprets prose in slightly different ways.

How to fix it
Make requirements **machine-readable and machine-usable**. 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 #4: Change impact and change-to-release are impaired

What it means
Manually finding what a change touches, making edits across multiple artifacts, and routing approvals stretches the time from decision to release. While the change waits, other work either stalls or proceeds on old information, setting up rework.

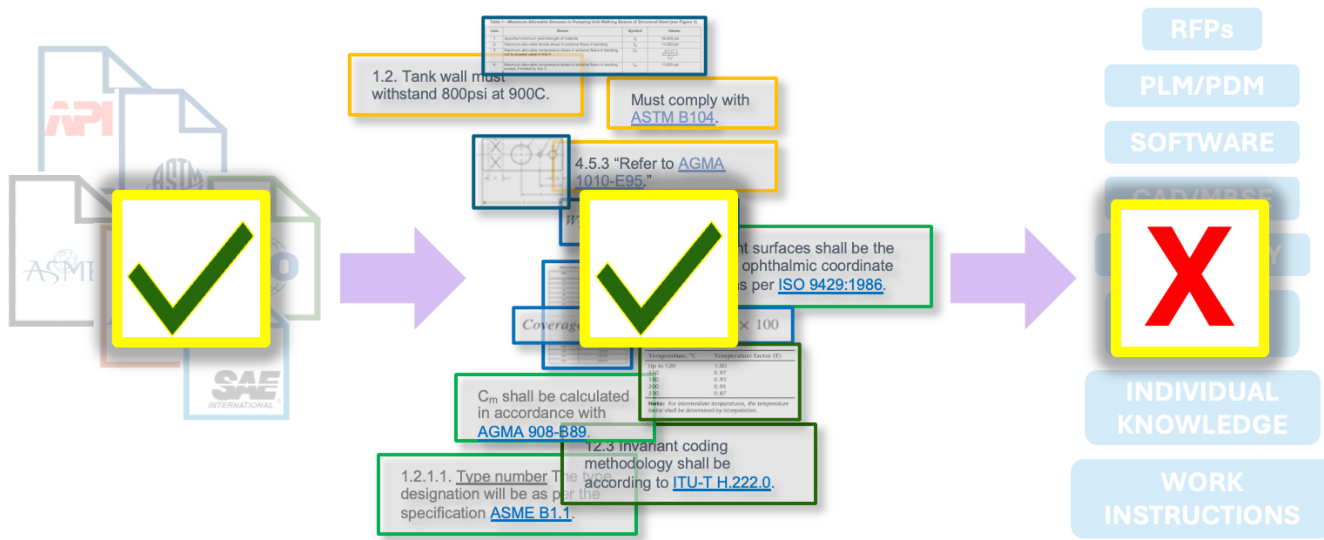
Pain Scale
Speed: 4 | Cost: 3 | Compliance: 3

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

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

What it costs
Longer approval queues; frozen or wasted work in other teams; more meetings and re-routes to reconcile edits; increased risk that old wording leaks into production while approvals are still underway.

How to fix it
See the blast 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 (**Windchill**, **Teamcenter**) 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 derivatives do not.

Hidden Cost #5: Copy/paste and re-key "tech debt" creates outdated specs and even more static data

What it means
Copying text from standards or drawings into internal documents creates **new** static data that drifts from the source. When the original changes, the copies don't. Company and supplier versions diverge, and copying large chunks of paid standards can add licensing risk.

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 external 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 clean-ups and retraining; re-issues to suppliers; risk of building to outdated rules; potential licensing exposure from reproducing paid content; general erosion of confidence in which version is "current."

How to fix it

Reference, 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 #6: Supplier adoption latency & supply-chain risk

What it means

After you release a change, suppliers still have to interpret it, update routers and instructions, and qualify before shipping to the new rule. Late discovery of restricted materials, obsolete specs, or long-lead processes turns into fire drills, especially for castings and forgings.

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

Days to weeks of schedule slip; expedite fees; WIP scrap and rework; strained supplier relationships; loss of schedule buffer on long-lead items that are hard to recover.

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 and regulated-material risks upstream**, so they're resolved before release, not after suppliers have started work, with acknowledgments you can see.

Hidden Cost #7: Rework & defect escapes

What it means

Ambiguous wording, stale references, or copy/paste slips lead to parts or tests that don't meet intent. The mistake isn't obvious until inspection or audit, so the fix is late and expensive – often requiring deviations, rework, or scrap.

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 static 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 a router, 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 – engineering, quality, supply chain, suppliers – no single budget "owns" it. 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 it doesn't have to be that way.

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 chalked up 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.

Why This Matters – And What to Do About It Now

Static documents are familiar, but they don't keep up with model-based engineering or the digital thread. Every copy, re-type, and manual chase is a small drag that compounds into missed windows, higher costs, and compliance headaches.

The good news: these 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.

SWISS uses domain-expert AI, deep industry ontologies, and proprietary data on engineering parts, materials, and manufacturing processes to transform static documents and drawings into model-based product definitions (MBPD) that enable rapid identification of engineering decision-making data and supply chain risks. It then delivers those modeled requirements by reference into your PLM, quality, manufacturing, and supplier workflows so everyone is working from the same, current always-updated source.

What to Read Next: 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 (e.g., PTC Windchill, Siemens Teamcenter), QMS, MES, and supplier packets – no copy/paste.
- **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.
- **Guardrails that matter:** provenance, version control, licensing hygiene, and supplier acknowledgements – so "current" truly means current.

Read the full paper on Model-Based Product Definition here.

For More Information

Exiger's SWISS platform uses domain-expert AI with proprietary ontologies and data on engineering parts, materials, and processes to transform static documents and drawings into interoperable, model-based "digital twins" that enable rapid identification and use of engineering and supply-chain decision-making data. The resulting model-based is organized and accessible via the SWISS Knowledge Graph and enables easy identification of regulated materials, critical minerals, outdated specs, and supply chain risks from previously static content. This model-based product definition integrates seamlessly into enterprise applications including PLM/PDM, MES, QMS, and SCM, facilitating unimpeded product data flowing from design to manufacturing to supply chain to sustainment.

Learn more at: https://www.exiger.com/products/digital-thread-and-parts-intelligence/