



# Top Five Use Cases of ALM-PLM Integration

**Building Trust and Excellence in  
Product Development**

---



# Table of Content

---

**Trust: The New Currency of Team Performance**

**3**

---

**Use Case #1: Manage Requirements Across the Product Lifecycle**

**5**

---

**Use Case #2: Improve Team Agility**

**10**

---

**Use Case #3: Effectively Manage Change**

**15**

---

**Use Case #4: Improve Product Quality**

**18**

---

**Use Case #5: Improve Project Governance**

**23**

---

**About PTC Codebeamer**

**27**

---

**About PTC Modeler**

**28**

---

**About Windchill PLM Software**

**29**

---

# Trust: The New Currency of Team Performance

---

A 2022 McKinsey survey of global consumers and business leaders highlights the central **role of trust** in building consumer acceptance and business growth. “Organizations that are best positioned to build digital trust are also more likely than others to see annual growth rates of at least 10%,” notes McKinsey.

Intangible yet imperative, trust is the invisible thread that links buyers and sellers, consumers and producers, and in the context of product manufacturing – teams of teams working collaboratively to deliver smart, complex products to the marketplace.

Smart, complex products combine software with physical products to deliver intelligent, connected, and evergreen capabilities. They represent some of the most significant advances of the past decade, from the Industrial Internet of Things to genetic sequencing, robotic surgery, and autonomous vehicles.

Smart, complex products are multifaceted and complicated to design, build, and manufacture. Their formation requires the specialized expertise of product management teams, systems, mechanical, electrical and electronics engineering teams, and software development teams working collaboratively towards common goals.

Successful teams rely on Application Lifecycle Management (ALM) and Product Lifecycle Management (PLM) disciplines to build trust in product teams, processes, and information.

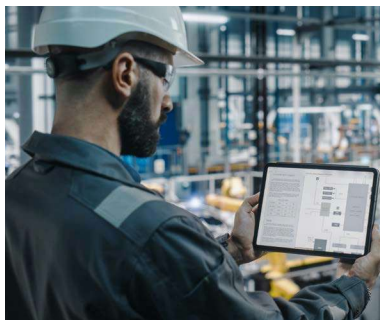
- **Application Lifecycle Management (ALM)** enables end to end management and tracking of software related processes, from inception, design, development, testing, deployment, and regulatory management through end of life. In the context of smart complex products, ALM manages the software that runs, controls, or reports on products. In the context of smart complex products, ALM, beginning with Model-Based Systems Engineering (MBSE), manages the software that runs, controls, or reports on products.
- **Product Lifecycle Management (PLM)** enables end to end product development and tracking of all product related processes, from product inception, design, development, and quality management out to manufacturing and service planning. PLM provides enterprise governance for these processes across the digital thread.



Combined, ALM and PLM provide a robust thread enabling collaboration, insight and governance to drive the product development cycle. Without this firm foundation, teams quickly lose speed and morale as they struggle to identify the location, version, and veracity of the millions of discrete pieces of information required to orchestrate the development of today's smart, complex products.

**Where to start your journey to improving team trust and excellence? This guide explores the top five use cases we see motivating PTC customers to integrate and improve their ALM and PLM maturity.**

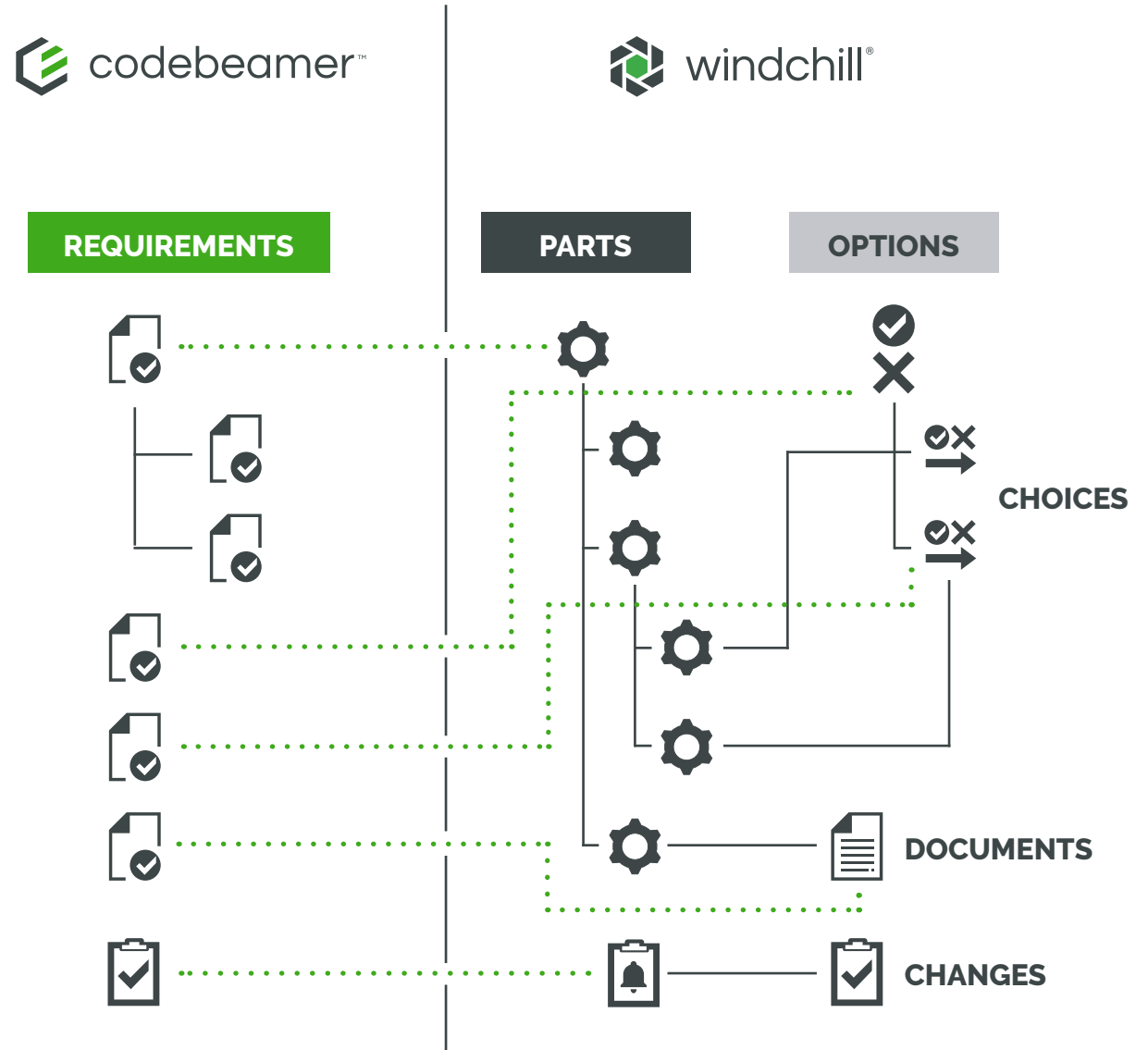
Of course, the use cases that may motivate your organization are unique to you. For this reason, we provide questions that can help guide your organization's prioritization and decision-making.



## Use Case #1

### Manage Requirements Across the Product Lifecycle

Requirements traceability empowers teams to understand which requirements are associated with any product or part, and trace any part back to its originating requirements.



# Use Case #1:

## Manage Requirements Across the Product Lifecycle

**Requirements management is the process of eliciting, analyzing, validating, and documenting market and user needs for a solution under development.**

Mature requirements management is a foundational discipline for building trust in product teams. Successful organizations create a shared vision of the product or application under development, and continuously refine that vision as the product evolves.

Requirements may include:

- **Functional requirements** that define product functions or capabilities
- **Non-functional requirements** which specify general attributes or performance needs, such as speed, torque, weight, or other attributes
- **Safety requirements** that define how the product will perform in response to specific hazards or threats
- **Regulatory requirements** which must be fulfilled in order to sell into specific industries, geographies, or markets
- **Epics and User Stories**, requirements elicitation techniques used by agile teams. These techniques capture needs and wants from the user's point of view, typically using first-person "I need, I see" narration to build an understanding of the users' world and empathize with their challenges.







## Benefits of ALM and PLM Requirements Integration

Integrated requirements management practices confer the following benefits:

### Alignment

Requirements traceability unifies individuals and teams around common objectives. Forwards traceability allows organizations to understand how requirements evolve from high-level vision through detailed specification and instantiation in any product. Backwards traceability empowers individuals to trace any work item back to its originating set of requirements.

### Collaboration

The more cross-functional teams can understand and internalize the “voice of the customer”, the more successfully teams can collaborate and the more likely the solution is to succeed. Modeling system requirements and allocating across teams is an important part of success.

### Enhanced Reuse

Requirements reuse supports platform and variant strategies that drive down costs and improve profitability. Successful reuse requires mature change and configuration management practices in order to manage requirements across a continuously evolving product landscape.

### Improved Quality

Verification and validation help ensure products continue to meet evolving user and market needs. Verification is the process of “building things right” by demonstrating that work products meet agreed-upon specifications. Validation is the process of “building the right thing” by confirming stakeholder buy-in and market demand. Mature teams continuously verify and validate requirements throughout the product lifecycle.

### Regulatory Compliance

Many regulated industries, including aerospace and defense, medical, pharmaceutical, and automotive impose specialized requirements governing safety and other product attributes. The ability to expose, track and manage these requirements is requisite for regulatory compliance.

## What It Looks Like: A Typical Requirements Integration Scenario

There is no one “right” way to manage requirements, as every organization is unique. Below is one example of project cadence for a hypothetical automotive manufacturer using an agile product development process.

### A Portfolio team

- Analyzes patterns and market trends
- Conducts focus groups to understand customer preferences
- Categorizes, analyzes, and ranks investment themes for profitability
- Defines a high-level vision of the product under development
- Communicates the high-level vision with a webinar, video and white paper that is accessible to all team members
- Continuously scans the competitive landscape and user groups to understand evolving needs – and updates vision documents as needed.

### Program Planning teams

- Decompose the product vision into tasks (epics) to be carried out by teams. Each product capability is associated with an investment theme and assigned to a team for further elaboration.
- Determining tasks and allocating them to teams.

### Physical Product Teams (Mechanical, Electrical, Electronic, etc.)

- Refine physical products requirements established by the program planning team, adding detailed specifications relevant to their engineering discipline
- Assign epics to sprints
- For each sprint, verify that deliverables meet requirement specifications

### Software teams

- Refine software requirements into user stories, adding detailed software specifications.
- Assign epics to sprints
- For each sprint, verify that deliverables meet requirement specification

### Quality teams (separate and/or integrated)

- Continuously verify that deliverables meet requirement specifications





## Questions for your team

- ? How well do your products reflect customer and business needs?
- ? Have you ever had to delay or rollback a product because a critical requirement was incomplete or missing? What was the impact in terms of revenue, customer satisfaction, or reputation?
- ? How consistent and repeatable are your requirements management processes?
- ? How easy is it for new team members to get up to speed?
- ? How easily can you reuse common requirements from one project to the next?
- ? How quickly can your team pivot to respond to new user or competitive information?
- ? How do you design or map requirements and domain-specific design?
- ? How do you allocate work across specific engineering disciplines?

“

**“We needed to capture customer needs and trace them down to product requirements, down to design, tests, and source code. Codebeamer was the only tool that allowed us to do that in a smooth way.”**

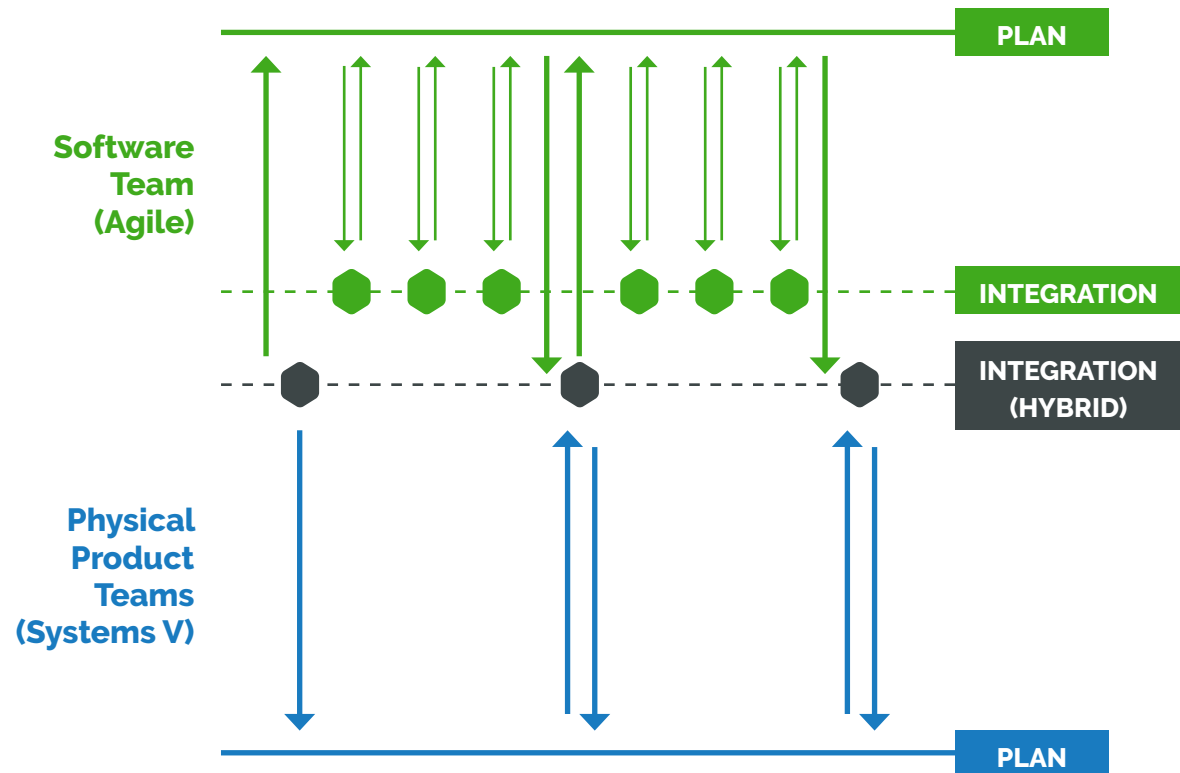
– Sarb Singh-Kaur, Director of Patient Care Software, Medtronic



## Use Case #2

### Improve Team Agility

In this hybrid model, software teams release every 4 weeks while physical (design, electrical and mechanical) product engineers release every four months. Both teams collaborate every 4 months to establish a new product baseline.



## Use Case #2:

# Improve Team Agility

**Agile development is the de facto development methodology of the software world – and is rapidly expanding to all engineering disciplines.**

Why? According to a May, 2021 [survey by McKinsey](#), highly successful agile transformations typically realized a 30 percent gain in efficiency. What's more, these organizations were 3 times more likely to become top-quartile performers.

How do companies build products that incorporate both physical and software elements? When it comes to these complex products, organizations can choose from a broad range of process models.

### Systems V

This Systems Engineering approach is widely used to manage large scale

projects that involve multiple engineering disciplines. The left side of the "V" typically decomposes requirements into designs and detailed engineering specifications, leading to component delivery, while the right side of the "V" integrates, validates, and verifies each lifecycle stage. The V-Model is a useful framework for managing complex products that require a high degree of rigor.

### Agile

This project management approach prioritizes flexibility and customer satisfaction. Hallmarks include short iterations (sprints), collaborative planning, regular reviews (retrospectives), and continuous delivery of value. Agile methods are ideal for projects with unclear or evolving requirements, as its iterative approach clarifies user needs as it delivers value.

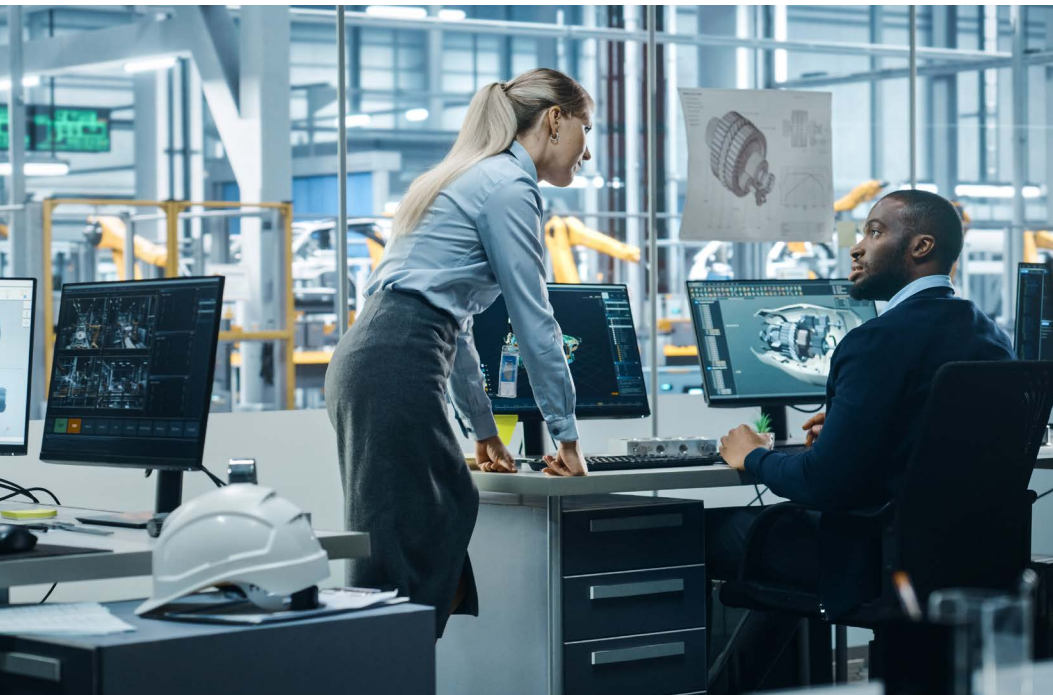
### What is Agile?

- **An iterative and incremental project management approach that prioritizes flexibility, collaboration, and adaptability**
- **A mindset centered around continuous improvement**
- **A set of frameworks that implement Agile values, including Scrum, Kanban, lean, and more**
- **A set of best practices that includes short iterations, collaborative planning, cross-functional teams, customer involvement, regular retrospectives, and continuous delivery of value.**

— Source: [Forbes](#), "[The Agile Mindset, Revolutionizing Software Development Team](#)"

## Hybrid

This model combines elements of multiple methodologies, and is probably the most common approach in modern manufacturing environments. One common approach of hybrid product development teams is for design, electrical and mechanical engineers to follow the systems engineering V-Model while software teams follow agile practices. Periodically, both sets of teams come together to integrate work and conduct a product-wide retrospective. The Scaled Agile Framework (**SAFe**) is one example of a Hybrid model.



## Benefits of ALM-PLM Process Integration

### Clear “Rules of the Road”

A unified process enables all team members to understand what they need to do, how to do it, and how their work fits into the bigger picture.

### Improved Team Alignment

Common understanding of objectives, design, timelines, and team members helps everyone understand the context of what they are working on, when it needs to be delivered, and why it's important for the whole product.

### Less Rework

A unified product development process clearly communicates how team output will integrate with other teams' work. Understanding these integration points is critical to building products right the first time.

### Repeatability, the Basis of Continuous Team Improvement

A continuous feedback loop is only possible if the process is documented and followed. At the end of each delivery, teams can come together to review what went right, what went wrong, and what changes could accelerate delivery and quality.

### Improved Quality and Auditability

In many regulated industries, proof of process adherence has been codified into a regulatory requirement, especially for software teams. The ability to enforce and demonstrate process compliance therefore becomes a regulatory requirement.



## What it Looks Like: A Typical Process Integration Scenario

It is beyond the scope of this white paper to show a detailed development process. By way of example, below are broad brush strokes of how a hypothetical manufacturer integrates their product development process.

### The Product Line / Portfolio team

- Analyzes patterns and trends and defines investment themes
- Categorizes, analyzes, and ranks investment themes for profitability
- Defines a high-level vision of the product under development

### Product teams

- Decompose the product vision into tasks to be carried out by teams.
- Assign work to individual physical and software engineering teams

### Physical teams

- Follow a V-Model process to deliver updates every three months, working from the requirements established by the epic planning team.

### Software teams

- Follow an agile process to deliver updates every three weeks.
- Refine requirements defined by the portfolio planning team into user stories.
- Define and execute sprints—a set period of time during which specific work is completed and made ready for review.
- Assign user stories to sprints, where they are implemented.
- Continuously integrate software with available hardware by testing with actual or simulated hardware at the end of each sprint.

### All teams

- Come together for project-wide alignment and integration of deliveries every three months, culminating in a “demo day”. In preparation for this event, teams conduct intensive integration testing and resolve any outstanding issues.

### At project conclusion

- Following product delivery, teams come together for a retrospective that reviews highlights and opportunities for improvement, reviews project metrics, and brainstorms on process changes that could improve product quality, team velocity, or both.



## Questions for your team

- How well integrated is your process across systems, software, hardware, electrical and mechanical engineering teams?
- Have you ever had to delay or rollback a product because of an error or omission in team hand-offs? What was the impact of that delay or rollback in terms of revenue, customer satisfaction, or reputation?
- How easy is it for new team members to understand the process for their team, and how it fits into the bigger picture?
- How enforceable is your process? How readily can you demonstrate to regulators that your organization has an established process that is followed by all team members?

“

**“Codebeamer out-of-the-box gives you real time traceability across the V cycle. It’s one of a few commercial products that actually do that. If I want to do the entire lifecycle in one tool, this was the answer.”**

— Tim Brennan, IT ALM Engineering Manager, Veoneer

**“We were planning on transitioning to Agile, and discovered that Codebeamer had the infrastructure to support that. We didn’t need to purchase another product to manage all the sprints and user stories and tasks – we could do all that embedded in ALM, so that was a great benefit for us.”**

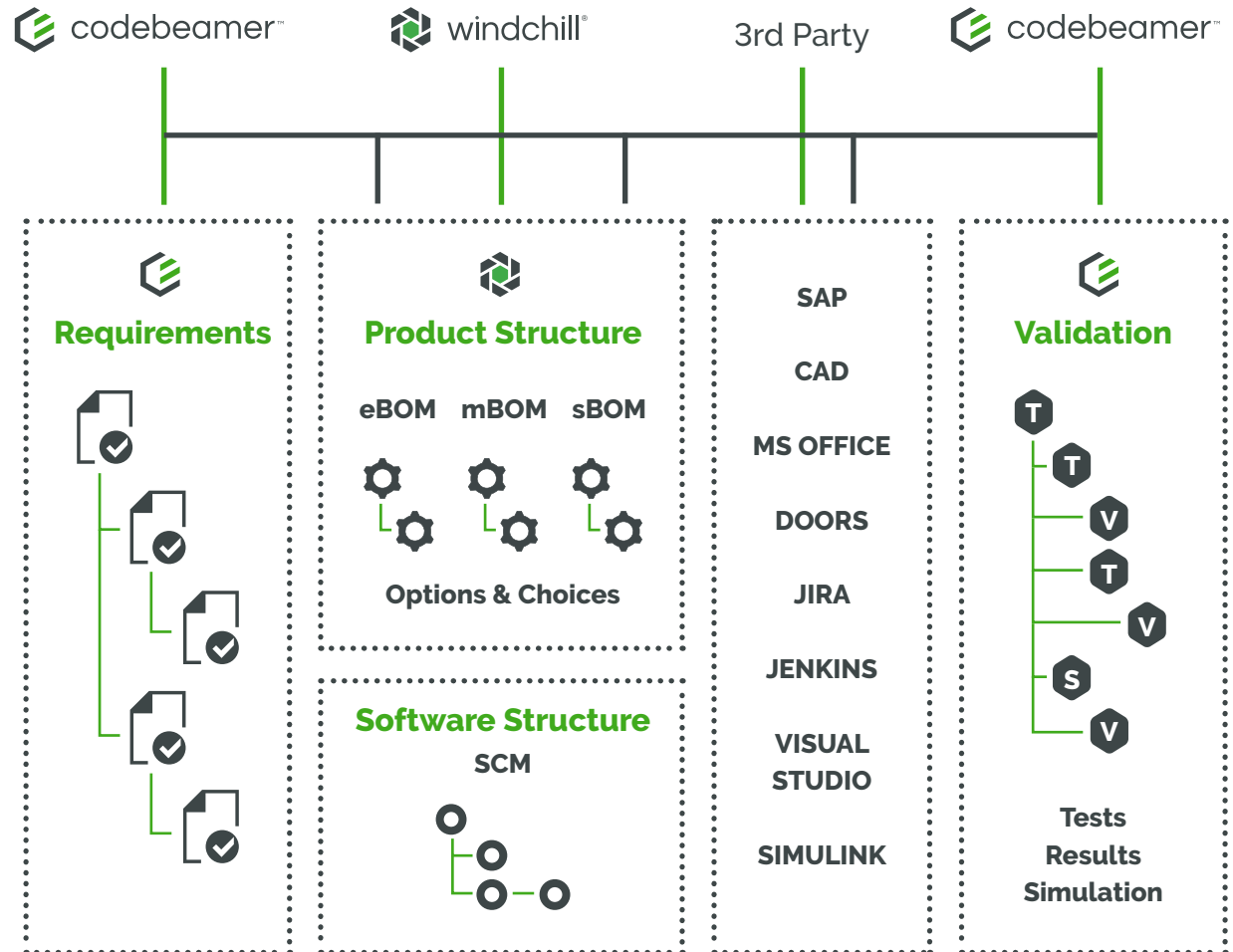
— Dr. Rita Hahn-Petschick, Leader, Analysis Solutions, Continental AG



## Use Case #3

### Effectively Manage Change

Efficient change management requires digital product traceability spanning requirements, design, and test assets.



## Use Case #3: Effective Digital Thread

---

**In the words of Winston Churchill, “to improve is to change; to be perfect is to change often.”**

Indeed, product teams must continuously update products in order to:

- Respond rapidly to evolving customer needs
- Pivot to meet competitive threats
- Mitigate errors in requirements, design, or even manufacturing
- Create one-off, concept, or customer special products

In short, change management is key to organizational agility, a precept for successful product companies.

Change management requires advanced traceability, navigation, and impact analysis capabilities across and between ALM and PLM domains. Specifically, it requires team members to:

- Traverse a robust digital thread of traced data including requirements, tests, documents, and logic. This enables cross-discipline impact analysis to ensure all enterprise needs are accounted for.
- Interrogate details related to the change and manage change-related issues and reporting

### Benefits of ALM-PLM Digital Thread

Organizations with mature and integrated change management practices:

#### Improve Team Agility

Mature change management helps teams respond rapidly to customer requests, changes in market conditions, and competitive threats. It enables organizations to pivot mid-stream to address evolving market needs.

#### Enhance Quality

Multi-domain changes are complicated. Without strong change management practices, it's easy to make a small change in a part or interface and forget to change the corresponding software – and vice versa. Change management enables teams to proactively manage quality even as requirements and products evolve.







#### Resilience

Organizations with the ability to embrace change are resilient in the face of evolving markets, supply chains, and competitive threats.





## Questions for Your Team

-  How resilient to change is your product development organization?  
How easy is it for you to make new information actionable?
-  How quickly can your organization pivot to counter a competitive threat?
-  How prepared are your engineering teams to manage late-stage changes?
-  How well do your teams collaborate on changes that span multiple teams?
-  How do you conduct impact analysis?
-  How do you prove end-to-end traceability?

“

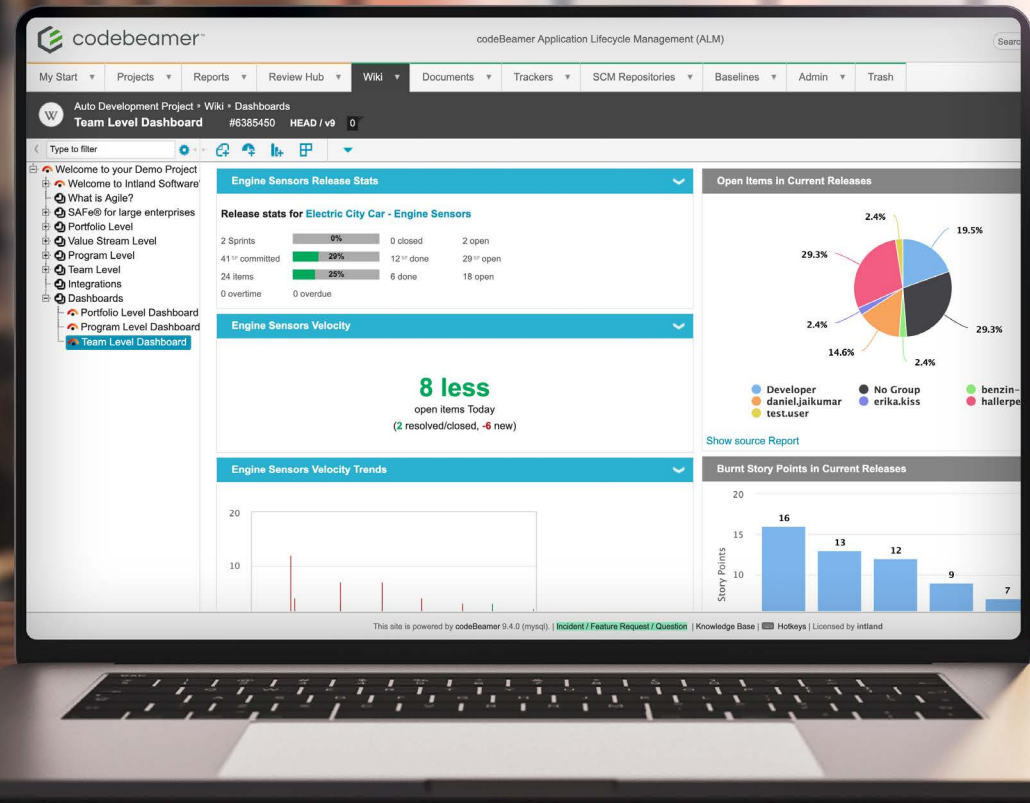
**“Codebeamer has provided us with a single interface to link and manage all sorts of objects, all of which are required to develop a highly complex product. The platform enables us to do this in a user-friendly and very flexible environment, linking all the functions of the company in a coherent manner.”**

— Jules Garbé, Head of Product, Navya

## Use Case #4

### Improve Product Quality

Individualized quality dashboards, like the one for this software tester, allow team members to focus on what is timely and relevant.



## Use Case #4: Improve Product Quality

---

**Quality management is the practice of managing all aspects of the quality lifecycle in order to verify product functionality, safety, performance, security, and availability. Mature quality management practices help teams and organizations build a culture of quality, a foundational principle of successful products and companies.**

In addition to defining good requirements and applying a suitable process, many different types of testing can collectively help ensure product quality:

- **Functional / Unit testing** verifies the functionality of a specific component.
- **System testing** exercises an entire system or product.
- **Simulation** model and test system or product behavior.
- **Security testing** is focused on the authorization, sign-on, threat mitigation and security systems of a product or system.
- **Usability testing** focuses on understanding and improving the user experience.
- **Regression testing** focuses on ensuring that changes didn't inadvertently introduce defects.
- **Aging testing** simulates exposure of materials in differing environmental conditions.
- **Spike or Stress testing** measures how the product performs in response to differing loads or other environmental or usage stressors.
- **Integration testing** exercises connections between components or with other products or systems.
- **Automated testing** is performed by computer or electronics systems trained to exercise products or systems, record results, and send error notifications.
- **Manual testing / inspection** is validation performed by people who manually interact with products and systems and record results.
- **Acceptance testing** verifies quality assurance from an end user's perspective, and is typically performed prior to release.

## Benefits of ALM-PLM Quality Management Integration

### Less Rework

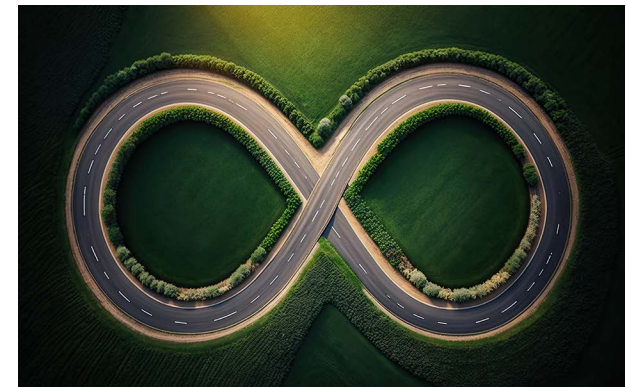
Building a quality product is everyone's responsibility, not just the QA team's job. Mature quality management practices empower team members to track quality throughout the product lifecycle and integrate a quality-first approach into daily activities.

### Reduce the Cost of Quality

Integrations with automated testing tools speed test creation and execution across the team. Team members should be able to automatically generate test cases from requirements, create work items from failed test cases, and see at-a-glance the quality status of their individual and team workstream.

### Maximize Reuse

QA efficiency climbs sharply when teams are able to parameterize test cases, organize test cases in libraries, and branch and merge libraries as needed to meet specific testing needs. The ability to version-control test assets is foundational for reuse. In mature quality assurance teams, test assets are subject to the same release, change, and configuration management discipline as any other asset.







### What it Looks Like: A Typical Integrated Quality Management Scenario

The following is a hypothetical scenario based on an imaginary company:

- Parts Corp prides on the quality of its hydraulic and anti-lock braking systems. To maintain this hard-won reputation, the company has a robust quality process.
- Requirements are created at the product-level before being decomposed into mechanical, electrical, and software requirements. Each set of requirements is validated and verified at each lifecycle stage.
- Software is released on a six-week sprint cycle, and undergoes both automated and manual testing to verify results at each stage. Changes to software are regression tested before being accepted into the current release stream. Software releases are regression tested against the latest hardware release or simulation.
- Physical product workstreams, which include mechanical and electrical workstreams, release every three months. At release time both software and hardware teams work together to overcome any integration issues.
- Once the hardware release is acceptance tested, it becomes the latest release by which all software releases are tested.
- System testing takes place throughout the product development cycle not just at the end.
- At any point, team members can trace test results back to test cases and originating requirements. They can also trace requirements forward to test cases and test results. This makes it possible for teams to verify that all requirements have been met prior to releasing the product.

## Questions for Your Team



Have you ever had to delay or rollback a product release due to quality issues? What was the impact of that on revenues, customer satisfaction, or reputation?



How do you know when your product is ready for release? How confident are you that all requirements have been verified and validated?



How automated is your quality assurance process? How could automation help your organization improve quality, reduce quality cost, or both?

“

**“The ability to perform full traceability, track the process and signatures/signoffs through the various reporting methods provided in Codebeamer allows us to clearly demonstrate and visualize our quality SOPs... The most positive value we got out of the software was the ability to clearly show process support and traceability.”**

— Ben Oberholzer, Manager of Software Development, DATATRAK





## Use Case #5

---

### Improve Project Governance

Merriam Webster defines **governance** as the process of overseeing the control and direction of something. Application Lifecycle Management facilitates a governance framework for software and overall projects, while Product Lifecycle Management enables governance over the whole product development cycle. Mature governance practices align teams around a shared product vision, provide clarity and transparency around decision making, and help build a culture of trust.



# Use Case #5:

## Improve Project Governance

---

### Benefits of ALM-PLM Governance Integration

#### Improved Focus and Alignment

An integrated governance framework is a powerful tool for aligning individuals and teams with the most important priorities. Without tailored information sharing, team members can easily become overwhelmed by the amount of information stored in an ALM or PLM system. That's why it's critical that team and individual dashboards display project information that is timely, relevant, and curated for each team and role.

#### More Consistent Project Deliveries

An integrated governance framework enables all team members to see at-a-glance whether deliverables are on track to meet project deadlines. It allows project managers or team leads to navigate escalations and identify potential problems earlier in the product lifecycle, when issues are less costly to resolve. Milestones may be time- or target-based.

#### Continuous Improvement

As the saying goes, you can't manage what you can't measure. An integrated governance framework provides insight into project quality and team velocity. Team metrics provide the foundation for continuous capability improvement for document best practices and a repeatable, agile process for any project.

#### Improved Auditability

A rigorous governance framework is a requirement in many regulated industries. In some industries, electronic signatures are the de facto standard for change management control. Mature governance practices make it easier to demonstrate that teams have established processes in place and that product development assets are secure.





## What it Looks Like: A Typical Project Governance Integration Scenario

Optical Inc specializes in designing and manufacturing optical solutions used in diverse scientific, robotics, and medical device applications. The company's electronic optical solutions combine hardware with electronics and software to "see" into far galaxies or the human bloodstream.

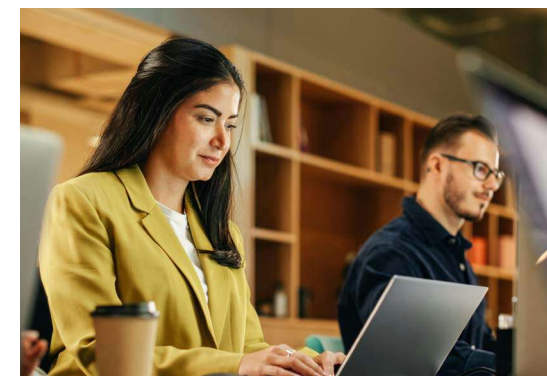
Optical Inc. follows a hybrid agile process called the "Optical Way". Software teams work in three-week sprints, while mechanical and electrical engineering teams work in nine week deliverable cycles. After each hardware delivery the teams come together for a cross-domain review and retrospective.

At the start of each new product, an internal project website is set up with information that defines the product, its target market, objectives, and key usage scenarios. The project website also includes a list of all teams and individuals working on the project.

Custom dashboards provide relevant information at the task, individual, team, project, and organizational level. Key visual assets for software sprints include:

- An Activity Stream that provides an aggregated view of all project activities
- A Burn Down Chart that displays open work items for a release or sprint
- A Burn Up chart that shows the number of resolved or closed issues, and the total number of work items over time
- Reports that track select workstreams or items of interest, such as recently submitted bugs or issues by status
- A list of Current Issues that have been assigned to an individual or team
- Team Velocity (completed issues per week per team)

By navigating information at the team, project, and organizational level, the organization has the information it needs to meet critical deadlines, improve project predictability, and track and continually improve performance over time.



## Questions for the Team

Consider the last time your organization failed to meet a product or project target date.



What was the impact of that in terms of team morale, revenue, customer satisfaction, and reputation?



If you could have identified the problem sooner, what practical steps would you have been able to take to make the target date?



How confident are you in your ability to predict team velocity, based on actual experience?



**“We chose Codebeamer because with this tool, it finally feels like we’re really working together. As the tool’s modules are organically integrated, all our requirements, testing, issue tracking activities, and artifacts are connected, traceable, and accessible to every member of our team.”**

— Jens Amberg, Vice CTO, Fritz Stephan

**“Codebeamer is an easy to use and cost-effective enterprise class collaborative software development platform that supports the needs of a distributed team. It is a system that is easy to maintain, and its metrics reporting capabilities provide us the kind of visibility we need to manage our projects effectively.”**

— Dr. Rita Hahn-Petschick, Leader, Analysis Solutions, Continental AG

# About PTC Codebeamer

---

**Codebeamer is a new generation Application Lifecycle Management solution for product and software engineering teams. This open platform combines requirements, risk, and test management with agile engineering capabilities. Simply put, Codebeamer accelerates software delivery while simplifying regulatory compliance.**

## Full Lifecycle Support

Codebeamer manages the complete software lifecycle from initial idea through design, development, testing, deployment, and ongoing change management. It empowers teams to:

- Establish a common software engineering process
- Define and govern requirements
- Manage risk and quality
- Efficiently deploy software
- Comply with industry regulations

## Innovation at Scale

Whether your team follows an Agile, Waterfall, Scrum or Hybrid process, Codebeamer has you covered. Visualize and manage tasks with Kanban boards. Analyze team performance using burndown and KPI charts. Or create custom team metrics to keep everyone informed, aligned, and focused on the highest priorities.

## Domain Knowledge Out-of-the-Box

Codebeamer excels at supporting industries with big ideas and stringent regulatory requirements: automotive, aerospace and defense, and life sciences. Pre-defined yet flexible templates built with deep knowledge of your industry allow teams to accelerate success and adopt industry-specific best practices.

## By the Leader in Digital Transformation

PTC is the global leader in software solutions that help product companies digitally transform how they engineer, manufacture, and service the world's products. PTC is dedicated to continuously improving its technologies to help customers thrive today and in the future. Codebeamer works independently or in concert with PTC's award-winning CAD, PLM, ALM, IoT, AR and SLM technologies to establish a digital thread that spans the entire product lifecycle.

For more information, please visit [ptc.com/codebeamer](https://ptc.com/codebeamer)

# About PTC Modeler

---

**PTC Modeler is a model-based systems engineering (MBSE) tool that helps organizations design and manage complex systems across multiple domains. It enables teams to collaborate effectively, ensuring consistency, traceability, and efficiency throughout the development lifecycle. It empowers teams to:**

- **Improve collaboration**
- **Enhance traceability & compliance**
- **Accelerate development**

## Visualization & Decision Making

PTC Modeler enhances visualization and decision-making by providing rich graphical representations of system architectures, interactions, and dependencies. Through SysML and UML diagrams, your teams can visually map out complex relationships, making it easier to identify inefficiencies, gaps, or potential design flaws early in development. These models serve as a common language between engineers, business analysts, and stakeholders, ensuring alignment and reducing misinterpretations. Additionally, real-time model validation and impact analysis tools allow teams to simulate changes, assess risks, and make informed decisions before committing to costly modifications.

## Scalability for Complex Systems

PTC Modeler supports large-scale, multi-disciplinary systems by providing a structured, scalable modeling environment. It supports hierarchical decomposition, modular design, and cross-domain integration, making it easier for your teams to manage interdependencies across software, hardware, and mechanical components. With its centralized repository and version control, it enables seamless collaboration among distributed teams while maintaining data integrity. This scalability ensures that as your projects grow in size and complexity, your organization can maintain clarity, consistency, and efficiency without overwhelming your development processes.

For more information, please visit [ptc.com/ptc-modeler](https://ptc.com/ptc-modeler)



# About Windchill PLM Software

---

**Windchill is a robust Product Lifecycle Management (PLM) solution that helps organizations manage product data, streamline workflows, and enhance collaboration across the entire product development lifecycle. It enables teams to efficiently control and track product information, from design and manufacturing to service and beyond. It empowers teams to:**

- **Unified Product Data Management**
- **Seamless collaboration**
- **Enhanced compliance & change control**

## Accelerate Time-to-Market

Windchill PLM speeds up time to market by centralizing product data, streamlining collaboration, and automating workflows. Teams can work with real-time information, reducing delays from outdated data and enabling faster decision-making and iteration, leading to quicker product launches.

## Reduce Costs & Errors

Windchill reduces costs and errors by ensuring data consistency, automating compliance, and improving change management. It minimizes rework and defects by providing a single source of truth, preventing costly mistakes, and optimizing designs throughout the product lifecycle.

For more information, please visit [ptc.com/windchill](https://ptc.com/windchill)



121 Seaport Blvd, Boston, MA 02210 : [ptc.com](https://www.ptc.com)

---

© 2025, PTC Inc. All rights reserved. Information described herein is furnished for informational use only, is subject to change without notice, and should not be taken as a guarantee, commitment, condition or offer by PTC. PTC, the PTC logo, and all other PTC product names and logos are trademarks or registered trademarks of PTC and/or its subsidiaries in the United States and other countries. All other product or company names are property of their respective owners.

588050\_top\_five\_use\_cases\_ALM\_PLM\_02\_2025