



Modernizing PLM Testing Frameworks: From Manual to Machine

eBook

 KEYSIGHT

Contents



CHAPTER 1

Introduction

[GO TO INTRODUCTION >](#)



CHAPTER 2

Product Lifecycle Management

[GO TO CHAPTER 2 >](#)



CHAPTER 3

PLM and the Role of Testing

[GO TO CHAPTER 3 >](#)



CHAPTER 4

Test Automation

[GO TO CHAPTER 4 >](#)



CHAPTER 5

Eggplant Test - AI Augmented Test Automation

[GO TO CHAPTER 5 >](#)



CHAPTER 6

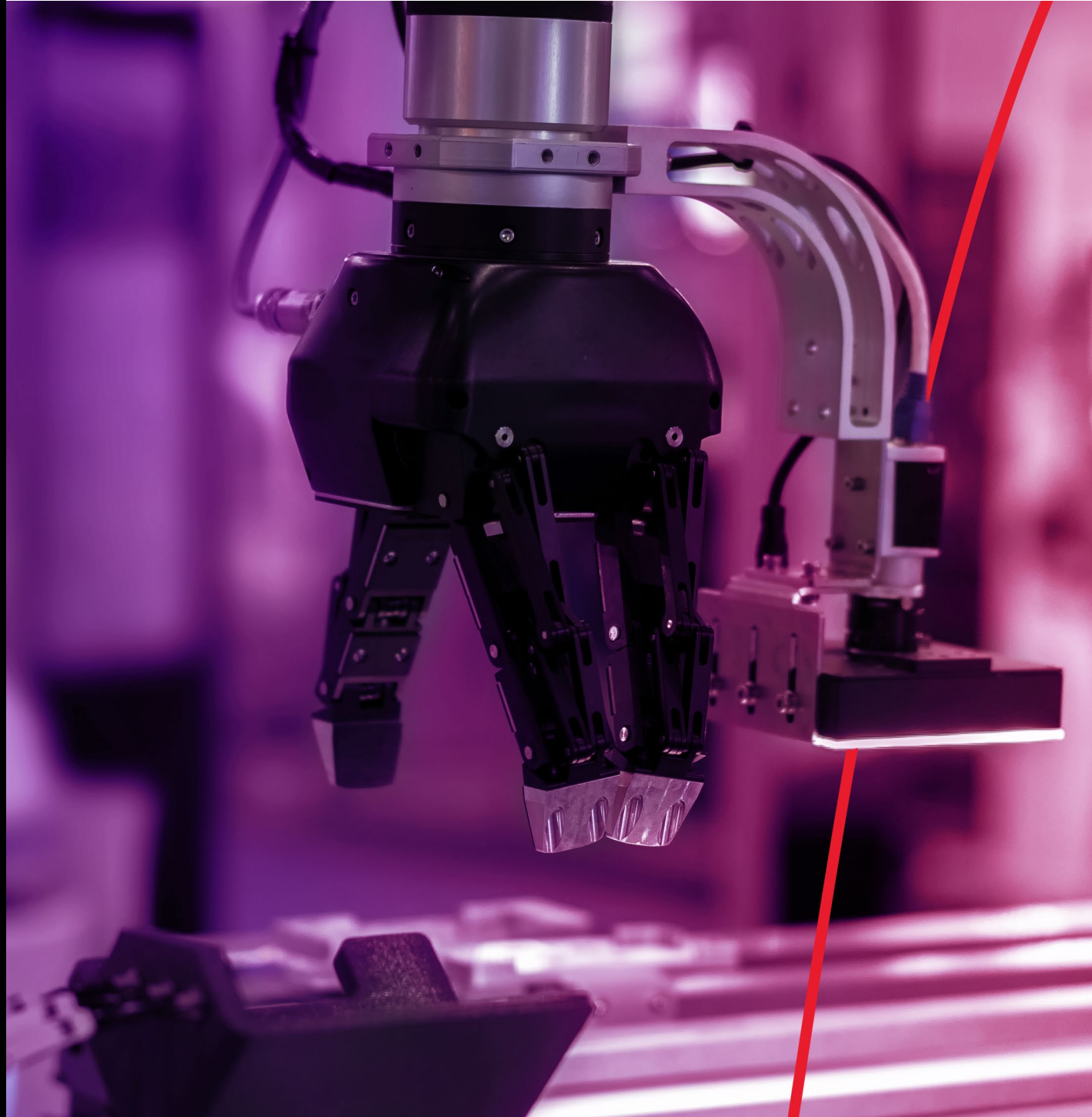
Conclusion

[GO TO CONCLUSION >](#)



Introduction

CHAPTER 1



Introduction

Product Lifecycle Management (PLM) software is critical for OEM's to manage a product's journey from initial concept to end-of-life, encompassing everything from design and manufacturing to support and disposal. The solutions combine people, processes, data and business systems to manage product lifecycle effectively.

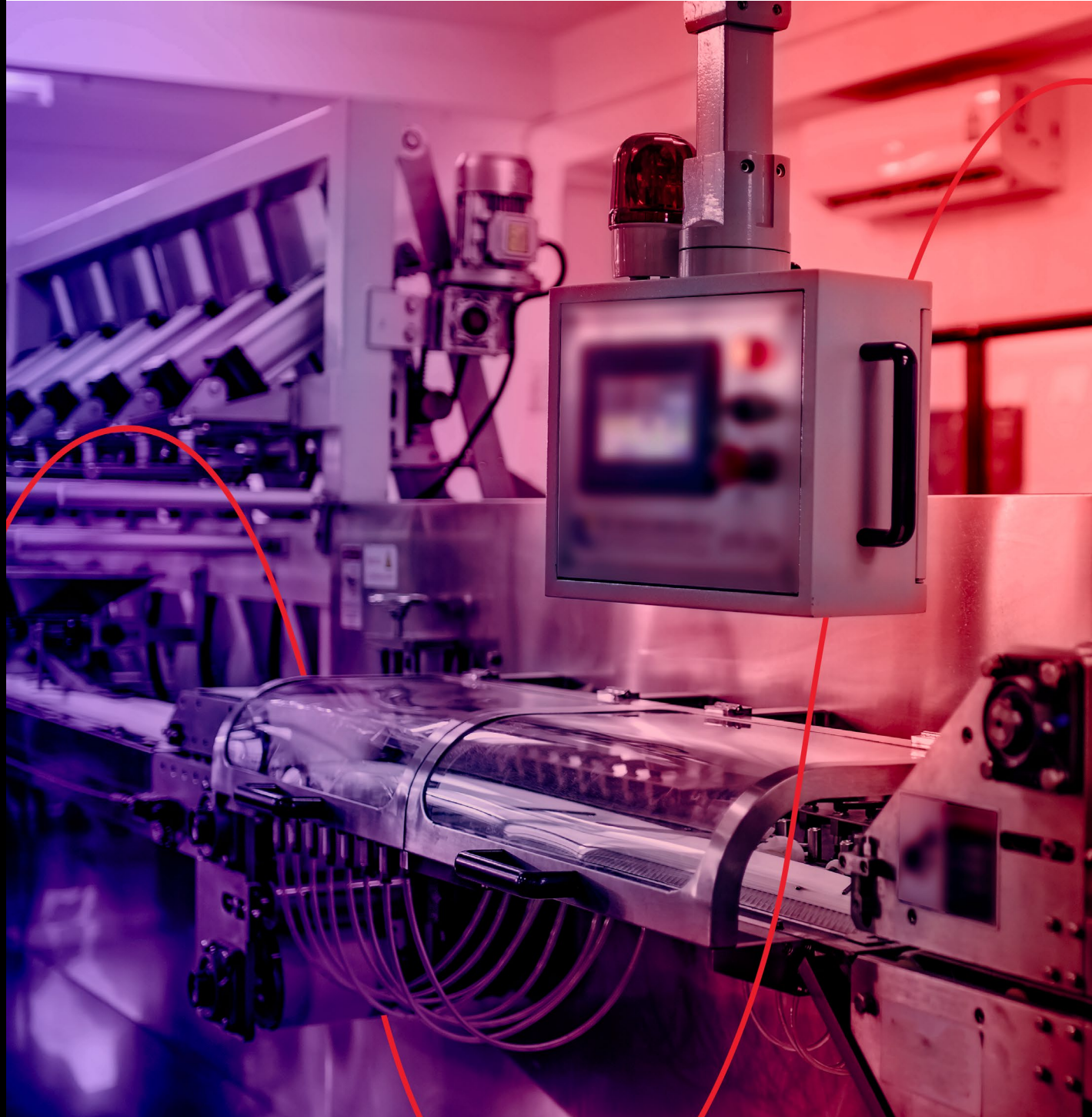
However, despite the inter-dependencies and complexities, manual software testing strategies dominate. This ebook explores the limitations of manual testing and the benefits of adopting automated frameworks, and the business case for harnessing AI augmented test automation in today's dynamic market landscape.





Product Lifecycle Management

CHAPTER 2



Product Lifecycle Management

Gartner defines PLM as a philosophy, process, and discipline supported by software for managing products through their lifecycle stages, from concept through retirement. The strategic framework integrates people, processes, and technology to manage a product's journey from concept through design, manufacturing, service, and eventual disposal. By connecting engineering, planning, purchasing, and manufacturing functions, PLM systems help companies organize, control, and process vast amounts of data and streamline workflows.

The global market for PLM is expanding rapidly and is expected to reach over **\$80 billion by 2034**. The software is the digital backbone of industries spanning manufacturing, aerospace and defense, marine and shipbuilding, life sciences, and healthcare.

Key components of PLM systems include:

- **Data management:** Centralizes product information with version control and traceability
- **Collaboration:** Enables cross-functional communication, breaking departmental silos
- **Workflow management:** Automates processes with defined approvals and tasks
- **Compliance:** Embeds regulatory requirements into development processes



PLM software providers

PLM software continues to mature with a handful of dominant providers, including Aras with its Innovator solution, Siemens Digital Industries with its Teamcenter/Xcelerator portfolio, Dassault Systèmes and its 3DEXPERIENCE platform, and PTC with its Windchill PLM solution.

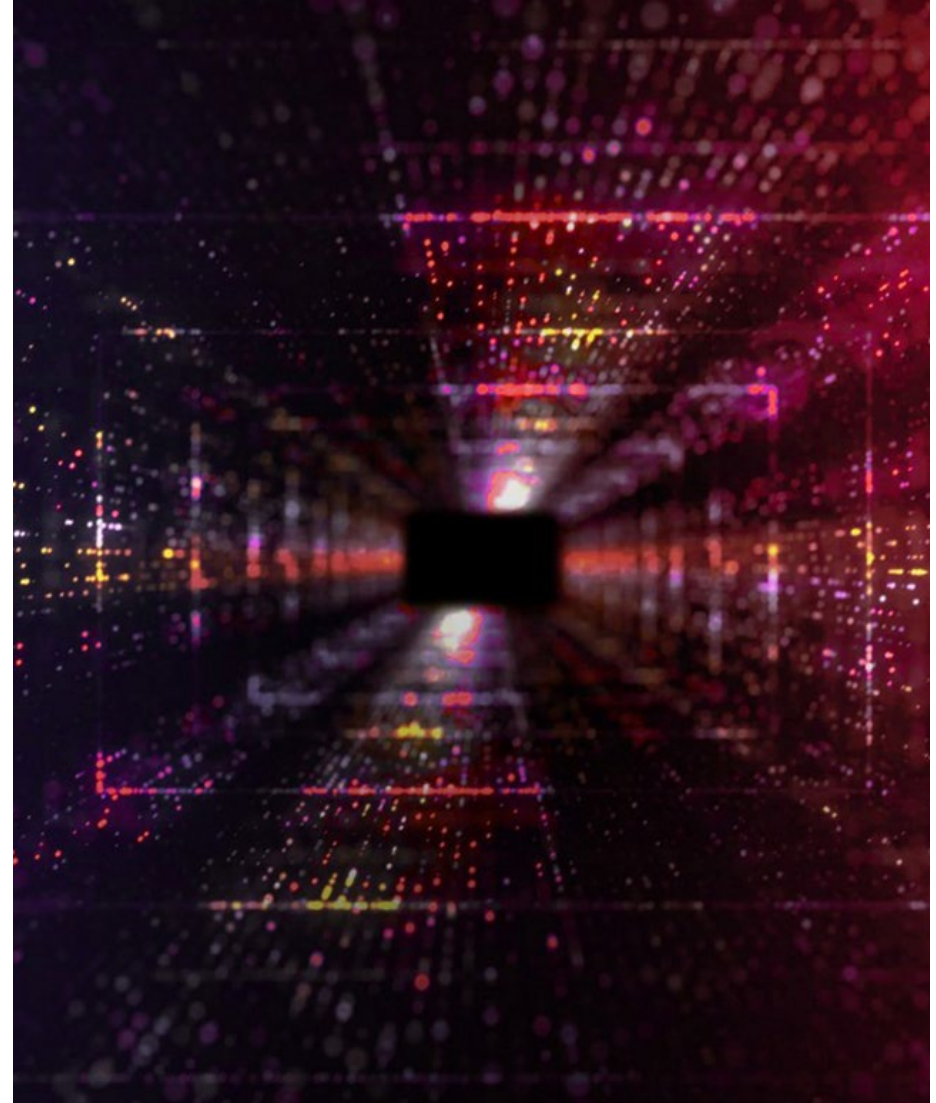
Other established vendors with PLM offerings include Autodesk Vault PLM, IBM Engineering Lifecycle Management, Propel PLM, Oracle Fusion Cloud Product Lifecycle Management, SAP Product Lifecycle Management, and Synopsys PLM.

Emerging companies such as OpenBOM, Propel, and Duro offer cloudbased PLM platforms. Another trend is companies focusing on vertical markets like Backbone PLM, Centric Software, and Coats Digital.

The above reflects the evolving PLM software market and how these solutions are shifting from an engineering tool to an enterprise-wide strategic platform. As product complexity increases, sophisticated PLM solutions are required to manage the data.

Innovative technologies like AI, IoT, and digital twins are reshaping capabilities by enabling real-time monitoring, predictive analytics, and enhanced decision-making throughout the product lifecycle.

Sustainability is now critical, with companies embracing circular economy principles. This requires PLM to support robust lifecycle tracking and environmental impact analysis.



Additionally, digital transformation efforts are accelerating investment in advanced platforms to enhance efficiency and compliance. As PLM software becomes more interconnected, robust testing is essential to validate system accuracy, ensure seamless integration with complex systems like CAD and ERP, and maintain performance during upgrades.



PLM and the Role of Testing

CHAPTER 3



PLM and the Role of Testing

PLM software is critical to orchestrate complex product development. However, there are significant hurdles to overcome in creating robust, scalable, and reliable environments.

Typical pain points span:

- **Implementation complexity:** PLM deployments are inherently complex, requiring customization to align with industry-specific workflows and legacy systems. The time-consuming nature of implementing the software can strain IT resources.
- **Upgrades and migration:** Upgrades and migrations are necessary to add features and bolster security. However, they introduce risk as they can disrupt operations, create inconsistencies, and require validation of processes.
- **Integration:** Product development relies on a diverse ecosystem of CAD, ERP, and MES tools. Ensuring a seamless integration with PLM is challenging, and can result in data silos, synchronization problems, and increased maintenance costs.
- **Digital mockups and visualization:** Without real-time visibility into the performance of PLM systems, it's difficult to proactively identify bottlenecks, predict failures, or optimize resources.

The testing imperative

Testing is critical to overcome the issues outlined, which necessitates navigating PLM complexity.

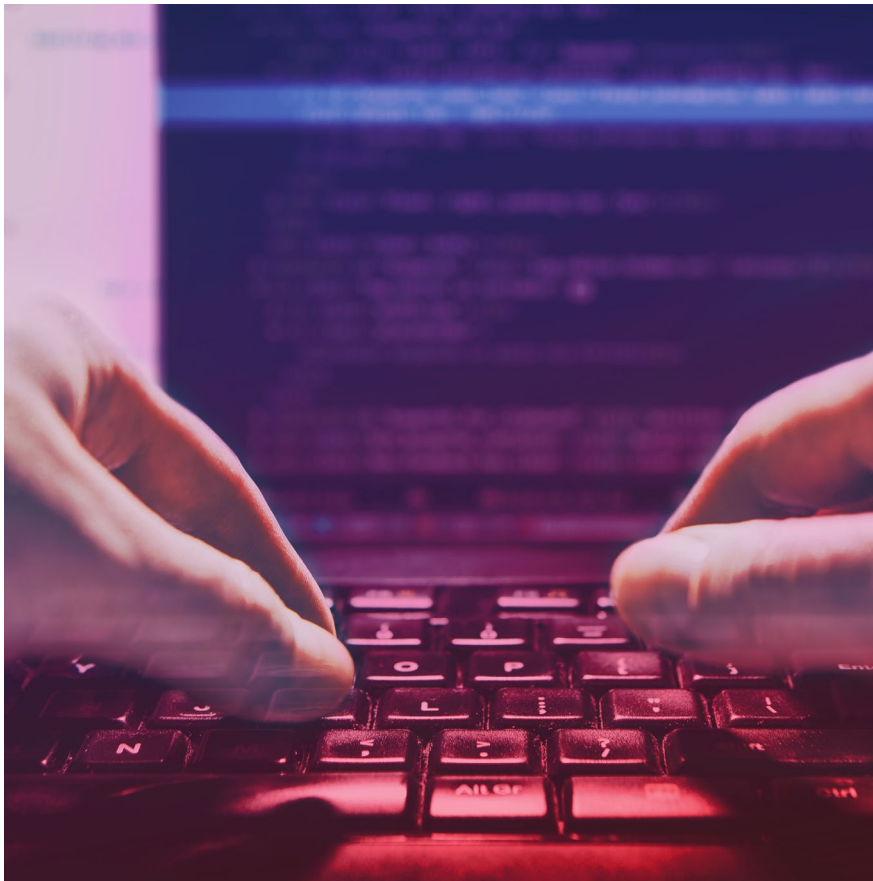
Here are some of the complex scenarios for testing:

- **Graphical interfaces and specialized applications:** PLM systems incorporate interactive GUIs and domain applications that require more than API or backend validation.
- **Data workflows:** The large volume of structured and unstructured information in CAD files, BOMs, and compliance documents necessitates robust data verification and regression testing.
- **Customization and configuration:** Every PLM deployment is customized and, as a result, requires continuous validation of the tailored workflows, business rules, and user permissions.
- **CAD and non-CAD data validation:** With CAD and non-CAD systems, ensuring accurate data exchange is vital for maintaining product integrity and supporting downstream processes.

Testing PLM systems is essential to address these problems, mitigate risk, accelerate innovation, and ensure that software investments deliver.

The state of manual testing

Despite the sophistication of PLM systems, testing strategies remain predominantly manual. This reliance is concerning given PLM's scale and intricacies, coupled with the fact that manual strategies are well established as slow, error-prone, and resource-intensive. All of these factors impact system reliability, speed, and business competitiveness.



Manual approaches involve step-by-step validation of workflows, data structures, and integrations. This requires deep domain expertise and coordination. As the complexity and reach of PLM systems expand, so does the data volume from sources like CAD files, compliance documentation, and intricate product hierarchies.

Manual frameworks struggle to keep pace with the speed and variability of changes including frequent system updates, customizations, and integrations with other enterprise tools.

Engineering.com carried out a study exploring the challenges of PLM system testing. It found that only 15% of respondents deploying manual frameworks believe their process is well-defined and running smoothly. This highlights that the vast majority of organizations lack confidence in their current practices. The problems cited span unclear protocols, inconsistent execution, and frequent post-deployment defects. These issues are amplified during upgrades or migrations, where manual frameworks frequently fail to find errors.

As PLM systems grow more interconnected, with an average of 5+ integrations, relying on manual validation of data flows and business logic is extremely challenging. This results in a backlog of untested changes that increase risk and drive-up maintenance costs. Frustration is growing among QA managers, test leads, and PLM engineers, with strategies that provide limited coverage and delay releases.

Manual testing limitations with PLM

Key survey findings include:

Manual dominates

As outlined despite innovations, when it comes to PLM, frameworks are predominantly manual. Almost 60% of teams rely on this approach, which contributes to slower release cycles and increased risk of errors. These strategies are also resource-intensive, often requiring significant coordination across teams and deep domain knowledge, which limits scalability and agility.

Testing strategy	Percent
Manually	57%
Internally developed automation frameworks	33%
Open-source automation frameworks	24%
Commercial automation tool	17%
Other	3%

Manual strategies cannot keep pace with the scale of modern PLM environments. With products often containing 10,000+ components and intricate relationships, comprehensive validation with a manual framework is almost impossible.



Low confidence test execution results

PLM testing challenges	Manual approaches
Modeling and simulation pre/post-processing	88%
Security concerns	75%
Cost of update	82%
Compliance and auditing	75%
End-to-end testing	88%
Complex GUIs	84%
Time needed to update	78%
Defect detection	87%
Reducing tools/frameworks required for testing	86%
Test case management	80%
Geometry manipulation	77%
Critical document upload and comparison	72%
Reporting	72%

Scalability and agility aren’t the only concerns; the study also uncovered other areas in which manual strategies pose challenges. The data demonstrates that over 70% of teams are encountering issues in each category, with end-to-end testing, defect detection, modeling, and simulation the areas of greatest concern. These problems are driving up dissatisfaction with manual capabilities, as highlighted in Chart X2.

Testing capabilities

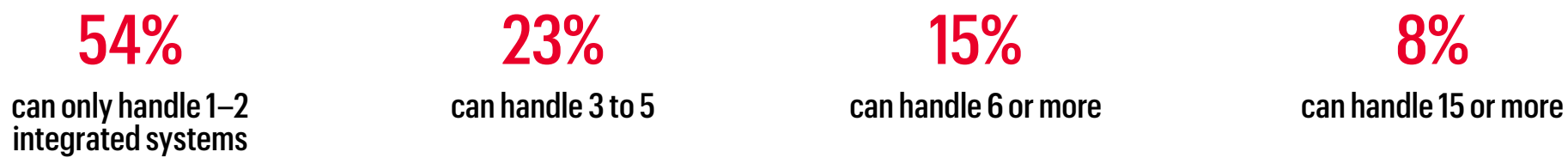
Dissatisfied

Ability to meet all test requirements before release date	60%
Scalability of current testing framework	63%
Data-driven testing	63%
Integration with existing tech stack	60%
Ability to automate workflows to increase job satisfaction and productivity	63%
Generation of testing analysis data	55%

The survey underscores that in every category, the majority of teams are unhappy with the results. 63% are dissatisfied with the ability to automate workflows, reflecting the tedious and time-consuming nature of manual strategies.

Limited coverage with complex systems

Integration testing is a significant challenge as PLM software connects to multiple enterprise systems. This makes manual validation of data flows and business logic extremely complex. The survey identified that 88% of teams find end-to-end testing challenging, leaving complex multi-system scenarios under-evaluated. Further investigation found that with manual approaches. The limited coverage means that bugs remain hidden in untested, integrated workflows. In contrast, those teams utilizing automated tools are able to increase the number of systems tested by 50%.



Delays the norm

Testing time impacts the speed of release, with 92% of organizations postponing or canceling releases because manual validation wasn't complete. Eighty-five percent of teams can only manage at most two major releases per year, with 81% shipping four or fewer minor patch releases annually; just over half (53%) only do 1–2.

This approach means that it takes longer to find and fix bugs. As changes accumulate between releases, each update is riskier to test. And if the slow release schedule delays new features or fixes, this creates a competitive disadvantage, which erodes confidence.

Regression testing is another PLM bottleneck, with 83% of all manual efforts focused on this task, limiting the amount of time for exploratory testing. The process is laborious, with 54% of teams needing at least one week for a complete pass and 40% requiring over two. This prevents skilled resources from focusing on innovation.

Testing teams frustrated

Unsurprisingly, teams are frustrated with the inadequacies of manual frameworks, with 85% stating their current strategy needs attention or improvement. The missed deadlines and coverage gaps reflect a broken process that fails QA's quality mission and negatively impacts morale.

The path forward for PLM

The limitations with manual approaches are not just technical; they also present strategic constraints that impact the entire product lifecycle and flexibility to respond to market changes.

As PLM systems are tasked with enhancing agility, mitigating risk, and maximizing business value, it's imperative to modernize testing strategies. If organizations rely on manual frameworks, bottlenecks will continue to endanger schedules, quality, team morale, and business performance.

Addressing the issues uncovered in the Engineering.com report requires a strategic shift toward automation, standardized frameworks, and continuous validation that can keep pace with the breadth and scale of modern PLM systems.





Test Automation

CHAPTER 4



Test Automation

Let's explore this shift further, looking specifically at why intelligent automation is a vital part of PLM system modernization. The approach uses software to automatically execute, verify results, and accelerate manual, repetitive tests without requiring human involvement. This provides numerous benefits, including improved software quality and efficiency, faster release times, and enhanced user experience. The Engineering.com report found that organizations deploying automated frameworks deliver an average of 3 major releases per year, compared to 1.4 with manual approaches

Along with faster release cycles, consistency and repeatability are other benefits. Tests can be created once and run repetitively, which is ideal for regression, functional, and performance evaluation. In addition, the approach delivers consistency and repeatability by ensuring these tests are executed the same way every time—eliminating human error.

Automation tools are not created equal

While these benefits are significant, it's important to recognize that not all tools are created equal. There is a wide range of solutions, many of which are failing to deliver value. Just 15% of automation adopters from the study are “very satisfied” with their tool's overall reliability, highlighting that much work remains.

The frustrations cited include:

- **Tool proliferation:** Seventy-six percent of teams struggle to reduce the number of tools they use, with separate frameworks typically required for web UIs, desktop, CAD applications, mobile apps, and other operating systems.
- **High skill barrier:** Because traditional automation requires deep programming expertise, manual testers and domain experts are unable to author or update tests. This quickly leads to a backlog, which impedes more widespread automation efforts.
- **Fragmented environment:** Many teams lack a unified, resilient platform, which leaves integration gaps and prohibits true end-to-end coverage.



Basic automation tools are particularly ill-suited for PLM testing, with challenges including:

Coverage gaps: First-generation tools can usually only validate scenarios that have been programmed into the scripts, meaning that possible user paths, edge cases, and configuration combinations may remain untested.

Testing the user experience: Most tools focus on individual components or functions, rather than the user experience across multiple modules, devices, or integrations. This is a significant issue for PLM, where workflows are interconnected and span various roles and systems.

Lack of intelligence: Because basic tools only test what they have been programmed to check, human involvement is still required for exploratory, usability, and other intelligent testing.

Open-source limitations

Some companies have turned to open-source solutions to address the challenges with basic test automation tools. Open-source certainly offers some benefits compared to the latter—better scalability and the ability to support more frequent updates, for example.

For PLM testing specifically, however, open-source falls woefully short. Key limitations include:

- **Web-only automation:** Most open-source tools were designed for browser based testing, offering limited automation support for thick client applications like CAD simulations, Windows applications, or integrations with the various tools that comprise complex PLM systems.
- **Issues with dynamic and custom UI elements:** Open-source solutions have difficulty evaluating dynamic interfaces, custom objects, and similar standard PLM software features. Many tools also offer no support for 2D and 3D graphical testing, which impedes efforts to validate the full user experience.
- **No comprehensive approach:** Because they often require integrations with other third-party tools for comprehensive testing, open source solutions can not support end to end scenarios traversing through multiple applications with different user interfaces.
- **Programming expertise required:** The majority of open-source tools require programming knowledge, particularly for advanced automation and customization.

No match for PLM complexity

It's clear that basic automation tools and open-source solutions cannot address the myriad testing imperatives of PLM software. Not only do these failures impact productivity and frustrate teams, they also increase risk, stymie innovation, and impede competitive advantage.



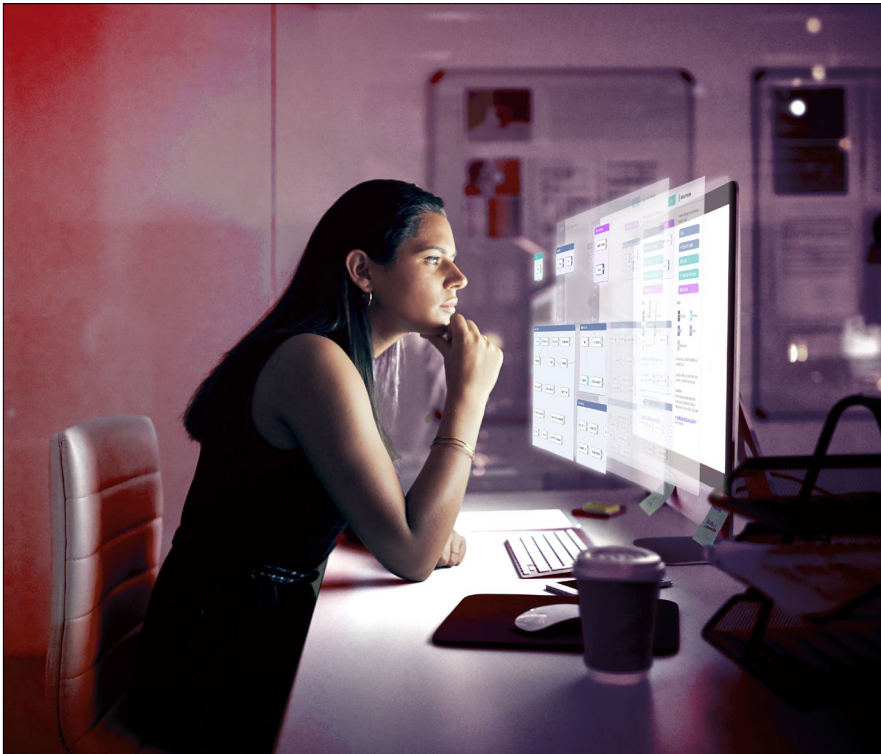
Eggplant Test – AI Augmented Test Automation

CHAPTER 5



Eggplant Test – AI Augmented Test Automation

Keysight's Eggplant Test overcomes the limitations of other industry solutions, enabling companies to maximize the value of test automation and ensure seamless PLM validation across CAD, MES, ERP, and other systems.



The AI-augmented, model-based platform delivers comprehensive and intelligent software testing at scale. Utilizing a digital twin approach, Eggplant builds a model of the system under test and deploys AI to interact with applications like a real user—without requiring access to the source code.

Through computer vision and Optical Character Recognition (OCR), Eggplant can test what customers actually see and validate the user experience across any device, platform, or operating system.

The scriptless, low-code interface allows non-technical users to create and maintain tests, removing the skills barrier with legacy automation tools. This reduces maintenance costs and accelerates release cycles.

The platform integrates seamlessly with CI/CD pipelines, supports exploratory and continuous testing, and provides ongoing insights that improve application quality and reliability. AI, and analytics to test and monitor the user experience, Eggplant Test analyzes apps and real data to auto-generate and execute user journeys.

This enables organizations to expand automation beyond execution to include creating test cases from a model of the application. The software uses learning algorithms to select the best paths to optimize tests for coverage and find defects. As it continuously learns, it can predict business impacts, enabling development teams to fix issues before they occur.

Navigating PLM complexity

These capabilities make Eggplant Test uniquely suited for complex PLM software. The platform supports true end-to-end testing of the entire product journey through:

- **Universal compatibility:** Eggplant Test can automate tests on any operating system, browser, device, or application interface, from legacy mainframes to modern cloud-based systems. This is essential for PLM solutions, which typically integrate with numerous enterprise tools and must function seamlessly.
 - **End-to-end validation:** Eggplant validates the entire PLM workflow—including CAD, MES, and ERP system integrations—in a single, automated test. This confirms that product data and processes flow correctly across all connected platforms and identifies any problems before they reach production.
 - **Full PLM UI coverage:** The platform tests the entire PLM user interface—including desktop, web, and mobile—ensuring the validation of all workflows and business processes, not just individual components.
 - **Advanced UI control:** Eggplant enables testers to interact with complex UI elements. The platform uses computer vision to recognize graphics, text, dials, sliders, etc.—even when their positions change or when they are accompanied by dynamic locators, moving 3D images, or widgets. This ensures accuracy even in the most complex and interconnected PLM systems.
- 
- **Visual recognition:** Eggplant automatically scans 2D drawings or 3D models displayed in the PLM interface and accurately recognizes and quantifies specific geometric artifacts within digital product models. This allows testers to confirm the number and placement of holes, slots, and other key product features, and catch any errors early in the development process.
 - **Automation for stability:** As mentioned, a failed PLM rollout or upgrade can impede shipments, incur regulatory violations, and negatively impact the bottom line. By thoroughly evaluating PLM systems, Eggplant prevents ecosystem disruptions, ensures security updates are properly applied, and helps companies avoid the risks and costs associated with failed deployments.

Overcoming traditional barriers

Eggplant eliminates the bottlenecks that have traditionally hampered PLM testing automation. Because the platform does not require source code access, it functions seamlessly even with proprietary systems, frequent UI changes, and complex integrations.

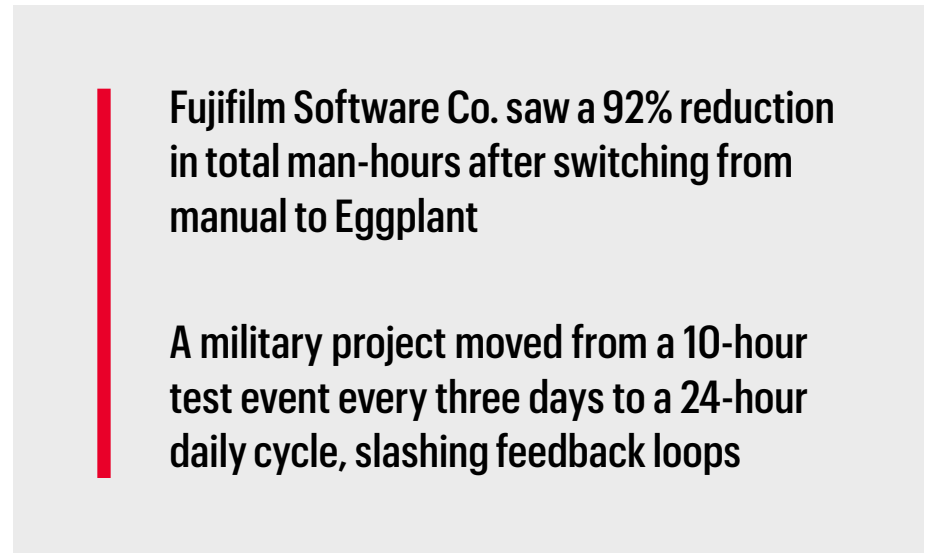
This model-based approach drives numerous benefits, including:

- **Enhanced security:** Minimizes the risk of exposing sensitive or proprietary code. In addition, it limits the attack surface by reducing potential threat avenues introduced during the process.
- **Legacy and third-party systems:** Validates functionality and user experience on these systems, many of which are impossible to test via traditional automation solutions, which need source code access.
- **Broader application and platform coverage:** Tests all PLM systems without needing access to the code.

Accelerating QA cycles

The days of testing being carried out solely by specialist engineers are in the rearview. With complex PLM systems, domain experts and other non-technical users need a seat at the table. Eggplant's intuitive, visual interface democratizes automation, enabling QA analysts, business managers, and other product stakeholders to design tests without deep programming expertise. This expands the tester pool, vastly reduces lead times, accelerates QA cycles, and lays the framework for ongoing enhancements

This graphic shows how Eggplant's AI-augmented automation delivers real-world improvements:



Fujifilm Software Co. saw a 92% reduction in total man-hours after switching from manual to Eggplant

A military project moved from a 10-hour test event every three days to a 24-hour daily cycle, slashing feedback loops



Conclusion

CHAPTER 6



Conclusion

PLM solutions are a critical backbone of digital engineering and manufacturing for managing complex graphical user interfaces, heavy process workflows, seamless process and data integrations. This environment will only grow in complexity in tandem with technology innovations, rendering manual approaches, basic automation, and open-source tools even more ineffective.

Manual frameworks are cumbersome, time-consuming, and incapable of scaling. They simply cannot keep pace with frequent updates, customizations, and integrations, which are the hallmarks of modern PLM. Manual efforts also fail to support continuous delivery and keep product teams bogged down with routine tasks that divert attention from more strategic initiatives.

Legacy automation tools may alleviate some of this work, but they lack the intelligence necessary to drive the business forward. They focus on components rather than the broader experience, often leading to downstream problems. Similarly, open-source solutions cannot deliver the end-to-end testing framework necessary for validating complex PLM software.

And in today's fast-paced market, any bottlenecks and release delays resulting from these limitations can quickly snowball into business-critical issues. That's why it's essential that PLM stakeholders view testing as an innovation accelerator and look for a partner to modernize their approach.

Keysight Eggplant is uniquely suited to PLM software solutions as the platform addresses the intricacies, configurability, and integration challenges that have historically stymied traditional automation solutions. By testing from the user's perspective, Eggplant ensures that the system meets business needs, delivers a high-quality experience, and reduces risk. Its low-code approach empowers a wide variety of product stakeholders to create, maintain, and scale tests across diverse PLM configurations, reducing both manual efforts and maintenance overhead.

This results in faster time-to-market, fewer defects, reduced costs, and productivity improvements with an immediate and long-term reward. Register for a demo today to learn how Eggplant Test can bring these and other benefits to your PLM.



Keysight enables innovators to push the boundaries of engineering by quickly solving design, emulation, and test challenges to create the best product experiences. Start your innovation journey at www.keysight.com.

This information is subject to change without notice. © Keysight Technologies, 2025, Published in USA, August 8, 2025, 7125-1062.EN