



SURVEY REPORT

Complexity Overload and Bottleneck Struggles: The Hidden Costs of Manual PLM Testing

Brought to you by:

 **engineering.com**



The State of PLM Software Testing

As engineering and manufacturing processes grow more complex, the need for efficient, reliable product lifecycle management (PLM) software testing has never been greater. At Engineering.com, we set out to better understand the current challenges and priorities facing engineering professionals in this critical area.

From February 7, 2025, through March 12, 2025, Engineering.com partnered with electronics test and measurement equipment and software manufacturer Keysight Technologies to survey our audience about their experiences with PLM software testing. This report reveals the realities of manual testing, the impact on release schedules and quality, and the growing interest in automation.

We are pleased to share these findings, providing engineering teams with valuable insights to help them navigate the evolving landscape of PLM software testing and to help them make informed decisions for the future.

— Rachel Pasini, *Editor-in-Chief*, **Engineering.com**

A Letter From Keysight

Choosing the right PLM software is only the start—it needs to be customized and integrated to ensure product data integrity across all systems; to do this effectively, these integrations must be rigorously tested.

PLM testing remains hampered by manual efforts, and as the survey responses show, teams are still struggling with complex, integrated environments that bottleneck every release. Regression testing can drag on for days or even weeks, since every update across CAD, ERP, or supply-chain modules must be painstakingly retested by hand.

Manual tests are prone to human error, demand intricate environment configuration, and fail to scale as project scope expands. Add compliance audits and detailed documentation requirements, and test cycles can swell from weeks into months, sapping morale and delaying schedules or feature cuts.

Partnering with Engineering.com to surface these findings underscores the urgent need for an intelligent, model-based automation approach that can compress test cycles into hours, maintain consistent coverage across intricate workflows, and free engineers from manual tasks to high-value work.

— Gareth Smith, *General Manager*, **Keysight Eggplant**

Executive Summary

Product Lifecycle Management Software – Manual Testing

Results from a comprehensive online survey reveal insights from 115 engineering professionals across diverse industries, including energy/utilities, aerospace and automotive. Respondents primarily hold management and engineering roles, with a focus on operations and product lifecycle management (PLM) processes. Representing organizations of varying sizes, most testing teams are small, highlighting the need for efficient tools and streamlined workflows.

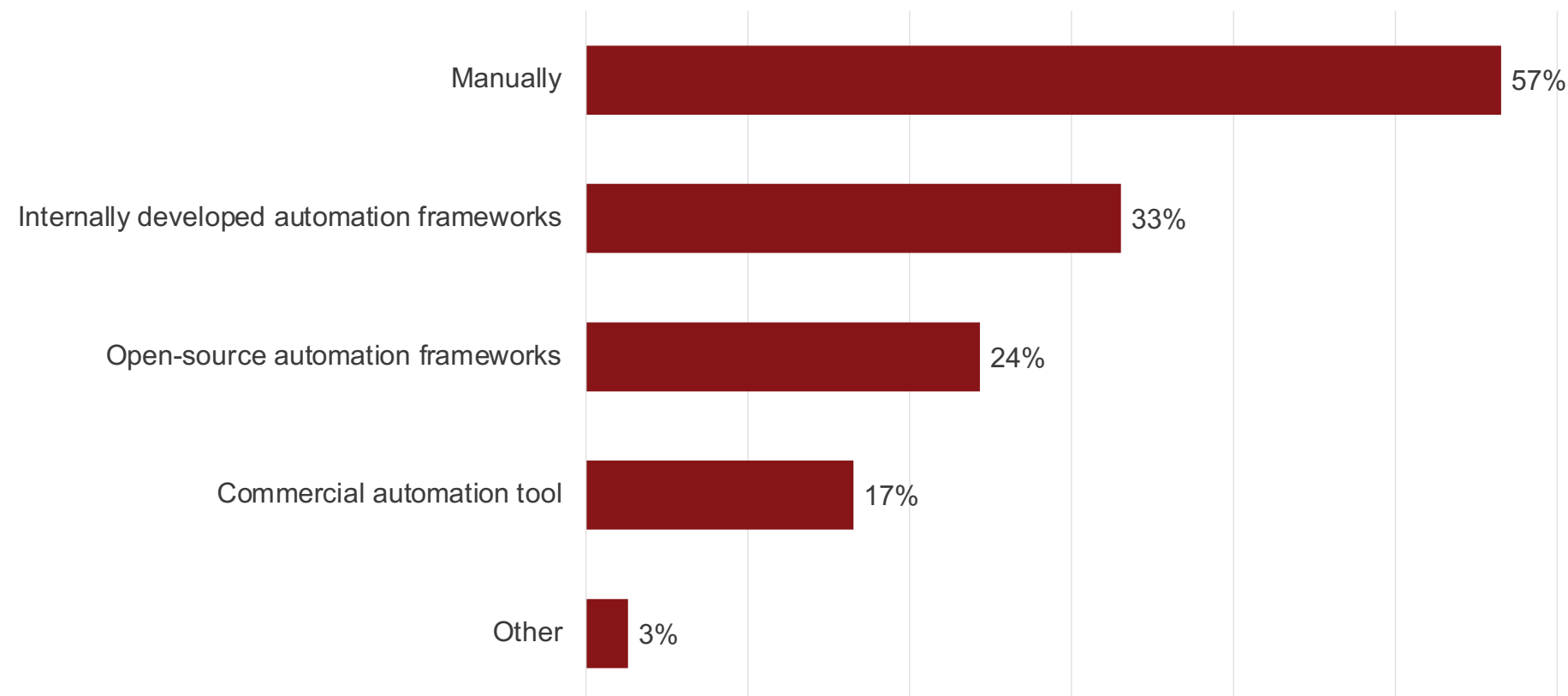
This PLM-focused survey explored three critical areas:

- ① Effectiveness and challenges in PLM testing processes
 - ② Testing methods and integration capability
 - ③ Priorities and satisfaction with testing capabilities
-

Introduction

Product lifecycle management (PLM) software serves as the backbone of complex engineering and manufacturing processes, yet its quality assurance (QA) practices often rely heavily on manual testing. This limited approach struggles to meet the demands of modern development cycles, where speed, quality and scalability are paramount. The survey data highlights critical issues with manual testing, including frequent release delays, limited test coverage and growing frustration among teams. Without significant change, these bottlenecks will continue to jeopardize schedules, product quality and team morale.

How do you test your PLM systems, tools, and software?

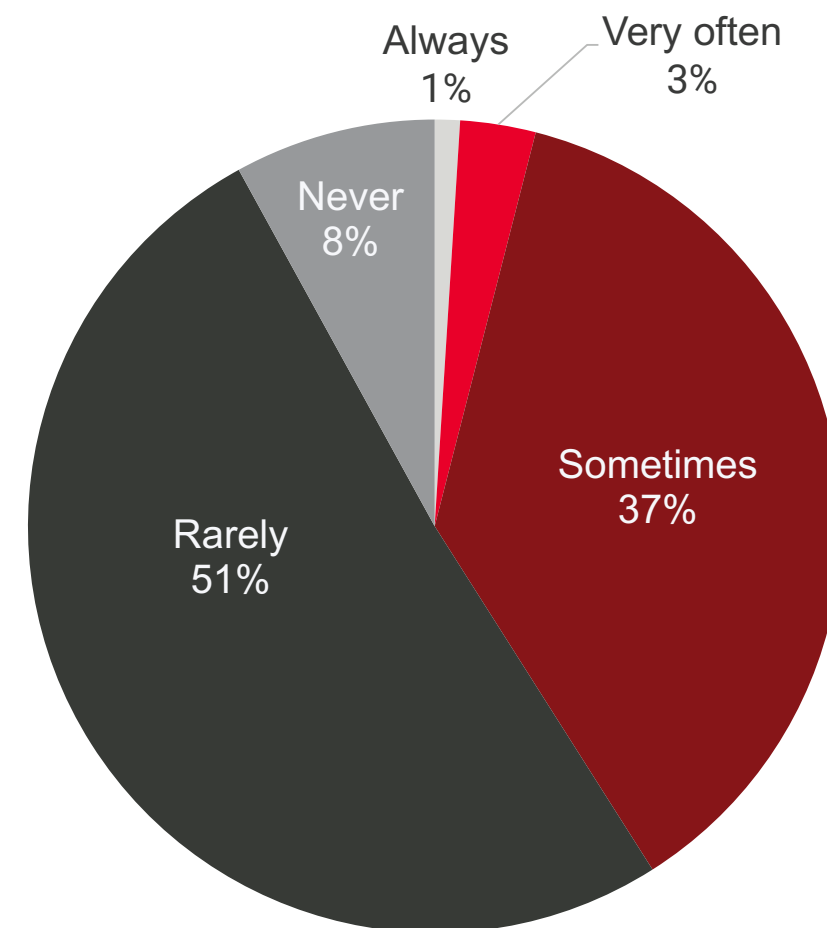


Chronic Release Delays Jeopardize Schedules and Product Quality

Survey results reveal that **92% of teams have postponed or canceled** releases because manual testing was not completed on time. Nearly every team has experienced last-minute delays or cancellations due to manual testing bottlenecks. These delays, which often span days or weeks, erode stakeholder confidence and incur significant opportunity costs.

Manual testing frequently becomes the critical path in release cycles, making missed deadlines a common occurrence. To accommodate these inefficiencies, teams often pad schedules or reduce scope to fit testing into tight timelines, slowing innovation and reducing competitiveness. The consequences of delayed releases are substantial: lost market opportunities, dissatisfied clients and strained team relationships.

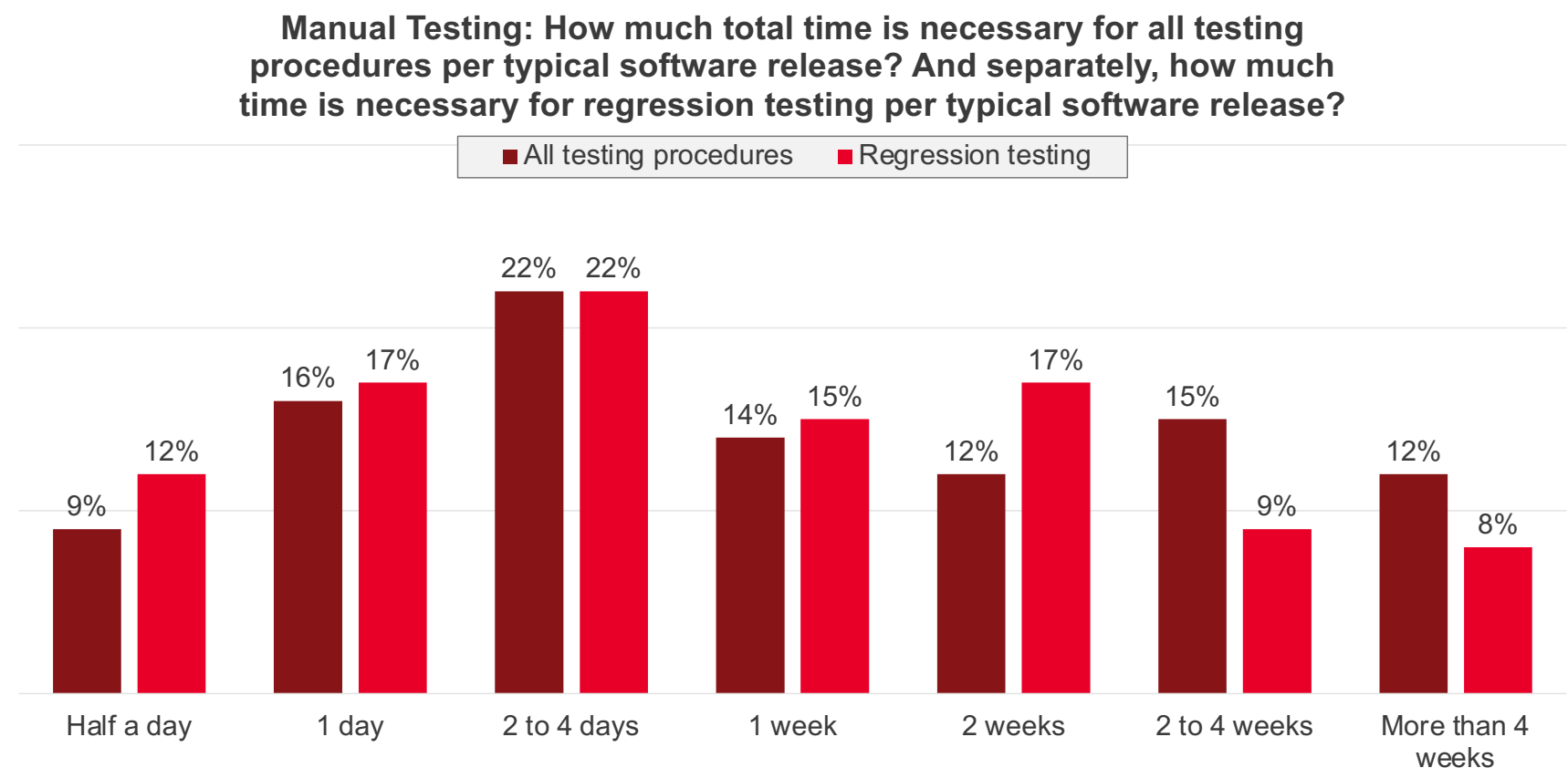
Manual Testing: How often are major software releases delayed or canceled due to critical defects?



Regression Testing Bottlenecks Drain Time and Resources

Regression testing consumes a disproportionate amount of time and resources in PLM software testing. According to the survey, **83% of all testing effort is spent on regression testing**, leaving little bandwidth for new feature development or exploratory tests. Regression suites are particularly large in PLM environments as they must cover core functionality, integration with other systems and compliance requirements.

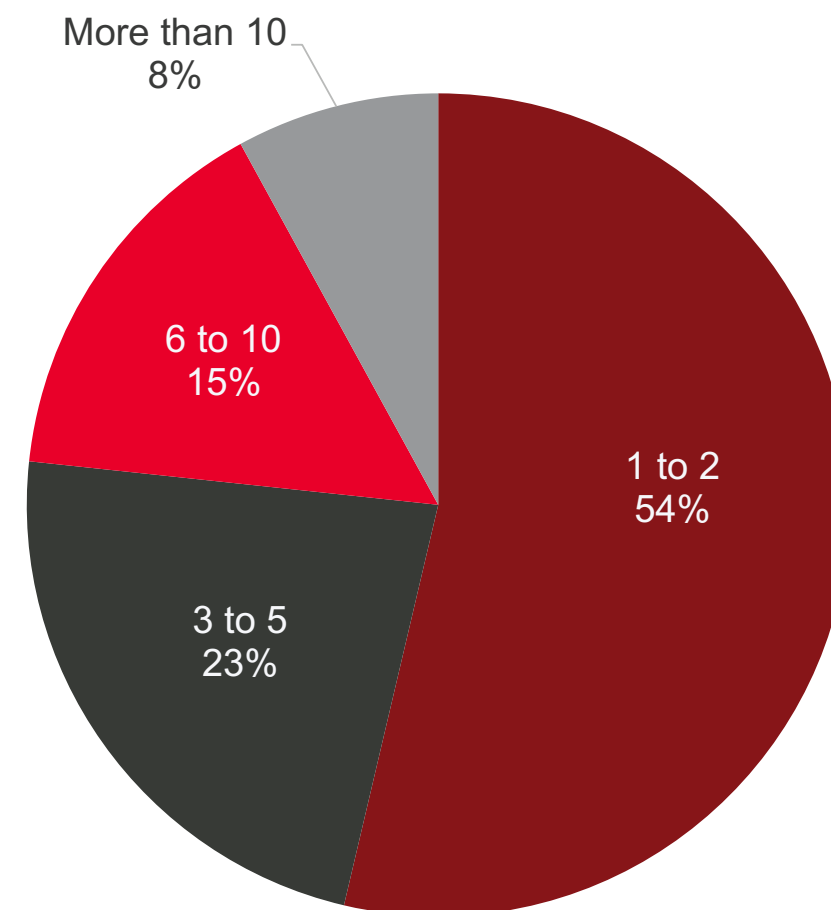
For many teams, regression testing is painfully slow: **66%** require at least one full week to complete a pass, while **34%** need two or more weeks. These prolonged cycles create serial delays and force testers to work late nights and weekends to meet deadlines. Under pressure, some tests may be skipped entirely, increasing the risk of undetected bugs. The heavy focus on regression drains skilled tester bandwidth, stifles innovation, and contributes to low morale across teams.



Hidden Bugs Lurk in Complex Systems Due to Limited Coverage

Modern PLM systems are deeply integrated with tools such as CAD software, ERP systems, supply chain platforms and more. However, manual testing struggles to keep up with this complexity. **Over half (54%) of teams** can only handle 1 to 2 integrated systems in their testing scope; 23% can manage 3 to 5 integrations; 23% six or more.

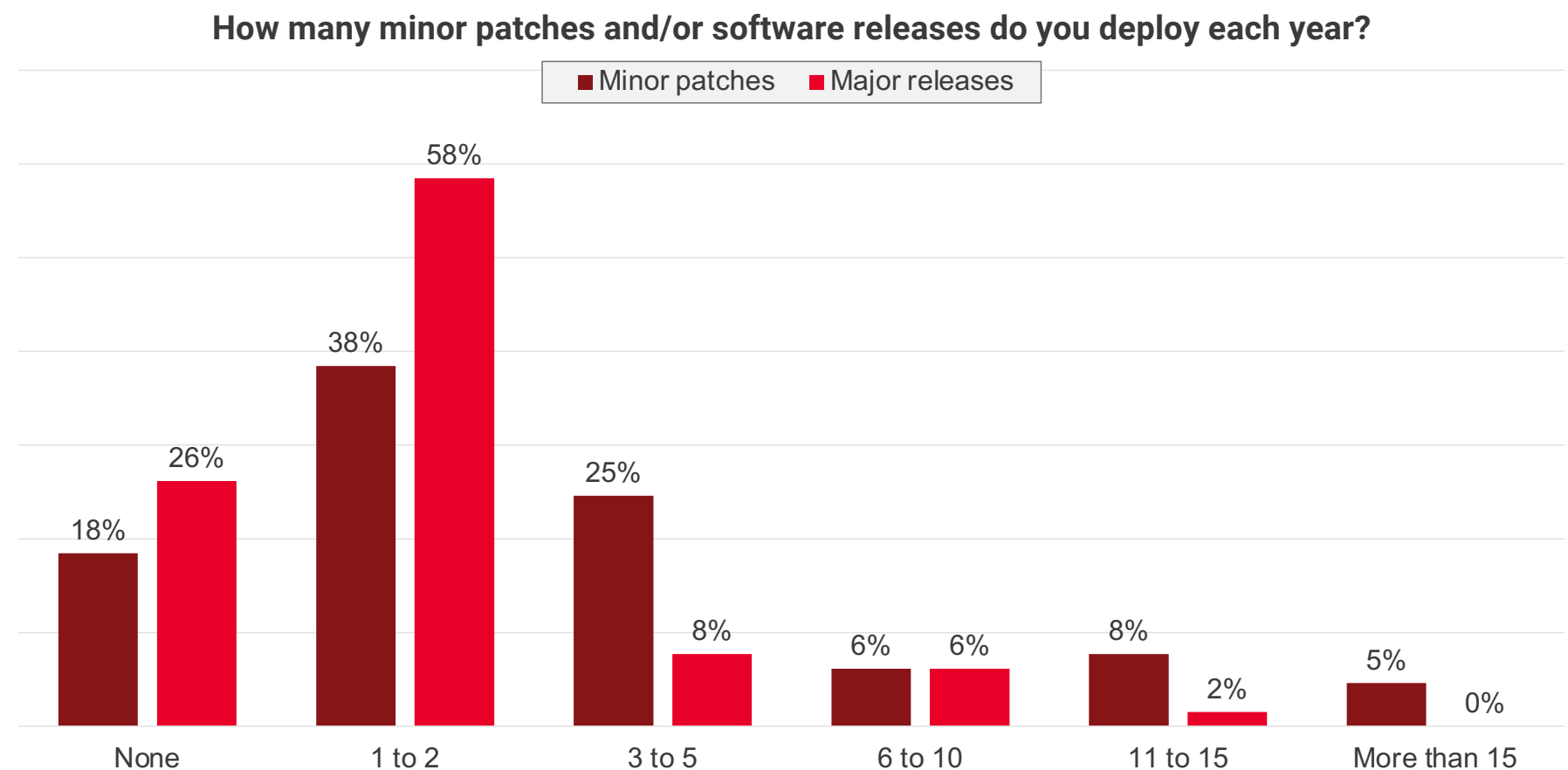
Manual Testing: How many systems are integrated with your PLM tools?



Slow-Release Cadence and Patch Backlogs

Manual testing overhead significantly slows release cadences for PLM software updates. The survey shows that **82% of organizations ship five or fewer minor patch releases annually**; 38% only manage 1 to 2 patches per year. Similarly, **85% of teams** can only deliver at most two major releases annually.

This slow pace delays critical bug fixes for months at a time due to lengthy testing cycles. As changes accumulate between releases, each update becomes riskier to test—especially when integration numbers grow—further compounding delays. This sluggish release cadence puts organizations at a competitive disadvantage when features or fixes are delayed by manual testing inefficiencies.



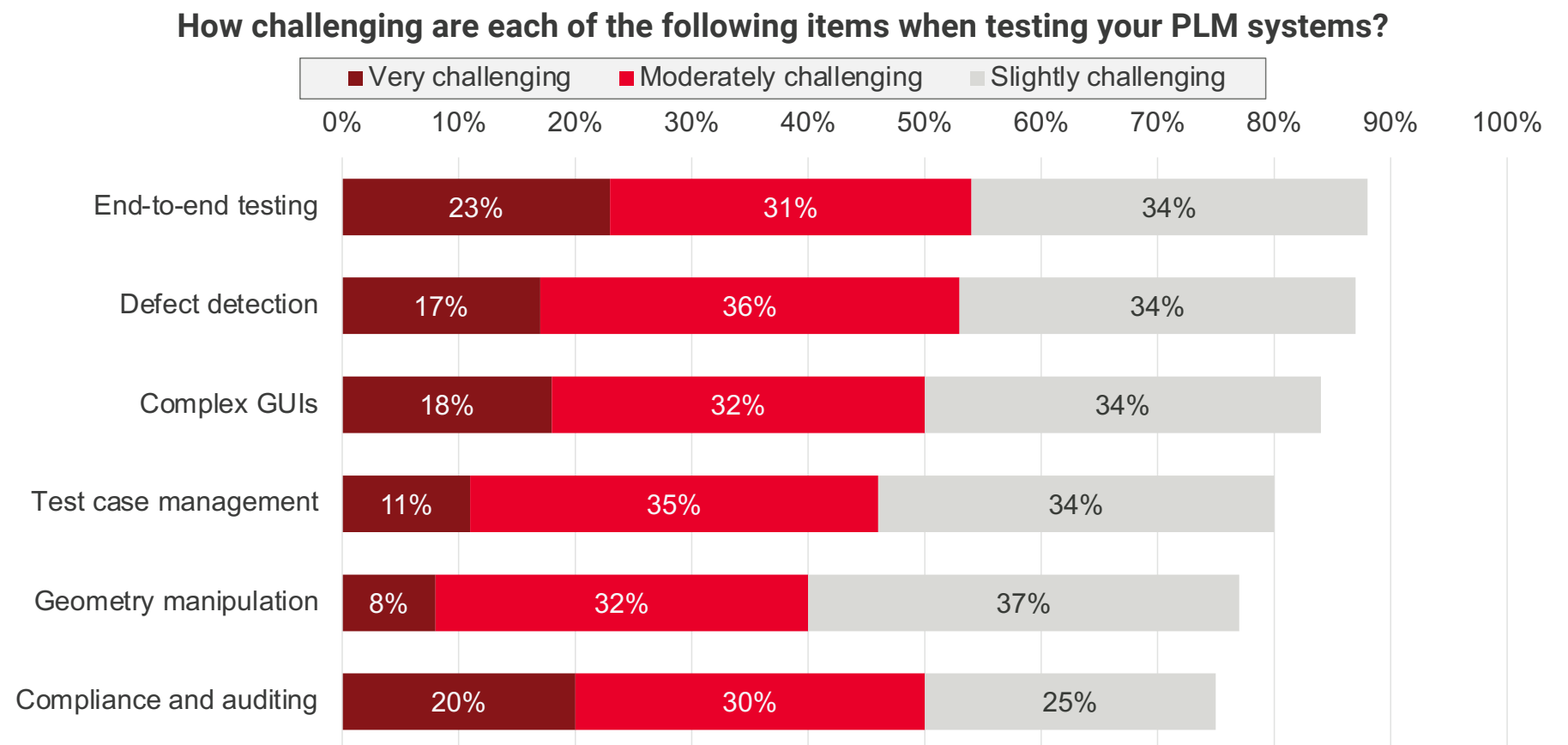
Overwhelming Testing Challenges Overburden Testers

Survey results highlight widespread challenges across nearly every aspect of manual PLM software testing.

For example:

- **88%** struggle with end-to-end testing of complex workflows.
- **87%** find defect detection difficult.
- **84%** cite issues with graphical user interface (GUI) complexity.
- **80%** face challenges managing and maintaining test cases.
- **77%** report burdens from complex data (e.g., 3D models).

Testers face high-difficulty tasks without adequate tools to support them, often operating in “crisis mode.” Compliance demands further exacerbate workloads for **75% of teams**. These pressures lead to burnout, errors in testing processes and high turnover rates among skilled testers.



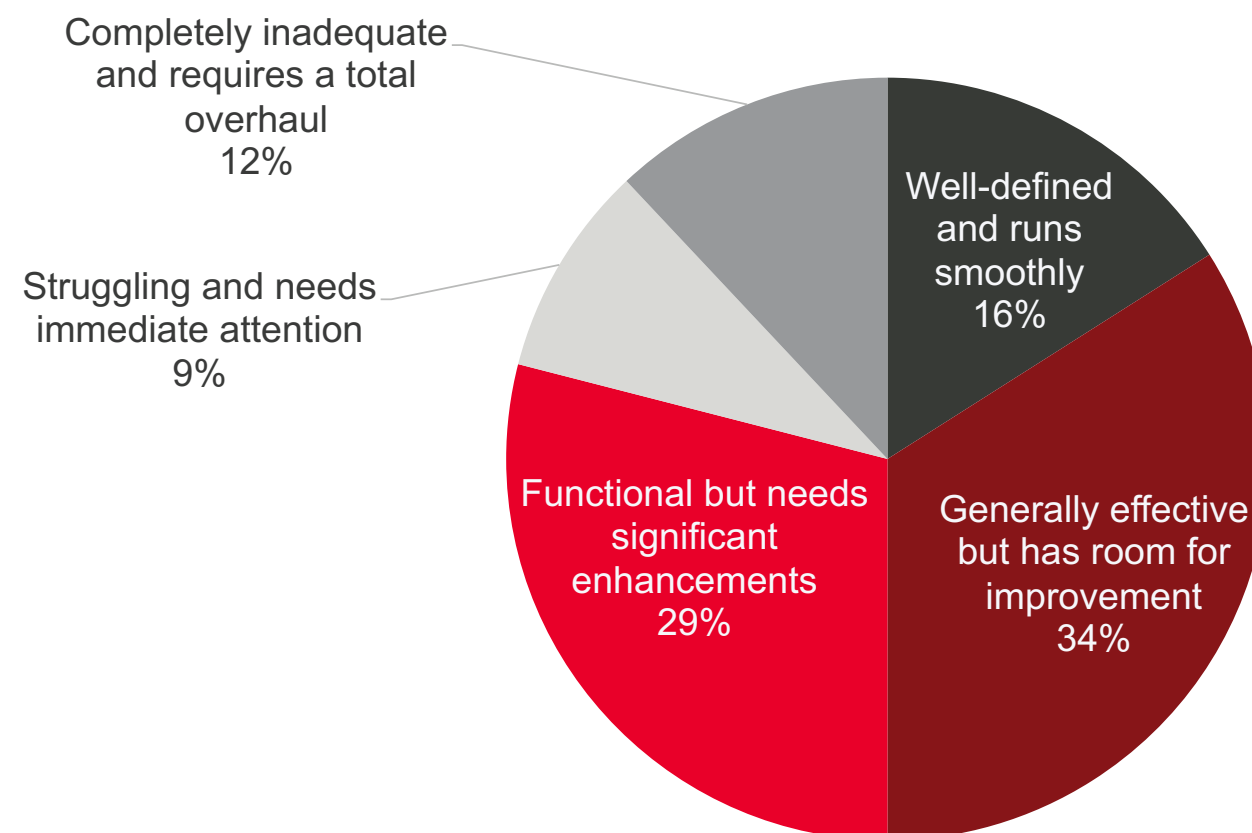
Widespread Dissatisfaction Signals Need for Change

The survey reveals an urgent need for change in PLM software testing practices:

- **84% of respondents** say their current approach "needs attention" or improvement.
- Only **16%** consider their processes well-defined and satisfactory.

Morale is low among testers and managers alike due to repeated missed deadlines and limited test coverage. Teams often operate in constant "damage control" mode—a broken process that fails both QA's mission to ensure quality and the well-being of its team members. High turnover rates and strained relationships further signal that manual testing processes are no longer sustainable.

How would you characterize the current state of your PLM software testing process?



Turning the Tide with AI-Augmented Test Automation

The survey makes clear that manual testing is a critical bottleneck for PLM software teams, resulting in constrained test coverage, chronic release delays, and widespread frustration. Key pain-point statistics include:

- 92% of respondents reported delayed releases due to incomplete manual QA
- 66% require at least one week—and 34% need two or more weeks—for a full regression pass
- 84% struggle with GUI complexity
- 80% cite test-case management as a major challenge
- 88% can only handle 1 to 2 integrated systems manually.

These figures underscore how labor-intensive testing processes prevent teams from keeping pace with fast-moving delivery demands. Intelligent, AI-augmented automation directly addresses these challenges by replacing heavy scripting with model-based, scriptless design. Rather than writing and maintaining extensive code, teams can use low-code tools that automatically adapt to UI changes and manage test data. Full regression suites that once took weeks can now execute in mere hours—making true continuous integration and delivery achievable. Moreover, a single, AI-driven test can span UI, API, desktop, and mobile layers, delivering reliable end-to-end coverage that manual efforts simply cannot match.

By automating repetitive tasks, testers are free to focus on designing strategic scenarios, analyzing results, and expanding test depth. This shift not only alleviates burnout and frustration but also allows teams to ship patches and updates more frequently, shrinking backlogs and eliminating last-minute crunches.

Keysight Eggplant dramatically reduces scripting effort with a low-code, scriptless interface while providing end-to-end coverage across UI, API, desktop, and mobile from a single platform. Its AI-driven models adapt when interface changes, validates product data and process workflows in real time, and unearths hidden defects. By eliminating manual bottlenecks and safeguarding data integrity, Eggplant transforms QA into a strategic asset that delivers faster, higher-quality releases—and greater confidence in every build.

The survey data clearly demonstrates that manual-intensive QA is unsustainable in today's fast-paced engineering environments. AI-augmented test automation offers a clear and urgent path forward, empowering teams to overcome the limitations of manual testing and achieve their quality and delivery goals.



Contact Us

We're happy to hear from readers with questions, comments, concerns and story ideas.

Email: rpasini@wtwhmedia.com

Contact Keysight to learn more:

www.keysight.com/gb/en/contact/eggplant.html

