

State of Test Automation

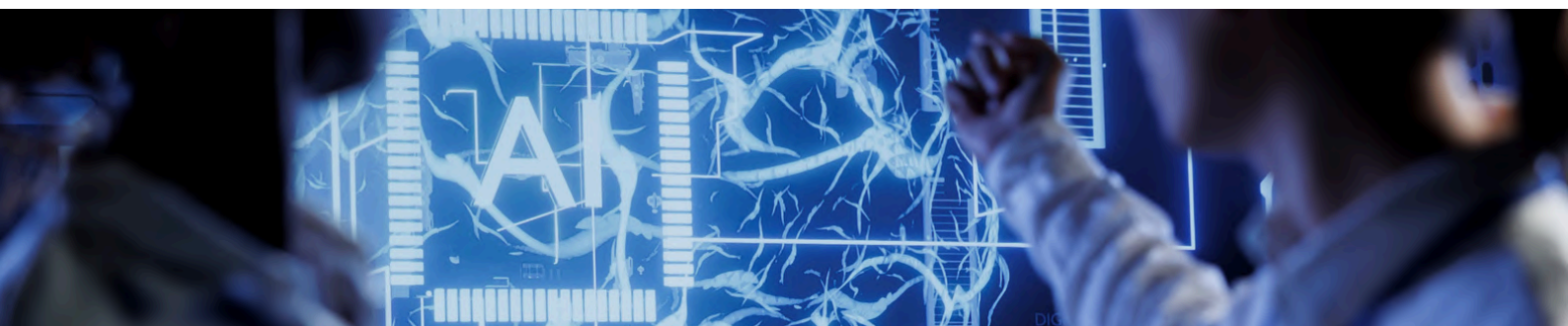
INDUSTRY INSIGHTS - JULY 2025

www.yethi.in



Table Of Contents

Introduction	3
Respondent Chart	4
Key Highlights	5
Section 1: Current QA Challenges & Practices	7
Q1. What's your biggest daily headache as a QA?	
Q2. What do you think is the biggest barrier to test automation adoption?	
Q3. How do you currently manage testing across different platforms (Web, Mobile, API, Desktop)?	
Q4. How often do test automation failures cause delays in your release cycle?	
Q5. What percentage of your total functional test cases are automated?	
Section 2: Tooling Decisions & Preferences	16
Q6. What matters most when choosing a test automation tool?	
Q7. Why do you use different tools for API, Web, Mobile, and Database testing instead of a unified solution?	
Section 3: AI & Codeless Automation	20
Q8. How important is codeless automation for your team?	
Q9. How do you measure the ROI of your test automation efforts?	
Q10. On average, how long does it take you to design test cases with adequate coverage?	
Q11. How confident are you in the accuracy of AI-generated testing outputs (e.g., test cases, scripts, defect reports)?	
Section 4: Future of Testing & Human Roles	26
Q12. How will the role of human testers evolve in the era of fully autonomous test creation and execution?	
Q13. What's the most crucial individual skill for QAs to thrive with AI's rise?	
Q14. Where do you think manual testing is still crucial?	
Conclusion	30



Introduction

For years, QA sat at the tail-end of delivery pipelines, validating what others had already built. That era is long over.

Today, quality has shifted **left, right, and everywhere in between**. What used to be a testing function is now an intelligence function, connecting user behaviour, system reliability, velocity, and business risk.

The reality is simple: you can't scale software without scaling trust. And you can't scale trust without rethinking how quality is engineered.

This report illustrates what that rethink looks like in practice, as experienced by QA analysts, testers, and teams operating in high-stakes environments around the world. The insights are drawn from LinkedIn polls conducted with a total of **2,917** respondents, averaging **208** responses per question.

Teams today are moving fast. They're expected to move faster. And while they know that test automation delivers both speed and accuracy, **trust levels in AI** remain a critical issue.

The rise of AI has added urgency but not complete clarity. Modern leaders aren't asking "**Should we use AI?**" anymore, but "**Can we trust what it gives us?**". That's a fundamentally different question, demanding different skills, different tools, and frankly, a different **mindset**.

What was evident from our data is that most QA teams are still in the middle of this transition. Some are pushing the boundaries, training AI models, automating strategy, and embedding quality deeper into development. Others are still stuck fighting flakiness, tooling friction, or cultural resistance.

But across the board, one shift is undeniably important:

QA is no longer judged only by how many tests are run, but also by how early and intelligently it intervenes. In other words, today, QA at its purest cannot afford to focus exclusively on either a shift-left or shift-right approach. It needs to be embedded throughout the STLC, both by preventive measures and active monitoring. By preventive, we mean measures that focus on preventing bugs. Active monitoring, on the other hand, helps ensure the quick remediation of new bugs in production. This is supported by analytics to predict areas where the occurrence of bugs is highly probable in the foreseeable future.

QA, moving forward, will be about **building and utilizing intelligent systems** to provide a consistent user experience.



Respondent Chart

Key Statistics

2917

Total Responses
Across 14 Questions

208

Average Responses
Per Question

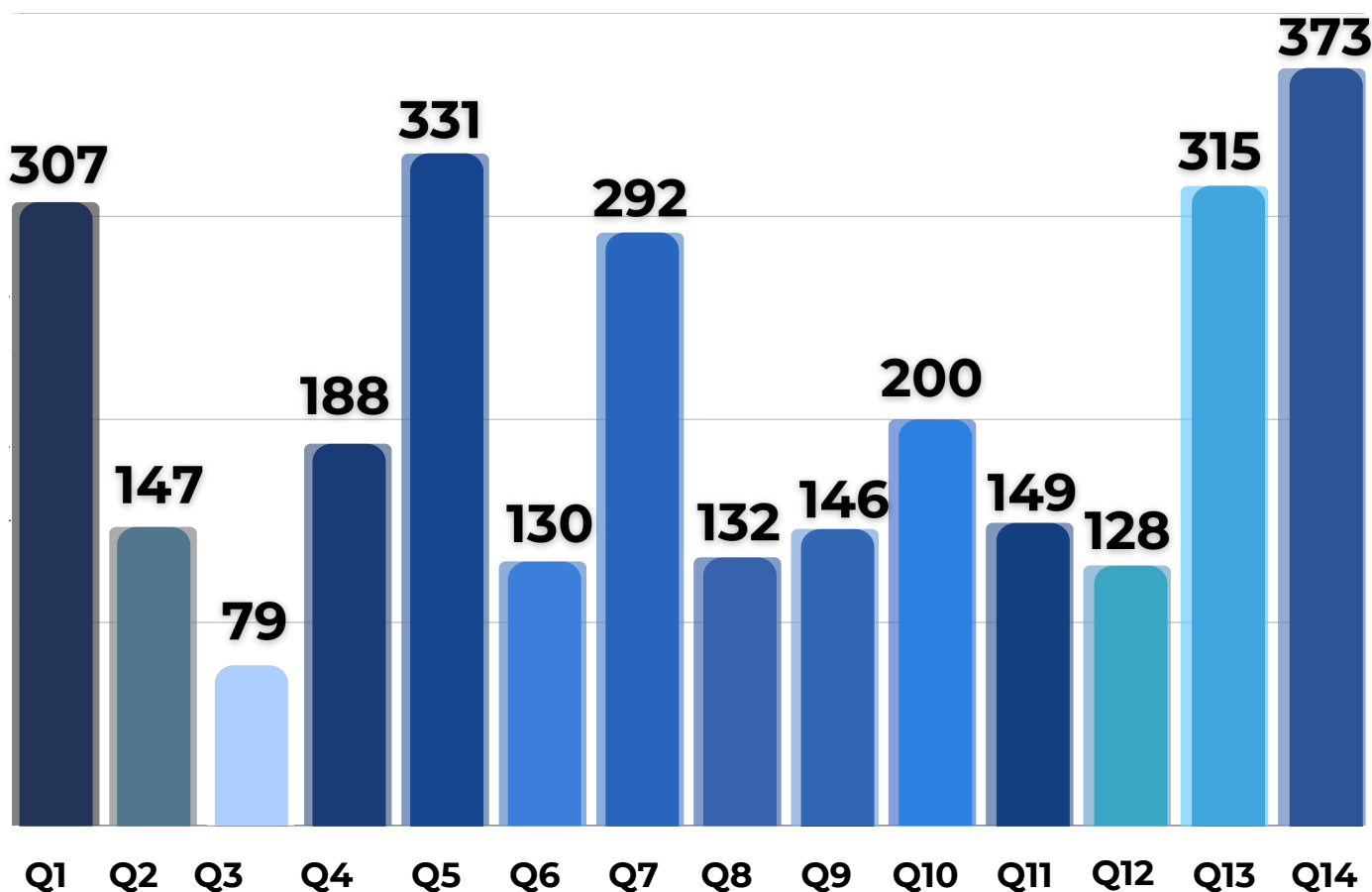
14

Number Of
Questions

>95% Respondents

QA engineers, test leads, SDETs,
and Automation engineers.

NUMBER OF RESPONDENTS (PER QUESTION)



- The study was conducted between **May and June 2025**, with each of the 14 polls deployed individually across **4** high-engagement Test Automation and QA communities. Our approach aimed at broadening the scope and diversity of insights, to capture a richer cross-section of global QA perspectives. Each poll had 4 options, which were designed to best benefit the context and gather a high number of responses from active QA professionals, maximizing statistical value for identifying real-world trends.
- Our survey captured insights from QA Engineers, Analysts, Managers, Automation Engineers, Test Leads, SDETs, and Business Analysts across key sectors, **Banking, IT Services, SaaS, Retail, E-Commerce, Consulting, Manufacturing, Healthcare, Pharma, Travel, Telecom, and Gaming**.
- Respondents span **global** testing teams and companies, offering an international view of current QA realities.

Key Highlights

A snapshot of the most decisive trends, patterns, and insights shaping QA, automation, and AI - based on what 2,500+ QA professionals say about the present and future of testing.

42%	<u>Release Pressure</u> feel that keeping up with fast releases is the biggest headache for QA teams.	65%	<u>Specialized Tools</u> prioritize specialized testing tools over a unified platform for their testing needs.
76%	<u>Manual Testing</u> respondents felt that manual testing will remain invaluable for exploratory testing and handling edge cases .	55%	<u>Preferred Approach</u> of teams favour a hybrid approach , combining codeless tools with code-based automation.
40%	<u>Most Desired Feature</u> Seek predictive analytics when choosing a test automation tool, making it the most in-demand feature for QAs.	52%	<u>Trust in AI</u> are confident in AI-generated testing outputs , but only after minor edits.
54%	<u>Delay Frequency</u> respondents reported that test automation failures cause delays in 20–40% of their release cycles.	14%	<u>Test Design Time</u> teams complete test design, on average in under 1 day ; test design remains a time-intensive task for teams.
39%	<u>Primary Adoption Barrier</u> of QAs felt that high upfront costs remain the biggest adoption barrier to test automation despite the long-term ROI automation brings.	52%	<u>Test Automation ROI</u> respondents measure ROI primarily through regression efficiency, and business stability : a clear nod to automation's value.
53%	<u>Strategic Shift</u> believe testers will transition to strategic roles , focusing on test strategy, while guiding AI-driven processes for smarter, more effective testing.	54%	<u>Automation %</u> of teams have automated between 25–75% of their functional test cases; still in the transition phase toward full automation.

SECTION 1

Current QA
Challenges & Practices



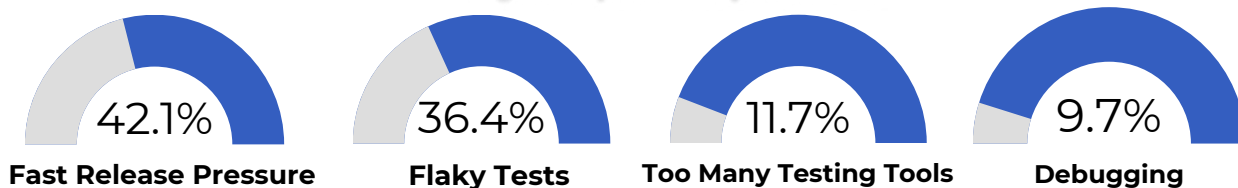
Understand how QAs choose tools and
balance between unified vs. specialized
solutions.



01

What's Your Biggest Daily Headache as a QA?

Figure 1 | 307 Respondents



Flaky Tests and Release Pressure Continue to Undermine QA Velocity and Trust

Despite advances in automation, QA teams continue to battle persistent operational bottlenecks that slow velocity and impair confidence in testing outputs. Chief among these are flaky tests and the mounting pressure of frequent releases, two compounding challenges that undercut both productivity and trust in automation pipelines.

36.4% of QA professionals cite flaky tests as a top daily blocker. These silent disruptors surface without clear code-level causes, wasting hours in **false-positive debugging** and slowly eroding trust in the very automation designed to boost velocity. In fast-paced CI/CD environments, false positives create friction and costly inefficiencies. Teams waste hours identifying “**phantom failures**” instead of focusing on quality coverage or risk-based testing. Flaky tests break the developer-QA trust loop, turning automation from an asset into a liability.

In parallel, **42.1%** of respondents cite rapid release cycles as their **most critical daily challenge**. With competitive pressures pushing software teams toward ever-shorter delivery timelines, testers must validate functionality in shrinking windows.

QA's role in this ecosystem is evolving from reactive bug catching to proactive, embedded quality enablers. But many teams are not yet equipped with the tools or processes to thrive under this velocity.

Fragmentation in Tooling Adds to QA Fatigue

While modern test automation ecosystems offer flexibility, they often come at the cost of tool sprawl. **11.7%** of respondents flagged the use of too many testing tools as a blocker. Fragmented stacks create context switching, integration overhead, and make it harder to scale consistent quality practices across teams. When Web, Mobile, API, and Performance testing all rely on siloed tools, efficiency drops and failure points increase.

The challenge is a **lack of consolidated solutions**. Best-in-class QA teams are now seeking platforms that offer end-to-end capabilities, simplifying test execution across modalities while reducing the operational burden on engineering.

Debugging: A Silent Time Sink

- **9.7%** cited debugging or root cause analysis as a top concern. Debugging could involve a real bug in the app, a flaky test, or even bad test data

While not always top of mind, debugging poorly configured environments, test setup issues, or integration glitches remains a daily drain on productivity.

This invisible tax compounds when flaky tests or misaligned test data further obscure root causes.

The implication for teams today is that debugging continues to act as a silent productivity killer, often requiring cross-functional coordination between QA, DevOps, and Product teams to fully resolve.

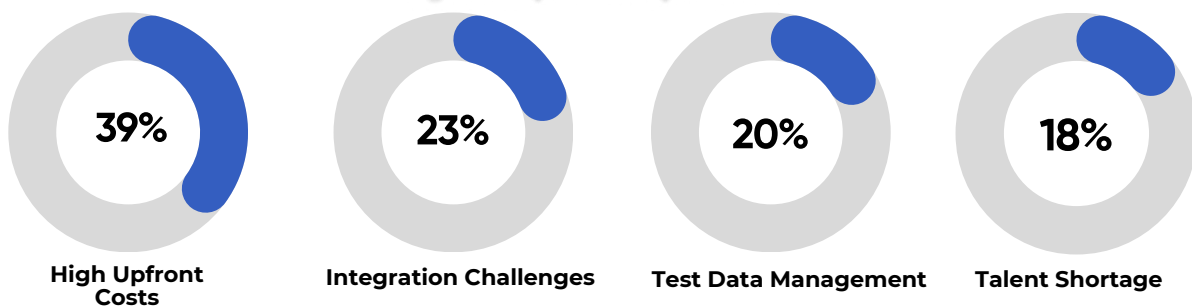
Strategic Takeaways

- **Mitigating flakiness** is foundational to restoring trust in automation and freeing up teams for strategic QA activities. This requires adding smart waits, test data stabilization, smarter retries (**Selective**), and intelligent orchestration.
- Continuous testing maturity is emerging as a core differentiator. To keep up with fast releases, apart from focussing on the **shift right side of testing**, teams must **shift testing left and automate earlier**, integrating QA seamlessly into the build pipeline.
- **Tool consolidation will be a critical success factor.** Enterprises must move toward unified test platforms that simplify toolchains and provide richer analytics to diagnose issues quickly.
- QA in 2025 about **engineering quality into every stage of the lifecycle** under increasing pressure and complexity.

02

What Do You Think is the Biggest Barrier to Test Automation Adoption?

Figure 2 | 147 Respondents



High Upfront Costs Continue to Deter Long-Term Gains

Despite the clear long-term benefits of automation, **39%** of QA professionals still cite **high upfront costs** as the primary adoption barrier. These could include infrastructure setup, licensing, training,

data transfer, and workforce reallocation.

While automation promises efficiency, scalability, and faster releases, the initial financial outlay remains a strong deterrent, especially for organizations with complex **legacy environments or budget-sensitive operations**.

This highlights a persistent disconnect between strategic ambition and operational readiness. Leaders understand why automation is needed, but lack a clear roadmap on how to implement it without friction. Even in 2025, the automation narrative is not **“Why Automate”**, it’s **“Is It Worth The Price?”**

Integration Complexities Compound Cost Concerns

With **23%** of respondents pointing to integration challenges, embedding modern automation tools into legacy stacks remains a technical roadblock. Integration often demands architectural overhauls, cross-team coordination, and deep platform customization, all of which inflate timelines and costs.

Talent Shortage Limits Operational Execution

Though less cited at **18%**, the talent gap is a strategic bottleneck. In regions like India, where automation demand is surging, the supply of skilled SDETs and automation architects is failing to keep pace.

Projections suggest AI-related roles will **double** the available talent pool by **2027**, making automation talent an enterprise-wide constraint. Organizations struggle both with execution capacity and quality.

Test Data Management Slows Real-World Progress

21.3% of respondents pointed to Test Data Management (TDM) as a silent but serious inhibitor. QA teams routinely deal with siloed, inconsistent, or sensitive data, challenges that lead to false positives, poor test coverage, and increased cycle times. Without robust TDM practices, even the best automation tools struggle to deliver reliable outcomes.

What Respondents Said

- 1 Upper management** prevents it from happening, not believing it would be valuable.
- 2 Introducing Additional Automation remains a challenge** without clear data of possible impact of newly automated processes

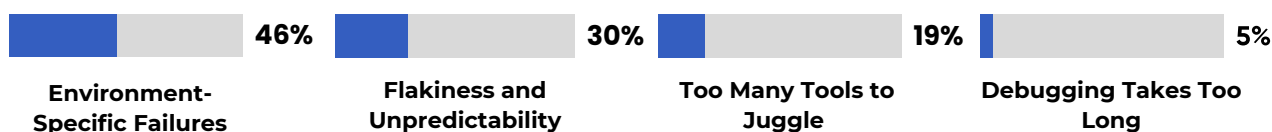
Strategic Takeaways

- While the promise of automation is widely accepted, practice still lags. The leading blockers cost, integration complexity, skill shortages, and data management reflect the growing pains of maturing ecosystems. Trust can be provided by showing tangible benefits of adopting automation.
- To move forward, organizations and tool vendors alike must simplify onboarding, integrations, and design **ROI-backed automation paths**. And as tools get smarter, to thrive testers must evolve too, from merely remaining script executors to strategists, guiding adoption from within.

03

What’s Your Biggest Frustration When Testing Across Web, Mobile, API, and Desktop?

Figure 3 | 79 Respondents



Environment-Specific Failures Continue to Undermine Test Stability at Scale

Cross-platform testing has now become an operational necessity. This is evident as **46%** of QA professionals cite environment-specific failures as the leading cause of frustration.

These failures stem from subtle yet impactful configuration mismatches, infrastructure inconsistencies, and data-related variations across staging, dev, and production environments. A common example in mobile testing is in testing applications for **iOS vs Android**. (Spacing, Permissions handling) The result: unreliable test outcomes that demand excessive investigation, often delaying releases or triggering regressions that weren't actual defects.

These findings underscore the growing need for smarter testing systems that can dynamically recognize and adapt to the unique needs of each environment. Whether testing across cloud platforms or microservices, teams increasingly require tooling that blends contextual awareness with intelligent decision-making. AI-driven testing engines, those capable of learning from environment-specific anomalies are fast becoming essential to maintain reliability.

Fragmented Toolsets Slow Testing Efficiency

19% of respondents highlight that juggling multiple tools across Web, API, and Mobile testing workflows creates unnecessary friction. This aligns with a broader trend noted across earlier findings: **fragmented ecosystems increase the cognitive and operational load on testers.**

From UI automation to performance testing and security validation, many teams use distinct tools that aren't inherently designed to communicate with one another. This patchwork leads to silos, slower feedback loops, and increased coordination costs.

Organizations seeking scale are now moving toward unified platforms with built-in support for multi-channel testing. Such consolidation minimizes context-switching and maximizes reuse, improving test orchestration and overall cycle time.

Debugging in Complex Setups: Still a Hidden Drag

Although only **5%** of respondents named debugging as the biggest frustration in cross-platform testing, the issue remains persistent. Variability in dependencies, test environments, and legacy configurations often complicate root cause analysis, especially when combined with asynchronous failures in mobile or cloud-hosted setups.

Time spent on identifying non-deterministic behaviours is time not spent on improving test coverage or driving innovation. Teams must adopt test practices that promote debuggability as a first principle, such as logging standardization and automated diagnostics embedded within test pipelines.

Flakiness, Especially in Mobile and API Testing, Persists

30% of respondents report test flakiness and unpredictability as significant frustrations, particularly acute in mobile and API-heavy ecosystems. Variations in device firmware, OS versions, or third-party service latency often result in inconsistent outcomes, making it harder to distinguish between real defects and transient issues.

What this signals is not just a tooling gap, but a process-level challenge: **modern QA needs to shift from reactive to resilient testing models.**

Practices like contract testing, virtual environments, and intelligent test retries can help reduce the noise caused by non-deterministic test behavior.

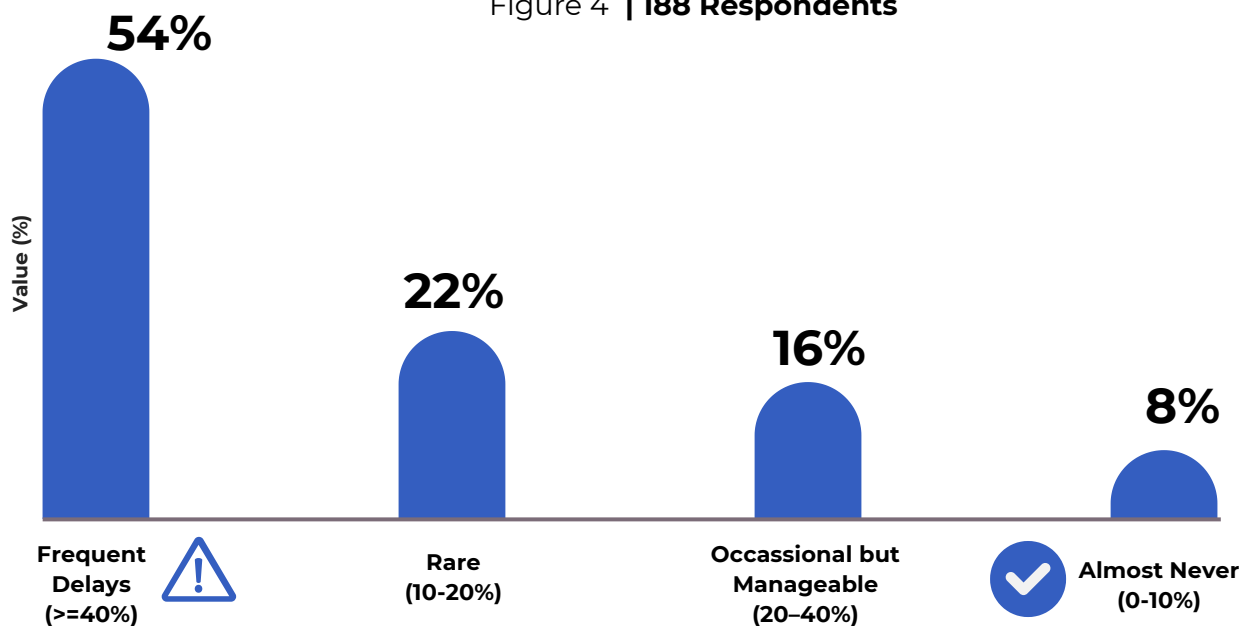
Final Thoughts

While environment-specific issues lead the charge, the common thread across all frustrations is operational complexity. The modern QA ecosystem needs a dual focus: **Unifying tooling and increasing the intelligence of testing frameworks**. For leaders, the takeaway is to **invest not just in faster automation, but in smarter infrastructure that anticipates, adapts, and scales**.

04

How Often Do Test Automation Failures Cause Delays in Your Release Cycle?

Figure 4 | 188 Respondents



Automation Delays Are Common, But Often Contained

Despite advancements in tooling and methodology, test automation failures remain a persistent factor in release delays. A majority (16%) of respondents reported delays occurring in 20–40% of release cycles, categorizing them as occasional but manageable. These failures are typically caused by brittle scripts, unstable environments, or dependencies that weren't properly mocked or isolated during test design.

This signals a nuanced maturity level:

many teams have built in the resilience and triaging workflows needed to respond quickly to breakages, preventing them from escalating into critical blockers. **Observability, Automated failure alerts**, structured debugging processes, and prioritization frameworks are now foundational to keeping delivery on track even when things go wrong.

Resilient Test Ecosystems Still Rare

Only **8%** of respondents say they almost never experience automation-induced delays. These organizations typically invest heavily in infrastructure, **version control**, CI/CD governance, and real-time quality observability. Their success reflects the value of consistent environments, strict data management practices, and test design patterns that anticipate variability across platforms and services.

This segment represents the gold standard:

a future where test automation is no longer a source of fragility but a strategic enabler of fast, frequent, and fearless releases.

The Middle Ground: Manageable But Costly

22% report rare automation failures (10–20% of cycles), and while not debilitating, these still consume time and distract from higher-value QA work like exploratory testing or root cause analysis. Organizations in this range have generally begun standardizing platforms, using CI-integrated dashboards, and adopting **self-healing automation** where applicable.

Frequent Failures Reflect Ecosystem Immaturity

At the other end of the spectrum, **54%** of respondents reported automation-related delays in more than **40%** of release cycles.

For these teams, automated testing may still be in a transitional state, struggling with poor test stability, high maintenance, and integration issues across staging or dev environments. The resulting noise from flaky tests, false failures, and environment mismatches creates overhead that slows delivery and erodes stakeholder trust. For these teams, reducing test failures is primarily about better tools combined with a focus on refining the process around test data, test ownership, and cross-functional alignment.

Toward Automation That Heals Itself

Across maturity levels, **automation failure isn't going away and must be managed.**

Industry leaders are shifting focus to building intelligent, adaptive test suites that can identify false positives, retry failed cases intelligently, and minimize manual intervention.

Investing in **self-healing automation frameworks, proactive environment monitoring, and integrated test observability** is becoming essential.

These innovations reduce maintenance load, restore confidence in automation outputs, and ultimately protect release velocity.

Final Thoughts

Test automation failures may not always halt the release train, but they shape how fast and confidently it moves.

As failure management strategies mature, QA teams are finding that the true differentiator is not always the absence of errors, **but how swiftly, intelligently, and sustainably those errors are resolved.**

05

What Percentage Of Your Total Functional Test Cases Are Automated?

Figure 5 | 331 Respondents



Mid-Level Automation Dominates, But Industry Maturity Still Lags

Despite the promise of automation, the majority of QA teams remain in a transitional phase, balancing between legacy manual processes and scalable automated testing strategies.

53% of respondents report automating between **25% to 75%** of their total functional test cases. This sizable middle tier reflects a **growing maturity curve**, where automation is no longer experimental, but also not yet comprehensive. Many of these teams are likely refining frameworks, resolving flakiness, and calibrating automation to business-critical test paths.

At the lower end, **26%** of QA professionals indicate they've automated less than **25%** of their functional test suites. These responses point to either early-stage adoption, high

dependency on manual validation (especially for visual/UI-based cases), or **as we earlier saw, organizational resistance** to investing in automation at scale.

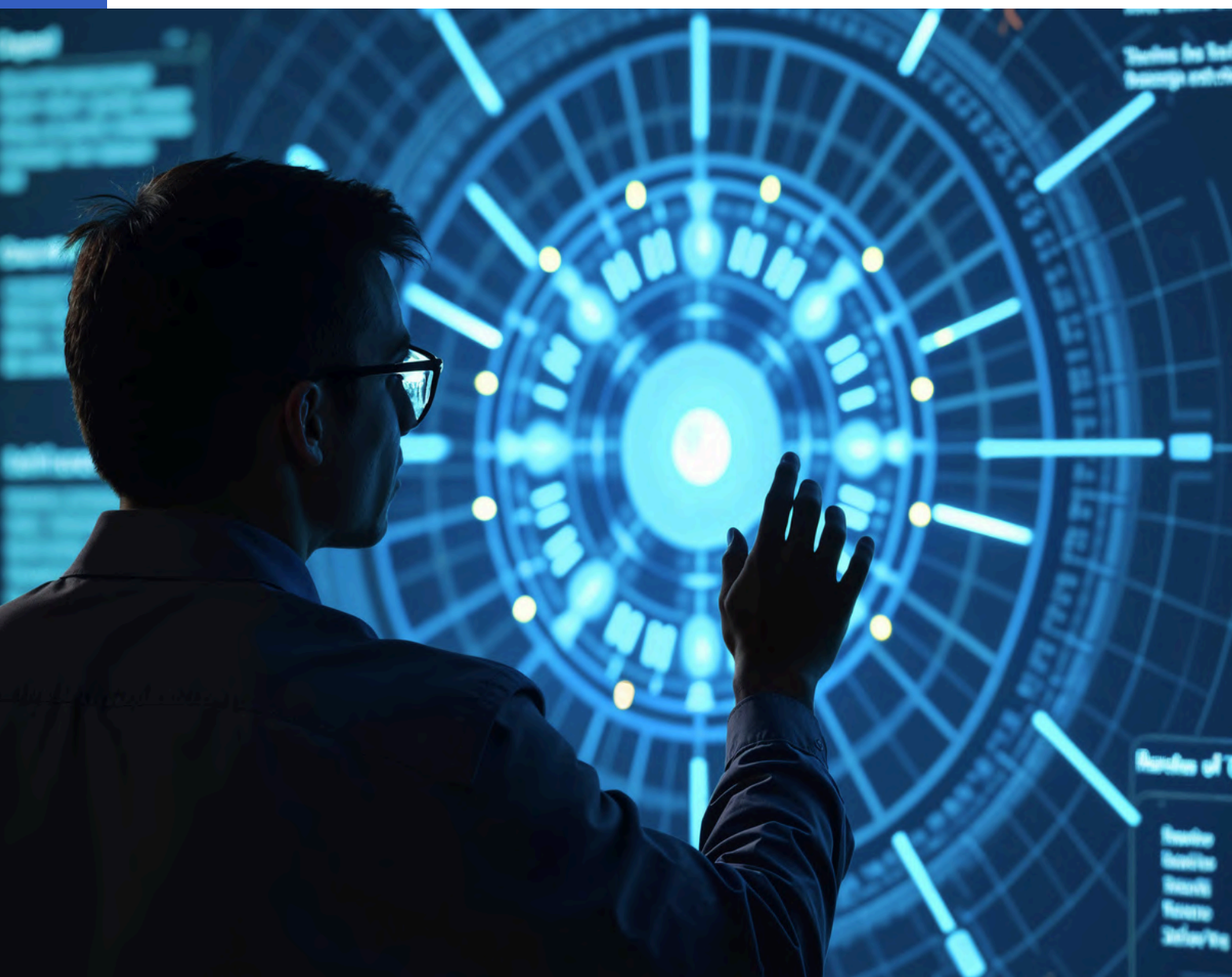
Conversely, **21%** report having automated more than **75%** of functional tests, a benchmark of **high maturity**. These teams are typically characterized by strong test governance, scalable infrastructure, reusable test frameworks, and a DevOps culture that understands that testing processes need to evolve with time.

Qualitative insight from comments strengthens this trend:

- 
 "I'd say that maybe **1%** of my tests are manual, and that 1% is for **visual confirmation of the UI**. **99%** are automated."
- 
 "Aiming for **80/20**."

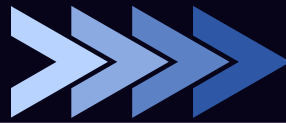
Strategic Takeaways

- **Partial automation** is the norm, showing growing traction but not full maturity.
- High-automation teams (75%+) are still a **minority, owing to trust issues**, highlighting the opportunity for platform-driven acceleration.
- **Gen-AI-based testing, paired with deep domain expertise** are potential unlocks for teams seeking higher coverage. Gen-AI enables **rapid test case design with multiple reruns for maximum coverage**.
- **Tool sprawl, flakiness, and lack of business-user collaboration remain key blockers** to moving beyond mid-range automation.
- As test automation adoption accelerates, the next frontier lies in making it more accessible, stable, and integrated. The journey to **75%+** automation will be driven by vendors communicating clear ROI, self-healing capabilities, and empowering QA as a strategic pillar that is to be embedded throughout the .

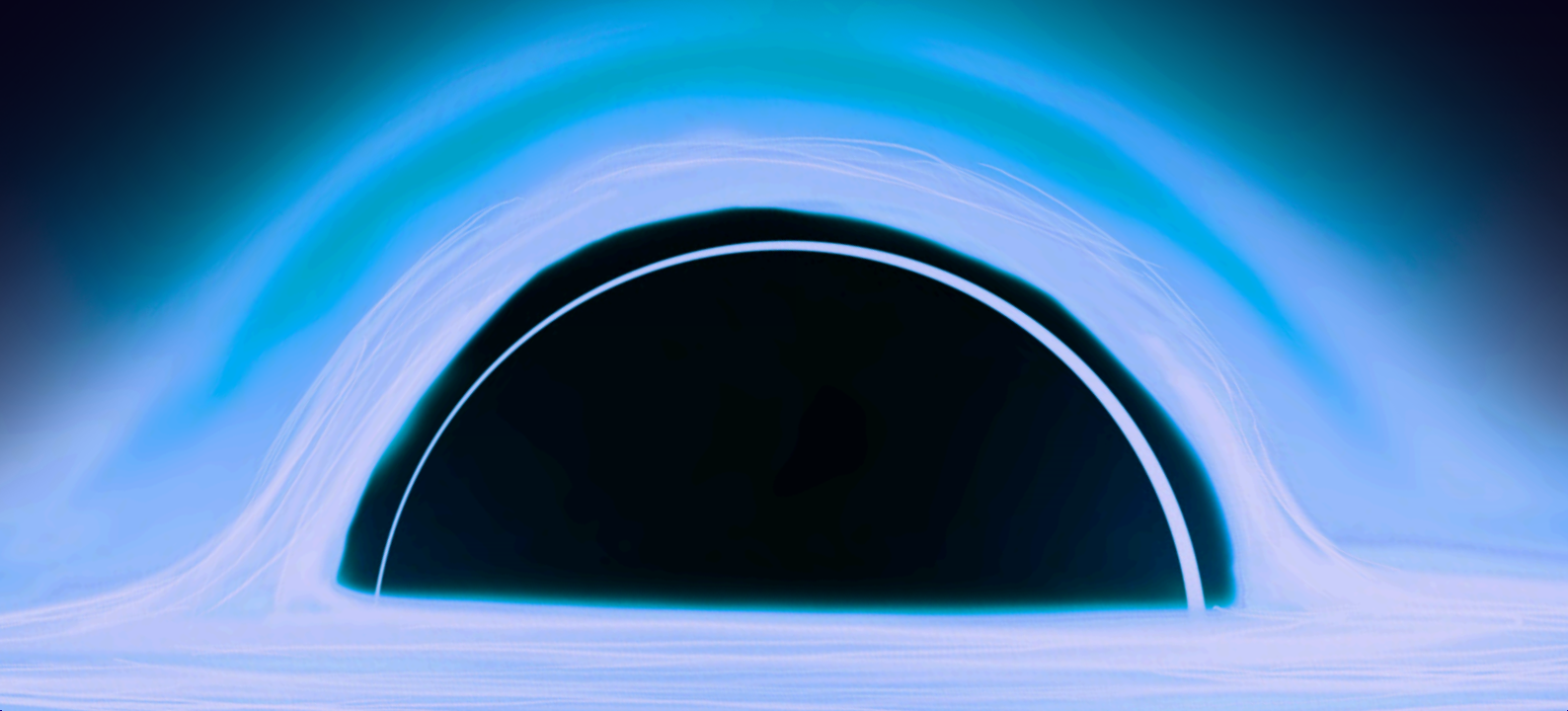


SECTION 2

Tooling Decisions & Preferences



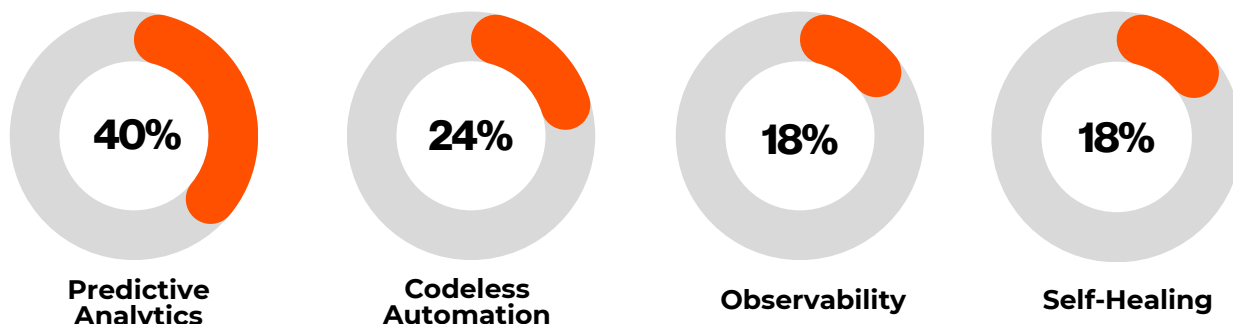
Understand how QAs choose tools and balance between unified vs. specialized solutions.



06

What Matters Most When Choosing a Test Automation Tool?

Figure 6 | 130 Respondents



Predictive Analytics Emerges as the Top Priority

40% of QA professionals ranked **predictive analytics as the most critical feature** when selecting a test automation tool, signalling a clear shift from reactive testing to strategic, proactive QA. This makes even more sense as **studies reveal that 49%** of banking consumers would switch to a competitor after just **one** bad mobile banking experience.

The rise of predictive QA reflects the industry's drive for faster, risk-aware releases and smarter test design decisions.

For solution providers, this is a strong call to **embed advanced analytics capabilities** that turn raw test data into strategic intelligence, enabling early risk detection and increased customer trust.

Other Key Features Gaining Ground

Codeless Automation

The adoption of codeless platforms underscores the need to **democratize** testing, empowering business analysts, product owners, and non-technical team members to contribute directly to test design through drag and drop workflows without having to rely entirely on SMEs, QA or development teams.

More importantly, it fosters true cross-functional collaboration. Instead of waiting for requirements to be translated into test scripts, stakeholders can now co-own quality from the outset. It's a structural improvement on how teams build software.

Lastly, by lowering the technical entry barrier, organizations **will be in a position to better leverage their existing workforce**, enabling a more agile, flexible approach to quality assurance.

Observability

Testing, once confined to pass/fail binary metrics, is now expected to offer continuous visibility into system health.

Tools with observability features, such as **unified dashboards for logs, traces, and metrics** are gaining traction.

Observability enables teams to trace defects to root causes faster and manage live systems with greater confidence. For banking, where even **52** minutes of annual downtime (99.99% uptime) is considered high-risk, **observability is a must offer feature for vendors**.

Self-Healing

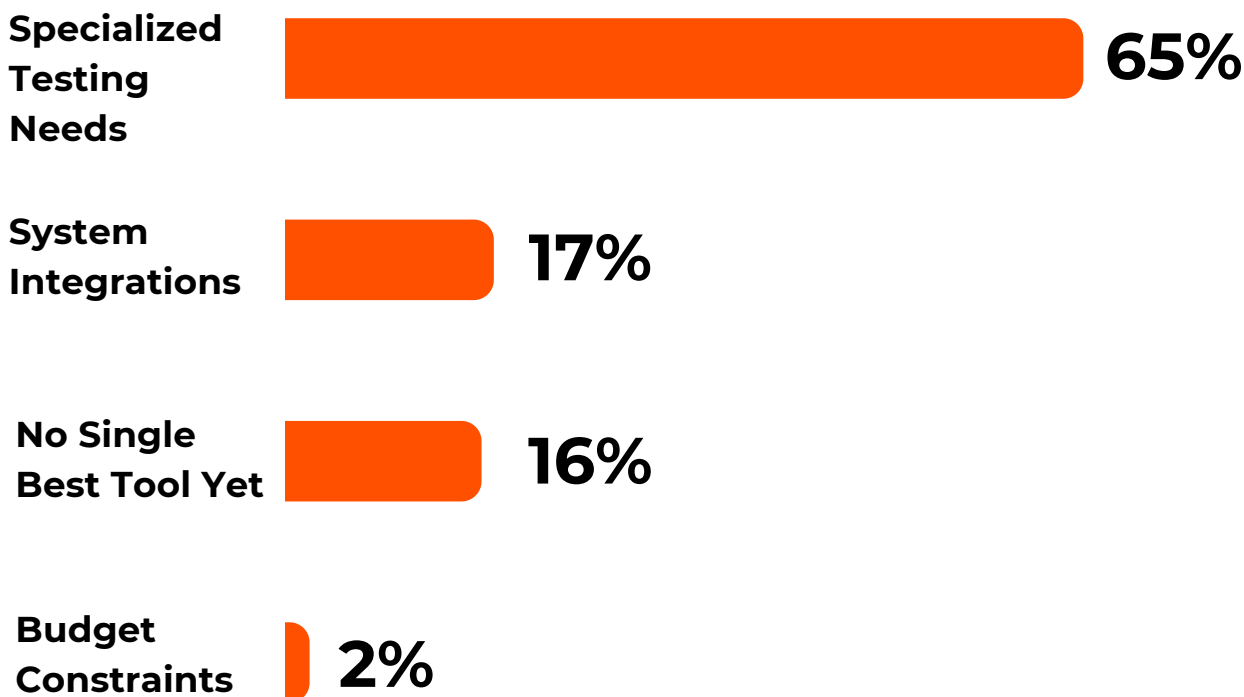
As release cycles shorten, teams are leaning on self-healing automation to maintain test stability.

These tools adapt to UI changes without manual updates, making automation more resilient and less brittle over time. This becomes vital in industries such as **banking**, where updates are released frequently.

In environments with frequent deployments, such as banking, self-healing tests protect ROI by ensuring consistency without rework.

07 Why do you use different tools for API, Web, Mobile, and Database Testing Instead of a Unified Solution?

Figure 7 | 292 Respondents



Specialization Trumps Unification: For Now

The data reveals a compelling insight: despite the growing availability of unified testing platforms, **65%** of QA teams continue to rely on specialized tools for different types of testing (API, web, mobile, database). This could be a matter of preference or a strategic necessity, based on the context.

At its core, the fragmentation is driven by capability, not convenience. Each testing layer comes with distinct technical complexities:

- **Mobile testing** requires **device emulation, gesture simulation, and responsiveness checks** across operating systems. Specialized tools simulate or give remote access to hundreds of real devices or high-fidelity emulators, saving cost and effort.
- **Web testing** involves DOM manipulation, cross browser testing, and UI validation.
- **Database testing** needs robust querying (fetching the right data), data integrity checks, and transactional verification.

Despite the existence of powerful unified platforms, respondents feel that no single tool, as of now, **delivers the depth of functionality required across all these domains without trade-offs.**

Integration Complexity

Even when a unified solution exists, respondents find integrating it into existing pipelines a challenge. Many QA teams operate in hybrid environments, with legacy systems, microservices, and layered CI/CD stacks. Switching platforms introduces risk, retraining overhead, and workflow disruption.

Lack of a Best-in-Class All-in-One Tool

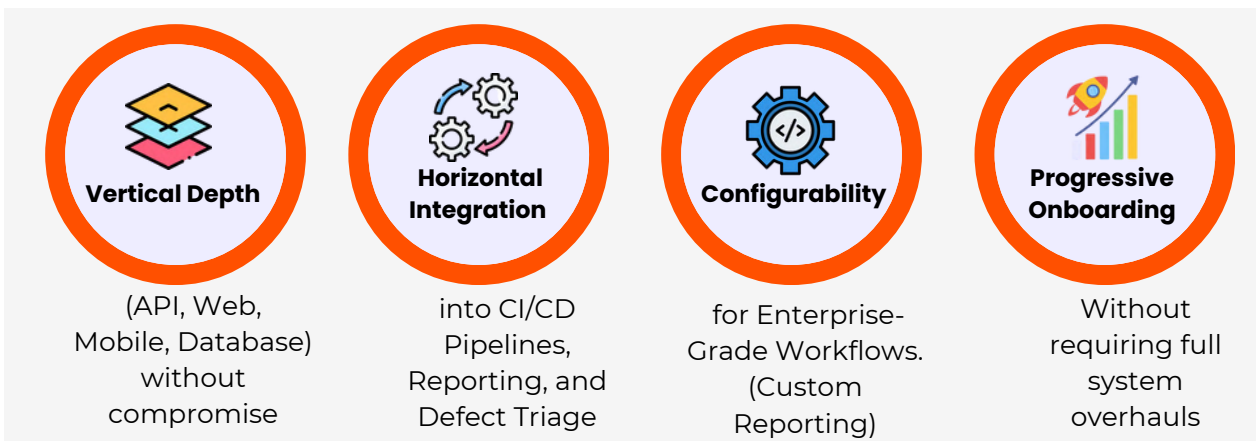
Teams are open to unification if a platform could meet the standards expected across each vertical. For high-stakes environments, QA leaders will not trade precision for platform simplicity.

To change this notion, there is a need in the industry for **greater awareness** around what advantages a modern, all-in-one testing solution can deliver in terms of coverage, scalability, and simplicity.

Budget Not a Deciding Factor

Interestingly, only **2%** of respondents cited **budget** as a key constraint. This reinforces a strategic truth: tooling decisions in QA are **capability-led, not cost-led**. When performance and reliability are at stake, teams are willing to invest, but only in tools that they feel will perform.

Therefore, the future of test automation lies in platforms that offer:



Strategic Takeaways

While fragmentation across testing tools is currently the norm, it's not necessarily by necessity, it's often by **perception**. Many QA teams continue to juggle multiple specialized tools because of **limited awareness, skepticism around depth, or lack of proven adoption stories of unified platforms**. This signals a crucial inflection point: trust is a primary bottleneck in multiple buying decisions, be it unified platforms or AI adoption.

To gain trust, vendors who already offer full-stack solutions must double down on 3 key aspects:

- Clearly articulate what their **unique value proposition** is and how their product can better serve the needs of the clients than a specialized tool **with numbers**.
- **Proof:** Providing robust case studies and benchmarks that demonstrate enterprise-grade scalability and precision
- **Ease of Migration:** Lowering friction in switching from fragmented toolchains through seamless integrations and support.

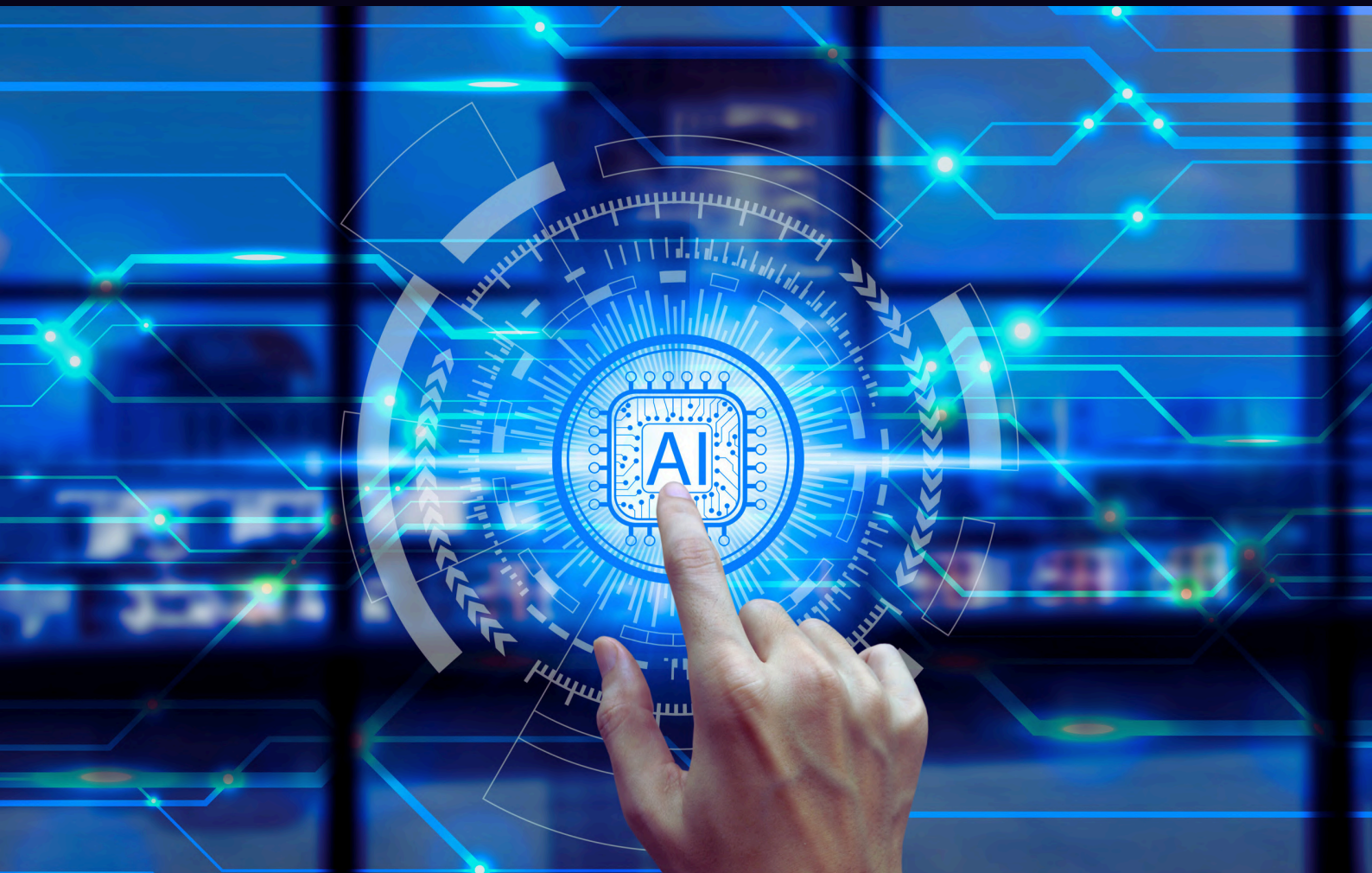


SECTION 3

AI & Codeless Automation



Gauge openness to AI-driven testing and the impact of low-code automation on QA workflows.

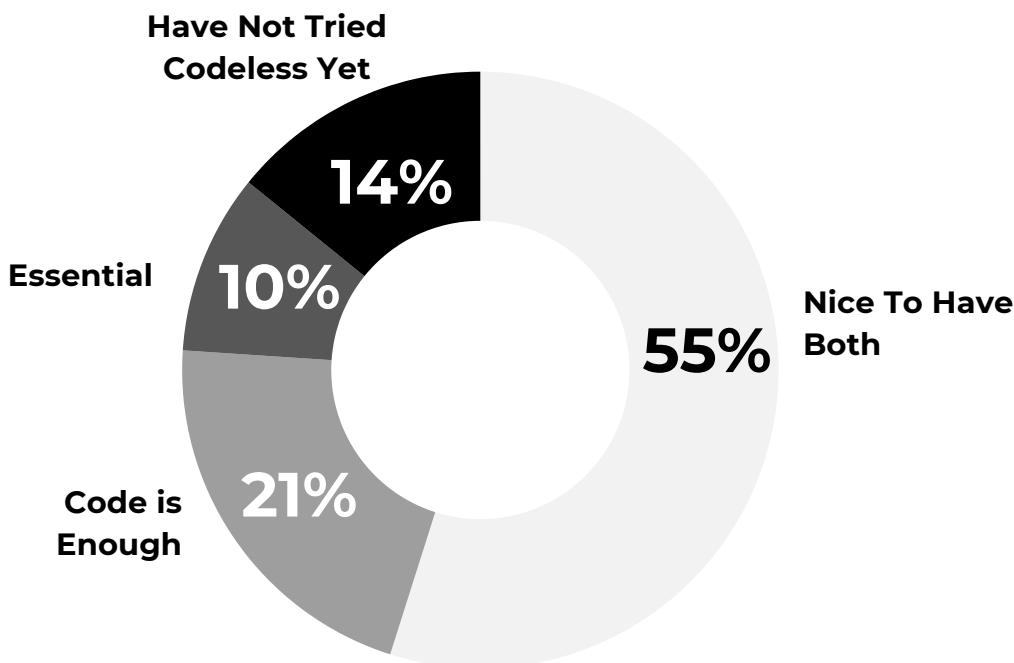


SECTION 3

08

How Important is Codeless Automation For Your Team?

Figure 8 | 132 Respondents



The Role of Codeless Automation in Modern QA: Tool or a facilitator of Transformation?

As test automation has gradually shifted into a standard conversation that most QA teams have, the primary shift lies in the focus. **Earlier, most teams focused solely on the ROI of test automation.** Today, while ROI remains vital, with QA becoming more inclusive, a secondary question that QA teams ponder upon when selecting tools is **the usability of test automation tools by non-tech professionals like business analysts.**

Codeless Is About Simplicity And Access

With the emergence of Gen-AI, a key strength of codeless platforms lies in unlocking automation for non-tech professionals like business analysts and product managers. These professionals can now build test cases with high coverage in only a few minutes.

The Case for Code Remains Strong

With that being said, **21.2%** respondents still use only code-based automation. This is often the result of legacy systems and complex integrations where coders prefer a deeper level of control and customization. For these teams, it is not always that code is a preference.

Codeless automation may not always be the most viable choice for testers, especially when dealing with complex logic.

Therefore, a common mode of operating includes a hybrid approach where teams utilize **codeless automation for repetitive tasks and code-based automation for complex tasks that require custom logic.** Our study backs this fact. **55%** respondents prefer having the flexibility of both code-based and codeless automation, owing to dynamic project requirements.

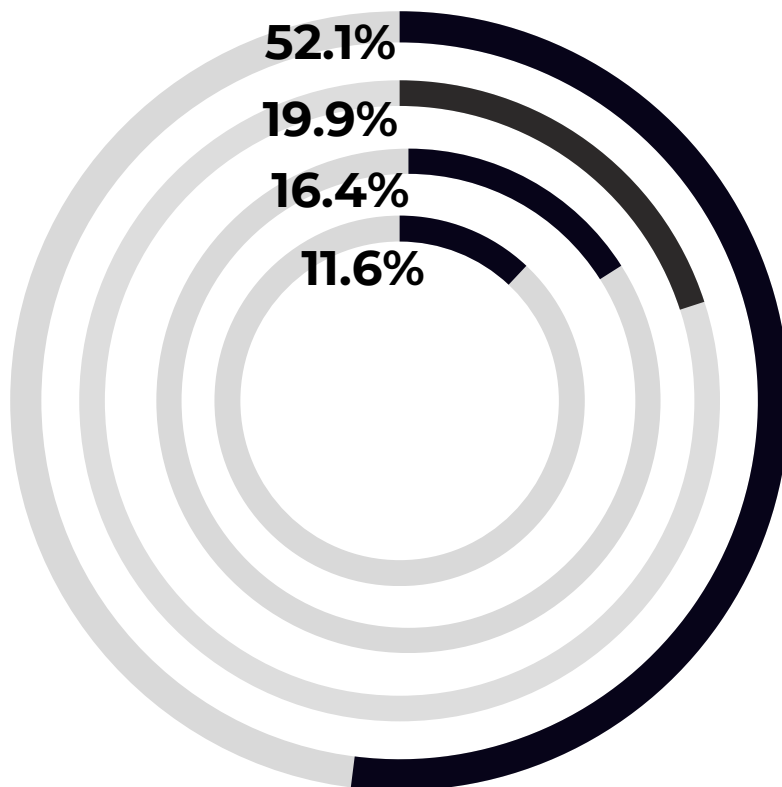
The Adoption Gap: An Opportunity

Interestingly, **14.2%** respondents are yet to explore codeless automation, revealing a latent opportunity in the market. But as we saw, this is not a problem of awareness, but **uncertainty**, as questions remain around the robustness and scalability of codeless platforms. Vendors need to justify how codeless automation can handle these issues to maximise prospect conversions.

09

How Do You Measure the ROI of Your Test Automation Efforts?

Figure 9 | 146 Respondents



LEGEND

- **Faster Releases & Regressions** ➔ **52.1%**
- **Reduced Manual Testing Time** ➔ **19.9%**
- **Improved Defect Detection** ➔ **16.4%**
- **Cost Savings** ➔ **11.6%**

Release Speed: Where ROI Matters The Most

Our data shows that **52.1%** of QA professionals rank **“faster releases and regressions”** as the single most important marker of automation ROI. Speed to market, especially with confidence in release quality, is now a direct competitive advantage.

The second tier of ROI drivers, namely reduced manual testing time (**19.9%**) and improved defect detection (**16.4%**), point to a deeper shift. Automation now revolves around re-allocating human effort.

By removing repetitive manual cycles, skilled testers are freed up for exploratory testing, customer experience validation, and high-risk scenario planning, areas where we discovered human insight still matters most.

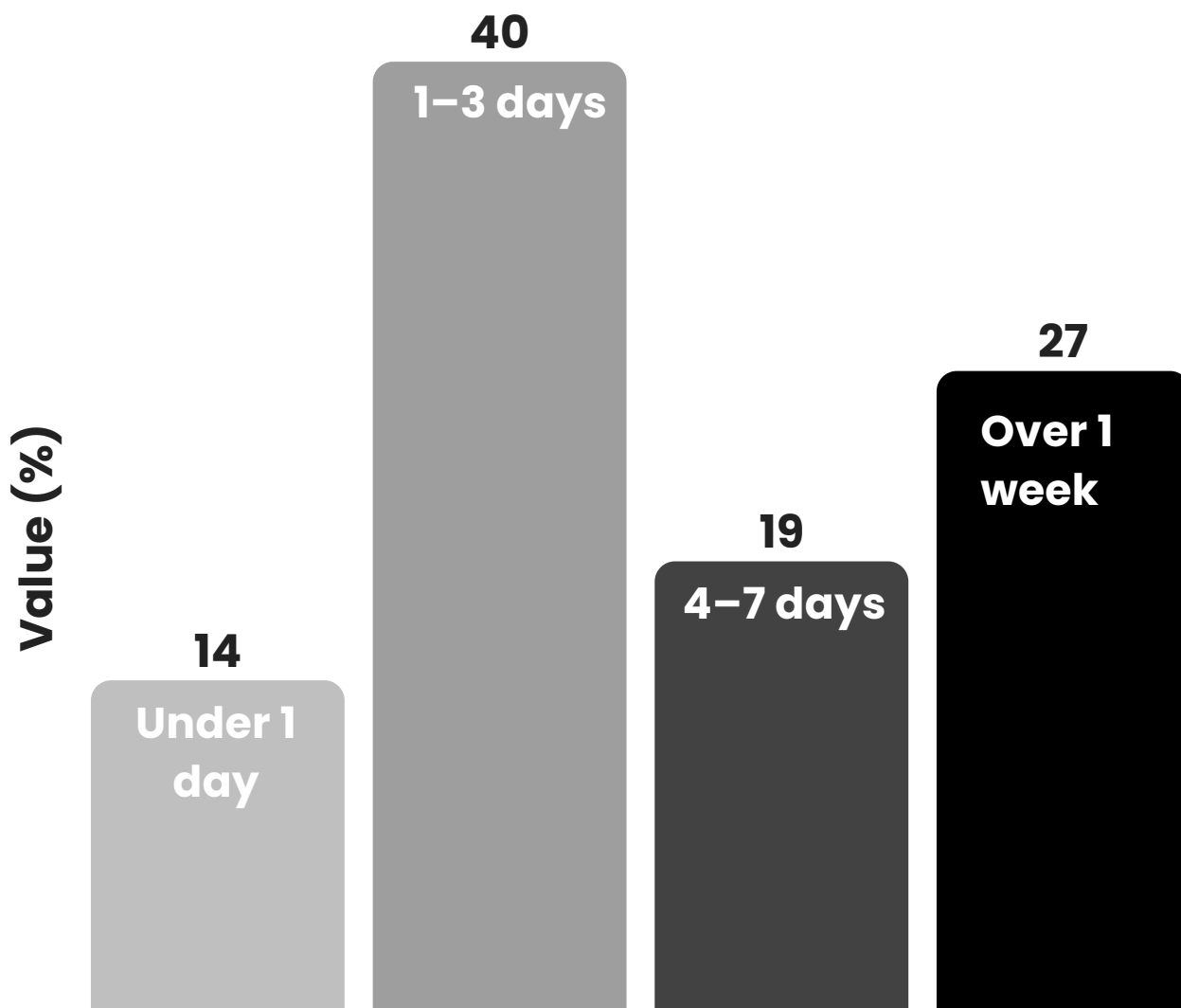
Early defect detection plays a crucial role here. When automated test suites catch failures earlier in the lifecycle, the cost of correction drops dramatically. Data reveals that fixing a bug after production is **30** times more expensive than addressing it earlier in the development process. Teams must also **shift-right** by course-correcting in real time, preserving quality and trust. Interestingly, cost savings ranked lowest at **11.6%**; a finding that reflects the maturing perspective of QA leaders. **Cost reduction** is still an important outcome, but not the core goal.

In large-scale engineering environments, automation is valued more for its ability to unlock scale, ensure consistency, and reduce business risk. It is seen more from a strategic front, and not just a tactical lever.

10

On Average Across Projects, How Long Does It Take You to Design Test Cases With Adequate Coverage (Covering Core Workflows / Edge Cases) for a New Application?

Figure 10 | 200 Respondents



Test Design Is Still a Bottleneck And That's a Strategic Opportunity

Despite advances in automation and tooling, test case design remains a time-consuming stage in the software test lifecycle.

Our data shows that nearly half (**46%**) of respondents spend **4+ days** on average just designing test cases, not executing them, not maintaining them, simply designing them across projects.

That figure alone demands attention. A primary reason for this could be the manual analysis of documents such as **functional specification documents, business**

requirement documents (often 100s of pages long), user stories, and ambiguous workflows.

Testers, with the help of subject matter experts (**SMEs**) are expected to translate vague requirements into structured, testable scenarios, a task that requires time, context, experience, and iteration. The result is a QA process that starts slow and stays reactive.

When test design is delayed, release cycles slow down. Teams fall behind on coverage. Gaps in quality widen, and alignment between business intent and test execution begins to fray.

GenAI Can Accelerate Test Design Without Sacrificing Accuracy

A new class of GenAI-powered tools is starting to close this gap. Instead of writing every scenario from scratch, QA teams can now input **business flows, user journeys, or requirements** and receive AI-generated test case drafts. These results come with a high level of accuracy and are structured, relevant, and customizable.

The role of testers shifts from **“authors” or “test case designers” to “editors” or “reviewers”**, reviewing and refining rather than building from the ground up.

The time savings are significant. But the larger benefit lies in expanding QA ownership. When GenAI handles the initial heavy lifting, business analysts, product

managers and domain experts can meaningfully participate in test design.

This leads to test suites that are not only faster to create but also better aligned to business outcomes. We also saw how most teams prefer a combination of code based and codeless automation.

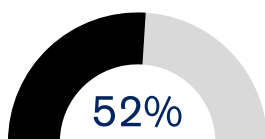
Teams that modernize their design workflows with Gen-AI will see faster time-to-coverage, higher test relevance, and smoother transitions from business requirements to validation.

For leadership, a key question today would be: **“Are we designing the right tests fast enough and involving the right people?”** That’s what will define scalable, future-proof quality assurance.

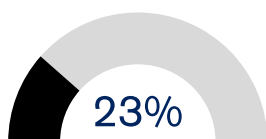
11

How Confident Are You In The Accuracy Of AI-Generated Testing Outputs (e.g., Test Cases, Scripts, Defect Reports)?

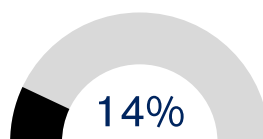
Figure 11 | 149 Respondents



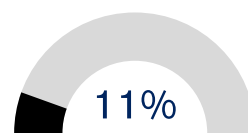
**Confident,
After Minor
Edits**



**Low trust,
Frequent
Reviews**



**Do Not Use
AI in
Testing Yet**



**Very
Confident,
Rarely Edit**



AI in Testing: Progress, But Trust Still Lags Behind

AI in test automation is maturing but it's **trust**, not tooling, that remains the final frontier for enterprise adoption.

Despite increasing usage, **37%** respondents either distrust AI-generated test outputs or haven't adopted AI at all. This hesitation stems not from a lack of exposure, but from concerns around **accuracy, data privacy and security, ROI, accountability, and leadership resistance**.

In high-stakes domains like banking, where customers are bombarded with options, trust becomes vital to minimize customer churn. With rising customer expectations, customers feel frustrated faster today when dealing with wait time and bugs as compared to earlier periods.

Insights Beyond the Numbers

What's most telling is how respondents use it. **52%** say AI-generated test cases are usable only after edits. This signals that **human-in-the-loop validation** has now become the norm. Teams accept AI as an assistive layer, not yet as a fully autonomous test author.

23% require frequent changes or express low trust. This group typically operates in regulated or safety-critical environments such as banking, where failure has legal, financial, or reputational consequences. In these sectors, AI explainability and precision still match, if not outweigh speed.

11% rarely edit AI-generated tests

This early adopter cohort either works in lower-risk digital-native environments or has access to advanced AI tooling deeply embedded into their workflows.

14% don't use AI at all. In many of these cases, we discovered that the blockers often include leadership skepticism to a lack of clear ROI communication or resistance from developer teams wary of losing control over test logic.

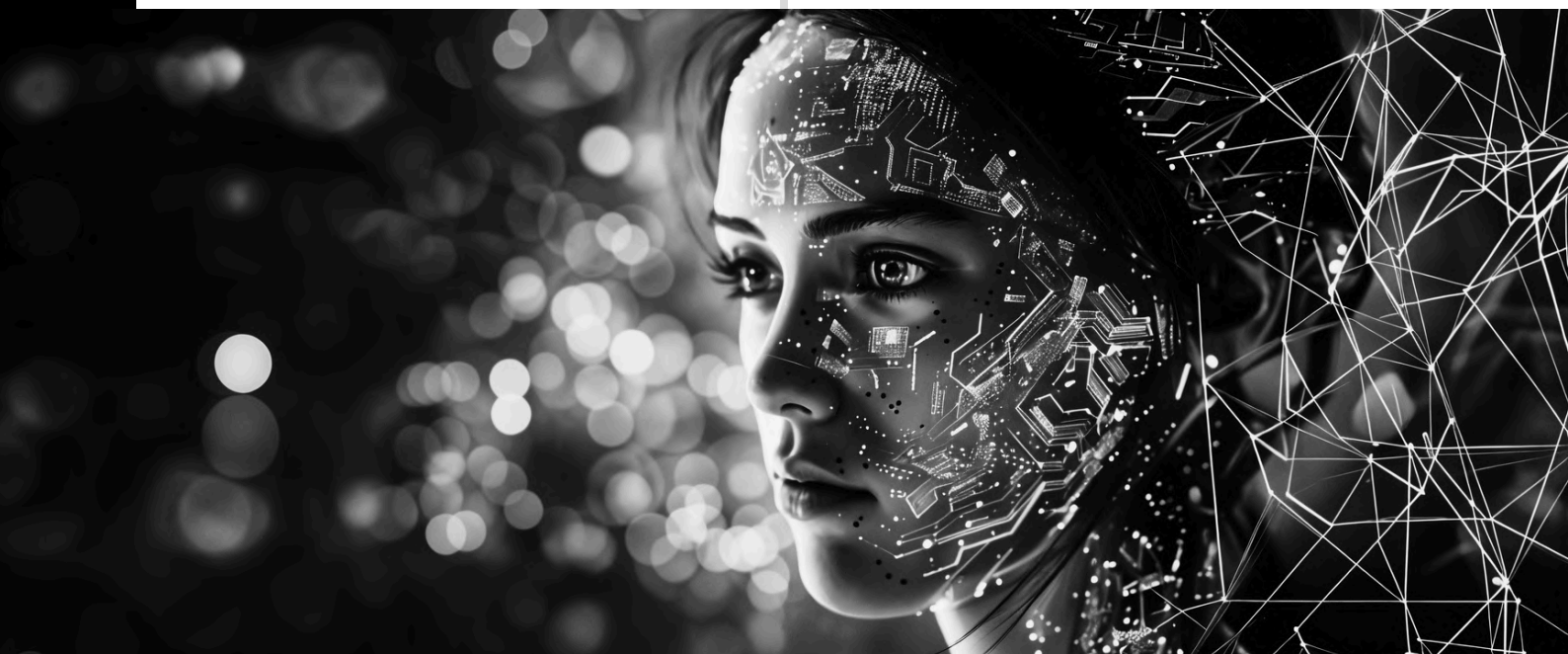
The Missing Link Is Not Necessarily Capability But Confidence

Tooling is ready. Testers are cautiously optimistic. **But the data conveys that trust is the primary currency AI hasn't fully earned.**

To close the adoption gap, vendors and leaders must move beyond showcasing features.

- As discussed earlier, proven, domain-specific case studies that demonstrate **ROI in clear numbers**.
- **Change management** frameworks to upskill teams and embed AI usage into everyday testing practices
- **Explainable AI outputs** that testers can interrogate, modify, and justify in audits

A common misconception or fear is that **AI will replace testers or take their jobs**. Skilled testers who carry **deep domain expertise** remain unaffected. The role of AI is only to support and give teams a strategic advantage. **QA teams that adopt a hybrid approach, i.e.,** AI combined with human expertise for critical asynchronous tasks will build trust through consistent outcomes.



SECTION 4

The Future Of Testing & Human Roles



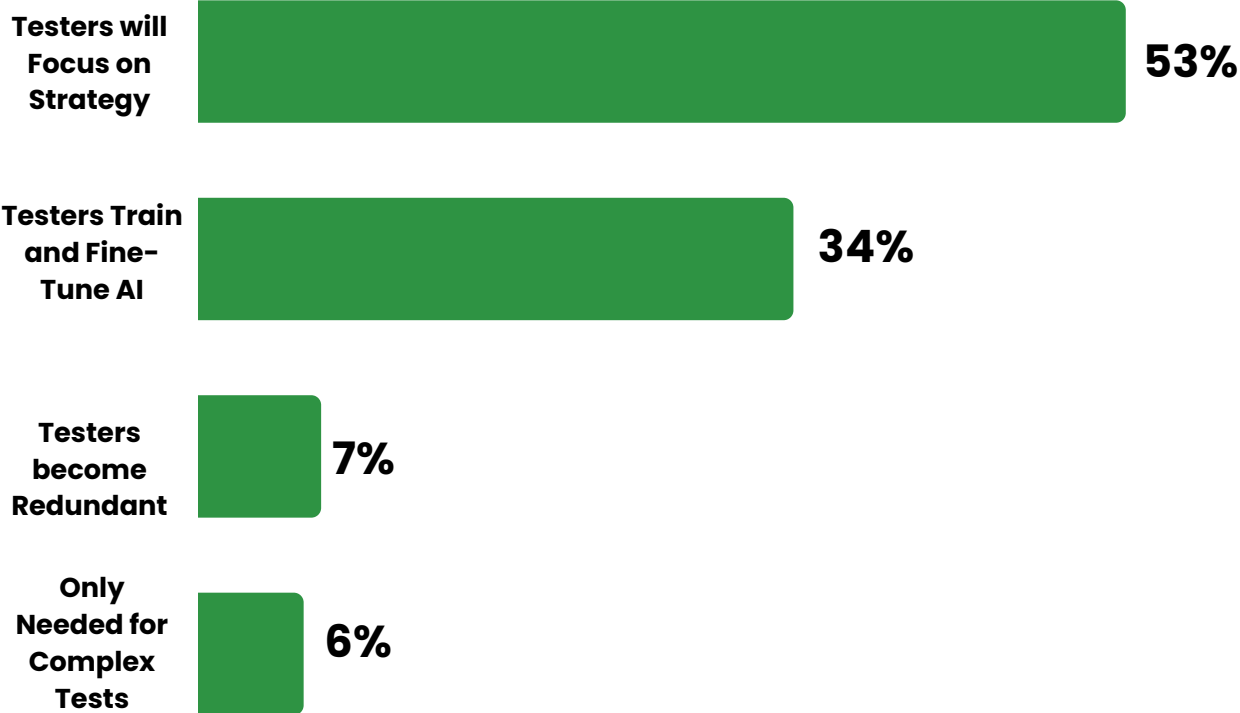
Explore how QA roles and skills are evolving in the age of autonomous testing.



12

How Does the Role of Human Testers Evolve in the Era of Fully Autonomous Test Creation and Execution?

Figure 12| 128 Respondents



Redefining the Role: AI Won't Replace Testers But Is Changing What They Do

The previous data revealed something subtle but significant: while AI adoption in testing is accelerating, it is not displacing testers. It is reshaping what they're expected to deliver.

This next finding makes that shift even more tangible. Over half (**52.8%**) of respondents believe testers will transition from hands-on execution to owning the strategic layer of testing, i.e., driving quality governance, risk analysis, and intelligent test orchestration.

- As AI takes on **repetitive tasks**, the human role shifts upstream, prioritizing which paths to test, evaluating trade-offs, and shaping what quality means for the business.
- 33.9%** believe testers will train and fine-tune AI. Many believe testers will serve as domain guides, helping AI models learn workflows, refine logic, and better mirror real-world conditions.

- A small group (**6%**) sees testers specializing in edge cases that require judgment. These are the cornerstones of exploratory and risk-based testing.
- Perhaps the most important statistic: only **7%** foresee testers becoming fully obsolete. This reinforces a broader truth that we discussed based on the findings in the previous question. AI is strong in executing repetitive tasks but cannot replace testers entirely for the time being.

AI Can Execute. Humans Must Lead

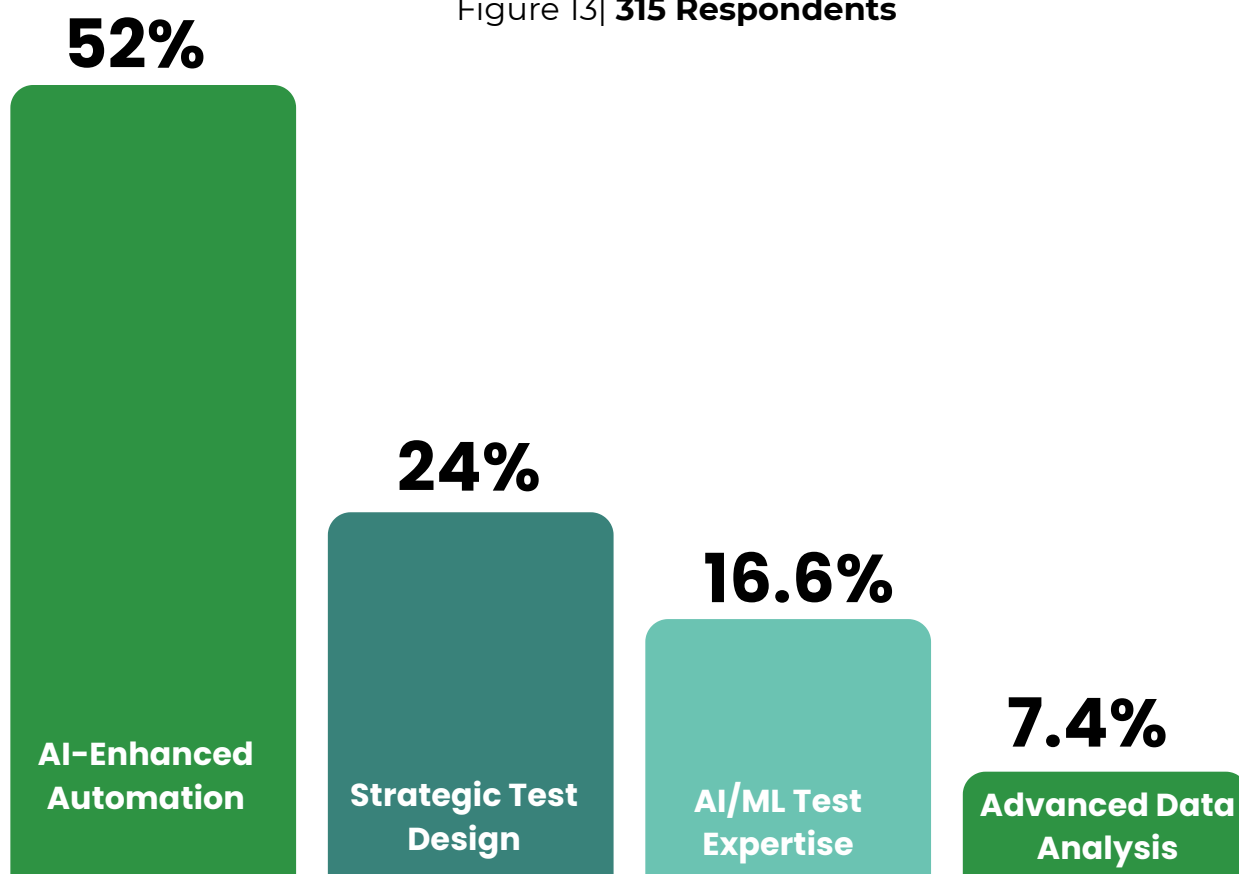
The data shows one clear key takeaway: AI is increasingly capable in "what" to test, but struggles with "**how**" and "**why**." That gap is where testers come in.

What is striking is that despite this reality, many AI tools still focus primarily **only** on accelerating test execution, not empowering or aiding strategic QA roles, possibly due to **capacity constraints**.

13

What's the MOST Crucial Individual Skill for QAs to Thrive With AI's rise?

Figure 13| 315 Respondents



The Rise of AI-Based Automation

The rise of AI-enhanced automation is redefining what QA is in the enterprise technology landscape.

It was surprising to note that **52%** respondents cited AI-enhanced automation as the most critical focus area, reflecting a deeply pragmatic view that AI is, first and foremost, an **execution** accelerator. It helps reduce redundancy, boost coverage, and speed up cycles.

However, we know that it is only one dimension of the transformation underway.

Beneath that surface, a more strategic shift is emerging, one that would require QAs to upskill themselves from being mere executors to quality architects of intelligent systems.

The Strategic Value of QA in an AI-Augmented Future

• AI/ML Testing Expertise (16.6%)

As organizations scale AI adoption, a pressing need arises: who validates the model? **16.6%** respondents now believe that the most important skill with AI's rise lies in **validating algorithms, training data, model bias, drift, and AI hallucinations**. This makes AI/ML literacy a premium skill set for future QA leaders, one that sets them apart from other QA testers.

• Strategic Test Design (24%)

As systems become more interconnected, dynamic, and personalized, the ability to think strategically about test coverage, user journey mapping, and business-critical failure paths becomes a core competency. Even with AI's rapid rise, strategy as a skill will remain invaluable, as it requires a healthy combination of intelligence, intuition, and experience.

• Advanced Data Analysis (7%)

Another skill that was interestingly underweighted, yet crucial. Advanced data analysis remains the foundation of intelligent QA. AI-powered test analytics, anomaly detection, defect clustering, and release-risk scoring require testers who can interpret patterns, extract insights, and turn data into decisions.

This **7%** figure likely reflects a skills gap more than a lack of importance, a space ripe for upskilling and leadership.

Today, AI can generate test cases, identify patterns, and prioritize test cases. These traditional markers of QA expertise are no longer differentiators. **But not strategy** as an adaptive, continuously-learning discipline. Strategy that interprets the signal from the noise. That prioritizes why to test, aligning software quality with

business-critical outcomes, user expectations, and systemic trust. **Meanwhile, the rote is going away.** The repetitive is being absorbed. Testers who centre their expertise solely on execution pipelines risk being outpaced not only by AI, but also by peers who learn to work with it.

Apart from technical knowledge and expertise, QA testers should equally focus on soft skills such **as communication, critical thinking, the hunger to continuously learn, and staying updated with industry trends.**

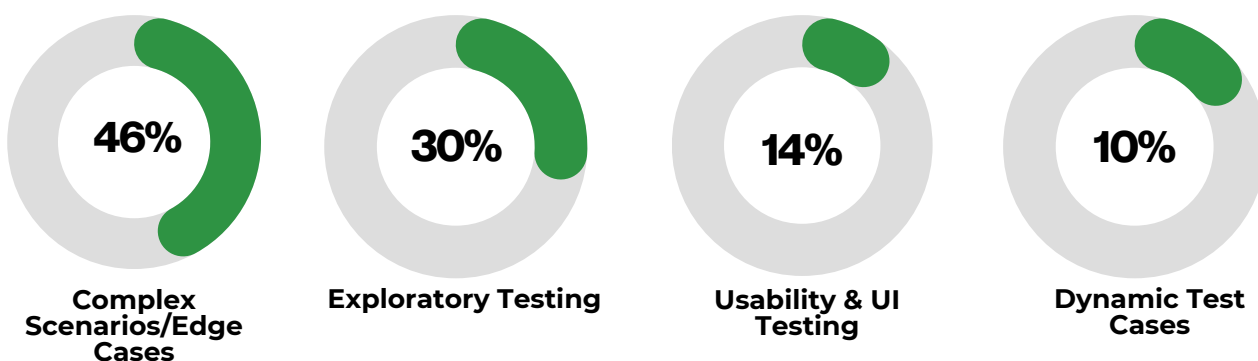
In this landscape, the QA function is a strategic assurance engine that is always running, always learning, always adjusting its risk lens.

AI will take over execution. But interpretation, prioritization, and contextual judgment? That's still human territory.

14

Where Do You Think Manual Testing is Still Crucial?

Figure 14 | **373 Respondents**



Manual Testing Will Remain Strategic

Exploratory Testing at 46%

Exploratory testing is a process of discovery. It thrives in ambiguity, allowing testers to follow intuition, chase anomalies, and surface issues not anticipated during planning. Especially valuable in early-stage development or rapid prototyping, exploratory testing remains one of the few areas where critical thinking outperforms coded logic. While exploratory testing is perceived as ad-hoc testing, it is done only after a clear plan in terms of scope and

resources are framed.

Usability and UI/UX Validation (14%)

Design is subjective, and human experience is emotional. While AI can check for functional correctness or visual mismatches, it cannot truly assess satisfaction, friction, or intent. That's why manual validation of UI and usability remains a critical human task.

Dynamic Test Cases (10%)

Highly dynamic workflows or rapidly evolving applications often require human intervention to adapt test logic spontaneously. In such cases, automation can lag behind frequent

updates, whereas manual testers can quickly adjust and interpret irregular outputs, reducing blind spots during test cycles.

Strategic Takeaways

- Enterprises should invest in automation for speed, but retain human-led testing where nuance, risk, and ambiguity dominate.
- That means developing **hybrid test strategies**, where human testers act as risk hunters and AI for more routine tasks.
- Upskilling testers for strategic thinking and test investigation is vital, especially in early design phases.
- Embedding exploratory testing into agile cycles as a non-negotiable quality activity.



5 Strategic Shifts Reshaping the Test Automation Landscape

Trust and Scalability Still Hold the Keys for Test Automation

Unified platforms that bridge capability gaps without losing depth will define the next generation of test tooling, saving time by eliminating tool sprawl. For vendors offering unified platforms, trust is the primary barrier that needs to be broken.

AI is Rewriting QA, But It's Not Writing Testers Off

AI literacy will become a core QA competency that testers need to survive. Leaders should prioritize upskilling in strategy formulation, AI workflows, bias handling, model validation, and system thinking.

Manual Testing Is Getting Smarter

The new manual is about intelligence. Leaders must stop **measuring manual vs. automation as a percentage**, and start measuring each by its value to risk coverage. Hybrid models that embed exploratory phases into sprints will outperform rigid pipelines.

Unified Testing Platforms Remain an Untapped Opportunity

There's a growing readiness for unified platforms that offer capability parity. The market is primed for tools that reduce cognitive switching and support plug-and-play integration without sacrificing specialization.

The Tester's Role is Moving From Execution to Intelligence

Testers will be measured by critical thinking; how well they interpret quality signals, design intelligent pipelines, and align testing with product outcomes. Across every response, one theme was clear: **value now lies in insight, not mere execution.**

CONCLUSION

QA is being recast as an engine of product confidence in an AI-augmented world. Across every signal, from the prioritization of speed and stability, to the hybrid rise of codeless tools, to the nuanced caution around AI outputs, one truth holds: **Technology is changing faster than trust.** And this gap is strategic.

Teams still default to fragmented tools because **specialization outpaces integration.** They **trust AI only when it's curated by human expertise.** They value test automation only when paired with contextual judgment. They embrace speed, but not at the cost of control.

The future will belong to organizations that stop treating QA as a cost centre or compliance necessity and start investing in it as a decision-making intelligence layer.

To future-proof QA in the years ahead, enterprises must make three bold shifts:

A) From tools to ecosystems

QA should be about building modular, adaptive ecosystems that blend AI, analytics, and human expertise as foundational capabilities.

B) From execution to foresight

Automation is not about replacing manual effort. It's about redirecting that effort toward what matters: predictive insights, failure anticipation, and continuous quality at scale.

C) From scripts to strategy

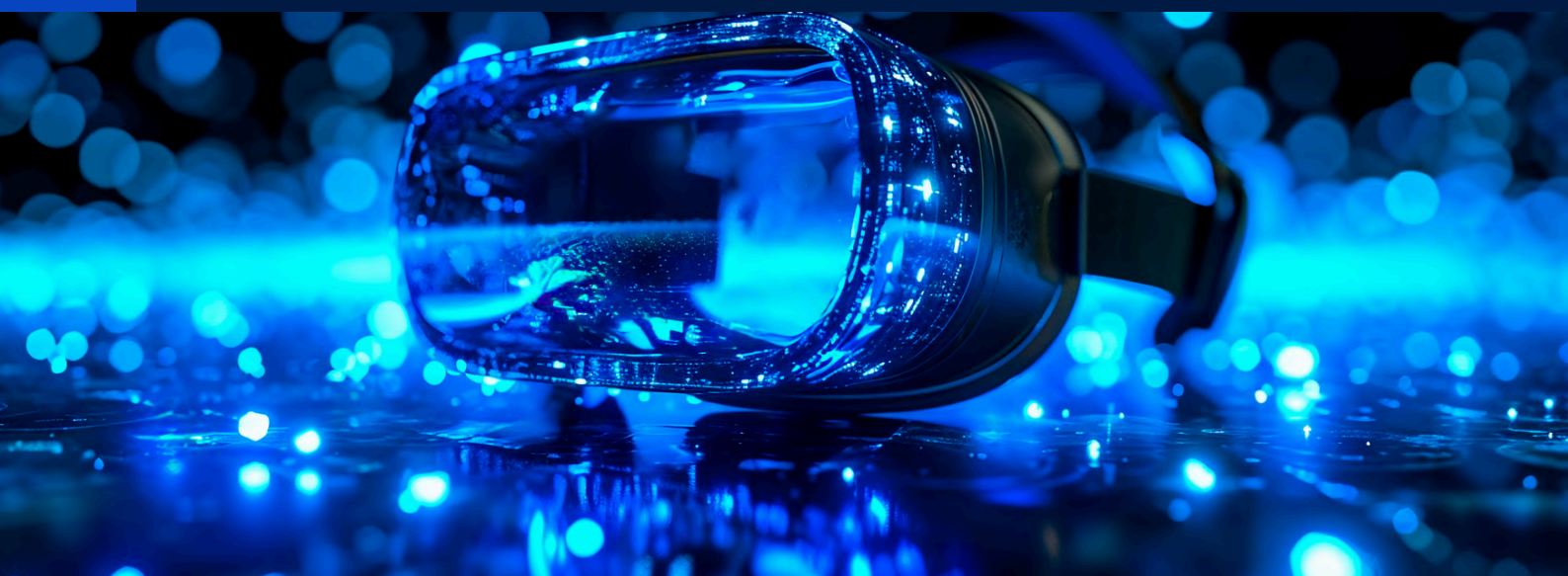
Testers will need to upskill not just in Gen-AI or codeless platforms, but in areas like model validation, data bias, and domain fluency. These are the new levers of differentiation.

Final Word: Quality Assurance is No Longer a Phase. It's an Intelligence Layer.

Modern QA is not just validating what's built. It's shaping what gets built. It's predicting bugs.

It's accelerating what matters. It's time to rethink how we define, deliver, and defend quality in the era of AI, acceleration, and always-on experiences.

The future of QA is proactive and intelligent. And it's already here.



CONTACT US TO LEARN MORE!

Our Product

- Tenjin Workbench

Our Services

- Consulting
- Testing Services
- TCOE

Contact Us



info@yethi.in



<https://yethi.in/>