VALA

AI doesn't remove classic QA problems, it makes them visible

VALA WHITEPAPER ON AI IN QA IN 2026

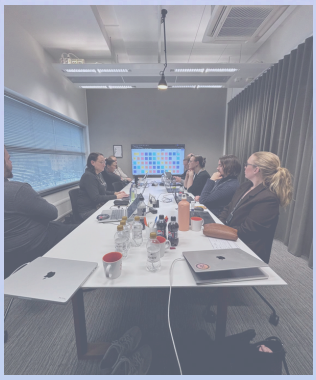
AUTHOR
Virpi Tuohisto

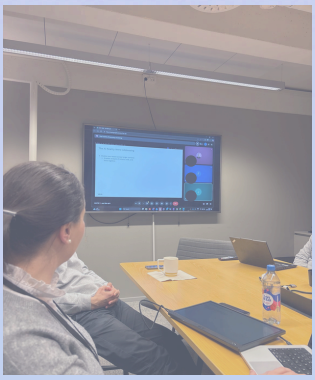
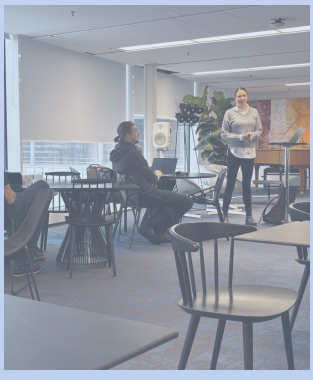
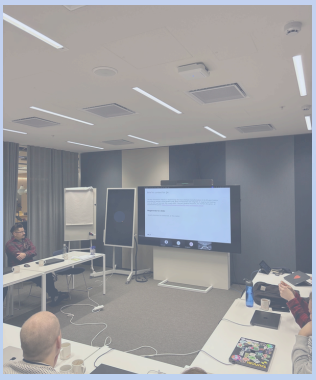
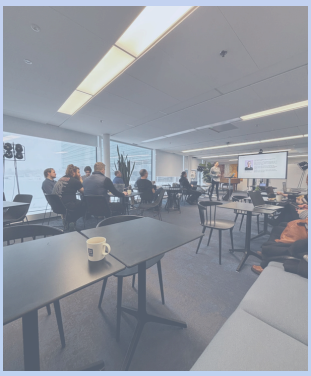
PUBLISHED
Mar 5th, 2026


AI
QA

Table of Contents



Why we wrote this

This publication offers a practical view of AI in QA in early 2026, based on real experiences from VALA' s clients. It explores current use cases, challenges, constraints, and opportunities related to applying AI in quality assurance.

The base of this paper was formed on observations from 16 artificial intelligence in quality assurance workshops facilitated with different teams and organisations. The workshops included testers, developers, test leads, business representatives and engineering managers.

This publication is not an AI hype piece. It does not claim universal truths or provide a single "right model". It documents patterns observed across workshops and customer discussions, and what those patterns may mean for leaders who buy QA and software quality services.

EDITOR-IN-CHIEF

Virpi Tuohisto

HEAD OF AI & TEST AUTOMATION AT VALA

About the workshops

These AI in QA workshops were designed to understand each team's quality challenges, share current information about AI in QA, and explore where AI could realistically support their work. The aim was to clarify constraints, opportunities, and priorities while outlining a few concrete AI initiatives for further exploration.

The working environments of the participants' varied significantly. Some teams worked in technically demanding systems with hardware dependencies, complex integrations, and large data volumes. Others operated in fast-moving development contexts with evolving requirements and limited time for test planning and review. In some cases, automation foundations were strong, but test sustainability, prioritisation, or visibility remained open questions.

> A team described a system that reacts to human eye movements, where behaviour varies and expected results are not fixed. This makes traditional test automation difficult. Looking ahead, AI raises open questions in this context: could it help learn behavioural patterns over time, simulate realistic variations in eye movement, or support testers in exploring "unexpected but acceptable" behaviour?

In the workshops, the discussions were not one-directional. While AI trends and use cases were introduced, we also learned from the participants; how AI was already being used in practice, where expectations were high, and where everyday realities shaped adoption. In several cases, potential AI initiatives were drafted during the workshop; in others, they were defined afterwards based on the discussions. Together, these discussions and drafted initiatives provide the foundation for the observations and themes explored in the following chapters.

Industry	Number of participating organizations
Information and communication	4
Administrative and support service activities	2
Human health and social work activities	2
Manufacturing	2
Wholesale and retail trade	2
Education	1
Electricity, gas, steam and air conditioning supply	1
Financial and insurance activities	1
Other service activities	1

Data from 16 Client Workshops (Jan-Feb 2026)

Problematic areas of testing in workshop companies

During the discussions, it became clear that the software context varied, but the underlying challenges were often familiar. Despite differences in domains and technical environments, similar challenges surfaced across teams:

Manual testing remains important in complex scenarios, especially where integrations, hardware elements, or long end-to-end chains are involved.

Test design and automation maintenance are laborious, often competing with delivery pressure.

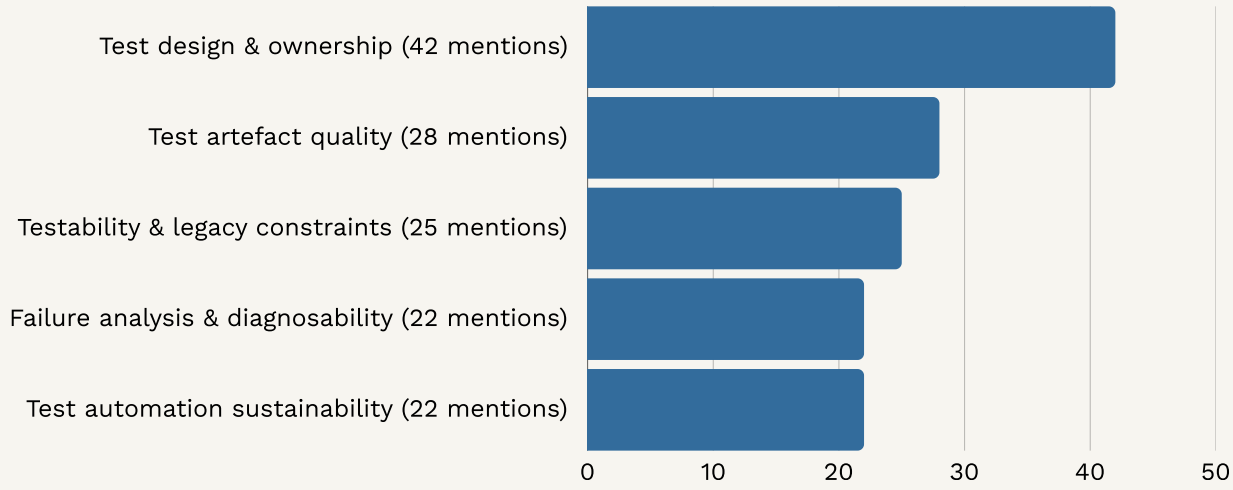
Requirements and acceptance criteria vary in clarity and maturity. This directly affects testability, prioritisation, and traceability. When expectations are vague or fragmented across tools, testing becomes reactive rather than methodical.

Test coverage and overall quality visibility are ongoing challenges. Teams struggle to answer basic questions: What is realistically covered? Where are the biggest risks? What changed in this release? What are the dependencies?

Many teams already used AI in some form. However, usage was often fragmented and difficult to prioritise. Rather than removing classic QA challenges, AI tended to make them more visible. When AI was asked to generate tests, summarise logs, or suggest priorities, underlying gaps in clarity and data surfaced quickly; how can AI successfully generate a test when no one has defined what's expected from the solution?

When workshop observations were categorised, certain themes appeared more frequently than others.

The most commonly raised challenging areas were:



The most frequently mentioned areas; test design and ownership, artefact quality, testability, diagnosability, and automation sustainability, are not new or AI-specific challenges. They represent long-known problems of software development that require structural clarity, shared practices, and technical discipline. These are not problems that generative AI can solve just like that. But embedded into a deliberate transformation of practices, AI may help strengthen clarity, visibility, and decision-making over time.

What teams expect from AI

As mentioned earlier, many participants were already experimenting with AI in some form. At the same time, they often expressed a desire for a clearer and concrete answer to AI in QA. Many asked, either directly or indirectly, for the much-desired silver-bullet AI tool. A tool that could be adopted easily and would meaningfully reduce effort or improve quality across testing activities. These expectations were shaped by real pressures. The participants described increasing system complexity, tricky requirements, heavy manual testing effort, and limited time. AI was seen as a potential way to cope with these constraints.

At the same time, discussions highlighted a recurring tension;

> **Many of the challenges teams hoped AI would address were closely linked to established QA and software development practices.**

This also shapes how useful any AI support could realistically be.

Rather than converging on a single “QA AI tool,” questions often shifted toward how current practices might need adjustment. In this way of thinking, AI is no longer viewed as something to add on top of existing work, but as something that makes current practices visible and can serve as a driver for more disciplined and effective ways of working.

Why most AI pilots fail to deliver impact

What the participants experienced in the workshops reflects a broader pattern seen in many organisations. Widely referenced research from MIT (State of AI in business 2025, MIT NANDA, 2025), suggests that most enterprise generative AI pilots failed to produce measurable business impact. Even months after its publication, this research continues to be frequently cited in discussions about AI transformation. It appears to resonate as many organisations recognise the same pattern: initial experimentation is easy, but translating pilots into sustained, measurable value is significantly harder.

> "Despite $30-40 billion in enterprise investment into GenAI, this report uncovers a surprising result in that 95% of organizations are getting zero return. The outcomes are so starkly divided across both buyers (enterprises, mid-market, SMBs) and builders (startups, vendors, consultancies) that we call it the GenAI Divide. Just 5% of integrated AI pilots are extracting millions in value, while the vast majority remain stuck with no measurable P&L impact."
>
> MIT NANDA| *State of AI in Business 2025*

> **Teams did not struggle because AI tools were ineffective. They struggled because AI was added on top of existing complexity, unclear requirements, and limited time for reflection and review.**

The reasons identified are not primarily technical. Instead, pilots struggle because AI is introduced without sufficient integration into existing processes, ownership structures, and decision-making practices. Learning gaps, unclear goals, and weak prioritisation prevent early experiments from turning into sustained value.

This wider signal helps put workshop observations into context. Teams did not fall short in AI adaptation, because AI tools were ineffective; rather, progress remained limited as AI was introduced on top of existing complexity, unclear requirements, and limited time for reflection and review.

Especially in a QA context, starting with small, clearly defined AI use cases was often seen as a prerequisite for success. Focused experiments, such as supporting log analysis, drafting test ideas, or improving test data quality, are easier to evaluate, measure, and adjust. This is not a lack of ambition, but a recognition that sustainable impact depends on clarity and ownership.

Seen this way, the high failure rate of AI pilots is not surprising. AI exposes organisational friction early. Where structures are unclear, pilots stall. Where ownership and priorities are explicit, small experiments have a better chance to evolve into something beautiful.

AI becomes less a solution and more a diagnostic tool.

AI doesn't break QA. It exposes it.

In many of the workshops, the participants described situations where AI highlighted problems they had already recognised but learned to work around. When AI was asked to generate tests, analyse results, or suggest priorities, gaps in clarity and structure had become visible quickly.

A team explored using agents to generate automated test cases, quickly producing a large number of tests. The tests were generally valid, but their quality and value varied. This raised a practical question: when test generation scales easily, who reviews the results? The discussion also highlighted that review practices themselves need to be intentionally designed; whether the review is done by humans, AI, or both.

Unclear or evolving requirements led to inconsistent AI-generated test ideas. Weak traceability between tickets, tests, and system behaviour made it hard to validate AI suggestions. In complex systems with many dependencies, AI trials have helped to surface interactions, but did not reduce the underlying uncertainty about impact or risk.

AI reacts directly to the quality of input it is given. When acceptance criteria are clear, test intent explicit, and ownership defined, AI-supported work is easier to review and reason about. When these foundations are missing, AI outputs tend to amplify ambiguity rather than resolve it. This leads to a useful reframing. AI is not breaking existing QA practices. It exposes where those practices are weak, implicit, or stretched thin. The discomfort that often follows is not a failure, but a signal.

Seen this way, AI becomes less a solution and more a diagnostic tool. It reveals how quality work is actually organised — and where assumptions, shortcuts, or gaps have accumulated over time.

A team discussed using AI to generate regression tests from production bugs to reduce manual effort. The idea quickly highlighted a limitation: bug reports varied in quality and structure. Without clearer reporting and context, automatic test generation would be hard to trust. The discussion highlighted how foundational practices often need strengthening before AI-driven innovation.

How AI is used in QA today

A recurring pattern in these discussions was that AI was rarely referred to as an autonomous tester. Instead, AI was described as a support tool embedded into everyday QA work.

Common uses included:

Drafting test ideas from acceptance criteria

Summarising logs and test results

Helping reason about system behaviour

Supporting test planning and reviews

These uses were typically exploratory and assistive. AI helped reduce writing effort, cope with large information volumes, and support thinking under time pressure. Final decisions, validation, and ownership remained with people.

When AI was pushed further, without clear rules or review practices, trust quickly became an issue. In practice, AI increased cognitive support more than test execution. It helped teams think faster and see more, but it did not remove the need for test design, prioritisation, or judgment.

More output, harder prioritisation

In many of the workshops, the participants described how AI made it easier to produce more test ideas, scenarios, and suggestions. Drafting test cases from requirements and code, generating checklists, or analysing changes became faster.

At the same time, deciding what to focus on became harder. AI often produces many technically plausible options, but does not reduce the need to choose what matters most. This highlighted a familiar challenge: AI does not solve prioritisation problems, but makes them visible. Without clear agreements on risks, priorities and value, faster output quickly turns into noise.

AI often surfaces plausible options, but does not reduce the need to choose what matters most.

AI was most helpful when it supported existing

rather than trying to replace them.

When AI helps – and ~~when it doesn't~~

A common theme in the workshops was that AI was seen as most helpful when it supported existing quality practices rather than trying to replace them. Teams with clearer acceptance criteria, shared testing principles, and established review habits had found it easier to use AI in a meaningful way. It acted as a sparring partner, not a decision-maker. Where these foundations are weak or implicit, AI often creates more work. Generated outputs are harder to review, trust is low, and teams struggle to decide what to keep or discard.

The difference is not the tool or the model. It is the surrounding discipline: AI amplifies what is already there; for better or for worse.

Testing AI-native systems

In some workshops and other customer cases, the discussion extended beyond using AI to support testing, toward testing systems where AI is an essential part of the product itself. These systems behave differently from traditional software. Outputs may vary, behaviour may change over time, and defining exact expected results becomes harder. This creates tension with established testing practices that rely on repeatability and clear assertions.

Among the participants, AI-native implementations were still relatively limited. Most AI-powered features in production were chatbots or AI-assisted functionalities embedded into otherwise traditional systems. However, several teams anticipated that more behaviourally adaptive and AI-driven systems would emerge in the near future, raising new questions for QA.

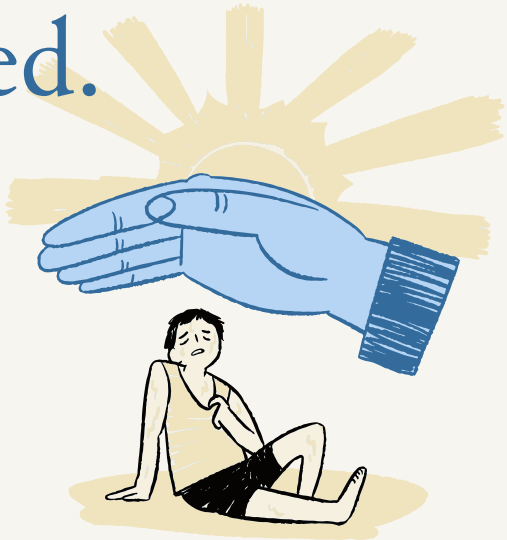
> In one customer case, the team raised a topic about systems that may evolve dynamically through AI in production. This led to a broader discussion: how do you assure quality when behaviour could change over time? Traditional pass/fail testing becomes harder to apply as variability increases.

> continuous observation, risk boundaries, and human judgment, rather than final release verification.

In safety-critical environments, such as industrial automation or medical systems, phenomena like hallucination or behavioural drift are not minor imperfections but unacceptable risks. In these contexts, tolerance for unpredictability is low, and expectations for traceability and accountability are high.

From a QA perspective, this does not reduce the need for discipline. It increases it. Questions about acceptance criteria, quality attributes, risk, and ownership become harder to avoid, not easier. Rather than introducing entirely new testing practices, AI-native systems tend to make existing ones more visible. Weaknesses in clarity, prioritisation, and responsibility surface earlier or at higher cost if left unaddressed.

Weaknesses in clarity, prioritisation, and responsibility surface earlier or at higher cost if left unaddressed.



When testing doesn’t end at release

During our sessions, many teams described situations where important quality signals appear only after release. Logs, test reports, monitoring data, and user behaviour already provide large amounts of information, but analysing it takes time and effort.

AI was discussed mainly as a support tool for post-release analysis: summarising logs, grouping results, and highlighting unusual patterns. The goal was not to replace testing, but to cope with volume and complexity. These discussions highlighted a shift in where testing effort is spent. Quality is not fully assessed at release time, especially in complex or evolving systems. Instead, testing continues through observation and interpretation of system behaviour.

This shift becomes even more relevant in systems that include AI-driven functionality. If behaviour can adapt, drift, or vary over time, post-release monitoring becomes part of quality assurance rather than an operational afterthought.

> “Quality is not fully assessed at release time, especially in complex or evolving systems.

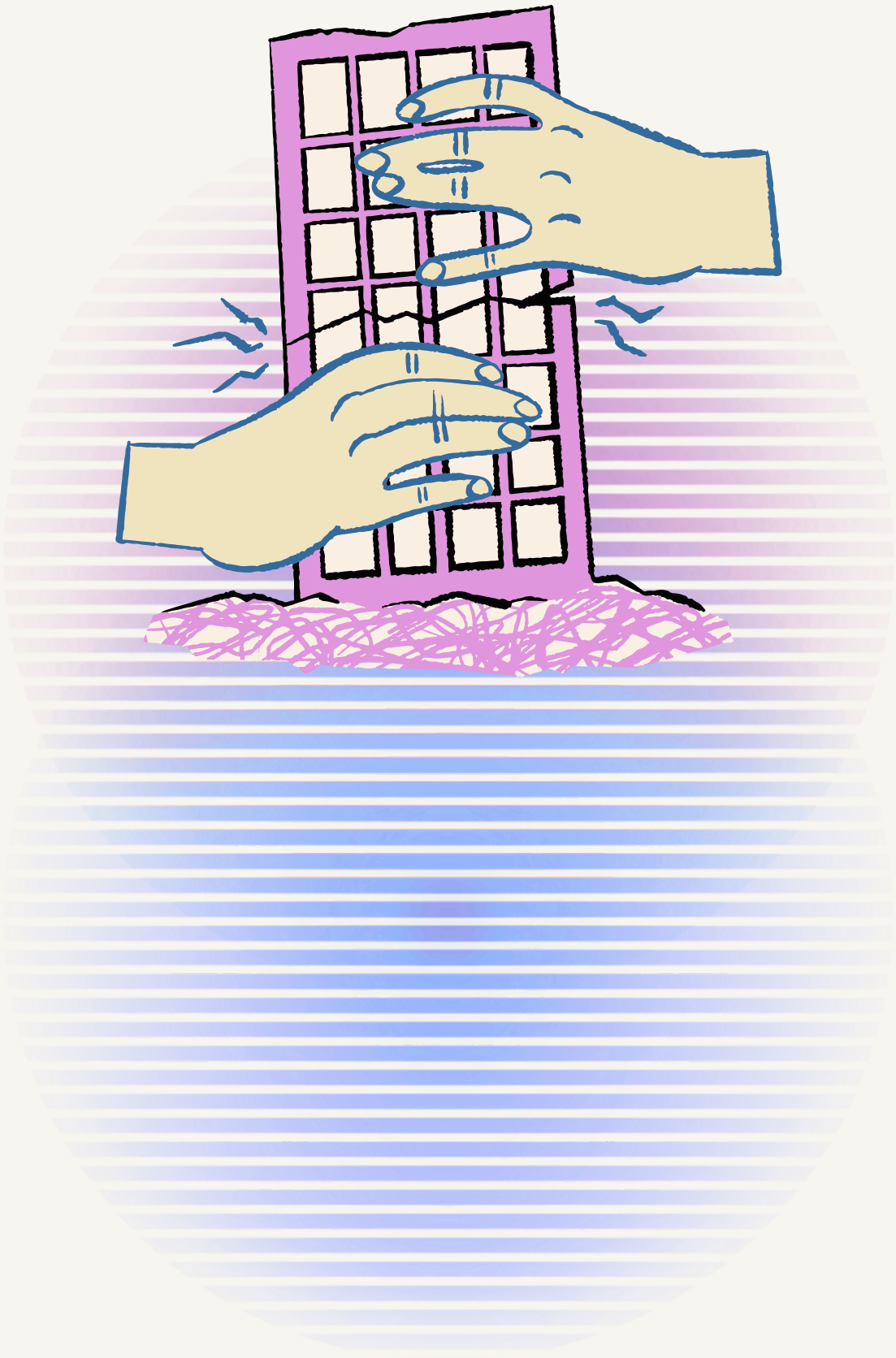
Are we ready to decrease building, and increase reviewing?

As AI becomes more capable of generating code, tests, and technical solutions, a shift in daily work becomes visible. Teams increasingly move from creating test assets to reviewing, adjusting, and approving AI-generated output.

This change is often framed as efficiency. In practice, it raises questions about learning, motivation, and long-term capability. Writing tests and building automation are not only delivery activities; they are how many professionals develop judgment, domain understanding, and problem-solving skills. There is also a creative aspect to this work; scripting, designing test logic, and exploring edge cases are often where engagement, learning and new ideas emerge.

Reviewing AI-generated work requires strong expertise, but it does not necessarily strengthen it. When relying too heavily on AI to produce initial solutions, the ability to critically evaluate those solutions may weaken over time. Sustained review-only work also raises questions about focus and motivation.

This is not an argument against AI support. It is a reminder that professional roles evolve with the tools they use. Organisations need to consider how hands-on building, expertise, and creativity are sustained as more work may shift toward review.



What leaders tend to underestimate

In many QA contexts, everyday challenges were rarely about missing tools. More often, discussions raised questions around ownership, prioritisation, review effort, and skills. AI introduces new kinds of output, but it does not remove the need for clear goals, shared rules, or accountability. In practice, it can shift effort rather than reduce it, increasing the importance of coordination and review.

Discussions also highlighted the role of expertise. AI-supported work still relies on people who understand the system and its risks. Maintaining that understanding requires continued hands-on involvement, even as AI support increases. This also applies to future capability: junior professionals need opportunities to design, build, and analyse, not only review, in order to develop into experienced engineers and testers.

Seen this way,

AI initiatives in QA are not just technical experiments.

They influence how quality work is organised, how decisions are made, and how responsibility is shared over time.

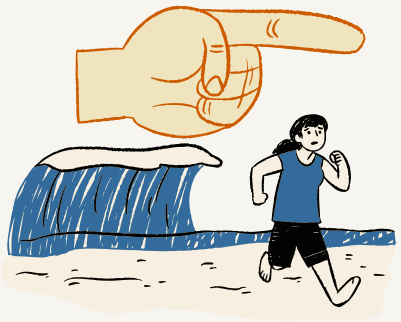

GREATEST PLACE TO WORK 2025
GREATEST PLACE TO WORK 2025

Conclusion: What this means if you buy QA

AI does not lower the bar for quality work. It increases visibility into how quality is achieved and where weaknesses exist.

Early value from AI is more likely to come from small, assistive uses that support existing practices, rather than from ambitious automation goals. These choices affect not only efficiency, but responsibility for quality outcomes.

AI-supported QA still requires clear ownership.



Decisions about acceptable behaviour, risk, and quality cannot be delegated to tools.

This also applies to evaluating AI itself: effectiveness needs to be assessed in context, based on whether AI improves clarity, decision-making, and confidence, or vice-versa.

Sustainable use of AI depends on maintaining expertise, review discipline, and shared understanding over time. AI initiatives should strengthen these capabilities, not quietly replace them.

“ Ultimately, AI does not replace the QA professional; it makes them more essential than ever.

Technology handles the execution, but the responsibility for quality – and the trust that underpins it – remains with us.

In 2026, software development is at a crossroads. While Generative AI promises to eliminate manual effort, 95% of corporate AI pilots fail to deliver real impact. Why?

This VALA whitepaper is not a hype piece. It is a practical analysis based on the real-world experiences of 16 organizations, exploring what happens when AI meets existing quality challenges. It reveals why in 2026 AI is a diagnostic tool rather than an automatic solution, and how the role of QA professionals is evolving from "builders" to critical "evaluators."

Read this to understand:

- Why AI testing fails without structural human discipline.
- How the explosion of code volume makes prioritization a strategic art.
- What buyers of QA must demand in an era where AI does the work, but humans carry the risk.
- Stop waiting and start understanding. Welcome to the new era of quality assurance.