# How AI copilots are transforming software testing

October 2025

# Introduction

As technology continues to evolve rapidly, it's imperative for software teams to maintain software reliability and deliver consistent quality. To meet stringent timelines and address the requirement gaps, quality analysis (QA) teams must maintain the quality of the software to prevent bugs at the time of its release. In such a scenario, artificial intelligence (AI) could become an important enabler for QA teams to automate repetitive tasks and leverage an AI copilot's support.

# Key challenges in software testing

Maintaining test automation framework is time-consuming and expensive as it requires a skilled workforce, costly tools and regular updates. Test execution processes often break down easily and generate incorrect results which slows down the software development process. Traditional tools can't keep up with quick changes in projects which require regular reviews. AI-based smart testing tools can assist testers in maintaining the automation framework and enable them to reduce the time and cost it takes to complete the process. An AI-driven QA copilot can monitor test frameworks, adapt scripts to changes, suggest fixes and reduce manual effort which helps in minimising cost and improving the efficiency of the process.

# QA and dev copilots – key differences

While dev copilots assist developers in writing and fixing code, QA copilots focus on testing. They help with tasks such as generating test cases, predicting bugs, and optimising test coverage. By automating repetitive tasks, they give QA teams more time to focus on tasks such as risk analysis and exploratory testing. The table below highlights the key differences between QA and dev copilots.

| Aspect | QA copilot | Dev copilot |
|---|---|---|
| Primary user | QA engineers/testers | Developers/software engineers |
| Main purpose | Assists in testing – generating test cases, scripts, data and coverage insights | Assists in development – suggesting, autocompleting and generating code |
| Input | Requirements, user stories, acceptance criteria | Code context, comments, function names |
| Output | Test cases, automation scripts, defect analysis, regression test selection | Code snippets, functions, boilerplate, bug fixes |

QA copilot is an AI-based assistant which produces test cases, code and reports to assist the testers and make the quality assurance process faster and more efficient.

These tools can:

- create and update test cases based on new features
- fix broken tests when the app changes
- suggest which parts of the app need more testing
- help write automation scripts faster
- spot gaps in test coverage.

Some AI copilots can even adjust tests for different user types or languages and manage testing across the whole build and release process, thereby saving both time and cost while minimising the scope of errors in the testing process.

# Use cases

QA copilots can be employed to complete various tasks such as test case generation based on requirements, writing automation scripts, providing test data, discovering coverage gaps and assigning regression tests. Some of the areas where QA copilots can be leveraged are:

**1.Test case design assistance:** Based on historical test data and well-defined user stories, AI copilots can help QA teams in creating the right set of test cases with better test coverage. They can re-process updated stories and modify the recommended test conditions in response to late requirements changes so that the test design stays updated as per the current requirements.

Based on the user stories defined on the agile board/test management tool testers can use fine-tuning models and prompt engineering to generate the test conditions along with the edge cases. Using natural language processing (NLP) and machine learning (ML) models, copilot takes reference from user stories, epics and features and suggests relevant test scenarios and test conditions along with edge cases. Before finalising, testers can review, update and approve the suggestions provided by the copilot.

**2. Automated defect triage:** AI agents help QA teams save time by automatically grouping and prioritising bugs based on previous data sets. Based on the severity and priority of the tasks, QA copilots provide suggestions on duplicate bugs and also highlight which aspect needs to be fixed on priority. The QA lead then reviews and confirms the suggestions given by the QA copilot. Further, QA copilots can also be integrated with test management tools where they can track the issues identified during the application testing stage.

**3. Test coverage gap analysis:** By using the continuous integration/continuous deployment (CI/CD) pipeline, AI copilot can identify the code changes based on which they can identify existing test cases which need to be updated or created to fill the test coverage gap. It can also be used during sprint reviews to assess test readiness.

Through application programme interface (APIs) or plugins, AI copilots interface with test management platforms and version control systems. They also scan differences or changes made in the source code during a commit in a version control system and correlate them with mapped test cases during code check-ins or pull-requests. It suggests pertinent test updates or new tests and highlights untested or inadequately covered code sections in real-time. Usually, the integration takes place during the pipeline's build or pre-deployment phase, giving teams the opportunity to respond to insights before the code moves to the staging or production environment. This integration guarantees that the quality checks are continuous and actionable within existing DevOps workflows.

**4. NLP script generation:** QA copilots turn the narrative requirements that people can understand into test scripts that the team can use to rapidly create tests. The tester still has to review, check the accuracy of the test script and make necessary adjustments before they can be used.

**5. Real-time test execution insights:** AI copilots review live test executions to detect issues like flaky tests or environment problems and provide answers which can help the testing teams to quickly solve the problem particularly for long CI/CD pipelines, frequent code changes and diverse testing environments. These copilots also help in overnight regression runs, where multiple teams are pushing codes through shared pipelines and manual failure analysis can be costly.

Through plugins or application programming interface (APIs), embedded in test orchestration platforms, AI copilots gather real-time metrics such as pass/fail rates, test durations, flakiness indicators, environment stability and root causes of failures. These insights are made accessible from centralised dashboards for easy comprehension.

In order to initiate a prompt reaction, alerts and recommendations are forwarded via email, chat platforms or in-app notifications. This integration connects QA and DevOps teams, enabling them to oversee and elevate test quality in real time.

# Bringing QA copilots into the workflow

AI copilots can be successfully integrated into the development lifecycle by taking the following measures:

1. **Define processes:** Organisations need to clearly define the QA processes that are most likely to get the maximum benefits from AI. This could mean working with AI vendors or creating custom solutions that match their specific requirements.
2. **Integrate QA copilots with the current workflow:** AI copilots can then be easily fused into the present workflows. For example, they may be connected to CI/CD pipelines, in which the commissioning of tests is done automatically as part of the development process. This ensures that quality checks are conducted in a sustainable and efficient manner while reducing the risk of errors at the production stage.

# How PwC can help

Testing is at the core of various technology projects. By developing and deploying AI agents as QA copilots, PwC's team implements  AI-powered strategies such as employing NLP, ML models and prompt engineering to not only create the basic test cases but also to produce edge cases and uncover untested areas. The team also assists testers in developing natural language script so that test cases can be presented to non-technical stakeholders.

# Conclusion

Deploying QA copilots with the right tools can help QA teams in delivering faster, more accurate software solutions. In future, the capabilities of QA copilots could be further extended to include facilitation of predictive testing, automatic defect resolution and effortless compatibility with newly developed AI-driven development pipelines. This shift will not only make testers more efficient but also accelerate the overall software development process.

# About PwC

**We help you build trust so you can boldly reinvent**

At PwC, we help clients build trust and reinvent so they can turn complexity into competitive advantage. We're a tech-forward, people-empowered network with more than 370,000 people in 149 countries. Across assurance, tax and legal, deals and consulting we help build, accelerate and sustain momentum. Find out more at www.pwc.com.

PwC refers to the PwC network and/or one or more of its member firms, each of which is a separate legal entity. Please see www.pwc.com/structure for further details.



## Contact us

**Debankur Ghosh**
Partner, Emerging Technologies
PwC India
debankur.ghosh.in@pwc.com

**Abhishek Verma**
Associate Director, Emerging Technologies
PwC India
abhishek.verma@pwc.com

**Krishanu Pathak**
Associate Director, Emerging Technologies
PwC India
krishanu.pathak@pwc.com

**Avantika Agarwal**
Manager, Emerging Technologies
PwC India
avantika.agarwal@pwc.com

## Contributors

**Bhagyashri Itankar, Anshuman Tripathy, Tamma Gnaneswara Reddy, Suranjika Dash**

pwc.in



HS/September 2025 - M&C 48723