

GUIDE

4 ways generative AI will transform the way you manage software testing

4 ways generative AI will transform the way you manage software testing

INTRODUCTION

With the meteoric advent of generative AI in the last 18 months, organizations on every level have seen incredible opportunities to improve their IT processes from a rapidly growing number of new tools.

According to [recent research from McKinsey](#), nearly 25% of C-suite executives are personally using generative AI tools for work, and more than 25% of companies using AI say generative AI is already on their boards' agendas for the upcoming year.

Generative AI can certainly be helpful in streamlining manual IT efforts and scaling software delivery, but how does one pierce all the buzz these days in identifying exactly where generative AI can help?

Generative AI, and what it can help software delivery teams accomplish, is evolving fast. That makes it important for everyone whose role touches quality engineering to understand the implications for how software is built and tested.

In this guide, we'll share four ways generative AI can boost your IT and quality engineering teams' testing efficiency and how test management will evolve as a result.

1. Accelerate test case generation

Generative AI represents a game-changer in the realm of test case creation. By harnessing the power of advanced algorithms and machine learning techniques, generative AI can automate the creation of test cases based on a variety of inputs, significantly reducing the manual effort required by QA teams. Theoretically, a tester could use many things to create prompts, from codebases to user behaviors, enabling AI systems to generate diverse and comprehensive test cases that cover a wide range of scenarios and edge cases. Today, test management tools exist that analyze connected requirements documents to generate test cases and test steps designed to validate those requirements.

Generative AI could also be used to generate already-automated test scripts by analyzing the frameworks used to automate previous tests, further streamlining the testing process and delivering time back to quality engineering (QE) folks. Regardless of what type of tests you're using AI to generate, it's important to note that they will often require human review and refinement to ensure accuracy and completeness. That's because AI currently struggles to grasp the complete context and intent behind the requirements. Therefore, it's best to think of AI as a powerful copilot in test case creation that can speed up your team's existing efforts.



Having AI writing the test cases and writing the test plans are the things that I'm excited about as a manager, especially tests built off existing tests' frameworks. Because we're moving so fast with deployments, I'm excited to see how these AI tools can increase testing productivity and save time and costs in the process.

— Ted Schneider | CEO at 7SIGNAL

2. Scale test coverage

With the right approach, generative AI can help testing teams achieve higher test coverage by generating a larger number of test cases in a shorter amount of time. Adequate test coverage can mean the difference between a successful and failed launch when identifying bugs and vulnerabilities across different parts of the software. The more tests you can generate that accurately map to your business-stipulated requirements, the more test coverage you can achieve, leading to improved software quality and reliability.

Scaling test coverage quickly will be vital to the success of your software delivery initiatives as AI sees wider adoption among developers. [A McKinsey study](#) shows developers may be able to complete coding tasks twice as fast with generative AI, implying there may be a corresponding increase in the amount of code generated. The result is that QA engineers will have to increase the speed at which they test and validate code. It will also be important to understand how AI can produce bugs in the code it develops, and to ensure rigorous testing processes are in place for AI-generated code.



We expect to see that ultimately, the preparation for testing will increase more than testing itself as generative AI becomes more widely adopted. It still takes time to develop good test plans, but with LLM in generative AI, you can define what you're looking to test (requirements) and it'll spit out some good test steps and test plans. Time spent in test design and review will increase alongside this increased test case output – validating that the test case accurately captures the business case, how it will be tested against, what the user flow will be for those things, etc.

— Ted Schneider | CEO at 7SIGNAL

2. Scale test coverage

To help scale test coverage at speed, generative AI can be a potent force. But it will be critical for quality engineers to carefully review its outputs, be aware of the nuances that AI can miss, and fill gaps in the test plan accordingly. AI may miss scenarios that cover edge cases or that reflect real-world user behavior, which only a human tester could understand and identify. Testing teams should think of AI-generated test cases as a rough draft that, like any piece of writing, needs a skilled editor before it's ready for wider consumption.



HOW WILL THIS CHANGE HOW A TEST ENGINEER SPENDS THEIR TIME?

With more time freed from administrative testing tasks such as building new test cases and maintaining test case library health, quality engineering folks can spend time in test strategy: which requirements are needed to map to, how to increase the communication loop with developers, creative ways to perform exploratory testing, and perhaps even considering new types of testing such as data integrity or performance testing.

— Paul DiGrazia | VP of Quality Engineering at Wolters Kluwer

3. Tailor test plans

Generative AI can help testers create test plans faster by intelligently identifying existing test cases in the library that map to a given requirement. This will help you more quickly identify relevant test cases and identify areas where coverage may be lacking. This tailored approach ensures that quality engineering efforts are focused on areas with less test coverage and not areas you've already tested, optimizing resource utilization and enhancing overall test effectiveness.

Identifying test cases in your library that will allow you to cover your requirement more quickly means that you can more quickly move onto creating new test cases for areas without adequate coverage. This will also minimize duplication and reduce the risk of unneeded rework.

In this fashion, generative AI can facilitate quicker test planning before a release, collating relevant assets before project launch so that your team can focus on filling gaps and ensuring that the overall test strategy is sound, ensuring coverage across business-critical requirements and edge cases, and performing any necessary exploratory testing to validate that the release will meet user expectations.

4. Analyze and improve test case libraries

Generative AI has the potential to help testing teams perform in-depth analysis of test case libraries to ensure they remain lean and relevant over time. Beyond quickly identifying existing test cases required for a given release, AI could be used in several ways to periodically clean up your test case library. Consider the following examples:

Identifying duplicates: If your test cases are housed in a central repository, an embedded generative AI tool could be prompted to identify duplicates. Within the context of a test management and reporting tool, you could then analyze which of the duplicated test cases are valuable and eliminate any that are brittle, flaky, or have not been used for a period of time, keeping only what's useful to your teams today.

Learning from past testing outcomes: In the future, it's likely that teams will apply generative AI to analyze executions for a given test plan or test suite over time, identifying commonly recurring patterns in test failures and suggesting opportunities for improvement.

Context awareness is critical

Note that each of the above use cases relies on the AI understanding the context of your requirements and test case library. Generative AI will be much more valuable to your testing practice if it can pull from your existing repositories. Offering that context will require your team to have a walled off generative AI tool whose outputs are private to your organization. Just as Stephen King wouldn't ask ChatGPT to edit the first draft of his latest novel, it would be unwise to feed an open-source AI tool the plans for a new product your company is building. Instead, opt for testing tools with built-in generative AI capabilities that will protect you from inadvertently disclosing confidential information or improving the model in a way that your competitors could take advantage of.

4. Analyze and improve test case libraries

AI MAKES QUALITY ENGINEERING INDISPENSABLE

Perhaps the most compelling aspect of generative AI in test management is its potential to shorten test cycles and create a faster path to release. By automating test case generation, scaling test coverage, and tailoring test plans, generative AI can shift your team's time from creating test cases to defining test strategy and ensuring it is executed against effectively.

This efficiency will become increasingly critical as AI-driven productivity gains on the development side pick up steam. As AI transforms the way software is built and delivered, strategic, quality-first thinking will become indispensable: Quality engineering teams will be faced with both a higher volume of code to be tested, as well as the new kinds of bugs and vulnerabilities that AI-generated code will inevitably introduce. Success will require scalable, automated systems for quality. A key piece of that will be understanding how to apply generative AI to the test cycle in ways that free up time to work on bigger, more meaningful problems.

