



Embracing Generative AI in Software Testing

Vishal Goyal
Chief Architect
Persistent





How to Increase Efficiency, Unveil New Insights & Drive Enterprise Value

Navigating the labyrinth of software testing, from the relentless pursuit of evasive bugs to orchestrating complex test matrices, takes resilience among software development teams. However, the emergence of Generative AI (GenAI) is sparking transformative zeal within developers in enterprises across various industries. GenAI provides developers with an astute software testing companion, one that not only amplifies efficiency but also weaves a tapestry of sophisticated analysis that was previously beyond their grasp.

At Persistent, GenAI revolutionizes our software testing toolkit with an array of Large Language Models (LLMs), offering interfaces as intuitive as ChatGPT, GitHub Copilot, and Cody, or as seamless as the APIs of Azure OpenAI, AWS Bedrock, and Google Vertex AI. Tailored to fit diverse use cases and preferences, these innovations promise to elevate the quality assurance landscape to unprecedented heights — and Persistent is collaborating with clients to bring them there.

In this white paper, we'll look at the advantages of GenAI in Software Testing, how to incorporate GenAI into an enterprise testing strategy, and where GenAI comes into play at every stage of the software testing lifecycle, based on Persistent's own groundbreaking work with GenAI.



What Sets **GenAI** Apart in Software Testing?

The core advantage of GenAI solutions like ChatGPT and Copilot lies in their sophisticated text processing and generation capabilities. This feature is particularly valuable in software testing, where the workload often involves handling large volumes of data, numerous test cases, and detailed bug reports.

Imagine an AI that can understand test scenarios, analyze bug reports, and even offer suggestions for improving test cases. In this context, Natural Language Processing (NLP) becomes crucial for converting user requirements into actionable and detailed test cases, thereby reducing the manual effort needed to interpret these requirements and allowing testers to focus more on actual testing and quality improvement.

Beyond text analysis and generation, GenAI tools can also perform predictive analysis based on their training data. For software testers, this translates to insights into possible points of failure, predictions about user behavior, and recommendations for

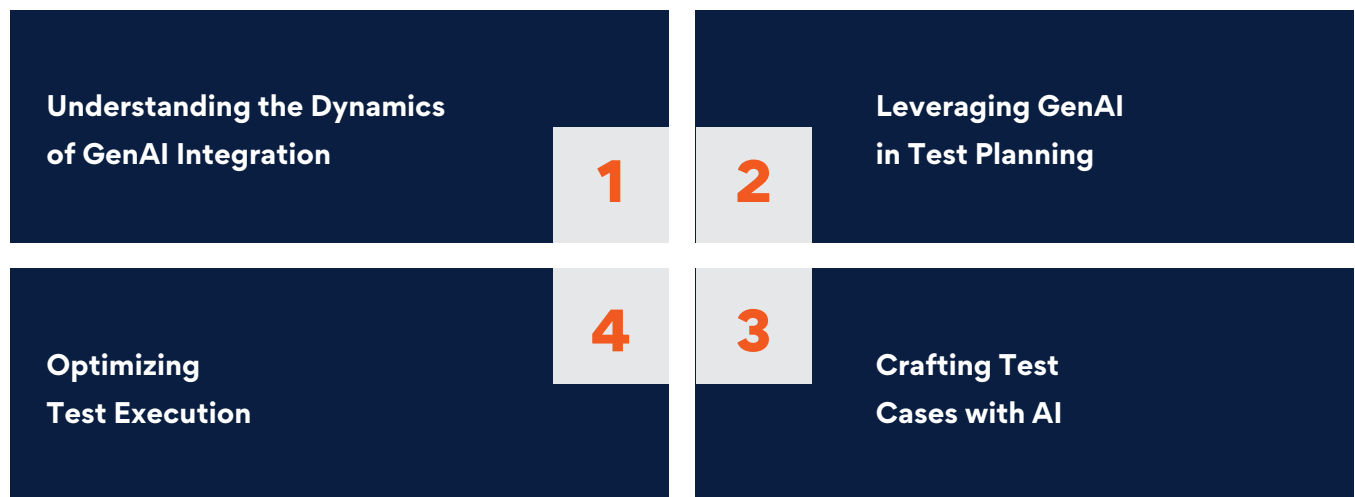
making test processes more efficient. These predictions are based on AI's ability to learn from past data and identify patterns, offering a way to anticipate issues before they arise.

A key capability of LLMs, which power GenAI tools, is their adaptability and capacity for learning. In the context of software testing, this means a GenAI tool can be customized to recognize specific software terminologies and workflows and adjust to the testing needs of different projects. The more these tools are utilized, the better they become at providing relevant and tailored support, making the software testing process more efficient and effective.

Creating a Software Testing Strategy with GenAI

In the realm of software testing, strategy is king. Crafting an effective testing strategy is a complex task that requires foresight, adaptability, and a deep understanding of both the software at hand and the methodologies best suited to test it. Overall, there are four primary areas that compose a GenAI-powered software testing project strategy (see below), each of which requires the input of stakeholders who have a deep understanding of a company's current / future software needs, the dynamics of its testing team, and what outcomes will have the most impact.

GenAI-Powered Software Testing Project Strategy



1 Understanding the Dynamics of GenAI Integration

- / **Initial Assessment:** The first step is to assess the specific needs of the testing project. What are the software's key functionalities? Which areas are most prone to errors? A full understanding of these factors helps to determine how GenAI can best be utilized, and level-sets a testing team on which software is in focus.
- / **Defining Objectives:** With ChatGPT, testing objectives can be more nuanced than just finding bugs. It can include enhancing the quality of test cases, ensuring comprehensive coverage, and even improving the efficiency of the testing process — all of which can be mapped out and signed off by all required stakeholders. Clarity on project objectives will result in clearer testing plans that produce beneficial results.

2 Leveraging GenAI in Test Planning

- / **Scenario Generation:** GenAI solutions like Sourcegraph Cody and GitHub Copilot can assist in producing a wide range of testing scenarios, especially those that mimic real-world usage. This ensures that tests are not just technically sound but also relevant to the end-users.
- / **Risk Assessment and Prioritization:** LLMs' ability to process vast amounts of data can be leveraged to identify potential high-risk areas in an application. This helps in prioritizing tests to focus efforts where they are most needed.

The table below is an example of test scenarios generated using Sourcegraph Cody. This enhances the quality of test cases by performing a very thorough testing of any application.

Test Case ID	Test Scenario	Expected Result	Test Type	Priority
TC001	Navigate to the Welcome Screen	The page is divided into two sections. The left section is blank. The right section has a blurred mountain image.	UI	High
TC002	Verify Welcome Back! text	The text Welcome Back! in bold is present on the left section.	UI	High
TC003	Verify arrow next to Welcome Back! text	An arrow in a small circle pointing right is present next to the text.	UI	High
TC004	Verify text below Welcome Back!	The text Please click on the Sign-in button to login to the ExtenSURE portal is present below the Welcome Back! text.	UI	High

Test Case ID	Test Scenario	Expected Result	Test Type	Priority
TC005	Verify Sign-in button	A blue rectangular button with white text Sign-in with Microsoft is present below the text.	UI	High
TC006	Verify button alignment	The welcome text, arrow, text and button are center aligned on the left section.	UI	Medium
TC007	Verify text in right section	The text ExtenSURE in bold white is center aligned on the right section.	UI	Medium
TC008	Verify text at the bottom	The text Digital Engineering Platform is present at the bottom of the page.	UI	Medium
TC009	Click Sign-in button	User is redirected to the Microsoft Sign-in page.	Functional	High
TC010	Enter invalid email format	Error message is displayed.	Negative	Medium

3

Crafting Test Cases with AI

- / **Automated Test Case Creation:** One of the most time-consuming aspects of testing is creating detailed test cases. GenAI can accelerate this process by generating initial test case templates based on the requirements and scenarios provided, which can then be modified as needed based on project objectives.
- / **Enhancing Test Coverage:** GenAI can also suggest additional test cases to cover edge cases or less obvious usage patterns, thereby improving the overall test coverage.

Below is an example of generating a Selenium script for performing functional testing of applications based on business requirements.

Prompt

I want you to generate test script in Selenium with Java. Firefox is my browser. My web application is **https://www.mystore.com/**. It has search functionality on its home page. Write the script so that these results are met when some one searches for **Red color throw pillows**.

1. Search for **Red color throw pillows** in the search box present on the home page.
2. Validate that “Showing results for **Red Throw Pillows**” is displayed on top of the search results and just below it the total results are displayed in figures.
3. Validate that all items displayed have red shades only.
4. Validate that all items have a price tag in dollars.
5. Validate that all items have a star rating.
6. Validate that the name of each item is displayed below it.

Response (Selenium Script)

```
public class MyStoreSearchTest {
    // Go to mystore.com
    driver.get("https://www.wayfair.com/");
    // Enter search term
    driver.findElement(By.id("searchInput")).sendKeys("red
    color throw pillows");
    // Validate search text
    String searchText =
    driver.findElement(By.className("search-
    info_query")).getText();
    assert searchText.equals("Showing results for \"Red Throw
    Pillows\"");
    // Validate total results
    String total = driver.findElement(By.className("search-
    info_total")).getText();assert Integer.parseInt(total)> 0
    // Validate red color and price
    for(WebElement item :
    driver.findElements(By.className("item-cell"))) {assert
    item.findElement(By.className("colors")).getText().contains
    ("Red"); assert item.findElement(By.className("prices")) !=
    null;}
    // Validate ratings, names
    for(WebElement item :
    driver.findElements(By.className("product-pod"))) {assert
    item.findElement(By.tagName("rating-stars")) != null;
    assert item.findElement(By.className("name")) != null;}
}
```

Below is an example of how testers can leverage GenAI to create complex API test cases step-by-step.

Prompt

I am a test automation engineer, recently onboarded on the project. My task is to automate API tests for an application by reusing an in-house developed API test automation framework. The test framework uses Rest Assured and is integrated with TestNG, Extent Report, AssertJ, etc. To begin with, list out the best API test automation practices to be followed.

GenAI Response

Here are some best practices for API test automation.

Use REST API testing frameworks like Rest Assured, Requests, etc. They simplify writing API tests.

1. Follow a BDD style for writing tests. Use Given / When / Then structure.
2. Reuse the test data using JSON files instead of hardcoding in tests.
3. Use Assertions to validate responses. Some good options are AssertJ, Hamcrest, etc.
4. Generate test reports using libraries like Extent Reports, Allure Reports, etc.
5. Group related tests into Test Suites for better organization.
6. Use before and after hooks to set up and tear down.
7. Follow naming conventions for tests, suites, etc.
8. Use version control system like Git to manage the tests.
9. Run tests in a CI / CD pipeline.

Let me know if you have any other questions!

The above example would include follow-ups through a series of prompts (examples below) to get final results as required.

- / How to go about API chaining? Need to save the value of 'id' received in the response as TestNG's ITestContext.
- / How to make sure multiple input parameters can handle all necessary data inputs?
- / How to go about testing the schema of this end point?

4 Optimizing Test Execution

- / **Scheduling and Resource Allocation:** GenAI can help plan the execution phase by suggesting an optimal schedule and resource allocation, balancing the need for thorough testing with the realities of project timelines and constraints.
- / **Feedback Loop for Continuous Improvement:** Integrating GenAI into the testing process creates an opportunity for a continuous feedback loop. Insights gathered from each testing cycle can be fed back into ChatGPT, helping to refine and improve subsequent rounds of testing.

Elevating Testing Quality and Advanced Applications with GenAI



Moving beyond basic test case creation and planning, GenAI's role in elevating the overall testing quality and its application in advanced testing scenarios is where its true value shines. Not only does it improve efficiency, but GenAI can also free up precious resources and time for software testing teams, so they can accelerate project completion and move onto other projects and tasks, delivering more overall value for a company.

Advanced Quality Assurance with GenAI

- / **Detailed Quality Checks:** GenAI can be used to perform in-depth quality checks on test cases and scripts, ensuring they are not only error-free but also aligned with best practices and testing standards.
- / **Enhancing Test Readability and Maintenance:** By refining test documentation, GenAI helps in maintaining clarity and consistency in test cases, making them more manageable and understandable for the entire team.

GenAI in Complex Testing Scenarios

- / **Handling Complex Test Data:** In situations where testing requires complex data sets, GenAI can assist in generating and managing this data, ensuring tests are both realistic and robust.
- / **Stress and Load Testing:** For more advanced testing like stress and load testing, it can help in designing and executing these tests, providing insights into system performance under various conditions.

Below is an example of generating a JMX script for performance testing and executing it using JMeter.

Prompt

Hello!, As a performance tester, create a .jmx script to perform load testing. Request details below:

Request1: "HomePage"

Method: "GET"

url: "https://www.mystore.com/"

Text Response Assertion: "Shop by Department"

Request2: "SearchItem"

Method: "GET"

url: "https://www.mystore.com/keyword.php?keyword=bags"

Text Response Assertion: "Showing results for "bags""

Request3: "AddToIdeaBoard"

Method: "POST"

url: "https://www.mystore.com/a/campaign_management/tag/return_tags_from_tracking_action" Payload:

{ "action": "AddToIdeaBoard", "data": { "sku": "W007365334", "page_type": null } }

Response (JMeter)

The screenshot shows the Apache JMeter 5.4.1 interface. The left sidebar displays a tree view of the test plan: Load Test > Ramp-Up Users > HomePage > Text Response Assertion > SearchItem > Text Response Assertion > AddToIdeaBoard > Response Code Assertion > CSV Data Set Config > Summary Report. The main window is titled 'HTTP Request' and shows the configuration for the 'AddToIdeaBoard' sampler. The 'Name' field is 'AddToIdeaBoard'. The 'Web Server' section shows 'Protocol [http]: https' and 'Server Name or IP: www.mystore.com'. The 'HTTP Request' section shows 'Method: POST' and 'Path: /a/campaign_management/tag/return_tags_from_tracking_action'. The 'Parameters' tab is active, showing a single parameter: '1 | {"action": "AddToIdeaBoard", "data": {"sku": "\${sku}", "page_type": null}}'. The 'Basic' tab is selected, and the 'Advanced' tab is also visible.

The screenshot shows the Apache JMeter 5.4.1 interface. The left sidebar displays the same test plan tree as the previous screenshot. The main window is titled 'Thread Group' and shows the configuration for the 'Ramp-Up Users' thread group. The 'Name' field is 'Ramp-Up Users'. The 'Action to be taken after a Sampler error' section has 'Continue' selected. The 'Thread Properties' section shows 'Number of Threads (users): 1000', 'Ramp-up period (seconds): 3600', 'Loop Count: Infinite' (checked), 'Same user on each iteration' (checked), 'Delay Thread creation until needed' (unchecked), 'Specify Thread lifetime' (checked), 'Duration (seconds): 3600', and 'Startup delay (seconds):'. The 'Summary Report' is visible in the sidebar.

Integrating GenAI in Security and Performance Testing

- / **Security Testing:** GenAI can be utilized to enhance security testing efforts by identifying potential vulnerabilities and assisting in creating tests that mimic various security breach scenarios. This provides a valuable layer of extra protection as software security threats are growing more numerous and sophisticated.
- / **Performance Optimization:** ChatGPT can analyze performance testing results, suggest optimizations, and predict potential performance bottlenecks.



GenAI in Test Automation, Debugging, and Problem-Solving



The adoption of GenAI in software testing extends beyond the realms of test planning and quality assurance. The technology's capabilities are particularly beneficial in test automation, debugging, and problem-solving — areas that are critical for efficient and effective software testing processes — by shifting a good majority of the work from overburdened humans to GenAI at machine speeds.

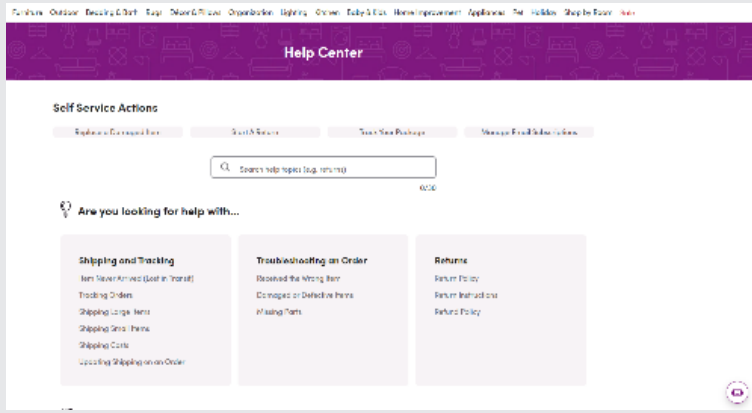
Enhancing Test Automation with AI

- / **Automating Test Scripts:** GenAI can assist in the creation and maintenance of automated test scripts. By generating code snippets or entire scripts, it accelerates the process of setting up automated tests, especially for regression and repetitive testing scenarios.
- / **AI-Driven Test Script Optimization:** Beyond just writing scripts, ChatGPT can analyze existing test scripts for potential optimizations, ensuring that automated tests are not only thorough but also efficient (and potentially faster) in their execution.

AI-Assisted Debugging and Issue Resolution

- / **Debugging Support:** LLMs' ability to process and interpret error logs and test results make GenAI an invaluable tool for debugging, as it can suggest probable causes of failures and recommend corrective actions.
- / **Problem-Solving Insights:** In complex debugging scenarios, GenAI's insights can guide testers toward effective problem-solving strategies, often highlighting issues that may not be immediately apparent to human testers.

Below is an example of gaining insights into accessibility checks against WCAG using GenAI. This helps generate multiple scenarios and improve quality of testing.



Prompt

I want to do comprehensive to do accessibility testing for this page of my web application. Assume there are no tools available in the market for doing this job. I want you to help me in this regard.

Response

- 1. Text Alternatives:** Check that all icons and images have alt attributes. For example, if there's an icon next to **Shipping and Tracking**, it should have an alt attribute that describes its function or meaning.
Resolution: Add meaningful alt text to each image and icon using the alt attribute within the image tag.
- 2. Keyboard Navigation:** Confirm that you can tab through all interactive elements: buttons like **Start A Return**, input fields, and links in the **Are you looking for help with...** section.
Resolution: Use proper HTML and ARIA roles to ensure all interactive elements are focusable and can be activated with the keyboard.
- 3. Readable Text:** Assess whether the text maintains high contrast with its background across all page elements to ensure legibility.
Resolution: Adjust colors to meet WCAG contrast ratio guidelines, ideally at least 4.5:1 for small text and 3:1 for large text.
- 4. Search Functionality:** Check that the search box has a properly associated label that screen readers can announce. Also, ensure that users can submit the search using the Enter key.
Resolution: Ensure the label element or aria-label attribute is used to identify the search field. Make sure the search can be submitted with keyboard input.
- 5. Heading Structure:** The page should have a logical structure, with one main h1 tag followed by H1-H6 tags as subheadings where appropriate.
Resolution: Use proper heading levels to convey the structure.

Streamlining Test Execution and Maintenance


- / **Execution Efficiency:** During test execution, GenAI can provide real-time support, offering suggestions for on-the-fly adjustments and optimizations to human testers for better test effectiveness.
- / **Maintenance and Updates:** As software evolves, so must the tests. GenAI can aid in the ongoing maintenance of test suites, suggesting updates and modifications to align with new features or changes in the software.

Collaborative AI in Quality Assurance

- / **Enhancing Team Collaboration:** GenAI's role extends to facilitating better collaboration within QA teams. It can serve as a central point for sharing insights, suggestions, and feedback, enhancing the collective testing effort.
- / **Integrating AI in QA Culture:** Embracing GenAI in software testing involves cultivating a GenAI-inclusive culture within Quality Assurance (QA) teams, encouraging the use of GenAI for more informed decision-making and innovative testing approaches. To enable a cultural change, GenAI should be positioned as a **virtual colleague** for humans involved in the testing process, one that is designed to make their jobs easier and more effective, not to replace their overall efforts. Once team members are exposed to GenAI's benefits, and how it can free up testing cycles and accelerate projects, many will see the inherent benefits of being paired with a GenAI colleague.



Challenges and Considerations in Integrating GenAI into Software Testing



While the integration of GenAI into software testing offers numerous advantages, it also comes with its own set of unique challenges and considerations. Like the use of most advanced technology, GenAI does have inherent limits and risks that must be considered in any GenAI-powered software testing project.

Understanding the Limitations of AI

- / **AI is a Tool, Not a Panacea:** It's important to recognize that GenAI, while powerful, is not a catch-all solution. It functions best when used as a complement to human expertise, not a replacement.
- / **Data Dependency:** The effectiveness of LLMs is largely dependent on the quality and quantity of the data on which they are trained. Inaccurate or biased data can lead to subpar suggestions and analysis, or suggestions or outcomes that are plainly wrong and not aligned to project goals.

Managing Change in Testing Processes

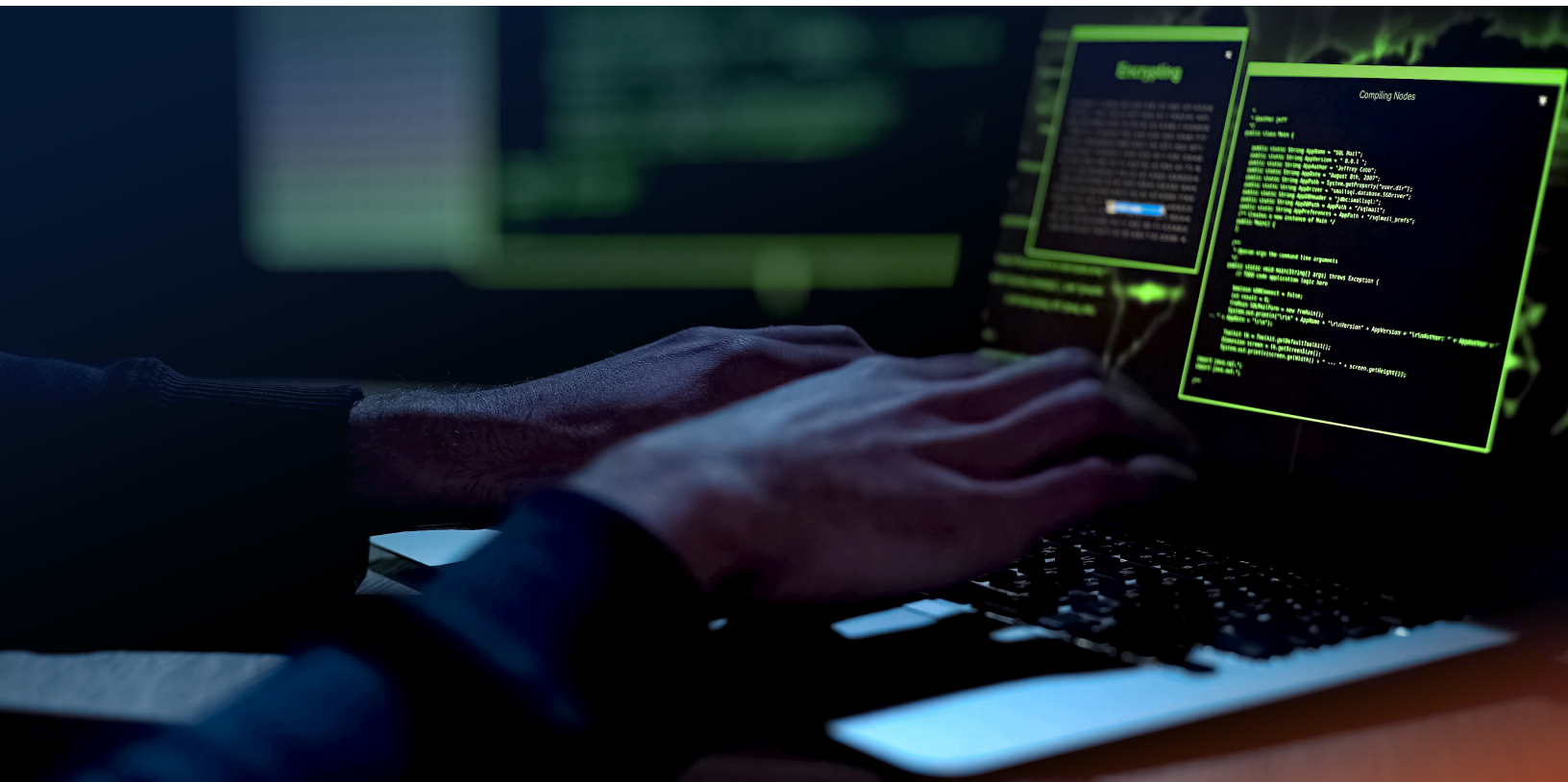
- / **Adoption Resistance:** It's an axiom to say that change is never easy. Introducing AI into established testing processes can meet with resistance among team members that are more comfortable with familiar or traditional training procedures. Proper training, education, and demonstrations of GenAI's value can help ease this transition.
- / **Adjusting Testing Strategies:** Integrating GenAI requires a shift in traditional testing approaches, necessitating flexibility and adaptability from QA teams. This adaptation will become easier over time as GenAI is brought into repeated software testing projects.

Ensuring Data Security and Privacy

- / **Handling Sensitive Data:** When using GenAI for testing, particularly in sectors with stringent data privacy regulations, ensuring the security and confidentiality of test data is paramount.
- / **Compliance with Regulations:** Compliance with data protection laws and regulations is crucial, especially when employing GenAI tools that process large amounts of potentially sensitive or proprietary information.

Technical Challenges and Resource Allocation

- / **Infrastructure Requirements:** Effective deployment of GenAI may require upgrades to existing technical infrastructure, including software and hardware resources, to run resource intensive LLMs.
- / **Resource Investment:** Cutting-edge technology inevitably comes with a cost. Implementing GenAI in software testing processes can necessitate significant initial investment in terms of time, training, and capital — investments that over time will become easier to justify as benefits in software testing are realized.



Embracing Change with GenAI

Incorporating GenAI into QA processes marks a pivotal advancement in the software development lifecycle, yet it demands an awareness of its limitations and a judicious approach to its deployment. The true value of GenAI emerges when its analytical capabilities are merged with human insight, forming a powerful synergy that can unlock unprecedented potential. The trajectory of AI within QA is laden with promise, propelled by ongoing GenAI enhancements that pave the way for more refined, efficient, and impactful testing methodologies. As we progress, the integration of AI in QA is expected to deepen, challenging us to continuously adapt our skills, strategies, and tools to stay abreast of technological progress.

ExtensURE from Persistent remains at the forefront of this dynamic field, vigilantly monitoring developments and innovating within the testing domain. By experimenting with a diverse array of tools and building various use cases, ExtensURE is dedicated to exploring the full spectrum of capabilities offered by these advanced technologies.

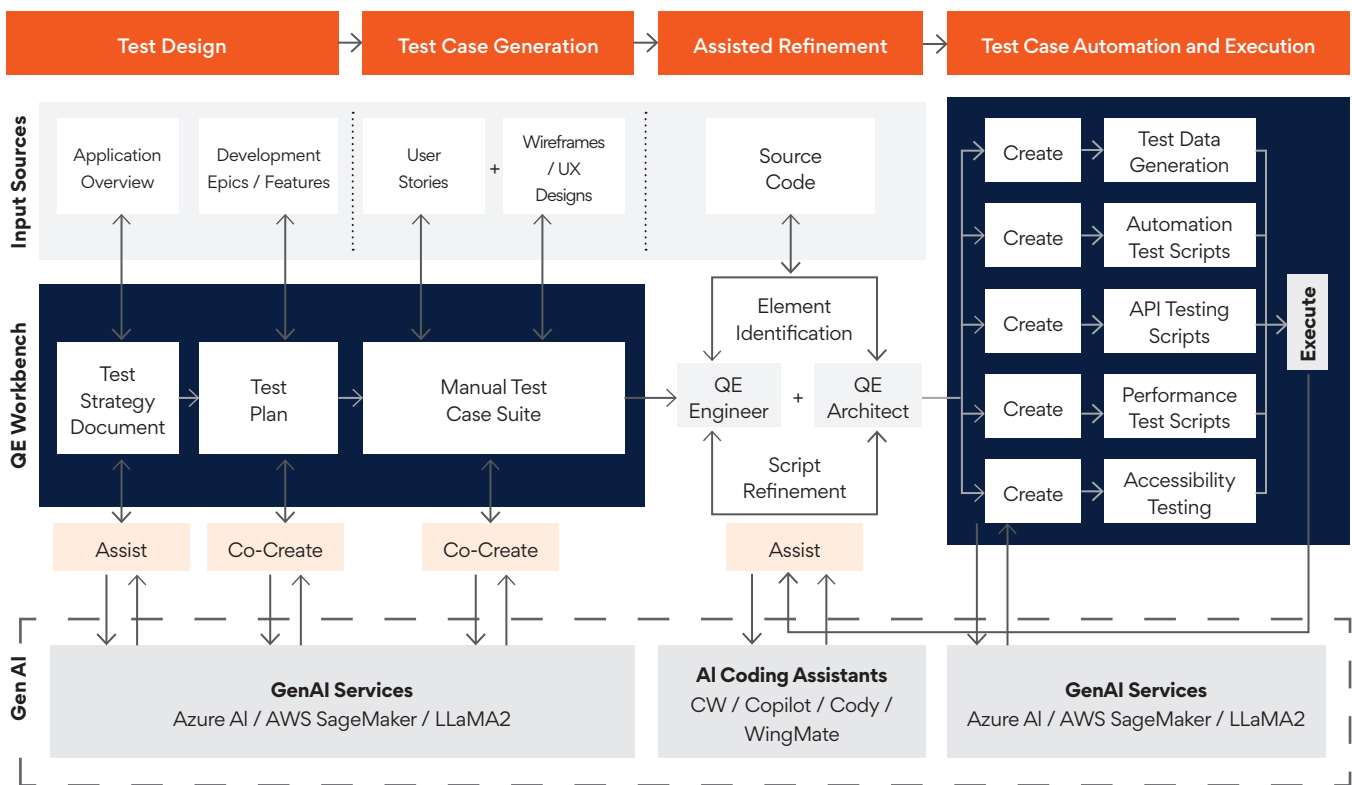


Figure: ExtensURE Process Flow for Testing Enhanced with GenAI.

Every business and organization that is looking at how GenAI can accelerate and optimize software testing should be exploring a way forward — and Persistent is engaged in such conversations with clients every day.

Reach out to Persistent and explore the possibilities of utilizing GenAI in software testing for your enterprise.

**See
Beyond.**
**Rise
Above.**

Learn more about our software testing approach
and ExtenSURE.

[Visit us](#)

About Persistent

With over 23,000 employees located in 21 countries, Persistent Systems (BSE & NSE: PERSISTENT) is a global services and solutions company delivering Digital Engineering and Enterprise Modernization. We work with the industry leaders including 14 of the 30 most innovative companies as identified by BCG, 8 of the top 10 largest banks in the US and India, and numerous innovators across the healthcare and software ecosystems. As a participant of the United Nations Global Compact, Persistent is committed to aligning strategies and operations with universal principles on human rights, labour, environment, and anti-corruption, as well as take actions that advance societal goals.

USA

Persistent Systems, Inc.
2055 Laurelwood Road
Suite 210, Santa Clara
CA 95054
Tel: +1 (408) 216 7010

India

Persistent Systems Limited
Bhageerath, 402
Senapati Bapat Road
Pune 411016
Tel: +91 (20) 6703 0000

