



Cloud Based Automated Functional Testing with AI Service Definition

GCloud15

Introduction

We deliver functional testing across IT systems and software applications - web, mobile and desktop. Utilising AI, specifically, Microsoft Playwright with Co-Pilot can not only speed up the build of automated functional tests but also open up automated functional testing to resources with limited experience of technical testing.

The Challenges of Functionally Testing Applications

AI & Automation – New Advances

Cucumber Gherkin promised to shake up test definitions by quickly being able to articulate the functionality and the intended outcome. Did it ever really live up to the promise? The bit that was missing was, now that we have a story, lets automate it. The automation process relied on the tradition approach with slowly building an automation framework, adding test cases as you went. The maturity of the completed automation test pack was dependent on two factors;

- The test tool and how it was able to support automated test cases. For instance, an automated test tool that was based on screen images could be built quickly, but when using screen locations rather than screen objects to locate an element on a screen, this was highly prone to failure whenever the location of objects on the screen changed. Being able to produce reliable and easy to maintain automated test code requires time, effort and experience to produce.
- The level of technical experience of the automated testing resources is important. Manual testers for instance can produce automation that generally can only be used once as they do not have the experience to build test scripts based on screen objects.

Recent developments in the middle of 2025 have changed the landscape. Playwright integrated with Co-Pilot can now leverage AI to record automated test scripts based on a prompt that looks very much like the structure of Cucumber Gherkin, effectively bridging the gap between being able to simplify the definition of functionality to test and having automation built directly from it.

Automated functional testing is now evolving into Prompt Engineering where less technical test engineers can now use AI to build functional automated test scripts that will be robust and solid where maintenance is limited and straight forward and again, mostly carried out by AI.

AI & Prompt Engineering

Effective AI performance depends heavily on prompt engineering, as modern models are highly sensitive to how instructions, context, and constraints are expressed. Well-designed prompts translate human intent into clear, actionable guidance, reducing ambiguity and improving output quality, consistency, and reliability. In many cases, strong prompt design delivers greater gains than changing models or adding system complexity.

Prompt engineering is most effective when it combines clarity, structure, and intent

alignment. High-quality prompts define the ai's role, desired outcomes, relevant context, and constraints such as tone or format. Breaking tasks into smaller steps, providing examples, and setting explicit evaluation criteria further improve accuracy and control by guiding reasoning and limiting incorrect assumptions.

Prompt engineering should be treated as an iterative discipline. Prompts must be tested, refined, and adapted based on real-world results, failure patterns, and expert feedback. As models and use cases evolve, continuously reviewing and updating prompts ensures sustained alignment, scalability, and dependable AI behaviour.

A reusable prompt template will be created. The template will be used when asking copilot or Junie chat and the playwright test generator to create or refactor tests. Playwright test generation – prompt template will include examples of the following:

Role & context, objective, scope & constraints, interaction requirements, prefer data-testid or accessible locators, assertions & validation, validate UI state, navigation, and error handling, structure & quality, and output format.

The template must also include the definition and context of any re-usable drivers and modular functions.

Testing Performance have years of experience across industries, we bring deep expertise in AI strategy and implementation. Our strategies are customized to fit the unique needs of your business, ensuring that AI delivers tangible value. From vision to execution, we are a trusted partner at every step of the AI journey.

Key Benefits

- AI-driven automation bridges the gap between functional requirements and robust automated test creation.
- Prompt-based test generation reduces reliance on highly technical automation engineering skills.
- Playwright with AI produces more resilient, maintainable tests using object-based and accessible locators.
- Prompt engineering improves test accuracy, consistency, and reliability without increasing tool complexity.
- Reusable prompt templates accelerate test creation and standardise automated testing quality.
- AI-managed test maintenance reduces effort when applications or user interfaces change.
- Automated testing becomes scalable and accessible to wider testing and delivery teams.
- Faster test creation and execution improves feedback cycles in agile and DevOps environments.
- Reduced script fragility lowers long-term maintenance costs and delivery risk.
- Proven AI strategy and implementation expertise ensures automation delivers measurable business value.

Return on Investment

- Reduced automation build and maintenance effort lowers long-term testing and delivery costs.
- Faster automated test creation accelerates release cycles and improves time to market.
- AI-maintained tests reduce rework caused by UI changes and fragile test scripts.
- Lower dependency on specialist automation skills reduces resourcing and training costs.
- Reusable prompt templates increase productivity across multiple projects and releases.
- Earlier and more reliable defect detection reduces costly late-stage fixes.
- Improved test stability decreases pipeline failures and wasted delivery effort.
- Open-source tooling combined with AI minimises licensing costs while maximising automation value.

Approach

The following activities are available with the cloud based AI Automated Testing.

- Assess application architecture and cloud environments to identify suitable candidates for AI-driven functional automation.
- Define functional testing scope, risks, and automation objectives aligned to cloud delivery models.
- Design prompt templates for AI-generated functional test creation and maintenance.
- Generate automated functional tests using AI-assisted tools such as Playwright and copilots.
- Implement robust locators prioritising accessible attributes and stable application objects.
- Integrate automated functional tests into cloud-based CI/CD pipelines for on-demand execution.
- Execute automated tests across cloud environments, browsers, devices, and configurations.
- Analyse AI-generated test results to identify functional defects and coverage gaps.
- Refactor and maintain automated tests using AI to adapt to application changes.
- Configure cloud monitoring and reporting for automated test execution and outcomes.

Training

Training is available and will take approximately 3 days.

Support

Support is available from 08:00 to 18:00 on working days in England. 1st

Line Support

1st line support will be available between the hours of 09:00 – 17:00 via the following channels;

- Web
- Phone
- email

2nd line support which will operate between the hours of 09:00 – 17:00.

Definition of SLA's (Will provide list)

- 0 – Critical – Within 1 Hour – Call back
- 1 – Severe – Within 2 Hours – Call back
- 2- High – 24 hours
- 3 – Medium – 48 hours
- 4 – Cosmetic – 72 hours