Workshop AI - Driven testing

Course code: ANWIA

This two-day interactive workshop focuses on the practical use of AI in the field of software testing.

What we teach you

On the first day, participants will be introduced to the basics of automated testing using the Playwright tool and will learn how to generate and optimize test scenarios with the help of AI. They will also get hands-on experience with Elastic and Kibana tools for data collection and visualization of test results. Through practical exercises, participants will create their own tests, dashboards, and experiment with basic AI prompting techniques. The second day will focus on advanced techniques such as autonomous testing, deeper AI integration into QA processes, and model management within CI/CD environments. Participants will gain insight into how to effectively leverage AI in the creation, maintenance, and evaluation of tests, and how to adapt QA processes to the new possibilities brought by AI.

Required skills

- Fundamentals of programming (ideally TypeScript/JavaScript for Playwright)
- Basic experience with QA and testing frameworks
- Willingness to try out AI tools (e.g., ChatGPT) and experiment with prompt engineering

Technical Requirements

- Environment for Playwright (Node.js, VS Code editor)
- Access to an Elastic/Kibana instance (or a demo version/Docker)
- Recommended testing dataset or a real example for test evaluation

Organizational and Technical Notes

The first day is highly interactive. It includes more exercises with prompting, creating and running tests in Playwright, basic reporting, and dashboards in Kibana.

The second day focuses on more advanced techniques and concepts (autonomous tests, AI-driven QA architecture, advanced prompting, integration with CI/CD).

Course outline

Day 1

- Introduction to the topic and interactive warm-up
- Using testing tools: Playwright in practice
- Reporting and analysis of test results – Elastic, Kibana
- Group discussion and summary of the first day

Day 2

- Generating and optimizing test scenarios with AI
- Autonomous testing
- Involvement in the QA process, maintenance, and development
- Final recap and Q&A

Day One

Introduction to the Topic and Interactive Warm-up

Workshop Introduction and Goals

GOPAS Praha
Kodaňská 1441/46
101 00 Praha 10
Tel.: +420 234 064 900-3
info@gopas.cz

GOPAS Brno
Nové sady 996/25
602 00 Brno
Tel.: +420 542 422 111
info@gopas.cz

GOPAS Bratislava
Dr. Vladimíra Clementisa 10
Bratislava, 821 02
Tel.: +421 248 282 701-2
info@gopas.sk

- Brief overview of the two-day program structure and expected outcomes.
- Discussion with participants about their expectations and specific testing challenges.

What AI Brings to Testing and Its Greatest Benefits

- Why AI is becoming prevalent in testing (scalability, speed, risk identification).
- Defining key concepts: machine learning, NLP, prompting (basics, how to "prompt" tools like ChatGPT to generate test scenarios).

Interactive Element:

- Participants try simple prompts to generate (or optimize) test steps in real time.

Using Testing Tools: Playwright in Practice — Basic Concepts and Technology

Core Features of Playwright

- Brief overview of what the tool can do (cross-platform end-to-end testing).
- Differences from Selenium/WebDriver and Playwright's strengths.

Hands-on: Creating Tests in Playwright

- Creating a simple test (execution and evaluation).
- Example of how AI (e.g., ChatGPT with proper prompting) can assist in designing test steps.

Interactive Element:

- Participants write their own mini test scenario in Playwright, which they can later extend.

Test Optimization and Maintenance of Test Code

- How to identify repetitive patterns in tests.
- Demo of how AI can suggest refactoring or add edge cases to test scenarios.

Reporting and Analysis of Test Results — Elastic, Kibana, (partially Grafana)

Data Collection and Its Importance for AI-Driven Testing

- What data can be collected from tests, logs, and application metrics.
- Using Elasticsearch to store and quickly query large volumes of test records.

Working with Kibana and Grafana

- Kibana: Creating basic dashboards, visualizing test results, anomaly detection.
- Grafana (partially): Example setup for monitoring and failure notifications.

Interactive Element:

- Participants create simple dashboards in Kibana, add filters, and display statistics from test data collected during Playwright exercises.

Prompting for Reports and Result Interpretation

- How to provide AI models with the right prompt to generate understandable reports.
- Demo of automatic generation of a brief natural language summary of a test run.

Group Discussion and Summary of Day One

- Recap of key points: prompting, Playwright, data collection, basics of Kibana reporting.
- Q&A: addressing unclear points, feedback, and topics for deeper exploration.
- Setting expectations for Day Two — advanced topics, autonomous testing, deeper AI integration.

Possible Closing Activity: Short quiz or team competition on newly acquired skills.

Day Two

Generating and Optimizing Test Scenarios Using AI

Prompt Engineering for Test Creation

- Details on crafting prompts to generate different test variants (negative tests, edge cases).
- Using metrics (coverage, risk areas) as AI inputs.

GOPAS Praha
Kodaňská 1441/46
101 00 Praha 10
Tel.: +420 234 064 900-3
info@gopas.cz

GOPAS Brno
Nové sady 996/25
602 00 Brno
Tel.: +420 542 422 111
info@gopas.cz

GOPAS Bratislava
Dr. Vladimíra Clementisa 10
Bratislava, 821 02
Tel.: +421 248 282 701-2
info@gopas.sk

Advanced Optimization of Existing Tests

- Methods AI uses to identify redundant or inefficient tests.
- Integration with CI/CD (GitLab, GitHub Actions, Jenkins) — when and how to trigger AI-driven test generation.

Autonomous Testing

Concept of Autonomous Testing

- What a "self-learning" test is, its limits, and benefits.
- Demo of tools moving in this direction (e.g., Mabl, Test.ai).

Practical Demos and Hands-on

- What an architecture of an autonomous test linked to Playwright or another framework might look like.
- Demonstration (if available) of a basic autonomous test that learns to find UI bugs on its own.

Integration into QA Process, Maintenance, and Development

Setting up the QA Pipeline

- How to align AI-driven tests with traditional QA processes.
- Role of QA engineer in the AI era: prompting, interpreting results, overseeing AI models.

Model Management and Maintenance

- Data preparation and continuous learning.
- Monitoring AI model quality and performance; identifying data drift.

Final Recap and Q&A

Summary of Key Takeaways

- Prompting as a key to effective AI use in testing.
- Integration of tools (Playwright, Elastic, Kibana/Grafana) into the AI-driven testing ecosystem.
- Opportunities for further development toward autonomous testing.

GOPAS Praha
Kodaňská 1441/46
101 00 Praha 10
Tel.: +420 234 064 900-3
info@gopas.cz

GOPAS Brno
Nové sady 996/25
602 00 Brno
Tel.: +420 542 422 111
info@gopas.cz

GOPAS Bratislava
Dr. Vladimíra Clementisa 10
Bratislava, 821 02
Tel.: +421 248 282 701-2
info@gopas.sk