Faysal Aziz

AI for Software Testing

Smart Test Automation & Evaluation with Generative AI & RAG Transform testing using intelligent agents, test-generation, evaluation, and real-world frameworks in 1 day

FA-0068 · 1 day

Agentic & Generative AI · Software Development

Introduction

The software industry's rapid evolution is driving a fundamental shift away from conventional manual and scripted testing methods toward AI-enhanced approaches. Modern testing now leverages Generative AI, large language models, Retrieval-Augmented Generation systems, and intelligent agents to create more efficient, responsive, and scalable test automation solutions.

These technologies streamline everything from test plan creation and case generation to identifying AI system hallucinations and anomalies. Drawing from three decades of hands-on industry expertise, the instructor focuses on practical, field-tested methodologies and cutting-edge tools that today's QA professionals need to stay competitive.

Learning outcomes

By the end of this one-day session, participants will:

- Understand the fundamentals of Generative AI, LLMs, RAG and their role in the Software Testing Life Cycle (STLC).
- Be able to generate test artifacts (plans, cases, data, bug reports) using AI and prompt engineering.
- Learn how to integrate AI agents into test automation workflows (codeless or code enhancement).
- Be able to evaluate and test RAG-based LLM systems using frameworks like RAGAS and metrics such as hallucination detection.

- Explore real-world tools (e.g. Playwright/MCP, TestRigor, DeepEval, Deepeval, vector DBs, PyTest integration) for AI-driven QA.

Prerequisites

- Basic understanding of software testing principles (manual and automated), STLC.
- Familiarity with Python scripting and test frameworks (e.g. pytest) is helpful but not mandatory.
- Comfortable using command-line tools and version control (Git).
- Curiosity about AI tools like ChatGPT, GitHub Copilot, or open-source LLMs.

Training outline

1. Foundations of AI-Powered Testing

- Generative AI vs traditional AI: LLMs, prompt engineering, RAG architecture
- Overview of agentic AI and AI Agents for QA

2. Test Artifact Generation with Generative AI

- Crafting test plans, test scenarios, test cases, test data using prompt-based tools
- Automatically generating bug reports and execution reports (manual and UI automation)

3. AI-Enhanced Automation Workflows

- Leveraging Copilot, GitHub plugins, and GenAI for code completion, fixing, optimization
- Implementing self-healing locators, Model Context Protocol (MCP) in Playwright or Selenium
- No-code QA automation with tools like TestRigor (English-based instructions)

4. Testing LLMs & RAG-Based Systems

- Architecture of custom LLMs using RAG; common evaluation metrics (accuracy, consistency, hallucination, latency)
- Introduction to RAGAS evaluation framework: integrating with PyTest to automate metric assertions

5. AI Agents for Autonomous Testing

- Building and orchestrating AI agents for browser-based QA: English-based commands to drive Playwright agents and MCP servers
- Integrating agents into existing test frameworks for dynamic, autonomous execution workflows

6. Tool Overview & Hands-on Demonstrations

- Real demonstrations of tools: DeepEval/RAGAS, LangChain, vector databases, Playwright, Selenium, TestRigor, Applitools, PostBots, Deepeval etc.
- Running LLMs locally vs cloud APIs; privacy and security considerations

7. Best Practices & Real-World Scenarios

- Selecting appropriate AI models, prompt design, managing hallucinations and biases.
- Integrating AI-enabled QA into CI/CD pipelines (e.g. Jenkins)
- Data governance, anonymization, credentials management when using AI in testing

8. Closing & Roadmap Forward

- Summary of key benefits: speed, coverage, adaptability.
- Practical steps to pilot AI-driven QA in your organization.
- Further learning paths in RAG evaluation, agentic testing, open-source LLM deployment.

This structure ensures a highly practical, deeply relevant one-day course—centered on delivering industry-demanded skills rather than academic theory. Let me know if you'd like to adjust the balance (e.g. more hands-on labs, deeper LLM evaluation, or inclusion of your company's specific tools).

© 2026 Faysal Aziz · Kuala Lumpur