Implementing AI in Software Testing | AI in Test Automation Training

Duration: **1 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Overview

Bring AI into your software testing workflow to accelerate test design, automation, maintenance, failure analysis, and risk-based test prioritization.

Artificial intelligence is changing the way software tests are planned, created, executed, and maintained. This course provides testers with practical techniques for introducing AI-powered tools into existing testing workflows without requiring an advanced background in artificial intelligence or machine learning.

Participants will explore how AI can support different stages of testing, from generating realistic test data and creating test cases to prioritizing tests and investigating failed executions. Hands-on exercises use tools such as ChatGPT, Testim, GitHub Copilot, Codeium, Mockaroo, and Faker to demonstrate practical scenarios that testers can apply in their daily work.

The course also covers flaky and redundant test detection, self-healing automation, visual regression testing, and AI-assisted failure analysis. Participants will learn how historical defects, commit activity, code changes, and complexity indicators can be used to identify higher-risk areas and focus testing effort where it delivers the greatest value.

You will also explore how AI-generated tests and insights can become part of existing CI/CD processes using tools such as GitHub Actions. Prompt design techniques are introduced throughout the course to help participants produce more relevant, consistent, and accurate AI outputs.

This instructor-led, one-day course is approximately 50% hands-on. It is designed for testing professionals who already understand fundamental software testing practices but are relatively new to applying AI in their testing activities.

Prerequisites

Participants are expected to have foundational experience in software testing. Previous experience with artificial intelligence or machine learning is not required.

Recommended knowledge and experience:

- Ability to read and review test cases
- Familiarity with functional or UI testing practices
- Basic experience with manual or automated testing tools and frameworks

Implementing AI in Software Testing | AI in Test Automation Training

Duration: **1 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Who Should Attend

This course is intended for professionals who want to incorporate AI-assisted techniques into their existing software testing practices.

It is particularly suitable for:

- Software Testers
- QA Professionals and QA Engineers
- Test Automation Engineers
- SDETs
- Manual Testers
- Technical team members involved in software quality processes

No advanced AI knowledge is required. Participants should, however, have a basic understanding of software testing concepts.

What You Will Learn

By the end of this course, participants will be able to:

- Generate realistic and varied test data with the support of AI tools
- Prioritize test cases according to code changes, historical test results, and risk indicators
- Create UI, API, and functional test cases using natural-language instructions
- Transform user stories and requirements into test scenarios and automation scripts
- Detect flaky and redundant tests and improve overall test-suite reliability
- Evaluate self-healing and AI-assisted test maintenance approaches
- Analyze failed tests and logs with AI to accelerate troubleshooting and triage
- Identify higher-risk software components using historical defects and code-change data
- Integrate AI-generated tests and testing insights into existing CI/CD pipelines

Outline

1. Introduction to AI in Software Testing

Implementing AI in Software Testing | AI in Test Automation Training

Duration: **1 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Understand the role of artificial intelligence in modern software testing and how AI can enhance existing testing practices.

- Understanding AI in the context of software testing
- Traditional testing compared with AI-augmented testing
- Benefits of AI for testing speed, coverage, and accuracy
- Applying AI to unit, integration, UI, and end-to-end testing
- AI testing tools for different levels of technical experience
- How AI changes and supports the tester's role

2. Generating Test Data with AI

Learn how to produce realistic, diverse, and structured data for different testing scenarios.

- The importance of high-quality test data
- Generating structured data with Mockaroo and Faker
- Creating edge-case and negative test data with ChatGPT
- Selecting the appropriate tool for different data requirements
- Maintaining realism while protecting sensitive information
- **Hands-on:** Schema-based test data generation with Mockaroo
- **Hands-on:** Generating varied test data through ChatGPT prompts

3. AI-Assisted Test Selection and Prioritization

Use project history and risk indicators to make test suites more focused and efficient.

- Prioritizing tests according to code and change history
- Identifying redundant and low-value tests
- Building smaller test sets while maintaining high impact and coverage
- Generating test scenarios from requirements
- Visualizing testing impact through dashboards and analytics
- **Hands-on:** Risk-based test case prioritization
- **Hands-on:** Detecting redundancy in test descriptions using AI

4. AI-Powered Test Generation and Automation

Turn requirements, user stories, and technical definitions into executable tests with AI assistance.

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

- Generating UI, API, and functional test cases with AI
- Converting user stories into test scenarios and scripts
- Improving generated results through prompt refinement
- Exploring GitHub Copilot, Testim, and Codeium
- Bringing AI-generated tests into IDE-based development workflows
- **Hands-on:** Generating UI tests from user stories
- **Hands-on:** Creating tests with Testim AI Recorder
- **Hands-on:** Generating unit and UI tests with Codeium

5. Intelligent Test Execution and Maintenance

Apply AI techniques to improve test stability while reducing the effort required to maintain automated tests.

- Automatically identifying and analyzing flaky tests
- Detecting testing bottlenecks through analytics
- Working with self-healing selectors and automation flows
- Applying AI to visual regression testing
- Tracking changing failure patterns over time
- **Hands-on:** Flaky test detection and analysis
- **Demo:** AI-assisted visual testing
- **Hands-on:** Exploring self-healing test scenarios

6. AI-Based Defect and Risk Prediction

Use historical project information to identify software areas that are more likely to contain defects.

- Predicting risk from testing and code history
- Connecting defect patterns with code churn and complexity
- Applying NLP techniques to commit information
- Visualizing higher-risk components with heatmaps
- Using predictive insights during test planning
- **Hands-on:** Forecasting defect risk from commit history
- **Hands-on:** Identifying high-risk software components

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

7. Integrating AI into Testing and CI/CD Workflows

Bring AI-assisted testing into everyday QA activities and automated delivery pipelines.

- Identifying practical AI entry points across the testing lifecycle
- Using ChatGPT to summarize test results and reports
- Incorporating AI into CI/CD workflows such as GitHub Actions
- Using LLMs to analyze and triage failed tests
- Developing consistent and effective prompting practices for testing
- **Hands-on:** Analyzing and summarizing test logs with AI
- **Hands-on:** Simulating AI integration within a CI/CD pipeline