TRAICA SOLUTIONS

The Logical Path to Solutions

Beginner QA Engineer Training Programme

Product Software Organisation | Duration: 6–12 weeks (recommended 8 weeks)

Programme Overview

Where to Start

Day 1 onboarding checklist and quick wins to build confidence in your first week

Core Training

16 comprehensive sessions covering fundamentals through advanced testing practices

Hands-On Labs

Practical exercises and deliverables to reinforce learning with real-world scenarios

Capstone Project

Full testing cycle demonstration showcasing all acquired skills and competencies

This structured programme transforms beginners into confident QA engineers through intensive hands-on practice, combining 60–70% practical work with essential theory.

First Steps: Your Day 1 Checklist

Technical Setup

- Create essential accounts: email, Slack, Jira, Confluence, CI dashboard
- Configure development environment: SSH keys, VPN access, browsers
- Install testing tools: Postman, necessary runtimes and dependencies

Product Orientation

- Review product documentation and recent release notes
- Understand user personas and customer journeys
- Shadow a senior QA engineer and observe their workflow

Quick Wins in Week One: Run smoke tests locally, file your first high-quality bug reports, and write manual test cases for a core user flow

Training Structure & Pacing

01

Session Format

Each 2–4 hour session combines lecture, live demonstration, and hands-on laboratory work

02

Practical Focus

60–70% of time dedicated to hands-on exercises, ensuring skills are immediately applicable

03

Weekly Rhythm

Lecture and demo, followed by lab work, then pairing sessions with experienced QA engineers

04

Continuous Support

Regular check-ins, code reviews, and mentorship throughout the 6–12 week journey

Foundation: Testing Fundamentals & Core Concepts

Testing Fundamentals & QA Mindset

Define quality, verification vs validation, test levels and types. Learn risk-based testing and heuristics.

SDLC & Agile Integration

Master sprint cadence, ceremonies, Definition of Done, and acceptance criteria in product teams.

Requirements Analysis

Apply equivalence partitioning, boundary value analysis, decision tables, and state testing techniques.

Test Case Writing

Write maintainable test cases with clear traceability to requirements and specifications.

Deliverables: Risk register, test plans, Gherkin acceptance criteria, comprehensive test case documentation

Practical Testing: Manual & Exploratory Approaches

Manual Functional Testing

Execute structured test cases systematically, documenting results and identifying deviations from expected behaviour.

Exploratory Testing

Use session-based test management (SBTM) with time-boxed charters to discover unexpected issues and edge cases.

Defect Reporting

Write actionable bug reports with clear reproduction steps, screenshots, logs, and environment details.

Practical Exercise: Conduct 60-minute exploratory sessions on demo applications like OWASP Juice Shop or TodoMVC, producing detailed session reports and high-quality bug reports.

Automation & Technical Skills Development

1. **Scripting Foundations**

   Programming basics, Git version control, test frameworks, and automation principles. Build first unit tests.

2. **Web UI Automation**

   Master Selenium or Playwright: locators, waits, Page Object Model, and flakiness reduction techniques.

3. **Android Testing**

   Master Espresso and UI Automator for Android app testing: emulator setup, instrumented tests, and mobile-specific scenarios.

4. **API Testing**

   HTTP fundamentals, authentication, Postman collections, and automated API test frameworks (pytest/JavaScript).

5. **CI/CD Integration**

   Integrate tests into GitHub Actions, GitLab CI, or Jenkins with automated reporting and failure notifications.

Deliverables include automation repositories, Postman collections, working CI workflows, and passing test suites on every pull request.

Advanced Topics: Performance, Security & Accessibility

Performance Testing

Learn load, stress and spike testing using JMeter. Interpret TPS, latency percentiles, and response times.

- Run JMeter scripts with 50+ virtual users
- Analyze bottlenecks and performance degradation

Security Basics

Top 10 overview and passive security scanning with tools like ZAP/Burp Suite to identify common vulnerabilities.

- Security checklist creation
- Basic penetration testing concepts

Capstone Project: Demonstrating Excellence

Requirements & Planning

Develop acceptance criteria and comprehensive test plan with risk assessment and coverage analysis

Manual Testing

Create manual test cases and execute exploratory charters with detailed session reports

Test Automation

Build minimum 10 automated tests covering unit, API, and UI layers with maintainable code

CI Integration

Configure continuous integration to run tests on every pull request with automated reporting

Specialised Testing

Execute basic load testing and accessibility scans, documenting findings and recommendations

Final Demonstration

Deliver repository, pipeline configuration, test artifacts, comprehensive report, and live demo presentation

Your Journey to QA Excellence

Success Criteria

- Passing test suite in CI for every pull request
- High-quality, reproducible bug reports
- Complete automation repository with documentation
- Postman collection with Newman integration
- Polished capstone demonstration

What You'll Achieve

Transform from beginner to confident QA engineer equipped with modern testing practices, automation skills, and professional communication abilities.

Ready to build quality into every release.

Questions? Connect with the training team to begin your QA engineering journey today.