

SQE Report

Report on the needs of TestOps, what we have covered so far, and what is missing in our framework

Waleed Ahmad 22I-2647

Soban Ahmad 22I-2642

Github Repository Link: [Waleed-Ahmad20/SQEProject](https://github.com/Waleed-Ahmad20/SQEProject)

Project Demonstration and Structure

Overview

This report consolidates the aspects of TestOps and its implementation based on assignments completed in our course “Software Quality Engineering”. The focus is on TestOps needs, the frameworks we've built, gaps in the framework, and recommendations for tools to address those gaps.

Framework Structure

The test frameworks developed during the course follow a modular architecture, each catering to distinct testing needs:

1. UI Test Automation Framework

Tools Used: Selenium, VSCode(Java), TestNG, Gradle

Key Features:

- Page Object Model (POM)
- Browser compatibility testing
- Gherkin language support for BDD
- Integration with Allure for detailed reporting

2. Unit Testing Framework

Tools Used: JUnit 5, Eclipse(Java), Mockito, ByteBuddy

Key Features:

- Test coverage for core application logic

- Integration with CI tools
- Mocking for dependency isolation

3. Web API Automation Framework

Tools Used: RestAssured, JUnit Jupiter

Key Features:

- Comprehensive CRUD API testing (GET, POST, PUT, DELETE)
- BVA and ECP techniques for input validation
- Automated assertion of response codes and data integrity

4. Performance Automation Framework

Tools Used: Apache JMeter

Key Features:

- Load testing with realistic user simulation
- Configurations: 50 threads, ramp-up in 10 seconds, loop count of 5
- Integrated reporting for performance metrics

TestOps Needs

TestOps (Testing Operations) is essential for modern development practices to ensure streamlined, scalable, and efficient testing in CI/CD pipelines.

Key Needs Include:

- Centralized test management and visibility.
- Real-time test execution monitoring.
- Automated reporting and defect tracking.
- Efficient test environment management.
- Seamless integration with CI/CD tools.

Framework Coverage in TestOps Context

Strengths of Current Frameworks:

1. **Automation:**
 - Automates functional, performance, and API testing.
 - Includes reusable modular test suites.
2. **Reporting:**
 - Allure provides detailed test reports for UI and API tests.
 - JMeter integrates performance metrics into reports.
3. **CI/CD Compatibility:**
 - Frameworks support integration with Jenkins, ensuring automated test execution in the build pipeline.
4. **Scalability:**
 - Modular architecture allows easy addition of new test cases.

Current Gaps:

- Lack of real-time monitoring dashboards for test execution.
- No centralized management of test environments.
- Limited support for parallel execution and distributed testing.
- Absence of historical trend analysis for defects and failures.

Recommendations for Addressing Gaps

TestOps Platforms:

- **Katalon TestOps**
 - Centralized test management.
 - Real-time monitoring with execution logs.
 - Seamless integration with CI/CD pipelines.
- **TestKube**
 - Supports distributed test execution.
 - Easy integration with containerized environments.
 - Interactive dashboards for reporting and metrics.

Additional Tools:

- **Zebrunner:** For live monitoring and defect tracking.
- **BrowserStack:** To improve cross-browser testing and parallel execution.
- **Grafana + Prometheus:** For custom dashboards to track performance and test

execution trends.

CONCLUSION

Adopting TestOps solutions like Katalon TestOps or TestKube will fill the gaps in real-time monitoring, test management, and historical trend analysis. These enhancements will make the framework more robust, scalable, and CI/CD-ready for modern software development needs.

REFERENCES

1. Katalon. *Katalon TestOps Documentation*. Katalon, <https://katalon.com/testops>. Accessed 28 Nov. 2024.
2. TestKube. *TestKube Documentation*. TestKube, <https://testkube.io/>. Accessed 28 Nov. 2024.
3. Balchin, Bradley. "TestOps: Bridging the Gap Between Testing and DevOps." *DevOps.com*, 18 Nov. 2024, <https://devops.com/testops-bridging-gap-testing-devops/>. Accessed 28 Nov. 2024.
4. Apache Software Foundation. *Apache JMeter User Manual*. Apache Software Foundation, <https://jmeter.apache.org/usermanual/index.html>. Accessed 26 Nov. 2024.
5. "Understanding Continuous Testing in DevOps." *TechBeacon*, <https://techbeacon.com/app-dev-testing/continuous-testing-devops>. Accessed 28 Nov. 2024.
6. *Selenium WebDriver Documentation*. SeleniumHQ, <https://www.selenium.dev/documentation/webdriver/>. Accessed 25 Nov. 2024.
7. RestAssured. *RestAssured Documentation*. <https://rest-assured.io/>. Accessed 27 Nov. 2024.
8. Zebrunner. *Zebrunner TestOps Overview*. Zebrunner, <https://zebrunner.com/>. Accessed 28 Nov. 2024.