

Automated Testing Framework - Team Guide

1. Project Overview

This project is an Automated Testing Framework designed for **UI, API, and performance testing**. The team follows a structured approach to ensure efficient testing and collaboration.

Key Tools & Responsibilities:

- **Google Drive:** Centralized storage for all test scripts, reports, and documentation.
 - **GitHub (Managed by Scrum Master only):** Stores automation framework, scripts, and configurations.
 - **Jira (Managed by Scrum Master only):** Tracks test cases, bugs, and project progress using CSV bulk import.
 - **Selenium & Postman:** Used for UI and API testing, respectively.
 - **Bulk Configuration Files:** Used for managing large test cases and Jira imports.
 - **ISTQB 4, TestNG/JUnit, JMeter/Gatling:** Used for structured testing approaches, reporting, and performance testing.
-

2. Team Roles & Responsibilities

Scrum Master

- Solely manages **GitHub** repository (handles all code merges and commits).
- Oversees **Jira**, ensuring tasks are properly assigned and tracked.
- Imports test cases and bug reports from **Google Sheets** into Jira using **CSV bulk upload**.
- Facilitates communication between developers and testers.
- Maintains bulk configuration files for automated test cases in **Jira**.

Developers (Test Engineers)

- Write and maintain test scripts for UI, API, and performance testing.
- Follow structured directory for file storage and upload relevant files to **Google Drive**.
- Develop and improve test automation framework (POM structure, utilities, etc.).
- Integrate structured development and **software testing principles**.
- Work with testers to debug and improve automation scripts.

Testers













- Execute test cases and report issues in **Google Sheets**.
- Upload reports and logs to **Google Drive**.
- Verify automation test results and collaborate with developers to resolve failures.
- Write and manage **bulk configuration files** for Jira test case imports.

3. Project Folder Structure & Responsibilities

Google Drive Folder Structure







Each team member uploads files to the correct folders:

AutomatedTestingFramework_GDrive

- |—  Config # Configuration files (e.g., Jira settings, test configurations, bulk CSV files)
 - |—  Tests
 - | |—  UI_Tests # Selenium scripts (.java, .py, etc.)
 - | |—  API_Tests # Postman collections (.json)
 - | |—  Perf_Tests # Load/performance testing scripts
 - |—  Reports
 - | |—  HTML_Reports # Generated test reports (Allure, Extent, etc.)
 - | |—  Logs # Log files from test executions
 - | |—  Screenshots # Screenshots of failed test cases
 - |—  Test_Data # Input data for tests (CSV, JSON, etc.)
 - |—  Docs # Documentation (Test plans, user guides, etc.)
 - |—  Test_Cases # Google Sheets test cases and bug reports (Exported as CSV for Jira import)
-

GitHub Repository (Managed by Scrum Master Only)

AutomatedTestingFramework_GitHub

- |—  config # Configuration files for automation
- |—  tests # Test cases (UI, API, performance)
- |—  framework # Page Object Model, utilities
- |—  reports # HTML reports, logs, screenshots
- |—  test_data # Input test data
- |—  docs # Documentation
- |— requirements.txt # Dependencies
- |— README.md # Project overview
- |— run_tests.java # Script to execute tests
- |— .gitignore # Ignore unnecessary files

GitHub Workflow

- Only the **Scrum Master** commits code to GitHub.
- Developers and testers upload all relevant files to **Google Drive**.
- The Scrum Master reviews and integrates files into the repository.

Jira Task Management

Jira is used for:

- **Bulk importing** test cases and bug reports from **CSV files** exported from Google Sheets.
 - Tracking test automation tasks and logging bugs.
 - Assigning tasks to the right team members.
 - Managing sprints and progress tracking.
 - Handling bulk configuration files for structured test case management.
-

4. Testing Phases & Timeline

Week 1: Setup and Test Case Development

- Set up Selenium/WebDriver.
- Organize project folder structure.
- Write initial UI test cases.
- Set up Postman for API testing.
- Integrate Jira for tracking test cases and issues.

Week 2: Functional Testing and Report Generation

- Develop detailed functional test cases.
- Automate functional test cases with Selenium.
- Implement TestNG/JUnit for report generation.
- Expand API testing in Postman.
- Set up JMeter/Gatling for performance testing.










Week 3: Performance and UI Testing

- Develop performance test cases.
- Execute load/stress tests.
- Enhance UI test cases for responsiveness and cross-browser compatibility.
- Log and track issues in Jira.

Week 4: Final Testing, Optimization, and Deployment

- Optimize and refactor the test framework.
- Execute the complete test suite.
- Generate final test reports.
- Document the framework and testing process.
- Integrate with CI/CD (Jenkins, GitHub Actions).

5. Guidelines for File Uploads

Type of File	Upload to Google Drive	Commit to GitHub
Selenium Scripts	 UI_Tests	✗ (Scrum Master only)
Postman Collections	 API_Tests	✗ (Scrum Master only)
Test Reports (HTML)	 Reports/HTML_Reports	✗
Log Files	 Reports/Logs	✗
Screenshots	 Reports/Screenshots	✗
Test Data (CSV/JSON)	 Test_Data	✗ (Scrum Master only)
Documentation	 Docs	✗ (Scrum Master only)
Test Cases & Bugs	 Test_Cases (Google Sheets, Export as CSV)	✗
Bulk Configuration Files	 Config	✗

6. Conclusion

This guide ensures a structured workflow and organized file management for efficient collaboration. By adhering to this structure, the team can maintain **consistency, clarity, and productivity** in test automation efforts while integrating **structured development, performance testing, and CI/CD automation**. 🚀