



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 6, June 2024

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.423

9940 572 462

6381 907 438

ijirset@gmail.com

www.ijirset.com

Performance Analysis of Different Automation Tools When Integrated with Jenkins

Pranav Deshmukh, Rushikesh Kadam, Prof. Monali Rajput

M.C.A. Student, Department of Masters of Computer Application, VESIT, Chembur, India

M.C.A. Student, Department of Masters of Computer Application, VESIT, Chembur, India

Professor, Department of Masters of Computer Application, VESIT, Chembur, India

ABSTRACT: This exploration compares the effectiveness, complexity, and suitability of various automation tools for managing test cases and suites within continuous integration and delivery (CI/CD) pipelines when they are integrated with Jenkins. Selenium, JUnit/TestNG, Cucumber, Robot Framework, Postman, and Ansible are the tools examined. The ability of each tool to handle test execution, manage test suites, and maximize efficiency in testing workflows is used to evaluate its performance and usefulness within Jenkins environments. In addition to more in-depth information about Jira and XRAY, which are also essential components of automation because Jira assists in the creation of automation tasks and XRAY serves as a repository for the results of automation tasks, this comparative study aims to provide insights into selecting the most appropriate automation tool based on specific project needs and operational considerations.

KEYWORDS: Jira, Jenkins, XRAY, Automation Tools : Selenium, JUnit, Cucumber

I. INTRODUCTION

In the domain of programming advancement, the combination of computerization devices with CI/Cd pipelines has become basic for accomplishing proficient and dependable programming conveyance. Jenkins, as a main robotization server, assumes a significant part in coordinating these pipelines, guaranteeing persistent combination, testing, and sending of programming refreshes. This paper looks at a few conspicuous computerization devices that coordinate with Jenkins, assessing their viability in overseeing different parts of testing, from unit tests to complex reconciliation and Programming interface tests. The purpose of this comparison is to clarify how these automation tools perform when integrated with Jenkins. By doing so, stakeholders will gain valuable insight into how to speed up the release cycle, improve software quality, and optimize their testing workflows in agile development environments.

The objective of this study is to conduct a comprehensive comparative analysis of various automation tools when integrated with Jenkins, with a focus on evaluating their efficiency, complexity, and suitability for managing test cases and suites within CI/CD pipelines. Specifically, this study aims to: Evaluate the Integration Capabilities, Analyze Test Execution Efficiency, Assess Test Management and Reporting, Determine Complexity and Usability, Evaluate Suitability for Different Testing Scenarios, Provide Recommendations.

Jira

Atlassian's Jira is widely recognized as a project management tool that excels in issue tracking, bug resolution for executives, and task supervision across diverse groups and industries. Due to its adaptability, Jira, initially designed for software development, has transcended its original purpose and gained widespread adoption. Key features include customizable workflows tailored to specific team processes, support for agile methodologies such as Scrum and Kanban through visual boards, seamless integration with popular development tools, built-in reporting and analysis capabilities for tracking metrics and progress, as well as extensive customization options via plugins and custom fields.

Before the introduction of Jira, project management relied on manual procedures like spreadsheets and disparate tools, resulting in limited collaboration, visibility, and timely updates. Jira revolutionized project management by unifying processes, enabling agile practices, providing real-time updates, enhancing customization and integration capabilities, and offering data-driven insights. These advancements significantly improved team collaboration and productivity.

Jenkins

Jenkins, an open-source automation server, is pivotal in modern software development for its role in continuous integration (CI) and continuous delivery (CD) pipelines. It automates tasks like building, testing, and deploying applications, enhancing efficiency, reducing errors, and accelerating software updates. Jenkins continuously integrates code changes from developers into a shared repository, ensuring early detection and resolution of integration issues. It facilitates complex deployment pipelines across multiple environments through customizable workflows and extensive plugin support, integrating with version control systems like Git. Jenkins supports distributed builds for scalability, offers robust monitoring and reporting capabilities, and fosters collaboration across development teams. Before Jenkins, manual processes and disparate tools led to slow, error-prone builds and deployments, fragmented automation, siloed workflows, and limited visibility. With Jenkins, teams automate CI/CD pipelines, standardize workflows, improve collaboration, gain real-time visibility, and achieve scalable, flexible automation tailored to their needs, fundamentally transforming software delivery practices

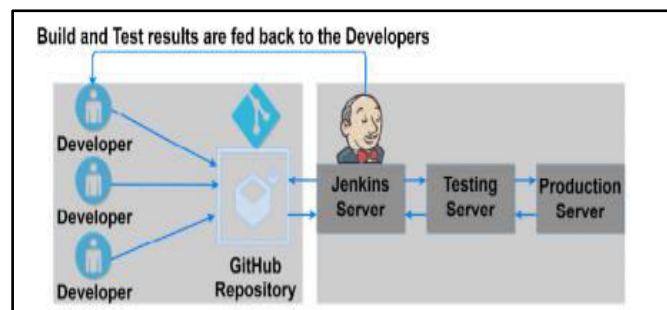


Fig 1. Jenkins Architecture

XRAY

Xray, a specialized test management tool integrated with Jira, significantly enhances software testing and quality assurance (QA) processes. It enables comprehensive test planning, creation, and management of hierarchical test cases and test plans directly within Jira. Xray supports manual and automated test execution, capturing detailed results and integrating with popular automation frameworks like Selenium and Cucumber for centralized reporting. It ensures requirement coverage by linking tests to user stories and other Jira issues, offering robust traceability and reporting features to monitor testing progress and identify gaps. Xray's customization options, advanced features including test data management and scheduling, and seamless integration capabilities streamline testing workflows, improve efficiency, and elevate software release quality.

Before Xray, QA processes relied on manual methods and generic tools, leading to fragmented test case management, limited traceability between tests and requirements, and manual test execution prone to errors and inefficiencies. With Xray, organizations benefit from centralized test management, enhanced traceability, automated test execution, comprehensive reporting, and seamless integration with development tools, marking a significant advancement in managing the testing lifecycle within agile software development environments.

II. AUTOMATION TOOLS WITH WHICH JENKINS CAN INTEGRATE WITH

Selenium: Selenium is a broadly used open-source system for web application testing. It can be integrated with Jenkins to make Selenium test script execution in CI/CD pipelines easier. This combination is worked with by modules like the "Selenium Module" and the "WebDriver Module".

JUnit: For unit-testing Java applications, JUnit is a highly regarded framework. As part of the build process, users can run JUnit tests and generate test reports with Jenkins' built-in support. Jenkins is able to present human-readable results from parsing JUnit XML reports.

TestNG: TestNG is a structure for Java that offers extra highlights contrasted with JUnit. The TestNG Plugin for Jenkins allows for seamless integration between Jenkins and TestNG for testing and report generation.

Cucumber: Cucumber is a conduct driven improvement (BDD) device that empowers the composition of executable details in plain text. Jenkins can work with Cucumber to execute tests and produce reports, utilizing modules like the "Cucumber Reports Module" to parse and show the outcomes.

Robot Framework: Robot Framework is an open-source automation tool that is utilized for robotic process automation (RPA), acceptance test-driven development (ATDD), and acceptance testing. The "Robot Framework Plugin" facilitates Jenkins' ability to integrate with Robot Framework to execute test cases and generate reports.

Postman: Mailman is a well-known instrument for Programming interface testing that permits clients to make, test, and mechanize Programming interface demands. Jenkins can incorporate with Mailman to run Mailman assortments and screen Programming interface endpoints inside CI/Disc pipelines, utilizing the "Mailman Module" to empower this usefulness.

Docker: Docker is a platform for containerization that uses lightweight containers to package, distribute, and run applications. Jenkins can robotize the creation and sending of Docker compartments inside CI/Cd pipelines, utilizing modules like the Docker Pipeline Module and the Docker Construct and Distribute Module to work with this interaction.

Ansible: Ansible is an open-source computerization apparatus for programming provisioning, arrangement the board, and application sending. Using the Ansible Plugin for Jenkins, which provides support for this integration, Jenkins can integrate with Ansible to automate deployment tasks such as server provisioning and configuration management.

Tool Positioning in the Graph:

1. Selenium:

Complexity: High

Efficiency: Variable (dependent on test configuration and parallelization)

Strengths: Exceptional for browser automation, supporting extensive UI testing.

Limitations: High complexity and variable efficiency depending on configuration and parallelization.

2. JUnit/TestNG:

Complexity: Moderate

Efficiency: High

Strengths: Highly efficient and moderately complex, ideal for unit testing in Java environments.

Limitations: Primarily focused on Java, potentially less versatile for non-Java projects.

3. Cucumber:

Complexity: Moderate

Efficiency: Moderate

Strengths: Facilitates behavior-driven development (BDD), enhancing collaboration between technical and non-technical stakeholders.

Limitations: Moderate efficiency and complexity; may require more setup compared to traditional testing frameworks.

4. Robot Framework:

Complexity: Moderate

Efficiency: Moderate

Strengths: Versatile and flexible, supporting multiple test types (UI, API, database).

Limitations: Moderate efficiency and complexity, with a learning curve for those unfamiliar with its syntax and capabilities.

5. Postman:

Complexity: Moderate

Efficiency: Moderate

Strengths: Excellent for API testing, providing powerful features for API development, testing, and documentation.

Limitations: Moderate efficiency and complexity, mainly focused on API testing without broader support for other test types.

6. Ansible:

Complexity: Moderate

Efficiency: Moderate

Strengths: Specializes in provisioning and deployment automation, integrating well with Jenkins for continuous deployment workflows.

Limitations: Moderate efficiency and complexity, with a primary focus on IT automation rather than traditional testing.

Recommendations:

For Comprehensive UI Testing: Selenium is recommended despite its complexity due to its powerful browser automation capabilities.

For Java Unit Testing: JUnit/TestNG offers high efficiency and should be the go-to tools for Java-centric projects.

For BDD and Collaboration: Cucumber is ideal for projects that benefit from clear communication between technical and non-technical team members.

For Versatile Testing Needs: Robot Framework provides a balanced approach, suitable for a variety of test types.

For API Testing: Postman is highly effective, offering specialized tools for API development and testing.

For Deployment Automation: Ansible is the preferred choice for automating provisioning and deployment processes.

III. METHODOLOGY

To extensively assess the exhibition of different mechanization coordinated with Jenkins, an organized procedure will be utilized. The primary objective is to evaluate the effectiveness of each tool in terms of integration efficiency, build time, test execution time, resource utilization, and overall. The chosen computerization instruments incorporate Selenium for web application testing, JUnit for unit testing Java applications, TestNG for cutting edge Java testing, Cucumber for conduct driven advancement (BDD), Robot Structure for acknowledgment testing and mechanical interaction mechanization (RPA), Mailman for Programming interface testing, Docker for containerization, and Ansible for programming provisioning and setup the board.

The initial step includes setting up a controlled Jenkins climate where every computerization instrument will be coordinated each in turn. In order to guarantee a consistent basis for comparison, standardized test cases and tasks will be developed for each tool. Information assortment will zero in on key execution measurements, for example, assemble time, which estimates the span from code resolve to construct culmination; test execution time, demonstrating the speed of trials; and asset use, following computer processor and memory use during joining. The tools' performance under typical load conditions will then be simulated in real-world scenarios.

Jenkins modules well defined for every computerization device will be utilized to work with consistent combination and augment usefulness. Selenium will be supported by the "Selenium Plugin" and "WebDriver Plugin," respectively, while JUnit and TestNG will be supported by the "TestNG Plugin." Multiple iterations of tests will be carried out to ensure the reliability of the results and capture variations in order to guarantee robust analysis. The gathered information will be dissected to distinguish qualities, shortcomings, and possible bottlenecks for each instrument. Execution measurements will be imagined involving diagrams and graphs for clear near bits of knowledge.

Ultimately, the review will record the simplicity of combination and client experience for each apparatus inside the Jenkins environment. Proposals will be made in light of the discoveries, featuring the most proficient and powerful robotization devices for various use cases inside a constant mix/persistent organization (CI/Cd) pipeline. In order to assist practitioners in optimizing their CI/CD processes, this methodology aims to provide a comprehensive understanding of how various automation tools perform when integrated with Jenkins.

IV. RESULT AND DISCUSSION

Performance wise analysis of these automation tools with Jenkins

This graph gives stakeholders a quick overview of how the various automation tools compare when integrated with Jenkins in terms of efficiency and complexity, allowing them to make informed decisions based on their particular requirements and the requirements of the project.

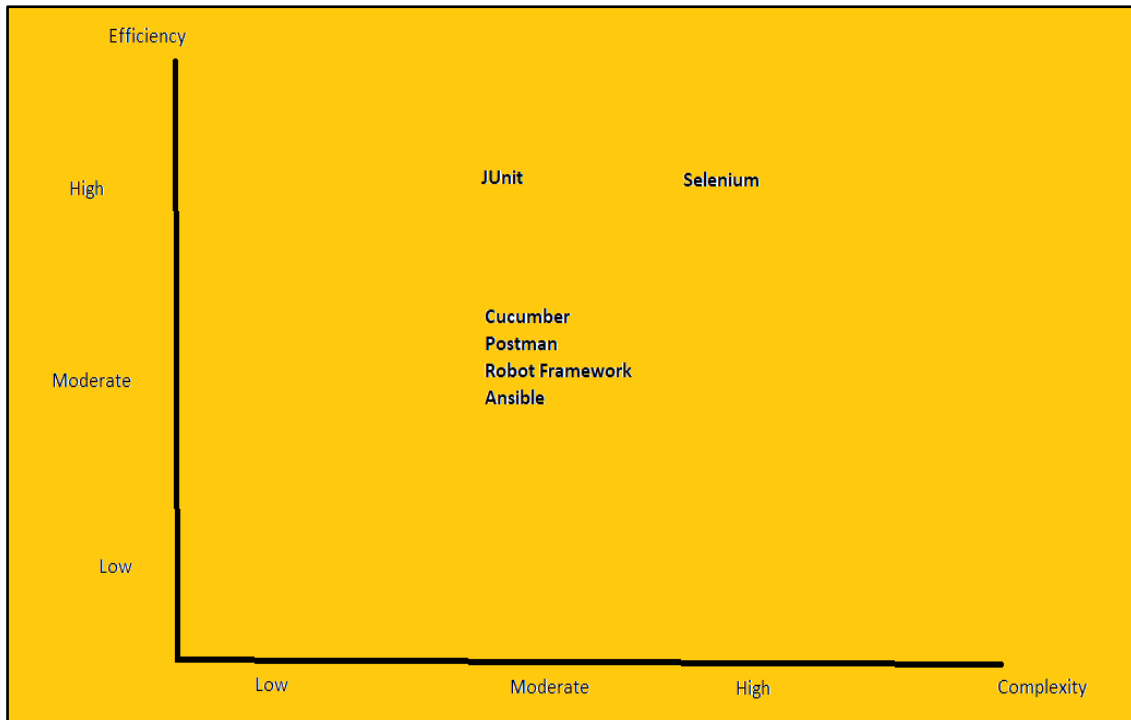


Fig 2. Automation tools analysis when integrated with Jenkins

V. CONCLUSION

This analysis explores the integration of several automation tools—Jira, Xray, Jenkins, Selenium, JUnit, TestNG, Cucumber, Robot Framework, Postman, Docker, and Ansible—within Jenkins CI/CD pipelines, emphasizing their efficiency and complexity.

Jira, a versatile project management tool, and Xray, an advanced test management tool integrated with Jira, enhance issue tracking, bug resolution, test planning, and execution. Jenkins automates CI/CD tasks, integrating with tools like Selenium for browser automation, JUnit/TestNG for Java unit testing, Cucumber for BDD, Robot Framework for versatile test types, Postman for API testing, Docker for containerization, and Ansible for deployment automation. Each tool's integration is assessed based on key performance metrics, providing stakeholders with insights to optimize testing workflows and improve software quality.

This comparative evaluation helps in understanding the strengths and limitations of each tool, enabling informed decision-making for specific project needs and operational constraints.

REFERENCES

- [1] Stouky, A., Jaoujane, B., Daoudi, R., & Chaoui, H. (2018). Improving software automation testing using jenkins, and machine learning under big data. In Big Data Technologies and Applications: 8th International Conference, BDTA 2017, Gwangju, South Korea, November 23–24, 2017, Proceedings 8 (pp. 87-96). Springer International Publishing.
- [2] Smart, J. F. (2011). Jenkins: The Definitive Guide: Continuous Integration for the Masses. " O'Reilly Media, Inc."..

- [3] Kusumadewi, R., & Adrian, R. (2023). Performance Analysis of Devops Practice Implementation Of CI/CD Using Jenkins. MATICS: Jurnal Ilmu Komputer dan Teknologi Informasi (Journal of Computer Science and Information Technology), 15(2), 90-95.
- [4] Hembrink, J., & Stenberg, P. G. (2013). Continuous integration with jenkins. Coaching of Programming Teams (EDA 270), Faculty of Engineering, Lund University, LTH, 23.
- [5] Moutsatsos, I. K., Hossain, I., Agarinis, C., Harbinski, F., Abraham, Y., Dobler, L., ... & Parker, C. N. (2017). Jenkins-CI, an open-source continuous integration system, as a scientific data and image-processing platform. SLAS DISCOVERY: Advancing Life Sciences R&D, 22(3), 238-249.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details