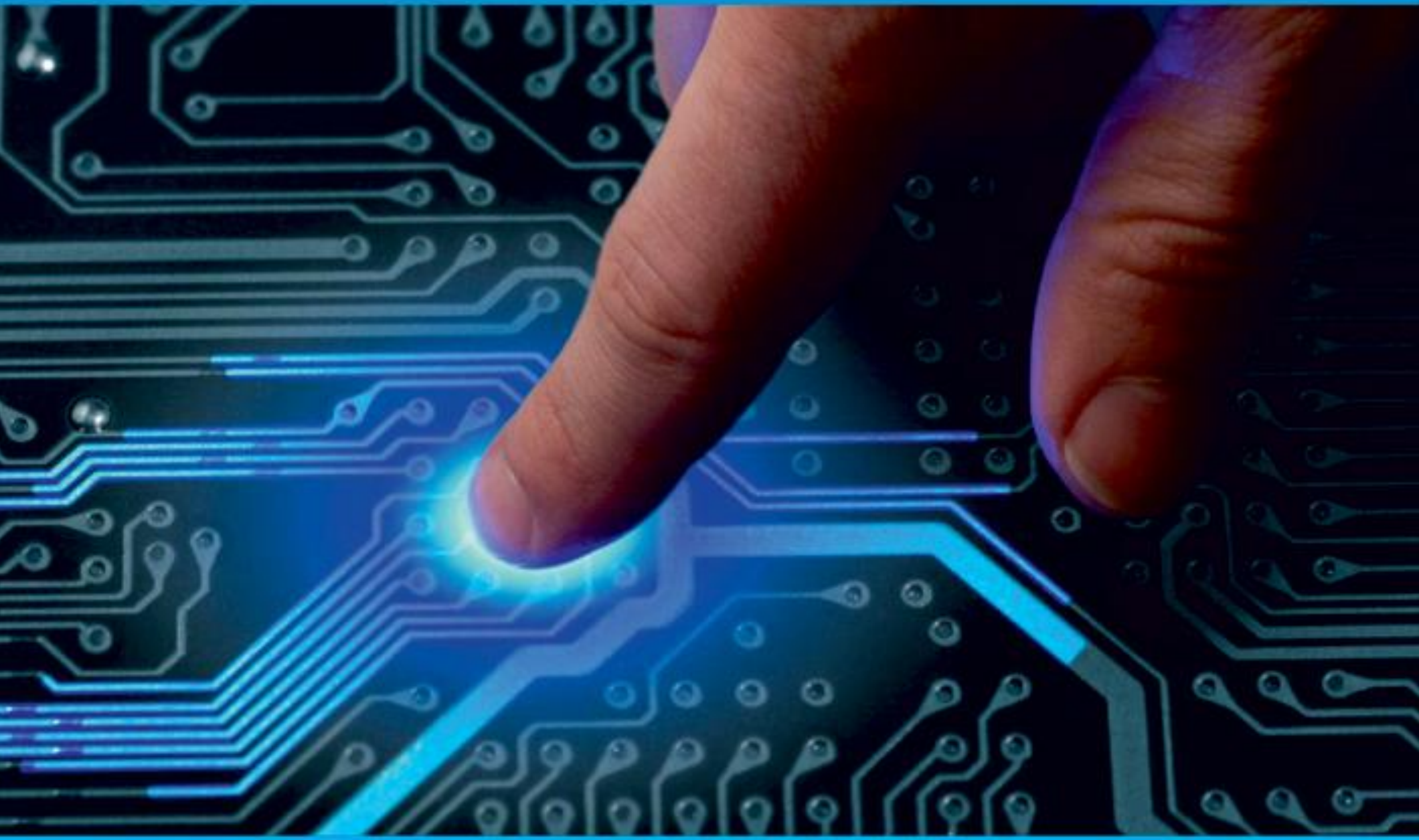
e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 11, Issue 7, July 2023

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 7, July 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1107021 |

Shift-Left Performance Testing

Mohd Abdul Hakeem

Quality Engineering Associate Manager, Accenture, Hyderabad, India

ABSTRACT: In the traditional performance testing approach, the load and performance testing comes into picture only at the final stage of the development life cycle when the complete build is ready. Whereas, in the Shift left approach the performance testing is included form the very beginning of the project and not as an afterthought. In the traditional approach, performance bottlenecks are identified and fixed just before the deployment or release which is an expensive exercise. The shift left approach has short delivery cycles, which is frugal to check every deliverable, however small it may be for performance. Thus integrating the tests in the continuous performance testing process is a better way of ensuring a quality deliverable.

KEYWORDS: Shift-Left Testing; Performance Testing Approaches; Early Performance Testing; Salient features of Shift-Left Testing; Software Testing Frameworks.

I. Introduction

The latest technological era is witnessing a major IT Transformation with a demand for a high quality and customer satisfaction with faster software deliveries. It is more a quality focused demand with a low risk of failures. Performance Testing plays an import role in achieving quality of the application. The performance critical bottleneck identified during the testing helps in optimizing and improving the performance of the software applications to a greater extent and thereby lowers the risk of application failures and outages which often is caused by critical performance issues. In general, the performance testing of software applications are often seen as a last mile or bottleneck activity in the traditional methods or approaches. The Performance testing task here is generally performed at the end of the software development life cycle which increases the risk of quality and failures.

The Old-School or traditional Performance testing approach has some limitations in accommodating the current style of requirements flow and not a good fit for the current Agile and DevOps practices. The traditional performance testing also has a limitation in accepting or handling the ongoing changes in the expectations and the requirements which results in negative outcomes such as, increased time to market, increase in costs and a lot of risk of failures with unexpected errors.

In the traditional performance testing approach, the load and performance testing comes into picture only at the final stage of the development life cycle when the complete build is ready. Whereas, in the Shift left approach the performance testing is included form the very beginning of the project and not as an afterthought. In the traditional approach, performance bottlenecks are identified and fixed just before the deployment or release which is an expensive exercise. The shift left approach has short delivery cycles, which is frugal to check every deliverable, however small it may be for performance. Thus integrating the tests in the continuous performance testing process is a better way of ensuring a quality deliverable.

II. Related Work

Shiping C et al. [1] have presented a software testing framework for the performance testing of software applications. This framework may be very much useful for small and large scale testing, especially for the load and performance testing. It is helpful in the Performance testing of software applications by separating the software application logic from the very common performance testing functionalities. It is prototyped on .NET & JAVA platforms. In this proposed model, three entities – Scripts, Configurations, Drivers (Test) and Results are being presented, which interact with each other. Scripts & Configurations are given as inputs and the driver generates the specified load based on the given inputs. After the test, the driver collates the test results and generates a summarized test report. This testing framework has been implemented on Java and .NET platforms and was demonstrated with a couple of examples. The metrics used while demonstrating the prototype are, Response Times, Latency, Throughput and other resource utilities like- CPU, Memory, Disk I/O and network traffic. To separate the application logics from other components, IRunner is used, which is a common interface. This interface pulls the common requirements for testing a variety of applications or technologies.

Srinivas Aditya et al. [2] have highlighted the benefits of adopting shift left testing in software development process. The presented study has shown the reduction in the development cost and time as the testing is done along with the development activity to avoid any delays in the process. The shift left testing method seeks the early identification of defects which lowers the total costs and also ensures the reduction of defects over time. This in turn leads to the customer satisfaction. The proposed methodology has many advantages as compared to the traditional testing. Challenges of Shift left testing were also discussed in this study. Different types of shift left testing were also explained and compared in this study like, Traditional Shift Left, Incremental Shift Left, Agile/DevOps Shift Left and Model-based Shift Left testing. The study says that the most important thing the testers focus on is the identification of bugs. This proposed Shift Left testing helps testers in identifying defects in early stage or phase which is a beneficial factor in the software development process.

Udhayakumar S P and Sivasubramanian M [3] presented a study on the early testing with Shift Left testing. The study says that the focus on Quality must start at an early phase in the development lifecycle to develop a defect free or bug free software. The proposed study is an attempt to get the underlying causes of defect injection and also to identify the ways to minimize or reduce the ways of defect injection. The study also analyses the requirements elicitation process and its effect to the Software bugs. It aims to identify the source of defect injection with a specific focus on process and people aspects. Phase Containment Efficiency (PCE) analysis was also done selected software projects using the phase injected phase detected categorization of defects. RCA (Root Cause Analysis) was also done on the same to identify the phase and causes of the injection. And finally, the cost of fixing these defects was also ascertained using the Industry standard cost data.

Jakob Engblom [4] has discussed about the pre-silicon software and system development using the virtual platforms in an ecosystem. Study says that Shift Left practice helps in breaking the dependency chains in software and hardware development so that the tasks can be started at an earlier phase and can run in parallel. It shortens the time to market and has the software ready with support for new hardware features when the hardware arrives. Inside the silicon companies, it usually means using the methodologies like - environment models, emulators, virtual platforms, FPGA prototypes, and hybrids thereof to allow firmware, boot code, drivers, operating systems, and application code to be developed, tested, and integrated before the chips arrive. Shift Left however cannot be just internal to the chip vendor, it is necessary to enable the ecosystem in the pre-silicon phase in order to build a complete product.

Ahamed Thahseen et al. [5] have presented their study on the importance of Software Testing in the development life cycle. Essential topics were identified and offered a methodology for quantifying the influence of Software Testing on the maintainability of Software, which was accomplished by performing an in-depth analysis of the corresponding or relevant literature and case studies based on the real world scenarios. The main objective of the study is to ascertain the obstacles linked with software testing and its maintainability and further propose optimal strategies to surmount these impediments. The study holds an academic significance as it offers valuable insights into the correlation between software testing and software maintainability. The study also aims to augment the extant literature on software testing and maintenance, thereby enabling software development teams to make accurate choices concerning the enhancement of software maintainability.

Dilsha D.S et al. [6] presented study on software testing frameworks. According to the study, software testing frameworks play a vital role in ensuring the software quality. It also says that selecting the appropriate testing framework is important for achieving the desired software quality goals. The presented study mainly evaluates the effectiveness of some popular software testing frameworks like – Jmeter, Selenium, TestNG, Cucumber and Cypress, on different software qualities, including maintainability, code quality, reliability, security, performance and usability. Study also includes a literature review of previous studies on the effectiveness of various software testing frameworks and an empirical study that compares the performance of these frameworks on the selected software qualities using a set of benchmark applications. Presented study shows that there is a potential for improvement of software quality with regard to user satisfaction with the testing frameworks. The majority of those surveyed claimed that using a software testing framework had improved the quality of their software.

Hutchison Steven J [7] has presented a study which highlights the importance of Shift Left in the early life cycle of software development. It states that to achieve the outcomes of Better Buying Power and deploy improved capability to the war-fighters in an effective and timely manner, we have to get the development done right and verify it through a rigorous developmental test and also evaluate the DTE before it is deployed to production. We have to Shift Left and change the paradigm which encourages the late testing in the life cycle. It also states the late testing means finding the bugs or issues or problems late which is very costly to fix.

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 7, July 2023 ||

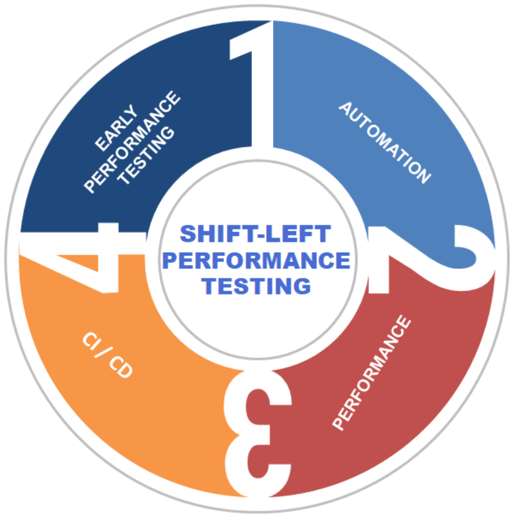
| DOI: 10.15680/IJIRCCE.2023.1107021 |

Kristian Bjerke-Gulstuen et al. [8] have described how the focus of test activities was shifted towards the earlier phases of development, which is called as “Shift Left”. This involved shifting the focus both within the iterations, and in how the overall testing work was organized. The study has finally described the results of this change and provided recommendations on how to organize the software testing activities in future for any large scale software development projects. The study herein described a large scale project where 11 development teams delivered a system in 12 three week iterations.

Mark Pitchford [9] has explained as to why it is important to design and test for security early and continuously when it comes to the embedded systems. Study says that the system connectivity has a very large impact across the embedded software development, whether safety is critical or not. It says, gone were those days when a system was developed, installed, and forgotten. The need to respond to the changing circumstances is paramount for many. Various pro-active approaches were explained. The concepts embraced by the shift-left principle are familiar to the individuals and teams developing safety-critical applications. It was observed that for many years, functional safety standards demanded a similar approach. Consequently, many best practices proven in the functional safety domain apply to security-critical applications.

VijayaShetty S and Sarojadevi H [10] presented study on the performance evaluation of Mobile and cloud applications. The data transactions between the Android mobile application and the online database were profiled and analysed for the performance issues. The profiling methodology uses open source profilers which is a cost effective one. To achieve a better performance, these profiling results can be used for optimizing the cloud based android applications. To achieve a good performance using the DDMS (Dalvik Debug Monitoring System) profiler and New Relic profiler, the SCALIBS (a kind of book store application) is optimized. The methodology used adopted the following steps – (a) Creating an online database in a registered server (b) Creating an android apk by using the developer android tools (c) Running the apk (Scalibs) in the android devices (d) Analysing the performance of Scalibs using DDMS and New Relics and (e) Collecting the results and analysing the application till the desired level of performance is achieved. Apache JMeter, which is an open source tool, has also been used in this study as a performance evaluation tool.

III. Shift-Left Performance Testing As A Solution



The Shift-left Performance Testing would be a better option in catering the performance testing needs and requirements for the latest technological stack and processes especially the Agile and DevOps processes.

The salient features of this approach include the following:
(a) Early Performance Testing
(b) CI/CD Pipeline
(c) Automation of Performance tests.

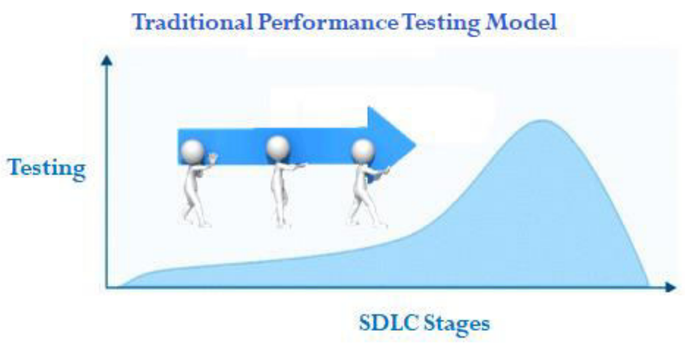
(d) Performance testing – Module or iteration-wise.

The Shift left approach with CI/CD pipeline performance testing not only helps in identifying Performance issues at the early stage of development life cycle, but also helps in optimizing and fine-tuning the application at each interval or iteration of the delivery cycle. Thus helps in meeting the performance SLAs before the build release.

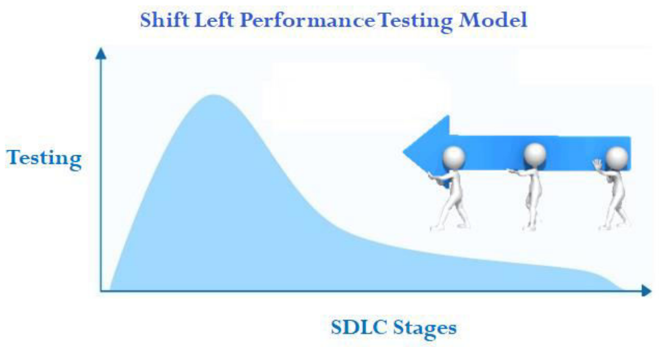
The Shift left approach involves automation of tests which avoids repeatability and human errors. The automation setup facilitates a lot in customizing the test scenario setup and test runs. Various load test scenarios and work load models can be easily customized and designed through this automated setup. With the automation set up, a good accuracy of results can also be achieved.

IV. Performance Testing – Traditional Versus Shift-Left Performance Testing Approach

In the traditional performance testing model, performance testing comes into the picture at the end of the delivery life cycle. The performance bottlenecks are identified and fixed at the final stage of the SDLC.



In the Shift left Performance testing approach, the performance issues or bottlenecks are identified and fixed at the early stage of the SDLC. Thus by moving the tasks to the left (as early as possible in the life cycle) a high level of quality is achieved.



V. Benefits Of Implementing Shift-Left Performance Testing

a) Then Shift-left approach enables the developers and Software Performance testers to conduct Performance testing at early stages of the Software Development Life Cycle.

b) This strategy allows smaller performance tests (ad hoc) against individual components as they are developed during the development life cycle. This way, the performance bottlenecks or issues are identified at an early stage. This in turn saves time, effort and budget.

c) In the CI/CD process, the shift left performance testing process helps the organizations in understanding the impact of the new components being added into the combine performance of the application. Thus it helps in identifying and uncovering the performance related issues at an early stage of the delivery lifecycle.

d) The trivial performance issues of early development cycle which takes a critical or gigantic form at the end of the development lifecycle if missed are easily fished out at an early stage and thereby saves a lot of extra effort and fits in the proverbial glove – “A stitch in time saves nine”.

e) The strategy facilitates development teams involvement actively in optimization enhancements within a short time after the detection of the application performance issues in the early stage in averse to waiting for the final completed build.

f) In this process an increased collaboration is seen between the developers and the testers. Both the teams work together with clear communication protocols for improving the quality and performance of the application. This helps in reducing the overall testing and re-testing cycle. This cultural shift in the work process helps in adding extra value to the whole development process.

g) Generally, the Key Performance Indicators (KPIs) for the performance are identified and specified at the application level. The shift-left performance testing approach defines the KPIs at the module and sub-module levels. This helps in fine-tuning and optimizing the performance of smaller units. Thus, helps in improving the performance of the application on the whole.

h) This approach has an automated set up for performance testing. The testing is conducted numerous times at different levels of the development life cycle. The automation of performance testing helps in achieving a greater efficiency and in avoiding human errors due to repeatability.

i) In the shift left performance testing approach a shared responsibility of testers and developers is seen in achieving and maintaining the quality by performing ad hoc performance tests in the development cycle. This helps in achieving and maintaining higher quality code base.

j) The CI/CD process with shift left performance testing sees a continuous performance tests conducted at module or component level for every code check-in as application is being developed. The final product thus developed will be of higher quality with very few performance issues.

k) This process enables the product teams in performing their daily tasks like, Testing, Providing feedbacks and in reviewing the changes and the progress. This way time to market is reduced due to integrated testing with the development teams with quick feedbacks.

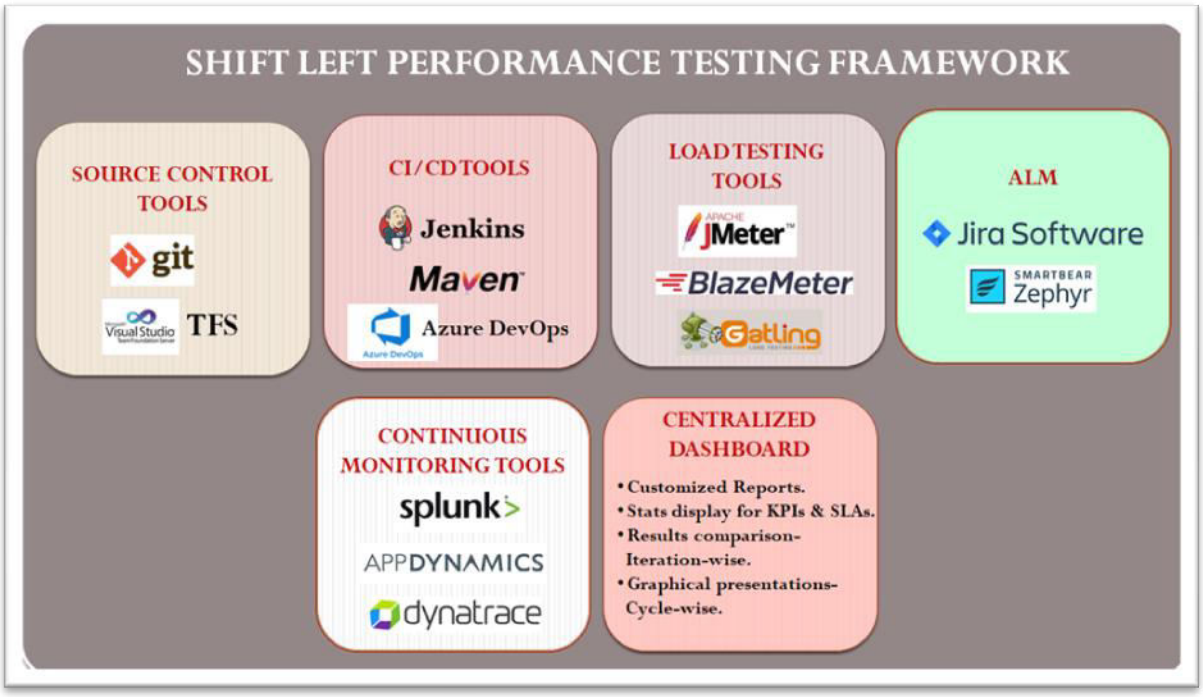
| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 7, July 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1107021 |

l) One of the major advantages of the Shift left Performance testing is the faster delivery of the software with less or fewer defects. This gives a better customer satisfaction with a better or improved user experience.

VI. Shift-Left Performance Testing Framework



The framework includes an integration of different tools across the organization. Git and TFS are used for version controlling. The CI/CD tools include Jenkins, Azure DevOps and Maven. These tools are integrated with the Load Testing Tools like Apache Jmeter, Blazemeter and Gatling. Continuous Monitoring is one of the salient features of this framework. The Continuous Monitoring provides immediate feedback and insight into the performance of the application and the interactions across different servers and environments. This helps in gauging the application performance of each test iteration and also helps in optimizing the application at module or component level. The Continuous monitoring set up also facilitates for the Infrastructure Monitoring, Network Monitoring and Application Monitoring. The Centralized reporting dashboard is an important module of this framework. The performance test metrics are well delineated at the dashboard with a drill-down option. The statistics for the performance KPIs are also displayed at the centralized dashboard. This centralized dashboard with access to the stakeholders helps them in understanding the incremental impact of the component performance tests with the trending information displayed. It also helps in providing immediate feedback. The Shift left performance testing framework is a plug and play model with a variety of tools and features enabling seamless and efficient integration of these tools.

VII. Conclusion and Future Work

The presented Shift-Left approach for Performance testing of various software applications and products verily helps in achieving a greater percentage of quality in the software by identifying the issues at an early stage of software development life cycle. It is also in line with one of the important software testing principles of ‘Early Testing’. The framework presented is a pilot framework designed to meet the requirements of a specific software project. Similar kind of a generic framework can be designed based on the project requirement and especially on the basis of Tech stack and availability of tools. The latest contemporary tools from each area of the framework section can be used to achieve the objectives of the shift-left approach.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 |

|| Volume 11, Issue 7, July 2023 ||

| DOI: 10.15680/IJIRCCE.2023.1107021 |

REFERENCES

1. Shiping C, David M, Surya N and John Z, "Yet Another Performance Testing Framework", 19th Australian Conference on Software Engineering, April 2008.
2. Srinivas Aditya V, Ramya T, Adithya Padthe and Pandu Ranga Rao A, "Shift-Left Testing Paradigm Process Implementation for Quality of Software Based on Fuzzy", Research Square, May 2023.
3. Udhayakumar S P, Sivasubramanian M, "Shift Left: Strengthening the Requirements Elicitation Process for Improving Quality Software in Software Development Projects", Research Square, May 2022.
4. Jakob Englomb, "Shifting Left Together, Enabling the Ecosystem with Virtual Platforms", Embedded World Exhibition & Conference, 2019.
5. Ahamed Thahseen, Navathanaraj A, Thakshila N, Arsath F, Dilshan S and Poojani G, "Analyzing the Impact of Software Testing on Software Maintainability", Research Square, May 2023.
6. Dilshan D S, H.M.P.P.K.H Samarasekara, Ariyarathna W.K.K.H, L.S.M.de Silva, W.L.P. de Silva, Hasara Methmini P.K, Wijewickrama G.R.P.S and Madushika H.A.J, "Evaluating the Effectiveness of Different Software Testing Frameworks on Software Quality", Research Square, May 2023.
7. Hutchison Steven J, "Shift Left! Test Earlier in the Life Cycle", Defense AT&L: September – October 2013.
8. Kristian Bjerke-Gulstuen, Tor Stalhane and Torgeir G, "High Level Test Driven Development – Shift Left", Springer International Publishing Switzerland 2015, C. Lassenius et al. (Eds.): XP 2015, LNBIP 212, pp. 239–247, 2015.
9. Mark Pitchford, "The Shift Left Principle", New Electronics, Vol. 54, Issue 14, Sep 2021.
10. VijayaShetty S and Sarojadevi H, "Performance Evaluation of Cloud and Mobile Application", 'International Research Journal of Engineering and Technology, Vol. 05, Issue 05, pp. 2974-2979, May 2018.

INNO SPACE

SJIF Scientific Journal Impact Factor

Impact Factor: 8.379

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

Scan to save the contact details

www.ijircce.com