

DESIGN SOFTWARE QUALITY ASSURANCE (SQA) FOR MOBILE APPLICATIONS QUALITY OPTIMIZATION USING AGILE DEVELOPMENT WITH CONTINUOUS INTEGRATION TOOLS AND TECHNIQUES

ALLAH BACHAYO BROHI⁴, ASIF ALI WAGAN¹, SHAFIQ-UR-REHMAN MASSAN², KAMLESH KUMAR⁵, *MUHAMMAD ADNAN KAIM KHANI¹, ASIF ALI LAGHARI¹, ABDULLAH AYUB KHAN^{1&3} AND IRFAN ALI KANDHRO¹.

¹Department of Computer Science, Sindh Madressatul Islam University, Karachi 74000, Pakistan.

²Department of Computer Science, Newport's Institute of Communications and Economics, Karachi, Pakistan.

³Department of Computer Science, Sindh Madressatul Islam University, Karachi, Sindh, Pakistan.

⁴Information Technology Center, Sindh Agriculture University TandoJam 70060, Sindh, Pakistan.

⁵Department of Software Engineering, Sindh Madressatul Islam University, Karachi 74000, Pakistan.

*Corresponding Author: adnankk12@gmail.com

ABSTRACT

Various open-source and commercial tools are available in the mobile software market today. The mobile application developers face many challenges to the quality and characteristics of the mobile application. Those mobile applications require continuous integration for delivery to develop software applications. Software Quality Assurance (SQA) The Continuous Integration (CI) is an agile software development practice which focuses on code integration and automation of the development process. Continuous integration combines the work of multiple developers at most mobile companies and is usually done at least daily to help detect any bugs early and to reduce the effort and cost, automated continuous integration tools for mobile Application plays a vital role in the software industry. Many automated constant integration tools are available hence selecting the appropriate tool is difficult in this research, we proposed a quality framework to select an appropriate continuous tool. A case study has been chosen by two automated tools Jenkins and JIRA to evaluate, and then the measured results of the software quality attributes in the framework are presented. Finally, we compare the software quality between the results before and after applying the continuous integration practices provided by the Agile development and ISO/IEC 9126 quality model is Usability, Testability, and Portability which are highly affected quality attributes of both models.

Keywords: Software Quality Assurance (SQA), Continuous Integration (CI) Tools and techniques, quality of software products (ISO/IEC 9126), agile development environment, Mobile Applications.

INTRODUCTION

The mobile utility marketplace has challenged builders to increase and launch best packages unexpectedly. This furious rivalry calls for application change to be done in a productive way. relentless incorporation (CI) is a product program advancement

rehearse which makes a claim to fame of computerization. making utilization of CI hone into the cell application change appropriately enables developers to have a compelling application change. with regards to a com rating report, more than 1/2 of all U.S. cell supporters possessed cell phones at the stop of 2012. The insights recommend the amount of cell phones endorser enhanced 29 percent from 2011, to more noteworthy than a hundred twenty-five million cell phone supporters. on a similar time, there were more prominent than one billion cell phone clients universally. The record gauge the number may bob to seventy-five billion out of 2014. A spic and span age wherein clients are always associated has unfolded. The Software Quality Assurance (SQA) turns into an essential part of today's industry and Software product quality (A. B Brohi et, all., 2017). At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics. The (Bhatti, S. N., 2005), in an agile characteristic and proposes the structure. The Continues Integration is the resulting artifacts are spontaneously created and tested. This process should identify errors as very timely in the process. Different phases multiple changes we are facing now days. The (Arcilla, R., et, al., 2014) Stage ware was a hard work which often results in code struggles hard to find bugs and even harder to fix them which led to significant delivery delays, Integrating the codes, integrating the models, Integrating the tools together to generate feedback loop, or testing for unit testing and functions test or every time you failed than generate the loop for feedback. This part depends on the foundation material will be exhibited to give a suitable viewpoint for the work in this proposal and this section contains characterize traces about programming quality, programming improvement lifecycles, persistent incorporation techniques, outrageous programming, programming joining and ceaseless combination. The (Baumeister, J., & Rutishauser, J., 2011, September) Continuous integration is a development tool that requires developers to integrate code into a shared repository code several times a day. Every developer checks the verification by automation built. Continuous integration does not get liberated of bugs, but it does make them melodramatically easier to find and remove the bugs. You can catch the bugs in time and completely transparent system since you create the feedback loop because of Jenkins. Continuous Integration is an individual parallel system (Beaumont, O., et, al., 2012, May). Developer commits the codes, CI help to identify the testers. Typical workflow of CI System. The core functionality of a continuous integration system for developing automated mobile application System is simple firstly build software than run and testing and after report the results. At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics. The build test and reporting can be performed through a script running from a scheduled task such a script would just check out a new copy of the source code from the VCS to do a build and

then run the tests. CI is a routine with regards to interfacing together subsystems or parts of programming to give a private bound together item. The product coordination improvement lifecycle the diverse parts of the product are assembled after they are executed find in figure No. 2.

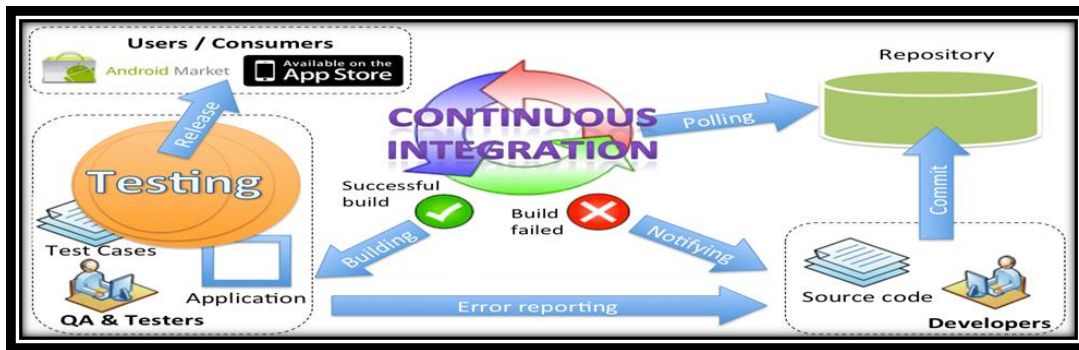


FIGURE NO.1. Continuous Integration Development Lifecycle (www.CI/testing.com)

The CI Development Lifecycle process allows multiple people and departments to work in harmony simultaneously. It does not matter which platform your app is getting launched on. CI can help to create a more efficient process whether it may have an Android app but this is especially true for the app industry, especially if hiring outside developers to work on their app. Continuous Integration speeds up the development and maintenance of mobile apps with automation. Developers will spend less time trying to figure out what is wrong with coding issues, that way they can spend more time focusing on improving the app and its functions.

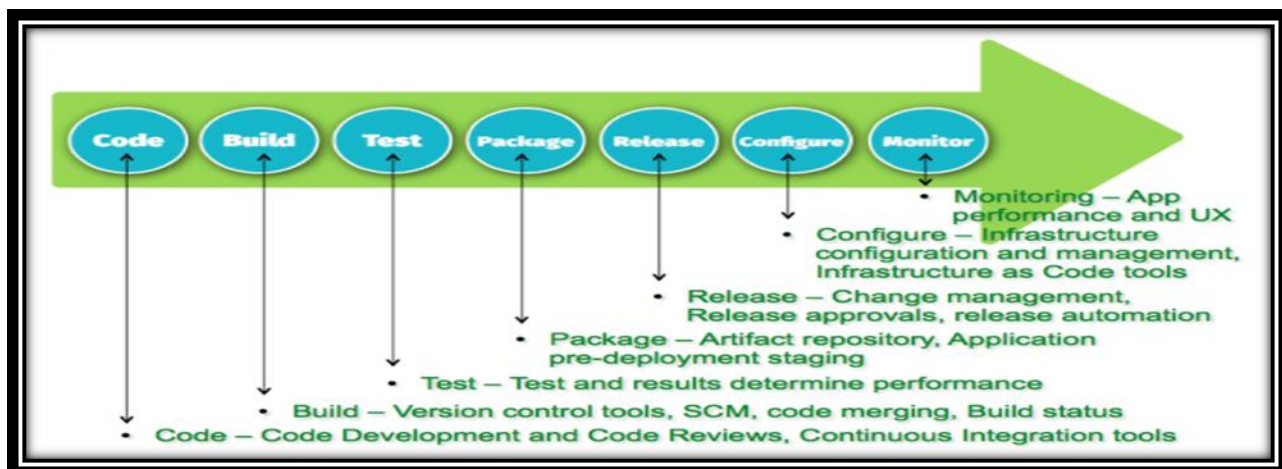


FIGURE No.2. Stages of Continuous Integration (www.Steage/ci.com)

The programmer may be writing code for the CI components. This prepares the automation for the subsequent stages. The goal here is to try and minimize the number of manual steps in show figure 2.

The system of integration, the software of integration, and tools of integration. The Software integration can be properly and simple practice when one designer executes all parts of the product. The framework is vast, and more than one engineer is taking a shot at the incorporating their segments together turn out to be massively multipart. The trouble of reconciliation is recognized in the product business. At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics. As per this issue engineers regularly and mix designers are left to a phase of the product coordination advancement lifecycle, after all segments are completely executed and are prepared to be assembled for testing. At that point this propose of the Continuously Integrating programming improvement was presented.

- Continues Integration include Implement Source Control system.
- Automate the Build Process.
- Frequent Commits, Code Consolidation, Fast builds.
- Automated Deployment.
- Self-testing the builds, Unit tests, Code Quality, Static Analysis.
- Testing in a Clone to Production environment
- Automate User Acceptance Test.
- Dashboard for Reporting.

2. LITERATURE REVIEW

(Betz, R. M., & Walker, R. C., 2013, May), this technique is targeted at the non-stop integration is a software program engineering exercise of often merging all developer operating copies with a shared foremost department, e.g., numerous instances an afternoon. That they had purposed a descriptive model for documenting and thereby higher knowledge implementations of the non-stop integration exercise and their differences. The utility of the model to an industry software development initiative then defined in an illustrative case look at a spiral model of software development and enhancement (Boehm, B. W., 1988). Researchers have solved this trouble they have evolved ARIE a cell robot gadget that mixes frontier-based total exploration with continuous localization. ARIEL localizes by means of matching its latest perceptions with the statistics saved and offered quantitative consequences that show that ARIEL can localize appropriately while exploring, and thereby construct correct maps of its surroundings. At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it

influences the product improvement process quality characteristics. This is not focused on continuous Integration Framework. In Washington (DC 20375-5337), order they have solved this trouble, they have advanced ARIEL, a cell robot device that mixes frontier-primarily based exploration with continuous localization. This studies paper addressed how to combine exploration and localization for cell robots. A robotic discovering and mapping a new environment requirement to understand its personal place and it can want a map so one can determine that vicinity. Researchers have solved this problem they have developed ARIE a mobile robotic device that mixes frontier-primarily based exploration with non-stop localization. ARIEL localizes with the aid of matching its current perceptions with the statistics stored and supplied quantitative consequences that reveal that ARIEL can localize accurately even as exploring and thereby build correct maps of its surroundings. This is not targeted on non-stop Integration Framework (Brandtner, M., et, al., 2015). The cause of this paper is targeted at the how adopting diverse Agile strategies enhance the improvement of mobile applications and if they may be used on the way to offer extra tailored system enhancements within an organization (Campos, J., et, al., 2014, September). A survey related to cellular software development procedure improvement became evolved. Simulation consequences show Doppler frequency shift has the most effective restrained effect on facts transmission performance over extensive frequency bands in excessive-velocity mobile surroundings whilst the issue vendors are time synchronized. The frequency aliasing will generate tons more interference between adjoining issue companies and therefore greatly degrades the bit errors fee performance of downlink facts transmissions. This paper is primarily based at the bit facts rat verbal exchange on the mobiles. Nashville (TN, united states), (Sarwar, A. L., & Humair Nawaz, Z. A., 2021)., this paper is supplied contributions to research on the use of SEM equipment to address business enterprise DRE device integration demanding situations. First, evaluated numerous tactics for combining non-stop integration environments with SEM equipment and defined Circuits which mixes the CUTS SEM tools with the Cruise manipulate. net non-stop integration surroundings. second, presented a case examine that shows how Circuits helps reduce the time and effort required to manage and execute integration exams that compare QoS metrics for a consultant DRE device from the area of shipboard computing. The outcomes of our case examine show that CiCUTS helps developers and testers make certain the performance of an example business enterprise DRE machine is inside its QoS specs at some stage in development, in place of waiting till system integration time is evaluated QoS. These researchers are based totally at the continuous environment, but they have not flowed non-stop Integration Framework (CI Framework) they are targeted on (Davis, CA 95616, united states) the large historical statistics on procedure metrics and effects in Git-Hub initiatives to determine the elects of one species innovation in method automation: non-stop integration. Such as Git-Hub supplied a new possibility on this

approach you will without difficulty discover big numbers of initiatives in diverse ranges of method integration and automation and gather facts on various influencing factors as well as productivity and first-class outcomes. In this paper, they are used massive historic records on method metrics and consequences in Git-Hub tasks to present the results of one specific innovation in process automation and non-stop integration. This research paper most important technique is that non-stop integration improves the productiveness of project groups, who can combine more out of doors contributions and without an observable diminishment in code satisfactory. These researchers are not primarily based on the continuous Integration Framework in this paper they have presented a realistic technique to m-studying and contact its cell Interactive getting to know gadgets (MILOs) which can be used inside a cell gaining knowledge of Engine (MLE) that runs on cellular telephones. The sign is used of cell telephones (in Europe regularly called Handles) enables a long-awaited getting to know at any area at any time. This not being tied to locations is as an instance thrilling in medication for vocational education of the medical body of workers and students. Milos is obtainable manifold opportunities for new kinds of conversation and explorative learning. This studies paper is based on them-gaining knowledge of cell smartphone, however, it not centered on the non-stop Integration framework. The (Duvall, P. M., et, al., 2007), approach of this paper is primarily based at the Hybrid method design technique and that they declare that this technique, the method used for its production, can facilitate the utility of a software program engineering technique to the production of cellular software structures. They have tested the challenges of developing software for cellular structures and starting via reviewing mobile systems characteristics and investigating the status quo of cell software development techniques. it has been shown that Agile methodologies are appropriate strategies for the improvement of such systems and primarily based on this assumption. They are diagnosed with specific requirements for a cellular software program improvement methodology primarily based on which a new agile technique has engineered the usage of the Hybrid method design approach. Those researchers are claimed that this system and the technique used for its construction can facilitate the software of a software engineering approach to the manufacturing of cellular software program structures. At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics. The (Elbaum, S., et, al., 2014, November) this system first is given a brief evaluation of non-stop and non-non-stop CA strategies, observed by means of two statistics aggregation schemes in bodily and medium get admission to manage layers. To gain up to one Gb/s peak information charge in future IMT-superior cell systems carrier aggregation generation is introduced with the aid of the 3GPP to support very-excessive-facts-fee transmissions over huge frequency bandwidths (e.g., as much as 100 MHz) in its new LTE-advanced requirements. Viable technical answers

for the asymmetric CA problem, control signaling layout, handover manage, and defend band placing are reviewed. (Franke, D., & Weise, C., 2011, March) Simulation effects show Doppler frequency shift has the handiest limited impact on statistics transmission overall performance over huge frequency bands in an excessive-pace cellular surrounding when the element companies are time synchronized. The frequency aliasing will generate a lot greater interference among adjoining issue vendors and therefore greatly degrades the bit error price performance of downlink data transmissions. This paper is based totally on the bit records rat conversation at the mobiles. In 2013, Portio considers posted a document known as "cell applications Futures 2013-2017, Portio looks into, 2013". In integration method the manner of mixing exclusive factors into an unmarried large unit or machine. it is miles frequently taken into consideration a confusing time period which causes the disorder. in the context of software development, integration suggests three one-of-a-kind processes. (Holck, J., & Jørgensen, N., 2003). The purpose of this paper is focused on the short leverages modern-day category algorithms to offer transparent and continuous cellular device protection. An exquisite characteristic is rapid's non-stop, user transparent put-up login authentication. they have tested the demanding situations of growing software for cell structures and beginning by means of reviewing cell systems traits and investigating the repute quo of mobile software improvement methods. it has been proven that agile methodologies are appropriate techniques for the development of such systems and based on this assumption. They are recognized unique necessities for a cell software program improvement methodology based totally on which a brand-new agile method has engineered the usage of the Hybrid methodology layout technique. These researchers are claimed that this system and the technique used for its creation can facilitate the utility of a software engineering technique to the production of mobile software systems. (Khani, M. A. K., et, al., 2022) on this non-stop integration improvement environments technique is primarily based on the software engineers regularly integrate new or modified code with the mainline codebases. The results of our case study display that CiCUTS enables developers and testers make certain the performance of an example agency DRE machine is inside its QoS specs during improvement, as opposed to ready until machine integration time is evaluated QoS (Brohi, A. B., et, al., 2019 July). Those researchers are primarily based at the continuous surroundings, but they do not go with the flow non-stop Integration Framework (CI Framework). System integration is the performing of consolidating the elements of an arrangement of subsystems, and programming, equipment or exclusively, to create a solitary and joined framework that fulfills the requirement for an establishment. At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics.

3. RESEARCH METHODOLOGY

The quality framework is proposed based on the results of the literature it has been exploratory that the quality component is the most critical element of each system that is utilized for the improvement of the software artifact. Today various open source and commercial tools are available in the mobile software market. The mobile application developers face many challenges as the quality and characteristics of the mobile application. Those mobile applications require continuous integration for delivery to develop software applications in an efficient manner. Continuous Integration (CI) is an agile software development practice which focuses on code integration and automation of the development process. The continuous integration tools for mobile application developers can develop a very useful and better-quality mobile application. Selecting a CI valuable or quality able tool is very difficult for continuous integration in the domain of mobile application. According to this research is searched on the internet for most demanding continuous integration tools are available for mobile application and find the comparison between three tools Jenkins and JIRA. Zhang, S. (2019, July). Software Quality Assurance: Tools and Techniques. In International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage (pp. 283-291). Springer, Cham.

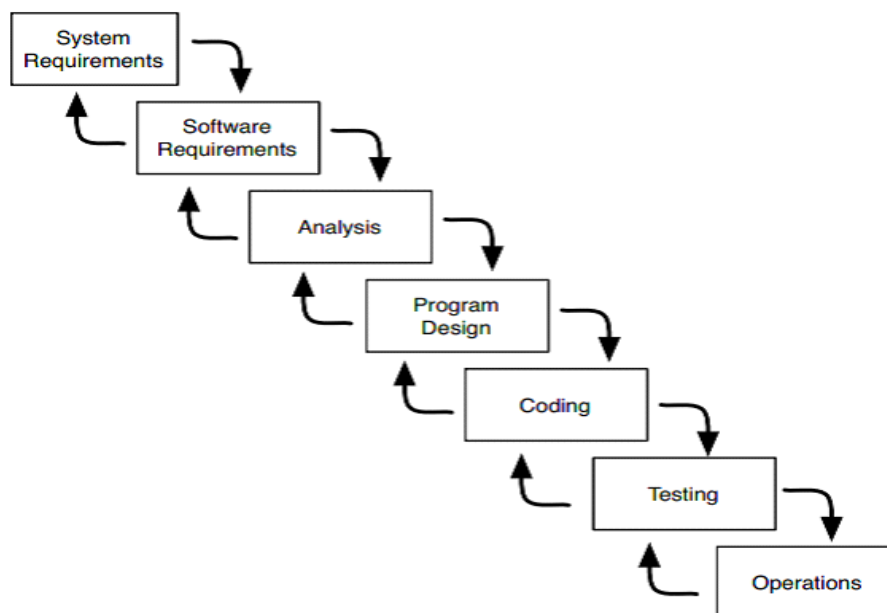


Figure No.3 Software life cycle Waterfall Model

There are numerous means of chance looking at and item prototyping. At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics. After those means are expert, the plain last item is finished after an improvement procedure that is much the same as the cascade form as demonstrated in figure 3.

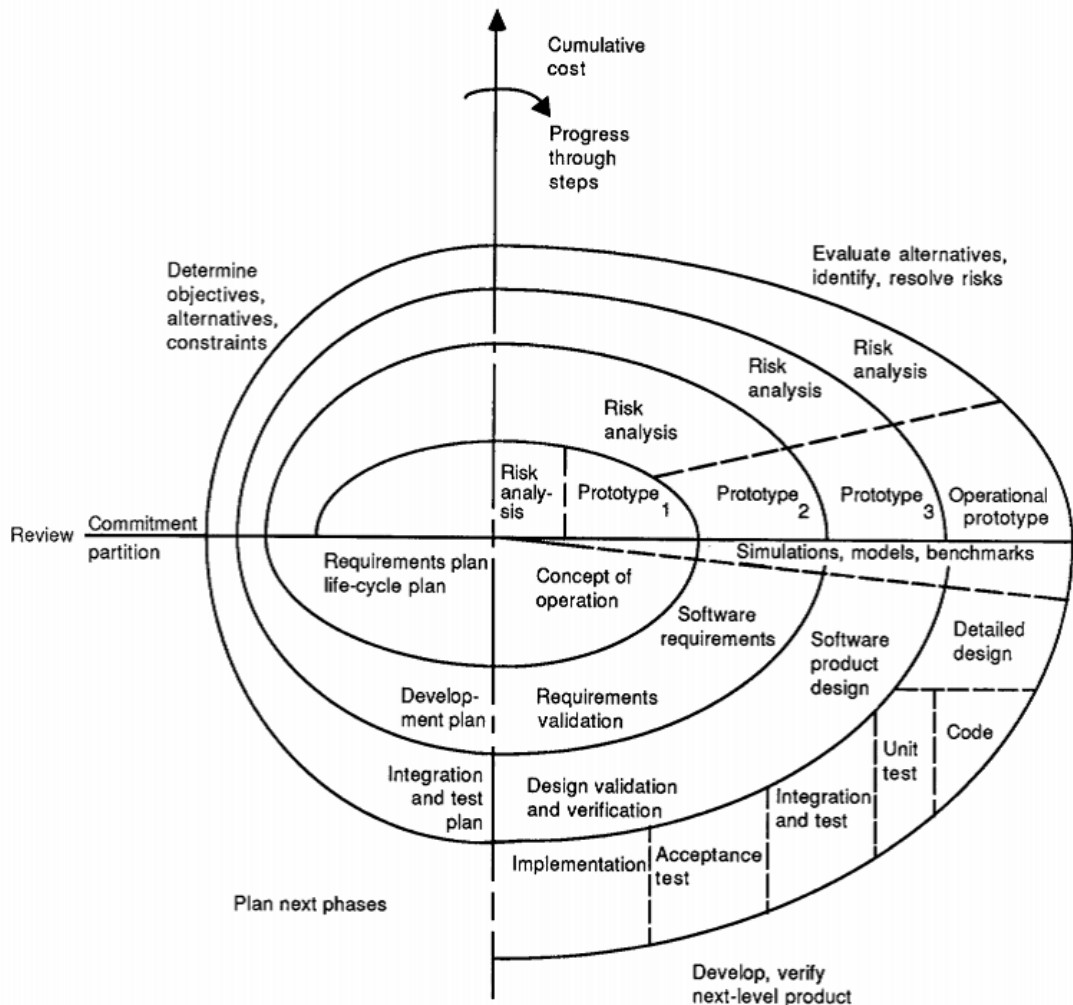


Figure No.4 Software life cycle Spiral Model

In an incremental model or a coordinated model, the advancement technique incorporates numerous little change life cycles. In everything about presence cycles, a moderate model of the product is progressed. The product program is venture by methods for step conveyed with new abilities through these transitional varieties. Totally

the item is done while its capacities are extraordinary and set up. As depicted and those change approach models are particular and have an unmistakable assortment of in general improvement ways of life cycles. At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics. The cascade show and the winding form have just a single enormous advancement presence cycle, however the incremental variant has comprised of two or three little change life cycles. The place of the product program reconciliation is the indistinguishable in every way of life cycle in each change models. After the source code is progressed and unit-inspected and the consequent objective is to join the arranged source code with other related added substances with precise designs. At this progression and all added substances or subsystems could be arranged to cooperate as it ought to be with the goal that you can make an operational gadget.

3.1 Continuous Integration of the Agile Methodology

The system of integration, the software of integration, and tools of integration. The Software integration can be properly and simple practice when one designer executes all parts of the product. The framework is vast, and more than one engineer is taking a shot at the incorporating their segments together turn out to be massively multipart. The trouble of reconciliation is recognized in the product business. As per this issue engineers regularly and mix designers are left to a phase of the product coordination advancement lifecycle. At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics.

3.1.1 Agile Methodology

An agile process is a set of values and principles used for built a mobile application. The agile methodology uses developing general purpose-specific software or tools.

3.1.2 Principles of Agile Methodology

- Our high precedence is to the pleasure of the customer thru the early and non-stop delivery of precious software.
- To convey the converting necessities even the overdue in development software.
- To deliver operating software often from multiple weeks to a couple of months with shorter timescale.
- The agile software program developer should work collectively daily during the project.
- We are running to the software program is a primary degree of progress.
- Agile processes promote sustainable development. The developers or users must be able to keep a consistent pace indefinitely.

- A continuous attention to technical excellence and precise design enhances agility.

3.2 Continuous Integration Framework for Mobile Application

According to this section, we proposed a Continuous integration framework for the mobile application. Sometimes mobile applications developers have difficulty in selecting the continuous integration framework tool for continuous integration delivery of the mobile application. At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics. We proposed a continuous integration framework that will help the mobile application developers select a continuous integration tool for mobile application from the huge list available for internet.

3.3 The Proposal of Continuous Integration Framework

Our proposed in continuous integration framework for selecting inappropriate tool of the continuous integration for mobile application. The main continuous integrating quality attributes adopted from ISO 9241 model are performance, usability, and technical support on the first layer of the continuous integration framework. Our Continuous Integration Framework is shown in Figure NO.5.



Figure NO.5. CONTINUOUS INTEGRATION FRAMEWORK

At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics. In the second layer, each attribute is further divided into the sub-attributes or sub-characteristics, these attributes will help to measure the percentage of the continuous integration tools and techniques on the bases of the metrics.

A software program developer is responsible for creating programming stock. They take a situation in getting the hang of, outlining, executing, and experimenting with programming in exceptional periods of a product change strategy. At that point they moderate for learning and outlining the objective of the product, an engineer should compose code or in a joint effort with different designers on his gathering to authorize and investigate this planned programming. The source code joins an amassed works document. At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics. New source code should be brought, and the old supply code may be refreshed while the product engineer endeavors to include new capacities or settling creepy crawlies inside the product. The situation of a product program designer in a CI perform is to devote changes to source code and additionally accepting comments from the CI contraption.

3.3.2 PERFORMANCE: A performance evaluation of the functional processes of a task and execution speed to find the percentage of performance. At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics.

Performance = Total average of efficiency + Total average of radiality + Total average of execution speed

3

3.3.1 EFFICIENCY: Efficiency is a sub-qualityattribute of the performance the agile requests depend on the performance. The following metric calculates efficiency.

$$\text{Efficiency} = \frac{(\text{Number of change codes})}{(\text{Total codes})} * 100$$

3.3.2 RELIABILITY: The reliability is a sub-qualityattribute of the performance of the reliability by providing the bug-tracking system through the continuous integration tool. This metric is calculated as follow.

$$\text{Reliability} = \frac{(\text{Number of bug-tracking from codes}) * 100}{(\text{Total Codes})}$$

3.3.3 EXECUTION SPEED: The Execution Speed is a sub-qualityattribute of the performance and Execution mentions. This metric calculated as follow.

$$\text{Average execution speed} = \frac{(\text{number of tasks run completed time}) * 100}{(\text{Total time})}$$

3.3.4 USABILITY: Usability is another basic quality properties in agile development. The percentage of the usability qualityattribute.

$$\text{Usability} = \frac{\text{Total Average of adaptability} + \text{Total Average of Tasks} + \text{Total Average of Satisfactions}}{3}$$

3.3.5 ADAPTABILITY: This sub-qualityattribute of usability shows the total percentage of users which used a different kind of tools. This sub-qualityattribute calculated as follow.

$$\text{Average adaptability} = \frac{\text{Total Number of the tool used users} * 100}{(\text{Total tools})}$$

3.3.6 NUMBER OF TASKS: This sub-qualityattribute of usability is defined as the total average of the number of tasks are executed. This sub-attribute calculated as follow.

$$\text{Number of Tasks} = \frac{\text{Total Number of executed tasks} * 100}{(\text{Total tasks})}$$

3.3.7 SATISFACTION: The Satisfaction is a sub-qualityattribute of the usability to provides. This metric calculated as follow.

$$\text{Satisfaction} = \frac{(\text{Number of question for satisfied}) * 100}{(\text{Total Questions})}$$

3.3.8 TECHNICAL SUPPORT: Technical support is that quality trait of a framework that permits a submission to work on different equipment or software stages. This qualityattribute metric can be calculated as follow.

Technical Support = $\frac{\text{Total Average of Supported languages} + \text{Total Average of supported platforms}}{2}$

2

3.3.9 SUPPORTED LANGUAGES: This is sub-quality attribute of the Technical Support can an analyzer comprehend. This sub-quality attribute metric can be calculated as follow.

Supported Languages = $\frac{\text{Total numbers of the supported languages}}{\text{(Total languages)}} * 100$

3.3.10 SUPPORTED PLATFORMS

The supported Platforms is a sub-quality attribute of the Technical Support. At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics. It is important to define the average score for each metric that will help in the recommendation of a better continuous integration tool for the mobile application and Table 1 shows a set of quality attributes that are common to the agile quality model and ISO/IEC 9126, and which were also used for this research.

If a transform is detected, the CI server retrieves documents of the most contemporary revision from the repository and triggers a construct. whilst a construct is brought about, the manufacture characters which configured for that construct is administered on the CI server. After the build script execution completes, the construct consequences are posted at the dashboard of the CI server. A CI server is not always a prerequisite for CI plays due to the fact we will write our own scripts to make a construct manually on all occasion modifications arise within the repository. however, it presents us with the blessings of sinking number of scripts that improvement teams need to write down. there are many alternatives for deciding on a CI server at the CI server market, ranging from business merchandise to open-source products. they are team metropolis, Bamboo, Jenkins, Hudson, and Cruise manipulate.

Table 1: Average Metrics of the Continuous Integration Tools and Techniques

S-NO	REMARKS	SCORE LEVEL
1	Excellent	100%
2	Very Good	80%
3	Good	60%
4	Fair	40%
5	Bad	20%
6	Very Bad	10%

The above table describes the important average value for each metric; this will help us to evaluate the quality of the continuous integration tool for the mobile application. At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics.

3.4 SOFTWARE SYSTEM OVERVIEW

The programming venture in this investigation is a budgetary framework that gives installment administrations. The task has begun over ten years prior, and its main working arrival was presented in October 2014. The framework is being produced and kept running by a data innovation organization, however, it is possessed by another element (alluded to in this proposition as the undertaking proprietor). The framework pursues an administration situated engineering. As the years progressed, new administrations were being added to the framework to adapt to the new needs and prerequisites of its clients. The software program challenged in this have a look at is a monetary device that offers payment offerings. The task has begun more than ten years ago and the primary running launch of it became delivered in October 2004. The gadget is being evolved and run with the aid of a records generation enterprise, but it is far owned by any other entity (stated in this thesis because of the assignment proprietor). The gadget follows a carrier-orientated structure. over the years, new offerings had been being introduced to the device to address the brand-new needs and necessities of its users. When new needs or administrations rise, the proprietors of the task convey this to the groups dealing with it. From that point onward, there are a few stages to experience before the new necessities get added to the framework. The means are:

- 1-The business endeavor examination team starts having necessity elicitation sessions with the proprietors to completely perceive their wants. Crafted by the undertaking assessment assemble takes around three months until the point when they have the pristine prerequisites very much reported with the achievable answers for the execution.
- 2-The business venture examiners' records are then given to the product program draftsmen to explore the most reasonable answer among the outfitted ones for making utilization of the fresh out of the box new alterations. Numerous gatherings and avocations take zone all through this portion. those gatherings regularly comprise of the business investigation group, the appropriate response designers, the product change supervisor and every so often the proprietors. The consequence of this segment is regularly the product structure record.
- 3-The configuration documents together with the proposed programming program structure are finished and given to the product program advancement team to begin the genuine usage. New administrations incorporate a few prerequisites. Necessities have assembled all things considered to make a discharge out of the venture.

4. IMPLEMENTATIONS AND RESULTS

The contrasts between the measurements of every quality factor are given, first and afterward the deliberate aftereffects of the product quality properties in the system are exhibited. At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics. At last, an examination of the product quality between the outcomes when applying the ceaseless reconciliation hones is given. The proposals, rules and best practices of this investigation depend on the outcomes introduced in this part. The hassle for databases is that manufacturing has offered venture-essential data that have to be continued. You cannot simply delete the binaries and update them with new ones. Therefore, to check the deployment of a new database model in an automatic technique, you want to automate a deployment process that preserves the country of the information. This is hard, but we now have a solution, which means that all through the improvement lifecycle, the development team can have confidence that they have completely worked and verified upgrade characters with the continuous transport surroundings attempt to indicate production scenario as lots as viable.

4.1 RESULTING SOFTWARE QUALITY ATTRIBUTES MEASURES

The season of improvement factor is estimated for the task by how much the framework client is happy with the time it took to build up the highlights. Prior to the nonstop

combination, the improvement time is the time when the venture is in the execution stage from its start until the point when it closes inside the product advancement group. With ceaseless reconciliation, the improvement stage likewise incorporates testing, construct, and arrangement of the item. When the grew, part is sent to the testing condition without issues, the advancement stage is finished for that necessity. The arrangement to finish the chose discharge was to take 7 months. Out of the arranged 7 months, a month and a half was the estimation by the undertaking administrator and the product advancement supervisor to finish the discharge execution, or, in other words, 30 days. While executing similar prerequisites and constantly coordinating parts of the code, the group burned through 26 days to finish this. The most troublesome thing that the group may confront is to get acquainted with the day-by-day reconciliations. Individuals from the group needed to incorporate even their fragmented parts of code without getting joining issues toward the finish of each working day. A few alterations must be made to the code after it was submitted yet this diminished, and the group will become acclimated to it as the work continues. Applying the Continuous Integration rehearse expects of mechanizing test and experiments each time a piece of the product included and is ensuring they cover the entire framework. Consistent Integration hone is including the recently included parts of the product additionally necessitates that the product gets tried and assembled consequently the testing result is sent instantly to the reaction as designers at the earliest opportunity in the working day. At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics

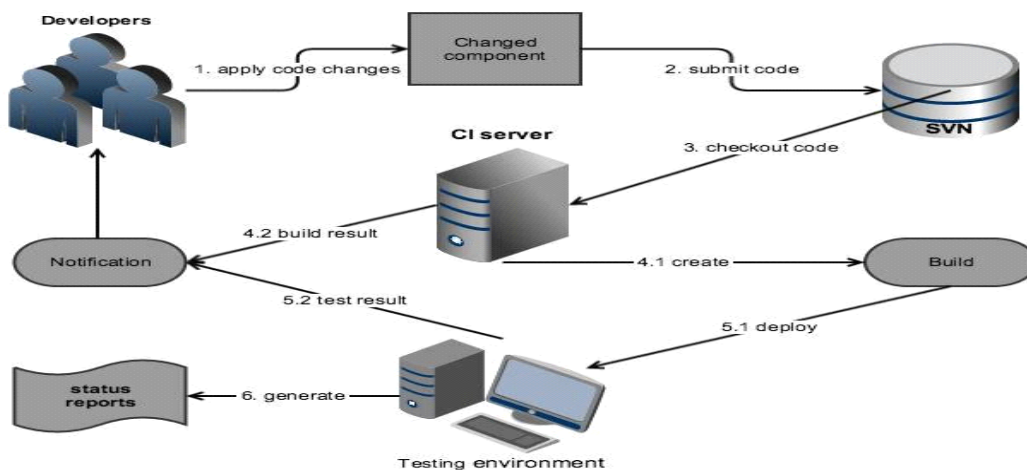


FIGURE NO.4. DEVELOPMENT PROCESS AFTER APPLYING CONTINUOUS INTEGRATION

It incorporates clarifications about the product advancement steps and Continuous Integration apparatuses. At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics.

4.2 PERFORMANCE OF JENKINS TOOL

A performance calculation of the functional processes of a task. Performance is an event in which a performer or a group of performers present. As followed. At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics. The consequence of this theory incorporates a structure to enhance the nature of dexterous programming frameworks being coordinated consistently. Adding examination to the zone of dexterous programming and constant incorporation adapts to the move of the universes' product engineers towards it. Looks into that demonstrate and depict the advantages picked up by the persistent mix of programming can energize and persuade programming makers to apply deft practices and enhance the nature of both their procedures and items. This examination gives a reasonable ordeal to an association that is as of now during the time spent moving from cascade programming improvement to light-footed programming advancement.

4.2.1 EFFICIENCY

Efficiency showed the total percentage of efficiency of quality of changed. The following metric calculates the efficiency of the Jenkins. At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics

Job					Total
Checkstyle	30	120	0	1	151
Dry	35	120	0	1	156
FindBugs	40	100	0	0	140
PMD	25	100	1	0	126
Static Analysis Collector	40	140	0	0	100
Static Analysis Utilities	30	130	3	4	107
Task Scanner	20	120	19	1	120
Warnings	30	130	0	0	100
Total	250	960	23	7	1,000

FIGURE. NO. 5. SHOWS THE CODE-CHANGED OF THE JENKINS TOOL



Show the Total Numbers of code changed = 960

$$\text{Efficiency} = (960/1000) * 100$$

$$\text{Eff} = 96\%$$

4.2.2 RELIABILITY: The reliability is a sub-attribute of the performance. This metric is calculated as follow. At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics.

Job	Warnings	Errors	Failures	Total
Checkstyle	30	120	0	151
Drv	35	120	0	156
FindBugs	40	100	0	140
PMD	25	100	1	126
Static Analysis Collector	40	140	0	100
Static Analysis Utilities	30	130	3	107
Task Scanner	20	120	19	120
Warnings	30	130	0	100
Total	250	960	23	1,000

FIGURE NO.6. SHOW THE TOTAL BUGS OF THE JENKINS TOOL



Show the Total Numbers of Bugs = 250

$$\text{Reliability} = \frac{(250)}{(1000)} * 100$$

$$\text{Reliability} = 25\%$$

4.2.3 Execution Speed: The Execution Speed is a sub-attribute of the performance. This metric calculated as follow.

Test	# Passed	# Skipped	# Failed	Time (ms)	Included Groups	Excluded Groups
Default suite						
Default test	1,080	0	0			
Class				Method	Start	Time (ms)
Total Default Test=1,,440						
Default test — passed = 1,080						

Figure. No.7. Show total Tasks executed at the time of the Jenkins Tool

$$\text{Average execution speed} = \frac{(1,080)}{(1,440 \text{ mint})} * 100$$

$$\text{Average execution speed} = 75\%$$

$$\text{Performance} = \frac{96 + 25 + 75}{3}$$

$$\text{Performance} = 65.333\%$$

4.3 USABILITY OF JENKINS TOOL: Usability is a process that specific users used a product, and it measured the quality of the Jenkins Tool. At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics.

4.3.1 Adaptability: Users who used a different kind of tools for the (CI) methods. This sub-attribute calculated as follow and find the Jenkins quality of the Jenkins tool.

$$\text{Average adaptability} = \frac{7 * 100}{(10)}$$

$$\text{Average adaptability} = 70\%$$

4.3.1 NUMBER OF TASK: This sub-attribute of usability is defined as the total average of the number of tasks are executed. This sub-attribute calculated as follow.

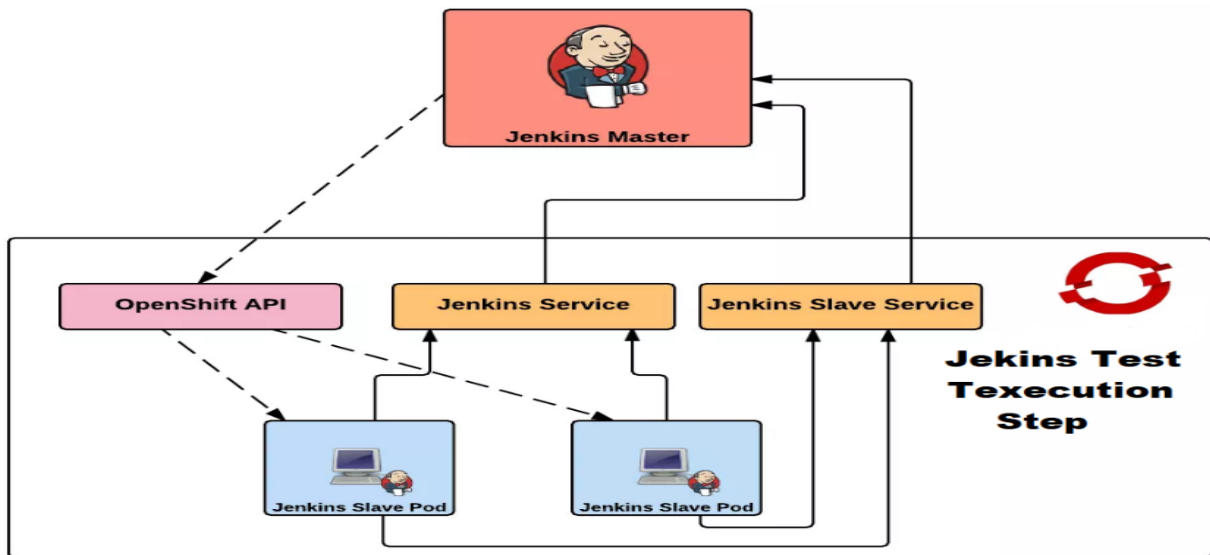


FIGURE. NO.8. CONTINUOUS INTEGRATION JENKINS EXECUTION

$$\text{Number of Tasks} = \frac{(98) * 100}{100}$$

$$\text{Number of Tasks} = 98\%$$

4.3.3 SATISFACTION: The many users used which kind of (CI) tool. it and find the quality of the Jenkins tool. This metric calculated as follow.

$$\text{Satisfaction} = \frac{(84) * 100}{(100)}$$

$$\text{Satisfaction} = 84 \%$$

4.4 TECHNICAL SUPPORT: Technical support is provided the average of the continuous integration tools are supported. This attribute metric can be calculated as follow.

4.4.1 Supported Language: This sub-attribute metric can be calculated as follow.

$$\text{Supported Languages} = \frac{(8) * 100}{10}$$

$$\text{Supported Languages} = 80\%$$

4.4.2 Supported Platforms

The supported Platforms is a sub-attribute of the Technical Support of the continuous integration tools. This sub-attribute can be calculated as follow.



Figure. No .9. Supported Platforms of Jira and Jenkins Tools

$$\frac{\text{Supported Platforms}}{80} = \frac{(68) * 100}{80}$$

$$\text{Supported Platforms} = 85\%$$

$$\text{Technical Support} = \frac{80 + 85}{2}$$

$$\text{Technical Support} = 165/2$$

$$\text{Technical Support} = 82.5\%$$

Table No.3. Show the Score continuous integration tools of the quality attributes

S-NO	TOOLS	QUALITY	SCORE LEVEL	REMARKS
1	Jenkins	Performance	65.333%	Good
	Jira	Performance	40.33%	Fair
2	Jenkins	Usability	84%	Good
	Jira	Usability	36%	Fair
3	Jenkins	Technical Support	82%	Good
	Jira	Technical Support	26%	Bad

The above table describes the important average value for each metric. This will help us to evaluate the quality of the continuous integration tool for the mobile application.

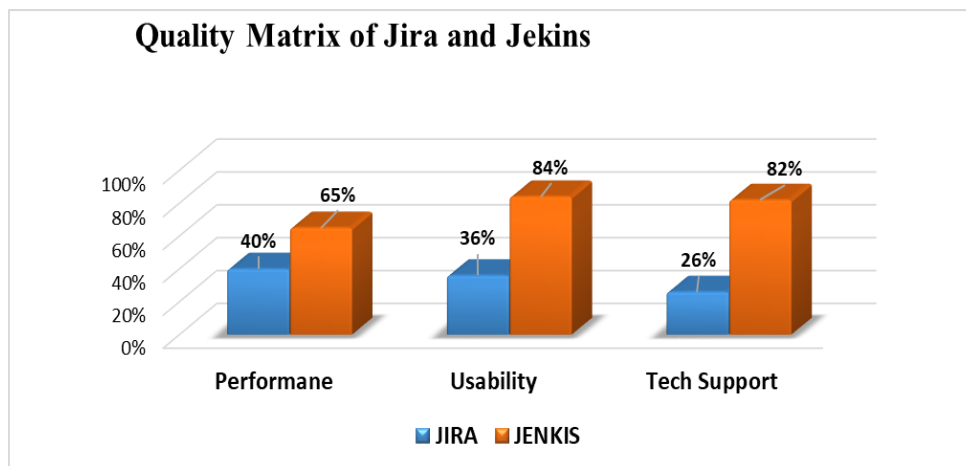


FIGURE NO.10. PERFORMANCE, USABILITY AND TECHNICAL SUPPORT OF CONTINUOUS INTEGRATION TOOLS

The above Figure No.10. Describes the important average value for each metric. This will help us to evaluate the quality of the Jenkins is best from JIRA in Ccontinuous lintegration tool for the mobile application.

It incorporates clarifications about the product advancement steps and Continuous Integration apparatuses. At last, it incorporates an area about the coordination period of the product mix and the distinctive strategies for incorporating the product parts and how it influences the product improvement process quality characteristics.

6. CONCLUSION

This part comprises three segments. The main area condenses crafted by this proposal. The second segment clarifies the commitment of this proposition to the field of programming building. The third and last part incorporates a few suggestions and future research bearings. Software producers are shifting Programming makers are moving their advancement hones towards consistent combination as it demonstrated a noteworthy change to the general programming quality. A significant number of the dangers engaged with the product coordination process were moderated because of incorporating parts of the created programming consistently once it is prepared. Continuous integration helped in estimating the strength of the product while yet being created on the grounds that tests are robotized as a piece of the combination procedure. At the point when the well-being status of the product is estimated consistently, remedial moves are proficiently made before the imperfections are aggregated. At the point when a deformity is found after a piece of the product is incorporated to a tried improvement condition, it confines the extent of researching the reason for the imperfection just to the recently included part. This helps find and fix the imperfection more productively and in less time.

FURTHER WORK: May be added to the difficulties and challenges in introducing continuous integration to a current development process for developers (especially experts) to change their daily activities and to adopt new practices checking in the created code to the variant control framework, surrenders are recognized prior, and a move is made promptly instead of finding identifies in a later phase of the product advancement lifecycle. The consistent reconciliation practices will be exhibited in this section and in conclusion a transient about the connection between programming quality and the ceaseless mix of the product framework.

REFERENCES:

1. Arcilla, R., Brown, D., & Herman, M. (2014). Continuous integration and deployment software to automate nuclear data verification and validation. *Nuclear Data Sheets*, 118, 422-425.
2. Baumeister, J., & Reutelshoefer, J. (2011, September). Developing knowledge systems with continuous integration. In *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies* (pp. 1-4).
3. Beaumont, O., Bonichon, N., Courtès, L., Dolstra, E., & Hanin, X. (2012, May). Mixed data-parallel scheduling for distributed continuous integration. In *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum* (pp. 91-98).
4. Betz, R. M., & Walker, R. C. (2013, May). Implementing continuous integration software in an established computational chemistry software package. In *2013 5th International Workshop on Software Engineering for Computational Science and Engineering (SE-CSE)* (pp. 68-74). IEEE.
5. Bhatti, S. N. (2005). Why quality? ISO 9126 software quality metrics (functionality) support by UML suite. *ACM SIGSOFT Software Engineering Notes*, 30(2), 1-5.

6. Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, 21(5), 61-72.
7. Bonichon, N., Courtès, L., Dolstra, E., & Hanin, X. (2012, May). Mixed data-parallel scheduling for distributed continuous integration. In *2012 IEEE 26th International Parallel*.
8. Brown, D., & Herman, M. (2014). Continuous integration and deployment software to automate nuclear data verification and validation. *Nuclear Data Sheets*, 118, 422-425.
9. Brandtner, M., Giger, E., & Gall, H. (2015). SQA-Mashup: A mashup framework for continuous integration. *Information and Software Technology*, 65, 97-113.
10. Sarwar, A. L., & Humair Nawaz, Z. A. (2021). Analysis Of Session Initiation Protocol with VoIP in Multimedia Conferencing System. *International Journal*, 10(3).
11. Campos, J., Arcuri, A., Fraser, G., & Abreu, R. (2014, September). Continuous test generation: enhancing continuous integration with automated test generation. In *Proceedings of the 29th ACM/IEEE international conference on Automated software engineering* (pp. 55-66).
12. Dromey, R. G. (1995). A model for software product quality. *IEEE Transactions on software engineering*, 21(2), 146-162.
13. Duvall, P. M., Matyas, S., & Glover, A. (2007). *Continuous integration: improving software quality and reducing risk*. Pearson Education.
14. Elbaum, S., Rothermel, G., & Penix, J. (2014, November). Techniques for improving regression testing in continuous integration development environments. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering* (pp. 235-245).
15. Franke, D., & Weise, C. (2011, March). Providing a software quality framework for testing of mobile applications. In *2011 fourth IEEE international conference on software testing, verification, and validation* (pp. 431-434). IEEE.
16. Dolstra, E., & Hanin, X. (2012, May). Mixed data-parallel scheduling for distributed continuous integration. In *2012 IEEE 26th International Parallel*.
17. Holck, J., & Jørgensen, N. (2003). Continuous integration and quality assurance: A case study of two open-source projects. *Australasian Journal of Information Systems*, 11(1).
18. Khani, M. A. K., Wagan, A. A., Laghari, A. A., Hyder, M., Mughal, Z. A., Sarwar, R., & Khan, A. A. DESIGN FRAMEWORK FOR THESUBVERSION (SVN) REPOSITORIES SYSTEM SOFTWARE.
19. Brohi, A. B., Butt, P. K., & Zhang, S. (2019, July). Software Quality Assurance: Tools and Techniques. In *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage* (pp. 283-291). Springer, Cham.
20. Matyas, S., & Glover, A. (2007). *Continuous integration: improving software quality and reducing risk*. Pearson Education.
21. A. B Brohi et, all, "Automated Software Testing Tools for Software Quality Model Using of Regression Testing Framework" in international conference at SAU Tandojam (2017).
22. Matyas, S., & Glover, A. (2007). *Continuous integration: improving software quality and reducing risk*. Pearson Education.
23. Rothermel, G., & Penix, J. (2014, November). Techniques for improving regression testing in continuous integration development environments. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering* (pp. 235-245).
24. Weise, C. (2011, March). Providing a software quality framework for testing of mobile applications. In *2011 fourth IEEE international conference on software testing, verification, and validation* (pp. 431-434). IEEE.
25. Zhang, S. (2019, July). Software Quality Assurance: Tools and Techniques. In *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage* (pp. 283-291). Springer, Cham.