IJCSE International Journal of Computer Sciences and Engineering Open Access

Survey Paper Vol.-7, Issue-6, June 2019 E-ISSN: 2347-2693

GUI for Continuous Integration and Automatic Bug Verification of Jenkins Server

Vaishnavi R. Mali[1*], **Anil R. Surve**[2]

[1,2] Department of Computer Science and Engineering, Walchand College of Engineering, Sangli, Maharashtra, India, 416415

Corresponding Author: vaishnavimali16@gmail.com

DOI: https://doi.org/10.26438/ijcse/v7i6.497500 | Available online at: www.ijcseonline.org

Accepted: 11/Jun/2019, Published: 30/Jun/2019

***Abstract*—**In software engineering, continuous integration (CI) is the routine with regards to blending all engineer working duplicates to a common mainline a few times each day. The principle point of CI is to avert coordination issues. Jenkins is a driving open source constant coordination server worked with Java. It is utilized to manufacture and test programming ventures constantly making it simpler to incorporate changes to the task. It provides more than 985 plugins to help to build and to test any project. Continuous Integration (CI) requires developers to integrate code into a shared repository several times a day. GUI will maintain all latest jobs with their build status and continuous notification referring to design issues, code, failures, etc. Once the project is configured in Jenkins then all future builds are automated. Hence GUI will help to represent basic reporting like run status (successful, failure, unstable) and weather reports (job health). GUI will also provide automatic continuous regression run which will give flag details of new relevant changes in build compared to last run and easy access for code change mapping and auto-selection. GUI will help to eliminate manual verification.

***Keywords*—**Jenkins, Continuous Integration and Continuous Delivery, Auto Build Trigger, Python Django, Perforce

I. Introduction

Jenkins is a free framework. It is written in Java. Jenkins mechanizes the software development process which can be a nonhuman part, with continuous integration and encouraging specialized parts of continuous delivery. It is a continuous integration server based framework that keeps running in different compartments, for example, Apache Tomcat. It supports source code management tools, for example, CVS, Git, Perforce and can execute arbitrary shell scripts and Windows batch commands [6].

Builds are activated with the help of different methods, for instance by submitting in a source code management framework, by scheduling by means of a cron-like system and by mentioning a particular form URL. It can likewise be triggered after different forms in the queue have finished. Jenkins usefulness can be reached out with plugins. Plugins can be accessible for coordinating Jenkins with respect to version control stretches out its utilization to project written in programming languages other than Java. Many developing tools are upheld through their individual plugins. Plugins can likewise change the visual of Jenkins or include new features. There are a number of plugins committed to the motivation behind unit testing that produces test reports in a different format and automated testing that supports various tests. Builds can produce reports of a test run in different format upheld by plugins and Jenkins can show the reports and create patterns and render them in the respective GUI and permits for email notifications for build results.

Jenkins has an extremely convoluted structure. So we generally need to do some setups manually before running builds and dealing with a Jenkins server needs exceptional consideration and a devoted engineer. This adds worker hours to the task and pushes up the general undertaking cost too. The organization has various components with a number of test cases, so there is a need for user-friendly platform i.e. GUI which will do all setups automatically and also integrate the code. The GUI will maintain all latest jobs with their build status and continuous notification referring to design issues, code, failures, etc. Once the project is configured in Jenkins then all future builds are automated [10].

Hence GUI will help to represent basic reporting like run status (successful, failure, unstable) and weather reports (job health). GUI will also provide automatic continuous regression run which will give flag details of new relevant changes in build compared to last run and easy access for code change mapping and auto-selection. The GUI will provide automatic build trigger from the Jenkins server. It will also maintain the latest changes from perforce and give user complete transparency. GUI will provide ease of access to records, results to the user. It will reduce manual efforts of configuration, verification, testing, etc.

In this paper, the proposed system is GUI for continuous integration and automatic bug verification of the Jenkins server. After the first two sections which is introduction and applications, the architecture is characterized in section IV, the applications and the research problems are characterized in section II, Related work in section III and project methodology are a reference to section V. Furthermore, conclusion and future work of this paper are in the last section.

II. Related Work

During the time spent data framework enhancement and redesigning in the period of distributed computing, the number and assortment of business prerequisites are progressively perplexing, keeps supported development, the procedure of persistent mix conveyance of data frameworks turns out to be progressively unpredictable, the measure of dreary work is developing. This paper centers around the persistent incorporation of explicit data frameworks, a community oriented work plot for constant coordinated conveyance dependent on Jenkins and Ansible is proposed. Both hypothesis and practice demonstrate that persistent incorporated conveyance agreeable frameworks can successfully improve the productivity and nature of consistent coordinated conveyance of data frameworks. The impact of the enhancement and overhauling of the data framework is self evident [1].

CICD pipeline to have three mechanization stages named benchmark, load test, and scaling. It limits the framework intrusion by utilizing test seat approach when framework benchmarking and it utilizes the creation traffic for burden testing which gives progressively exact outcomes. When benchmark and burden test stages are finished, framework scaling can be assessed. At first, the pipeline was created utilizing the Jenkins CI server, Git archive, and Nexus store with Ansible computerization. At that point, GoReplay is utilized for traffic duplication from generation to test seat condition. Nagios checking is utilized to break down the framework conduct in each stage and the after effect of test seat has demonstrated that scaling is able to deal with a similar burden while changing the application programming, yet it doesn't enhance reaction time of use at critical dimension and it lessens the danger of use organization by coordinating this three-stage approach as CICD robotization broadened include. In this manner, the examination gives a compelling approach to oversee Agile based CICD ventures [2].

Continuous Integration (CI) is the most widely recognized technology among programming engineers where the work is coordinated into a benchmark every now and again. The business is confronting immense difficulties while creating software at numerous locales and tried on various stages. The most ideal approach to make CI faster and increasingly productive is to mechanize the assemble and process of testing. Jenkins is an open source CI apparatus that aides in computerizing the total star cess, lessening crafted by a designer and check the improvement at each progression of software development. In this paper, the usage of Jenkins for programming patch mix and discharge to the customer is examined. A genuine situation is considered, how the product improvement is done in corporate endeavors and how Jenkins spare engineers/integrators critical work hours via robotizing the total procedure. In this paper, Jenkins is executed in a manner of a master/slave, where Jenkins server is master and slaves are the Jenkins customers [3].

Programming setup devices are getting to be well-known step by step. Here paper portrays an open source ceaseless combination instrument: Jenkins, which is a server situated game plan that keeps running in a servlet. The structure, usefulness, and use of Jenkins are introduced. The point of this exploration paper is to stress on the Jenkins Integration Development Environment (IDE) and think about five programming combination instruments to decide their ease of use and adequacy [4].

Definite setup of a test domain which is utilized for Near Field Communication (NFC) Integrated Circuits (ICs) at ams AG. The test condition is utilized for the check and approval of the NFC ICs, just as for pre-verification trials, and can use in two different ways: (1) manual execution for each designer to have the option to run tests around their work area before submitting code changes and to repeat coming up short experiments and (2) computerized execution which is essential for the Continuous Integration (CI) approach which is pursued during advancement and to guarantee that all tests are kept running against the Device Under Test (DUT). Initial a depiction of the framework which will be tried is given. A short time later the utilized instruments and techniques to execute the different test seats are talked about. The obstacles which appeared during the way toward setting up the earth for manual just as computerized execution are clarified and the utilized arrangement is talked about [5].

III. Methodology

A. Architecture

The system architecture has mainly three levels Frontend, Middleware, Backend as shown in Fig.1.

GUI is the main component of frontend level which is built in Angular with Clarity Module. Django REST API is called by GUI, so GUI will have all the information from Jenkins server and it also contains directory information from Perforce. GUI will populate all this information in a more representable manner. GUI will provide various options like run test cases, view infrastructure required for specific Protocol component. It will give options to the users of selecting Branch, Changelist number of their choice. It will also show test case results, queue information, test case run history, etc.

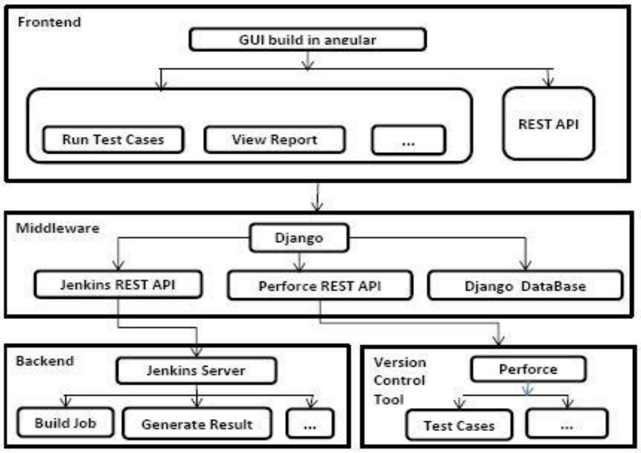

Figure 1. System Architecture.

Django is the component of middleware which have integrated SQLite DB in it. Django is written in python. it will call Jenkins REST API to GET/POST information to the server. It will get all directory structure by calling P4(Perforce) REST API. All the information from P4 and Jenkins server will be stored in Django DB. Jenkins server will act as a backend for GUI and Git/perforce is a source code management software which is also known as version control.

B. Agile Methodology

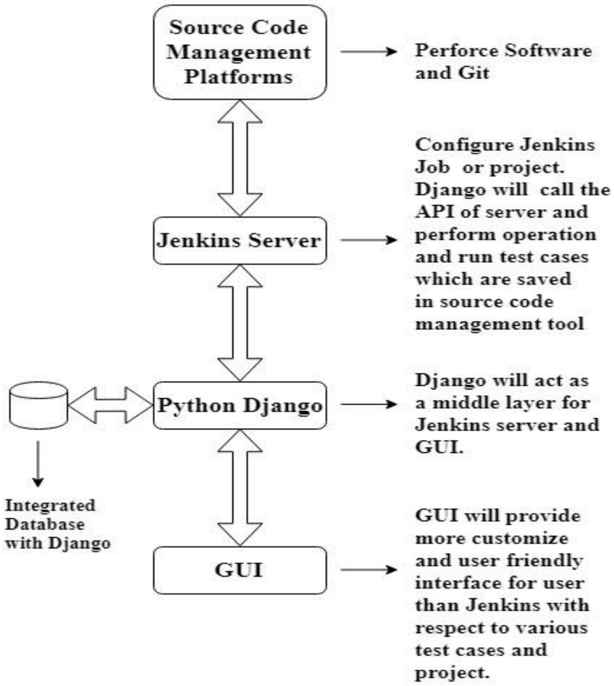
Source Code Management Platforms
Perforce Software and Git
Jenkins Server
Configure Jenkins Job or project. Django will call the API of server and perform operation and run test cases which are saved in source code management tool
Python Django
Django will act as a middle layer for Jenkins server and GUI.
Integrated Database with Django
GUI
GUI will provide more customize and user friendly interface for user than Jenkins with respect to various test cases and project.

Figure 2. Workflow diagram.

AGILE methodology is a preparation that advances steady accentuation of progress and testing between all the item headway life cycle of the endeavor. Both development and testing practices are at the same time is not normal in the Waterfall model [7]. Agile method is for dealing with better program building and finding necessities and arrangements to develop. It provides versatile arranging, transformative advancement, early conveyance, and consistently improvement, and it gives support to the changes which are fast and adaptable.

C. Jenkins Server

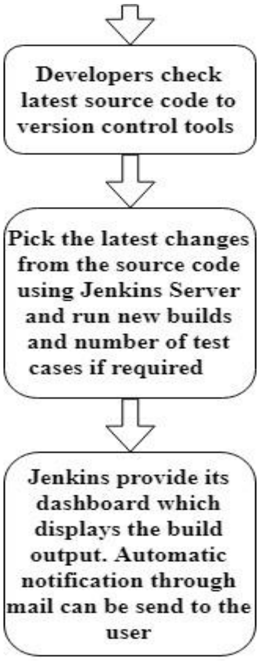
Developers check latest source code to version control tools
Pick the latest changes from the source code using Jenkins Server and run new builds and number of test cases if required
Jenkins provide its dashboard which displays the build output. Automatic notification through mail can be send to the user

Figure 3: Jenkins Generalize Workflow.

Jenkins is a free framework. It is written in Java. Jenkins computerizes the automatic part of the product advancement process, with consistent joining and encouraging specialized parts of ceaseless conveyance. It is a continuous integration based framework that keeps running in various compartments, for example, Apache Tomcat [6].

D. Python Django and Django REST framework

Django is based on Python which open and free source web framework, which pursues the model-view-template (MVT) compositional pattern. It is kept up by the Django Software Foundation (DSF). Django's essential objective is to facilitate the formation of very complicated sites which requires databases. The structure stresses reusability and pluggability of components, less code, low coupling, quick advancement, and the rule of don't do the same thing by yourself. Python is completely utilized for setting up documents and informative models. Django also gives administrative operations like CRUD (create, read, update, delete) interface that is produced progressively through configured by administrator models. Django REST structure is an incredible and adaptable toolkit for structure Web APIs. The Web API which can be used with enormous ease and win for your designers [7].

E. Source Code Management Platform

Git, a subsidiary of Microsoft, is an American online facilitating administration for source code management also known as version control utilizing Git. It is mainly utilized for computer coding. It offers the majority of the circulated version control and source code management (SCM) usefulness of Git just as including its very own highlights. It gives access control and various collaboration significations such as error and bug finding, feature requests, work management, and information for every project. Perforce is a source code management tool wherein clients associated with a shared repository. Perforce is utilized to share files between the record archive and individual clients workstations [8], [9].

IV. CONCLUSION

We have presented GUI for continuous integration and automatic bug verification of Jenkins server, The GUI will provide automatic build trigger of Jenkins server. It will also maintain the latest changes from perforce and Git. It will give the user complete transparency. GUI will provide ease of access to records, results to the user. It will reduce manual efforts of configuration, verification, testing, etc.

ACKNOWLEDGMENT

We would like to express our gratitude to all the authors of research papers. Also towards the Computer Science and Engineering Department of Walchand College of Engineering, Sangli for the support and guidance.

REFERENCES

[1] W. y. PetroChina, Z. t. PetroChina and G. y. PetroChina, *"Design and implementation of continuous integration scheme based on Jenkins and Ansible"*, In the Proceedings of the 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD), **Chengdu**, pp. **245- 249, 2018.**

[2] S. A. I. B. S. Arachchi and I. Perera, *"Continuous Integration and Continuous Delivery Pipeline Automation for Agile Software Project Management"*, In the Proceedings of the 2018 Moratuwa Engineering Research Conference (MERCon), **Moratuwa**, pp. **156-161, 2018**

[3] N. Seth and R. Khare, *"ACI (automated Continuous Integration) using Jenkins: Key for successful embedded Software development"*, In the Proceedings of the 2015 2nd International Conference on Recent Advances in Engineering and Computational Sciences (RAECS), **Chandigarh,** pp. **1-6, 2015.**

[4] P. Rai, Madhurima, S. Dhir, Madhulika and A. Garg, *"A prologue of JENKINS with comparative scrutiny of various software integration tools"*, In the Proceedings of the 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), **New Delhi**, pp. **201-205, 2015 .**

[5] S. Puri-Jobi, *"Test automation for NFC ICs using Jenkins and NUnit"*, In the Proceedings of the 2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW), **Graz**, pp. **1-4, 2015.**

[6] C. Mamatha, S C V S L S R. Kiran, "A Novel Approach for Cloud Computing Environment *"Implementation of DevOps Architecture in the project development and deployment with help of tools"*, International Journal of Scientific Research in Computer Science and Engineering, Vol.**6**, Issue.**2**, pp.**87-95, 2018.**

[7] C. Sisodiya P. Sharma, *"Scrutiny to Effectiveness of Various Software Process Model"*, International Journal of Scientific Research in Computer Science and Engineering, Vol.**5**, Issue.**2**, pp.**88-93, 2017.**

Authors Profile

Vaishnavi R. Mali is research scholar completed Bachelor of Engineering from Amravati University, Maharashtra, India in 2017 and pursuing Master of Technology(Computer Science and Engineering) from Walchand College of Engineering, Sangli [Maharashtra].

Anil R. Surve is a faculty in Computer Science and Engineering department of Walchand College of Engineering, Sangli [Maharashtra]. He is currently pursuing Ph.D. (Computer Science and Engineering) in the area of SOA based context-aware smart Pervasive Computing. He has 18 years of engineering academic experience where he has dealt with most of the Computer Science & engineering core courses. He has mentored more than 300 undergraduate and 40 postgraduate student projects on the latest innovative technologies. His research publications include 29 papers in national/international conferences and 10 in international journals.