CI Automation for HausKhaas

Saket Thaploo[1], Jelson Chacko[2], Rushikesh Musmade[3], Yogesh Kalme[4]
[1, 2, 3, 4] Department of Information Technology
[1, 2, 3, 4] D. Y. Patil College of Engineering, Akurdi

Abstract-*Continuous Integration also abbreviated as CI in the cooperate world, is the most common practice among software developers where their work is integrated into a baseline on a periodic basis. There are various challenges faced by the industries, while developing Software (S/W) s for various purposes at multiple sites and tested on multiple platforms. Automating the build process along with the testing of the software, are the best ways to make Continuous Integration efficient and faster. Jenkins is a CI tool that helps in automating the complete process, reducing the work of a developer and check the development at each and every step of S/W evolution. We proposed the implementation of Jenkins for software patch integration and release to client. In order to understand the current scenario, a real-life scenario has been considered in which the software development process in corporate ventures is studied and how Jenkins can be used to save developers / integrators crucial work hours by automating the complete process. Jenkins can pull the code from subversion repository and then can be used to build the code. Information must be accurate, readily available, easily found, and all team members must have access to all this and notified on real time basis in order to succeed in a world where technologies, requirements, ideas and the tools being used for the project are constantly changing. In order to overcome or suppress these challenges, modern software development has evolved in such a way that key concepts of adaptability to change and data-driven project management are considered.*

Keywords-Continuous Integration, DevOps, Review Board, Subversion, Jenkins, HausKhaas.

I. INTRODUCTION

Software development processes are fundamentally based on efficient and effective communication. Communication between engineers, between engineers and managers, and between teams and clients are all essential components of a successful project. Requirements must be effectively transferred from client to engineer, specifications must be transitioned from architect to engineer, and constant communication between project team members, managers, and clients throughout the project life cycle is critical to the success of projects of any complexity [1]. To succeed in a world where technologies, requirements, ideas, tools, and timelines are constantly changing, information must be accurate, readily available, easily found, and ideally delivered constantly, in real-time, to all team members. To meet these challenges, modern software development has evolved to encompass key concepts of adaptability to change and data-driven project management. A recent movement dubbed DevOps [2] has attempted to use automated systems to bridge the information gap between project team entities and to enforce rigorous processes to ensure real-time communications. DevOps (a clipped compound of development and operations) is a culture, movement or practice that emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals while automating the process of software delivery and infrastructure changes. It aims at establishing a culture and environment where building, testing, and releasing software can happen rapidly, frequently, and more reliably. Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository several times a day [3]. Each check-in is then verified by an automated build, allowing teams to detect problems early. Continuous delivery (CD) is a software engineering approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time. It aims at building, testing, and releasing software faster and more frequently. The approach helps reduce the cost, time, and risk of delivering changes by allowing for more incremental updates to applications in production. A straightforward and repeatable deployment process is important for continuous delivery. Jenkins helps automating the non-human part of the whole software development process [4], with now common things like Continuous Integration, but by further empowering teams to implement the technical part of a Continuous Delivery. Builds can be triggered by various means. It can also be triggered after the other builds in the queue have completed

II. LITERATURE SURVEY

Heavy automation of the software development project tasks defined can be significantly improve communication rate and knowledge transfer accuracy, thus increasing overall software project productivity. DevOps gives us insights as to how to optimizing software development through interactions on the system that are executed effectively and efficiently. The communication problems that occur between the developers and the testers are overcome using modern DevOps techniques. Designing effective and efficient software development process is a challenging problem. A recent movement in the software development industry attempts to bring operations and development teams into closer collaboration by relying on automated systems as the interface for project artefacts and knowledge between these teams. [6]

Continuous Delivery decouples a database Continuous Delivery decouples a database change from an application release by making changes backward and forward-compatible with each other. Defining the right thing and making it easy to adopt through compliance testing is the main motive to be achieved. By defining deployment standards for development and operations teams and making them easy to adopt, the company drastically improved the average release cycle time. [2]

Integration is a vital component of Software Development for any Software industry. Automation of these integration tasks is of at-most importance as these tasks need to be executed continuously. Jenkins provides a better solution for CI automation. Jenkins provides a better solution for CI automation. Jenkins being a powerful tool for the automation of the build process can be used for software development. [8]

III. PROPOSED SYSTEM

A. Current Scenario

In the present scenario, the developers involved in developing the application, HausKhaas, have a tough time reviewing the codes of each of the developers. It is time consuming, congested and a complicated problem. Whenever modifications are made on a code, the older versions of the code are hard to maintain. In most of the cases the older versions are deleted, ignored or lost. Due to this it becomes difficult to re-evaluate the code for the mistakes that had been made earlier.

In the build process, the files of which the application is made is used in various other files in order to execute the application. The process of which code pulls what code is done manually. This makes it a tedious task.

B. Proposed Scenario

In order to overcome all the drawbacks in the present system which was a complicated and an unprofessional way of developing applications the following ideas have been proposed

Introducing a Review Board which is a web-based collaborative code review tool, available as free software under the MIT License. Review Board integrates with Bazaar, Git, Mercurial, Perforce, and Subversion. Review Board can be installed on any server running Apache and is free for both commercial and personal use. There is also an official commercial Review Board hosting service, RBCommons. Review requests can be posted automatically or manually using either a Python script or a REST Web API.

Automating the build process using build automation utilities. Build automation utilities allow the automation of simple and repeatable tasks. When using the tool, it will calculate how to reach the goal by executing tasks in the correct, specific order and running each task. The build tools differ two ways, these are task orient vs. product-oriented. Task oriented tools describe the dependency of networks in terms of a specific set task and product-oriented tools describe things in terms of the products they generate.

C. DevOps

DevOps is a culture or practice that emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals while automating the process of software delivery and infrastructure changes

D. Tools

- Automated Review System:

It detects the new code and automatically assigns it to be reviewed by a senior developer. With automatic management of the code reviews, we ensure that no piece of code fails to have multiple developers review it, before it ends up in the final version of the software product.

- Apache Subversion(SVN):

Apache Subversion is a software versioning and revision control system. It is used to maintain current and historical version of files such as source code, web pages, documentations, etc.

- Jenkins:

Jenkins, originally called Hudson, is an open source Continuous Integration tool written in Java. Jenkins helps developers to automate Continuous Integration process as much as possible up to the point of Deployment.[4]

IV. SYSTEM ARCITECTURE

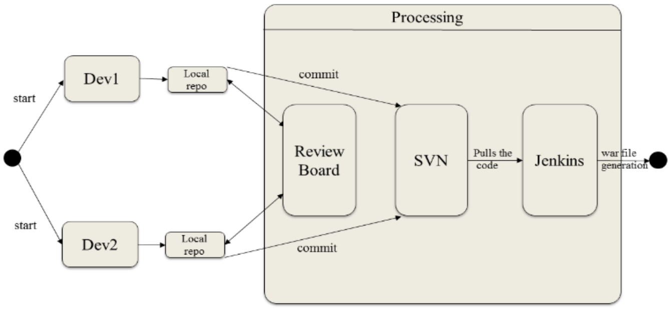

Figure 4.1 Detailed Design

Developers will be given the access to the review tool and a repository where they can store their code which has to be reviewed before the build process. With automatic management of code review, we ensure that no piece of code fails to have multiple developers review it before finalizing it. After review process, the code is then stored and maintained in Apache Subversion (SVN) which is then pulled by Jenkins for automated build process. This whole process is based on continuous Integration which focuses on integration of procedures which is reproducible at the very least, and in fact largely automated.

VI. CONCLUSION

HausKhaas is an online food ordering site developed by Calsoft Pvt. Ltd. There is always scope for the improvement of the application in various ways. In order to implement these changes there is a need to introduce various modifications in the code. Due to these continuous modifications, there is a need to check for its correctness and only if it is correct the code must be committed.

Integration is a vital component of Software Development for any Software industry. Automation of these integration tasks is of at-most importance as these tasks need to be executed continuously. Jenkins provides a better solution for Continuous Integration automation. We have presented the implementation of automation script for build process using Jenkins. The challenges faced during automation of this task have been discussed in detail. Process work-flow for automation for build process has been discussed. These automation scripts can be optimized, allowing to merge different patch branches using git or SVN and test and then push to remote repository for deployment using Jenkins.

ACKNOWLEDGMENT

We take this opportunity to thank Mr. A. J. Patankar, the Head of the Department (Information Technology) who was also our project guide for his valuable guidance and for providing all the necessary facilities, which were indispensable in the completion of this project report. We are also thankful to our project mentor Mr. Saptak Takalkar for his valuable and priceless support. We are thankful to all the staff members of the Department of Information Technology of D. Y. Patil College of Engineering, Akurdi for their valuable time, support, comments, suggestions and persuasion. We would also like to thank the Institute for providing the required facilities, internet access and important books.

REFERENCES

[1] J. H. a. A. Cockburn, "Agile software development: the business of innovation," IEEE Computer, vol. 34, no. 9, pp. 120, 127, Sep 2001.

[2] M. C. a. A. Spillane, "DevOps Making It Easy to Do the Right Thing," in IEEE, May/June 2016.

[3] M. R. a. D. F. A. Schaefer, in Continuous Integration and Automation for DevOps, Springer Netherlands, pp. 345 - 358.

[4] J. F. Smart, Jenkins The Definitive Guide, Sebastopol, CA: O'Reilly Media, 2011.

[5] C. Haight, "DevOps: Born in the Cloud and Coming to the," in Gartner Research, 2010.

[6] J. Y. A. C. Constantine Aaron Cois, "Modern DevOps: Optimizing Software Development Through Effective System Interactions".

[7] M. S. D. M. a. A. G. P. Rai, "A Prologue of JENKINS with Comparative Scrutiny of Various Software Integration Tools," in 2nd International Conference on Computing for Sustainable Global Development (INDIA Com), 2015.

[8] R. K. Nikita Seth, "ACI (Automated Continuous Integration) using Jenkins: Key for Successful Embedded Software Development," in Proceedings of 2015 RAECS UIET Panjab University Chandigarh, 2015.

[9] E. R. E.M.A. Rauf, "Software Test Automation: An Algorithm for Solving System Management Automation

Problems," in Proceedings of the International Conference on Information and Communication Technologies ICICT, 2014.