Developers Simplicity by Automating Code

Parth Patel
Dept. of Information Technology
MVP'S KBTCOE
Nashik, India
patelparth@kbtcoe.org

Bharat Sharma
Dept. of Information Technology
MVP'S KBTCOE
Nashik, India
sharmabharat@kbtcoe.org

Nupur Rathod
Dept. of Information Technology
MVP'S KBTCOE
Nashik, India
rathodnupur@kbtcoe.org

Swarali Tannu
Dept. of Information Technology
MVP'S KBTCOE
Nashik, India
tannuswarali@kbtcoe.org

***Abstract*—DevOps is a collection of patterns derived from agile approaches that aim to increase communication between development and operations teams. The major goal of this paper is to explore how DevOps practices have influenced software quality. The other goal is to figure out how to improve quality in an effective manner. The objective of the project is to build a Devops pipeline in the business and on college campuses that will automatically balance load and manage several servers at the same time. Developer's task in this system is to push code to GitHub, after which automation will begin. The system maintains and saves data in Kubernetes-managed Docker containers. We'll need to provide an API to manage database operations. The system will have its own features that would alert developers if there are any flaws. The developer will then go over the code again, looking for flaws and errors. Until then, users should not experience any downtime or delays when using our WebApp, thus it will be automatically scaled to the previous WebApp image and the previous website will be launched without delay. Our WebApp will be launched on AWS servers, with several Docker images running on those servers, all of which will be managed by Kubernetes-Orchestration.**

Keywords: DevOps, Cloud Computing, Cloud Development, Cloud Testing, Agile

I. INTRODUCTION

Over the last two decades, the software has evolved into a critical component of the business. Researchers in software engineering introduce a new generation of languages, software architectures, development tools, and technologies such as Cloud Computing, Crowdsourcing, APIs and Rest services, Big Data, and the Internet of Things.

In the IT sector, software development process models follow a common framework for structuring, planning, and controlling the process of developing an information system in an effective and productive manner. A variety of software development life cycle models have been established and designed to be used in the software development process. Each process model follows a set of stages that are specific to its type, assuring software development success. The Waterfall Model, Iterative Model, Spiral Model, V-Model, and Big-Bang Model are all traditional software development life cycle models. Agile was introduced to the business in 2001, and organizations began to employ agile ideas like Scrum and Kanban. Industry developed DevOps a few years ago, which was an upgraded version of Agile that focused on operational concerns.

DevOps is a set of methodologies for communicating and collaborating between developers and operations to provide software and services more quickly, reliably, and with greater quality. The term 'DevOps' was coined by combining the words 'Dev' and 'Ops.' DevOps is the division of tasks and responsibilities among members of a team who are fully responsible for their service and its underlying technology stack, from development to deployment and support.

Cross-functionality, shared responsibilities, and trust is encouraged in a DevOps context. DevOps fundamentally extends the agile movement's continuous development aims to continuous integration and release. DevOps emphasizes the automation of change, configuration, and release procedures in order to handle continuous releases.

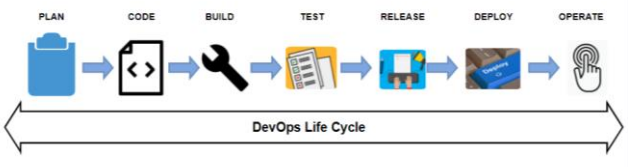

Fig 1

The goal of this study is to see if software quality improves as a result of software companies implementing DevOps. This investigation has revealed the variables that must be considered to increase software quality and how firms might implement DevOps.

Customers of Electric Bike can use the WebApp and Android App to track their bike's status, such as battery percentage, battery temperature, maintenance due date, and any faults.

Through the AWS RDS Database, which is receiving information from the Raspberry Pi that we are using for the Bike's Digital Dashboard, there will be Real-Time Updates on the WebApp and Android App.

II. LITERATURE REVIEW

The conceptual framework was created by the researchers based on the important findings of the literature review and the main goal of this study. The conceptual framework is separated into two portions, one of which contains the essential factors for DevOps implementation and the influence on software quality, and the other of which contains the impact on software quality.

The major goal of this study is to figure out how to improve software quality in a DevOps context. After evaluating variables such as Culture, Automation, Measurement, and Sharing, qualitative questions and interviews with DevOps and Quality Assurance specialists were done to find how to improve the quality. Customer pleasure hinges on product quality. To improve quality, companies should use a DevOps-aligned approach. According to the conclusions of the study, automation is the most important success element in improving quality. Before beginning automation, the software team must first determine the Return on Investment (ROI). Because some businesses automate without thinking, which may not be sustainable in the long term.

Companies must find teams with automation or technological expertise and train them to improve their automation capabilities. When there is a shortage of automation resources, management should hire individuals with the necessary expertise as a first step in implementing DevOps.

Test driven development (TDD), behavior driven development (BDD), and acceptance test driven development (ATDD) are best practices to follow when practicing DevOps. TDD is a development practice that starts with writing tests before you write any code. The flow with TDD is that one starts with the desired outcome written as a test. BDD encourages working with the business stakeholder to describe the desired business functionality of the application and expresses the test in a DSL that is pretty close to the natural language. ATDD builds on TDD and BDD, and it is involved in finding scenarios from the end user perspective. Quality engineer has to write automated tests with examples of these use cases, and then run them repeatedly while the code is being developed.

It's crucial to set up a good environment for automating scenarios before you start automating them. Jenkins for Continuous Integration, Cucumber for BDD, Junit for TDD, GIT for configuration management, Quality Center, JIRA, ALM for Test lifecycle and defect management, Selenium, QTP, and UFT for automation are some of the tools to employ when practicing DevOps.

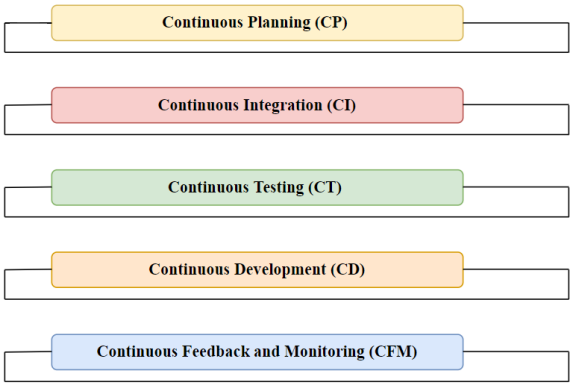

Fig 2

III. METHODOLOGY

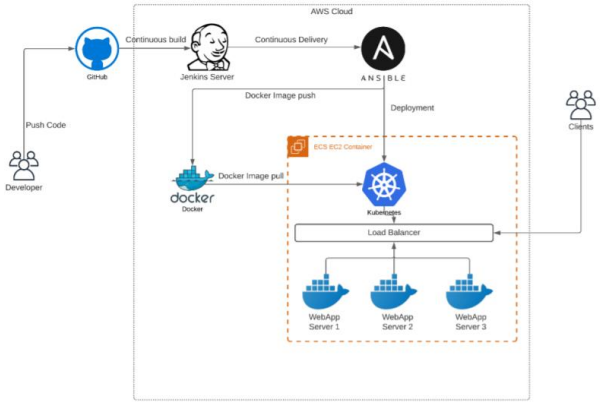

Fig 3

The appropriate methodology for implementing DevOps is defined by following procedures outlined in Figure 3.

To begin, the developer will write the code and upload it to GitHub, a code hosting platform that allows for version control and collaboration. It will allow users to collaborate on projects from any location. GitHub Actions allows integrating continuous integration (CI) into your repositories easier than ever. Jenkins, a continuous integration/continuous delivery and deployment (CI/CD) automation software DevOps tool, will then be triggered by github's webhooks, which allow you to build or set up integrations.After that, webhooks will invoke Jenkins, which will check if there are

any errors in code. If the code contains any errors, Jenkins will send an email describing the error and will continue to use the previous code. However, if there are no errors in the code, Jenkins will push it to all servers using Dockers.

Docker allows you to separate your applications from your infrastructure, allowing you to swiftly release software. Docker is a containerization platform. It allows developers to package applications into containers, which are standardized executable components that combine application source code with the OS libraries and dependencies needed to run the code in any environment.The developer will design a Docker image that will behave as a server with its own operating system, dependencies, and application enclosed within a container.After designing, the container will be launched within a second. There would be no downtime and no customer churn for large enterprises who have large traffic. The fact that these containers must be controlled manually is a disadvantage. To handle this problem we use Kubernetes which is a container management system. Kubernetes itself handles the load balancing and also launches a new container by itself when necessary . It creates clusters to ensure absolute zero downtime and remove all manual errors. Pods are Kubernetes' smallest and most basic deployable objects. In your cluster, a Pod represents a single instance of a running process. One or more containers, such as Docker containers, are contained in pods. When many containers are running in a Pod, they are handled as a single entity and share the Pod's resources. For example, if a server can hold a thousand participants but suddenly has a load of 1200, kubernetes will launch another server with all of the same properties as the previous one, and once the load drops from 1200 to 1000, kubernetes will shut down the newly created container and move all of the participants to the previous containers. Finally the website will be hosted without any downtime.

IV. CONCLUSION

To adopt DevOps methodologies for pilot projects with Cloud Computing proves to be a great success in terms of the following ways:

With integrated operations and development teams, applications may be produced and delivered quickly. The impact of application modifications is easily grasped by DevOps engineers. Software fixers are, on the whole, speedier. With cloud projects, DevOps continues to stress overall success by constructing a bridge to make work more efficient. The major fundamental features of DevOps best practises are uniform production and automated tools. DevOps also provides cloud-scale infrastructure while speeding up testing and deployment procedures.DevOps approach promotes more frequent code releases as both development and operation cycles are associated with this approach in cloud platforms. Deployment of DevOps are generally isolated and targeted, so bugs are easier to spot and in turn easier to fix and implement further. DevOps with cloud computing projects focuses on speed-to-delivery of application development so that business needs meet faster with lowering the cost of development, operations, testing and deployment.

V. REFERENCES

[1] Susarla A, Barua A, Whinston AB. A transaction cost perspective of the" software as a service" business model. Journal of Management Information Systems 2009; 26(2):205–240.
[2] Fox A, Patterson D. Engineering Software as a Service: An Agile Approach Using Cloud Computing. Strawberry Canyon LLC, 2014.
[3] Dyck A, Penners R, Lichter H. Towards definitions for release engineering and devops. Proceedings of the IEEE/ACM 3rd International Workshop on Release Engineering. 2015, doi:10.1109/RELENG.2015.10.
[4] Debois. Opening Statement. Cutter IT Journal 2011; 24(8):3–5.
[5] Bass L, Weber I, Zhu L. DevOps: A Software Architect's Perspective. Addison-Wesley Professional, 2015.
[6] Rowe M, Marshall P. The business case for collaborative devops. https://www.ibm.com/ developerworks/mydeveloperworks/blogs/invisiblethread/entry/the_business_ case_for_collaborative_devops Jul 2012.
[7] Loukides M. What is DevOps? O'Reilly Media: Sebastopol, CA, 2012.
[8] Allspaw J. 10+ Deploys Per Day: Dev and Ops Cooperation at Flickr. Retrieved 21st of February, 2016. http://www.slideshare.net/jallspaw/10-deploys- per-day-dev-and-ops- cooperation-at- flickr/ 2009.
[9] Rembetsy M, McDonnell P. Continuously deploying culture: Scaling culture at etsy. Retrieved 11th of March, 2016. http://www.slideshare.net/mcdonnps/continuously-deploying- culture- scaling-culture- at-etsy-14588485 2012.
[10] Erich F, Amrit C, Daneva M. Cooperation between information system development and operations: A literature review. Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, 2014; 69, doi:10.1145/2652524.2652598.
[11] Erich F, Amrit C, Daneva M. A mapping study on cooperation between information system development and operations. Product-Focused Software Process Improvement,Lecture Notes in Computer Science, vol. 8892, Jedlitschka A, Kuvaja P, Kuhrmann M, Mnnist T, Mnch J, Raatikainen M (eds.). Springer International Publishing, 2014; 277–280, doi:10.1007/978-3- 319-13835-0\21.
[12] Erich F, Amrit C, Daneva M. Report: Devops literature review. http://www.utwente.nl/mb/iebis/ staff/amrit/devopsreport.pdf Oct 2014.
[13] Cruzes DS, Dyba T. Recommended steps for thematic synthesis in software engineering. Proceedings of the 5th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, 2011; 275–284.
[14] Petersen K, Feldt R, Mujtaba S, Mattsson M. Systematic Mapping Studies in Software Engineering. Proceedings

of the 12th International Conference on Evaluation and Assessment in Software Engineering, 2008; 68–77.
[15] Charmaz K. Constructing Grounded Theory (Introducing Qualitative Methods series), Second Edition. SAGE
Publications, 2014.
[16] Merton RK, Kendall PL. The Focused Interview. American Journal of Sociology 1946; 51(6):541–557.
[17] Kvale S. Doing Interviews. SAGE Publications, Inc, 2008.
[18] Yin RK. Case Study Research: Design and Methods (Applied Social Research Methods). Fourth edition. edn., Sage
Publications, 2008.
[19] Glaser BG, Strauss AL. The Discovery of Grounded Theory: Strategies for Qualitative Research. Aldine de Gruyter:
New York, NY, 1967.
[20] XMind. Xmind mind mapping software. http://www.xmind.net/.
[21] Bang S, Chung S, Choh Y, Dupuis M. A grounded theory analysis of modern web applications: Knowledge, skills,
and abilities for DevOps. Proceedings of the 2nd Annual Conference on Research in Information Technology,
Orlando, FL, 2013; 61–62, doi:10.1145/2512209.2512229.
[22] Hosono S, Shimomura Y. Application lifecycle kit for mass customization on PaaS platforms. Proceedings - 2012
IEEE 8th World Congress on Services 2012, Honolulu, HI, 2012; 397–398, doi:10.1109/SERVICES.2012.80