



A LITERATURE SURVEY ON DEVELOPMENT AND OPERATION

¹NIYA N J, ²JASMINE JOSE

¹MSc.SCHOLAR, ²ASSITANT PROFESSOR

¹DEPARTMENT OF COMPUTER SCIENCE,

²ST.JOSEPH COLLEGE(AUTONOMOUS) IRINJALAKUDA, KERALA, INDIA

Abstract: DevOps, originated in the context of agile software development. It is the suitable approach to enable the CI/CD of working software in small organization. Organizations are taking significant interest in adopting DevOps ways of working. It is based on the agile development movements and supports rapid development and deployment cycles. DevOps is the most buzzing term around the IT industry.

Index Terms - Continuous integration, Continuous delivery, devops tool.

I. INTRODUCTION

DevOps stands for the Development and Operation collaboration. Agile was instrumental in Development regaining the trust in the business, but it unintentionally left IT Operations behind. DevOps is a way for the business to regain trust in the entire IT organization as a whole. It is a set of practices that automates the process between software development and it teams in order that they t estand release software faster and more reliably. It is a new software development approach.

II. WORKING

DevOps is a culture or framework among development operation. It is a union of process, people and working product that enable continuous integration and continuous delivery (CI/CD) to the enduser. DevOps is a software development method that combines the quality assurance with the operation and development. The goal of DevOps is to change and improve the relationship by advocating better communication and collaboration between these two teams. DevOps helps to increases an organizations speed to deliver applications and services. It allows organizations to serve their customers better and compete more step market. DevOps is used because of its predictability costefficiency, maintainability, reproducibility, greater quality etc.



The main stages of DevOps works in the software industry are:

Continuous Delivery:- Continuous delivering means development and practices daily and apply build software development in automated test.

Continuous Testing:- Continuous Testing means when it can run the application or software and can enable or not any risk in its future and also can analyse the all properties in that project.

Continuous Integration:- Continuous Integration Of DevOps is the development and developers can merge the code then changes into the central storage area that which automated tested and run.

Continuous Monitoring:- Continuous monitoring is the carry out used for defining risks and weaknesses in the application. Increase DevOps agility with improved database monitoring. It is an powerful approach to increase both the speed and quality of software deployment.

Continuous Feedback & Optimization:- Many examples of applications are related to the optimization of parameters of mathematical models represents systems of nonlinear differential equations.

DevOps is a framework which applies agility and lean principles across entire software delivery. It's a kind of building culture to have high degree of collaboration among all the departments. In devops models the QA and security teams become more tightly integrated with development and operations team throughout the application life cycle. When security is the focus of everyone on a devops team, then it is sometime called as the DevSecOps.

III. METHODS

CONTINUOUS INTEGRATION AND CONTINUOUS DELIVERY(CI/CD)

Continuous integration(CI) is a software development practice where developers regularly merge their code changes into a central storage after which automated builds and tests are run. CI most often refers to the integrative software release process and requires both an automation component and a cultural component. The key goal of these is to find and address bugs more quickly, improve software quality, and reduce the time.

Continuous integration focuses on smaller commits and smaller code changes to integrate. A developer commits code at regular intervals, at minimum once a day. The developer pulls code from the code repository to ensure the code on the local host is merged before pushing to the build. At this stage the build served runs the various tests and either accept or rejects the code commit.

Continuous delivery(CD) is a software development practice where code changes are automatically built, tested, and prepared for production release. It expands on continuous integration by deploying all code changes to a testing environment, a production environment, or both after the build stage has been completed. Continuous delivery can be fully automated with a workflow processor partially automated with manual steps at critical points. When continuous delivery is properly implemented, developers always have a deployment or easy build artifact that has passed through a standardized test process.

IV. CI/CD BENEFITS AND CHALLENGES

4.1 CI/CD BENEFITS

Faster identification and resolution of defects:- CI/CD allows way to establish the appropriate quality gates in the development and test. A fast feedback loop to the developers ensures that are addressed early in the development cycle.

Reduced assumptions:- CI/CD replaces testing assumptions with knowledge. So it eliminate all cross-platform errors at the development stage.

Better quality assurance:- CI/CD enables QA teams to release deployable software. Without it the projects are prone to delayed releases because of unforeseen issues which arise at any point in the traditional development and test process.

Software health measurability:- By establishing continuous testing into automated integration process, software health attributes such as complex it can be tracked overtime.

Reduced overhead cost:- Finding a bug at the development stage is the cheapest possible way. If the same bug was to be fixed in any other environment, it would cost more. CI/CD requires some upfront overhead cost.

Faster time to market:- Faster test and QA cycles enable organizations to get quality products and services to market faster and more efficiently.

4.2CI/CD CHALLENGES

Varied Application Development Stacks:-Need to built a platform that creates a CD pipeline foreach application.No single vendor provides a comprehensively suite of

tools:-Continuous delivery(CD)means stitching together a tool chain spanning multiple vendor and open source tools

Minimal automation in order to meet throughput and consistency goals.It usually required to automate a bulk of the manual steps in the process

Limited Environments:-Plan to be running delivery pipelines at scale,a dynamic pool of available,“clean” target environments is required

Lack of visibility into SDLC Pipeline:-Key decision makers and cross functional teams lack unified visibility in an enterprise SDLC dashboard

Organization Inertia:-Align the organization to understand, adopt, engage and leverage CD and management willingness to“do things right”.

Continuous Integration(CI) and Continuous Delivery(CD)are foundational practices being adopted by agile organizations inorder to accelerate their development and test processes to meet the new demands of mobile and IoT.While they are distinct disciplines,most organizations find the greatest impact by adopting them together. The benefits of this approach include reduced time,risk,and expense of software delivery.continuous integration and delivery (CI/CD)is a set of software practices and technique that enable the frequent release of small at code changes,with extensive visibility and traceability.It typically involves the creation of a largely pipeline.And it ultimately leading to deployment in production.

V.DEVOPS TOOLS

1. *Gradle:-Gradle allows faster shipping and with configuration possibilities*
2. *Git:-Git allows you to track the progress of the development work.*
3. *Jenkins:- Can iterate and deploy new code as quickly.It also allows to measure the success of each step of pipeline.*
4. *Bamboo:-Bamboo can save a lot of configuration time.I also have a more intuitive UI with tool tips,auto-completion,and other handy features.*
5. *Docker:-Docker integrates with Jenkins and bamboo.If it usetogether with one of these automations servers,you can further improve your delivery workflow.Docker is also great for cloud computing,technology ecosystem and nonstandard tracking metrics.*

VI.CONCLUSION

DevOps helps the development team to launch their products into the market more fastly.It reduces the conflict between the two teams ie, development and operation team and improve the development efficiency and quality.Thegoalsof successful DevOps are increased automation,standerdized processes collaborated teams continuous improvement reduced cost and effort and time to market.The key challenges that come in the successful DevOps are inconsistent release management processes lack of team collaboration.

VII.ACKNOWLEDGMENT

I would like to take this opportunity to acknowledge the contribution of certain people without which it would not have been possible to complete this paper work. I am thankful to the Principal Dr. Sr.Isabel, Guide and Coordinators for their support, encouragement and suggestions. I would like to express my special appreciation and thanks to my guide Mrs.Jasmine Jose, you have been a tremendous mentor for me.

REFERENCES

- [1] LEONARDO LEITE, CARLA ROCHA, FABI OKON, DEJAN MILOJICIC, PAULO MEIRELLES. In "A Survey of DevOps Concepts and Challenges"
- [2] https://www.researchgate.net/publication/320161713_Is_DevOps_another_Project_Management_Methodology
- [3] https://www.researchgate.net/publication/267330992_Report_DevOps_Literature_Review
- [4] <http://www.scrummaster.dk/lib/AgileLeanLibrary/Topics/DevOps/Introducing-DevOps-to-the-Traditional-Enterprise-final.pdf>
- [5] <https://raygun.com/blog/best-devops-tools/>
- [6] *Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices* Mojtaba Shahina, Muhammad Ali Babara, Liming Zhuba CREST-The Centre for Research on Engineering Software Technologies, The University of Adelaide, Australia
Data61, Commonwealth Scientific and Industrial Research Organisation, Sydney, NSW 2015, Australia
mojtaba.shahin@adelaide.edu.au, ali.babar@adelaide.edu.au, liming.zhu@data61.csiro.au
- [7] "Practicing Continuous Integration and Continuous Delivery on AWS Accelerating Software Delivery with DevOps" by Asif Khan

