Automation of Software Development using DevOps and its Benefits

Sushma T N
Computer Science and Engineering
R V College of Engineering
Bangalore, India

Dr. Ashok Kumar A R
Computer Science and Engineering
RV college of Engineering
Bangalore, India

Abstract— **DevOps is a software development method that collaborates both development and operations teams to increase the quality of software delivered.It helps development teams with continuous planning,continuous integration continuous deployment and continuous feedback.for example In IT companies each one in the team works on a single feature and then combines their work to make it a complete application. After the development of an individual feature, it becomes a very tedious job to integrate all of their code and it takes much time. So, DevOps comes with an idea of integrating team code after at regular intervals thus making final application development easier and faster.Like above example continuous deployment,planning and feedback also achieved.This paper explains the various tools available in the market to make software development easier, DevOps architecture, and its benefits.**

Keywords— DevOps; CI CD; DevOps Cycle;

I. INTRODUCTION

In software development, development team members write code for a new feature and check the quality of code. The operations team receives the application files from the development team and they deploy it on the production environment. In the traditional method of developing software, the development and operations team worked separately. this caused some major issues such as

1. The development team had to wait to get the feedback from the operation team
2. The application doesn't get deployed successfully if some packages are missing in the production environment, as many software components get tightly packaged.

DevOps methodology states that the development team should know how the application deployment is done and the Operation should have the basic knowledge of how software has been developed about language, required packages, etc. So, when these two teams work together the software release happens at a faster rate. Then the release management process starts to validate, monitor, verify in releasing and deploying on all the stages of the software.

II. DEVOPS CYCLE

The development and Operations teams each have four phases. Development phases are plan, code, build, and test. The Operations phases are Release, Deploy, Operate, and Monitor.
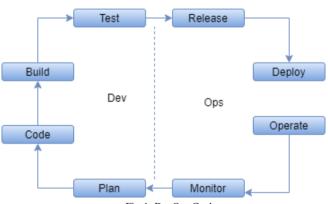

Fig-1: DevOps Cycle

The Cycle Includes Plan, Code, Build, Test, Release, Deploy, Operate, and Monitor that are shown in the Fig-1.

The Cycle includes Plan, Code, Build, Test, Release, Deploy, Operate, and Monitor. In this cycle of the process, the development team of the company splits its team for various operations on the working of each software component to cover all the stages. Whenever there is a need for improvement in any component the team plans a new feature for the implementation and once the plan is accepted, the process moves to the next phase the development team code the requirements that are defined in the planning phase. After the successful completion of the coding team will build the same to move into the Test phase in the release and deploying management.

The managers or coordinators of component release is responsible for scheduling, monitoring, validating the release for checking the quality of the new features that are added to the component and approve it to the deployment phase. Then once it passes all the tests this will be binding back with the main package.

III. DEVOPS OPERATION

This section explains the major DevOps Operations which needs to be followed for boosting the quality of software development. This section also explains the few tools available in the market to make that operation easier.

Fig-2 shows Collaborative Development, Continuous Testing, Continuous Release, Continuous Customer Feedback, Continuous Monitoring, and Continuous Development that are the DevOps operations.
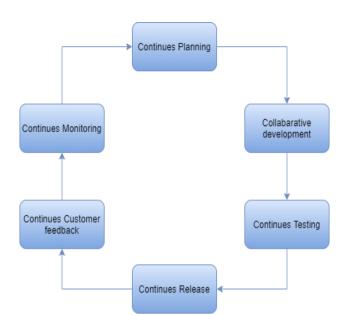
Fig-2: DevOps Operations

The Collaborative Development a phase where stakeholders from a software development project can work together to discuss, document, and produce project deliverables. Continuous Testing is a way to reduce the waiting period for developers to provide feedback by incorporating environmentally induced tests and more conventional developer tests. In addition to these, Continuous Release, Continuous Customer Feedback, Continuous Monitoring, and Continuous Development also help in the proper execution of Development that uses the reliable and most recently available resources available in the industries.

The following section will give an explanation of a tool currently in the market that can be used for DevOps operations. By using these tools one can automate the software development.

JIRA: Jira is a project management tool developed and maintained by Atlassian. Using Jira the issues, bugs etc can be tracked easily. Team members have to individually log details of the issue,bug they are working on and by when it gets completed. It helps the team manager to know hat exactly each one is doing,it kind of gives transparency.

GitLab: It is a distributed version control system used for storing the source code developed by the team. Repository is the place where code is saved to related to a project.The code developed by each team member is stored in different branches and they are merged after integration testing is successful.GitLab uses SHA Algorithm for the secured data transfer.

Jenkins: Jenkins is a continuous integration and continuous deployment tool. It takes the code from the git repository and does integration testing after which the code will be merged. Jenkins acts as the main tool which can be integrated with gitlab,sonarqube,Jfrog and UCD for the automation.

SonarQube: SonarQube is a tool developed by SonarSource for the continuous quality of code.SonarQube does only static code analysis that means it reports duplicated code, bugs,unused variables by comparing with the standards of programming language

JFrog: It is basically a tool for storing and managing the executable files like .war, .jar.From JFrog artifactory executable files are sent for deployment.

IBM Urbancode deploy: Tool developed and maintained by IBM to automate applications deployment on development, test and production environments. Urban code does proper versioning of the deployments done. Urban code can be integrated with Jenkins by installing the plugin.

IV. BENEFITS

The following are the benefits of using DevOps method in software development. They are

- Increased Software Delivery: That is which ensures the validation of software components in release management and deploying management.
- Integration of developed code in regular intervals: This is one of the process in deploying management that validates each component and makes them tightly packages for each new update as the version in the software.
- Increased quality of software: This happens because all the changes happening in the software component is well known and monitored with the help of this system.
- More responsibility on the team: As code developed by each member needs to be integrated regularly and it needs to go through phases like integration test, static code analysis, user acceptance testing etc.In case of any failure the whole team will be responsible.
- It aligns IT and Business: It increases the speed of the industry or company towards the short-term and long-term goals.

V. CONCLUSION

As explained above in this paper how DevOps principles can really optimize and benefit the organization's software delivery capability if adopted in the software development life cycle.DevOps not only involves change to processes but also change the method of software development. Organizations adopting DevOps principles will surely have an edge over all those organizations that do not run this DevOps wave.DevOps allows the organization to reduce market time, Adapt to continuous feedback Effectively balance costs and quality Have more predictability in releases Increase efficiency of the organization as a whole.

DevOps not only helps the organization by increasing the development speed but also lets clients know how much of development has been completed and it's quality. In the market, there are many tools available for DevOps automation but choosing the one with many features and which provides continuous support is very important.

REFERENCES

[1] DevOps for Dummies – by Sanjeev Sharma, https://www14.software.ibm.com/webapp/ iwm/web/signup.do?source=swg-rtl-sdwp&S_PKG=ov18162

[2] Adopting the IBM DevOps approach for continuous software delivery, http://www.ibm.com/developerworks/libr ary/d-adoption-paths/

[3] Understanding DevOps – Infrastructure as a Code, https://sdarchitect.wordpress.com/2012/12 /13/infrastructure-as-code/

[4] Understanding DevOps & Bridging the gap from Continuous Integration to Continuous Delivery mvirmani@in.ibm.com Fifth international conference on Innovative Computing Technology (INTECH 2015)

[5] Pulasthi Perera, Roshali Silva, Indika Perera " Improve Software Quality through Practicing DevOps" 2017 International Conference on Advances in ICT for Emerging Regions (ICTer): 013 - 018 978-1-5386-2444-9/17/$31.00 © 2017 IEEE

[6] M. Huttermann, "Beginning DevOps for Developers," in DevOps for Developers, 2012, pp. 3-13.

[7] 2019 IEEE 41st international conference on software engineering: Companion Proceedings(ICSE-Companion) "Improving the software logging practices in DevOps Boyuan Chen York University, Toronto, Canada chenfsd@eecs.yorku.ca

[8] L. Bass, I. Weber, and L. Zhu, DevOps: A Software Architect's Perspective, 1st ed. Addison-Wesley Professional, 2015.