

Enhancing software automation using DevOps

¹Gokul Patil, ²Nisha Magar, ³Vaishnavi Gangurde, ⁴Zarka Khan, ⁵Srushti Magar

^{1, 2, 3, 4, 5}Department of Computer Engineering

^{1, 2, 3, 4, 5}SITRC, Sandip Foundation, Nashik, India

Email ID: ¹gokul.patil@sitrc.org, ²nishamagar.n@gmail.com, ³vaishnavi616@gmail.com, ⁴zarkanashin786@gmail.com, ⁵srushtimagar1098@gmail.com

Abstract - DevOps is a software practice that aims at unifying development, operations and quality assurance process. It strongly advocates automation and providing the cloud security. DevOps that stands for development and operations along with quality assurance is a technology which is used for the automation purpose. The sole purpose of this technology is to provide automation throughout the entire software development process right from the planning phase, along with construction, integration, testing and also the release of the product to the customer. DevOps provide with different tools such as Git, Jenkins, Chef, Selenium, etc. which help us throughout this automation process. The platform used for this technology is Visual Studio Team Services (VSTS) and the Microsoft Cloud Azure. It includes the Continuous integration, continuous delivery, Continuous Testing and Release Management processes. Due to the strong automation, it saves a lot of time and manual effort which serves as a great benefit to a lot of organizations. All the stakeholders associated with the project can keep a track of all the activities going on throughout the development process. It bridges the gap between Continuous Integration and Continuous Delivery.

Keywords - TFS, VSTS, IIS, Release Management, Continuous Integration, Visual Studio, Triggers, Build and Release, Manage Connections.

1. INTRODUCTION

DevOps as a term came across 2-3 years ago as a moment, as a philosophy to help the development team and operations to work better together, to have better collaboration and communication and the goal was to build trust. DevOps is a software practice that aims at unifying development, operations and quality assurance process. It strongly advocates automation. DevOps is basically the combination of development and quality assurance (QA). In DevOps idea is at the design stage also we have requirement enhancement, request and bug fix. After deploying the software product to the customer, the customer will give feedback to the deployment team and according to that feedback, the deployment team will make changes in the product. The team can improve two things the product and the deployment process itself. The process is usually improved in two ways i.e. reduce the amount of rework that needs to be done for the specific project and reduce the amount of overhead in the process in general.

Abbreviations and Acronyms

VSTS:-Visual Studio Team Services (VSTS) is an extension of the Microsoft **Visual Studio** architecture that allows it to encompass development teams, with special roles and tools for software architects, developer specialties and testers. It is a collection of hosted DevOps services for application developers.

TFS:-Team Foundation Server (TFS) is a Microsoft product that provides source code management either with Team Foundation Version Control or Git, reporting, requirements management, project management automated builds and lab management. It is the on premises version of VSTS that you can install and manage on your server.

IIS:- Internet Information Services is an extensible web server created by Microsoft to serve requested HTML pages or files. It hosts the website and other content on the web.

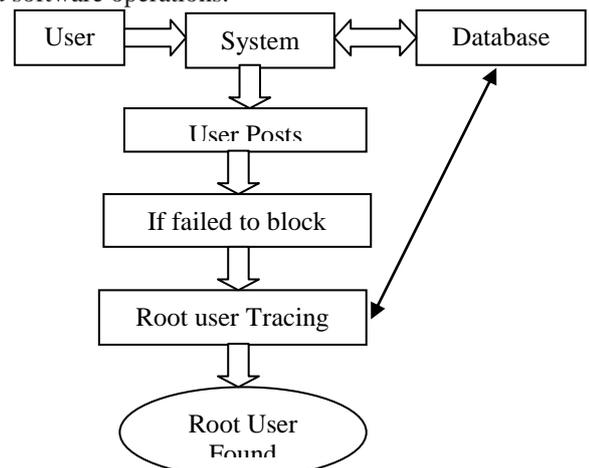
Release Management: - **Release Management** is the **process** responsible for planning, scheduling, and controlling the build, in addition to testing and deploying releases it is used to develop and deploy applications regularly to any platform.

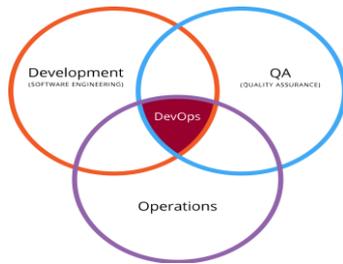
Continuous integration:- Continuous Integration (CI) is a development practice that requires developers to **integrate** code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.

Builds and Releases - A **“build”** is a developed application for the customers that are given by development team to the software testing team. A **“release”** is an official launch of the application for the customers. A **build** when tested and certified by the software testing team is provided to the customers as **“release”**.

Description:

- DevOps is a software engineering culture and practice that aims at unifying software development and software operations.





- DevOps as the name itself suggests, is an integration of Development and Operations, along with quality assurance. It is a software practice which is used for the purpose of automation.
- The sole purpose of this technology is to strongly advocate automation throughout the entire software development process.
- The utmost goal of DevOps is to have better communication and collaboration between different business units within the organization, therefore developing a quality product.
- DevOps provide with different tools such as GIT, Jenkins, Selenium, etc. which help us throughout this automation process.
- DevOps is used to automate the entire workflow, thereby increasing the productivity of the product and delivering it at a faster pace.
- DevOps is a practice of optimizing development and operational activities through structured process, collaboration and automation.
- DevOps's central concept is to manage end to end engineering processes.
- Due to increased communication and collaboration, DevOps allows the members of the organization to deliver the product to their customers within a short time, in order to achieve total customer satisfaction.
- DevOps presents a significant broader perspective in delivering software. The perception of DevOps being a mere tool has been formulated into a change in culture, paradigms in automation, clarity in measuring risks and felicitating ease of sharing. [11]
- In simple words, DevOps can be defined as an alignment of development and IT operations with improved communication and collaboration.

Why is DevOps Needed?

- Before DevOps, the development and operation team worked in complete isolation and the communication was carried out with maximum effort which led to a bad collaboration between different departments.
- Testing and Deployment were isolated activities done after design-build. Hence they consumed more time than actual build cycles.
- Without using DevOps, team members are spending a large amount of their time in testing, deploying, and designing instead of building the project.
- Manual code deployment leads to human errors in production

- Coding & operation teams have their separate timelines and are not in synch causing further delays.

How Is Devops Different From Traditional IT

Let's compare traditional software waterfall model with DevOps to understand the changes DevOps bring.

Old Process	DevOps
After placing an order for new servers, the Development team works on testing. The Operations team works on extensive paperwork as required in enterprises to deploy the infrastructure.	After placing an order for new servers Development and Operations team work together on the paperwork to set-up the new servers. This results in better visibility of infrastructure requirement.
Projection about failover, redundancy, data center locations, and storage requirements are skewed as no inputs are available from developers who have deep knowledge of the application.	Projection about failover, redundancy, disaster recovery, data center locations, and storage requirements are pretty accurate due to the inputs from the developers.
Operations team has no clue on the progress of the Development team. Operations team develops a monitoring plan as per their understanding.	In DevOps, the Operations team is completely aware of the progress the developers are making. Operations team interact with developers and jointly develop a monitoring plan that caters to the IT and business needs. They also use advance Application Performance Monitoring (APM) Tools
Before go-live, the load testing crashes the application. The release is delayed.	Before go-live, the load testing makes the application a bit slow. The development team quickly fixes the bottlenecks.

Why Is Devops Used?

DevOps allows Agile Development Teams to implement Continuous Integration and Continuous Delivery. This helps them to launch products faster into the market. The time it has been realized that this optimization as part of continuous integration - alone - is just not helping to make the entire delivery lifecycle efficient or is not driving the organization efficiency. Unless all the pieces of a software delivery lifecycle work like a well oiled machine - efficiency of organization to optimize the delivery lifecycle can not be met. This is the problem which DevOps tries to address. [12]

Other Important reasons are:

- 1. Predictability:** DevOps offers significantly lower failure rate of new releases
- 2. Reproducibility:** Version everything so that earlier version can be restored anytime.

3. Maintainability: Effortless process of recovery in the event of a new release crashing or disabling the current system.

4. Time to market: DevOps reduces the time to market up to 50% through streamlined software delivery. This is particularly the case for digital and mobile applications.

5. Greater Quality: DevOps helps the team to provide improved quality of application development as it incorporates infrastructure issues.

6. Reduced Risk: DevOps incorporates security aspects in the software delivery lifecycle. It helps in reduction of defects across the lifecycle.

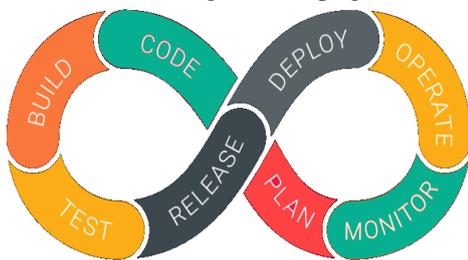
7. Resiliency: The Operational state of the software system is more stable, secure, and changes are auditable.

8. Cost Efficiency: DevOps offers cost efficiency in the software development process which is always an aspiration of IT companies' management.

9. Breaks larger code base into small pieces: DevOps is based on the agile programming method. Therefore, it allows breaking larger code bases into smaller and manageable chunks.

DevOps Lifecycle

DevOps is the combination of Development, operations and quality assurance. DevOps lifecycle is the integral part without which understanding of DevOps gets difficult.



DevOps is a set of methods in which developers and operations communicate and collaborate to deliver software and services rapidly, reliably and with higher quality. DevOps is sharing of tasks and responsibilities within a team empowered with full accountability of their service and its underlying technologies stack; from development, to deployment and support. [1]

Here is brief information about the Continuous DevOps lifecycle:

1. Development

In this DevOps stage the development of software takes place constantly. In this phase, the entire development process is separated into small development cycles. This benefits DevOps team to speed up software development and delivery process.

2. Testing

QA team use tools like Selenium to identify and fix bugs in the new piece of code.

3. Integration

In this stage, new functionality is integrated with the prevailing code, and testing takes place. Continuous development is only possible due to continuous integration and testing.

4. Deployment

In this phase, the deployment process takes place continuously. It is performed in such a manner that any changes made any time in the code, should not affect the functioning of high traffic website.

5. Monitoring

In this phase, operation team will take care of the inappropriate system behavior or bugs which are found in production.

DevOps Principles

Here, are six principles which are essential when adopting DevOps:

1. Customer-Centric Action: DevOps team must take customer-centric action for that they should constantly invest in products and services.

2. End-To-End Responsibility: The DevOps team need to provide performance support until they become end-of-life. This enhances the level of responsibility and the quality of the products engineered.

3. Continuous Improvement: DevOps culture focuses on continuous improvement to minimize waste. It continuously speeds up the improvement of product or services offered.

4. Automate Everything: Automation is a vital principle of DevOps process. This is not only for the software development but also for the entire infrastructure landscape.

5. Work As One Team: In the DevOps culture role of the designer, developer, and tester are already defined. All they needed to do is work as one team with complete collaboration.

6. Monitor And Test Everything: It is very important for DevOps team to have a robust monitoring and testing procedures.

Summary:

- Before DevOps operation and Development team working in completed isolation.
- Manual code deployment is mostly never accurate and has errors to some extent, because humans are not that precise.
- Earlier, the operational team has no idea regarding the progress of the Development team as they work in total isolation.
- After DevOps was introduced, both the teams are aware about other team's activities as they work in collaboration. The purchase, monitoring and planning is accurate.
- DevOps offers Maintainability, Predictability, Greater quality cost efficiency and time to market.
- Agile process focuses on functional and non-functional readiness while DevOps focuses on that IT infrastructure aspects.
- DevOps life cycle includes Development, Testing, Integration, Deployment, and Monitoring.
- DevOps engineer will work with development team staff to tackle the coding and scripting needs.

- DevOps helps organizations in shifting their code deployment cycles to weeks and months instead of years.

2. METHODOLOGY:

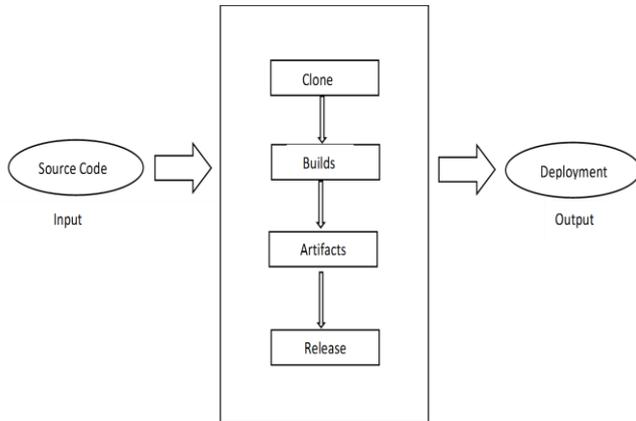


Fig. Release Management Cycle

The above figure shows the representation of entire release management cycle of DevOps. The input consists of the source code which must be coded in offline Visual Studio. This input is then cloned (copied) onto the online Visual Studio, following which the build pipeline is created. A “**build**” is a developed application for the customers that are given by development team to the software testing team. A build represents one execution of a pipeline. It collects the logs associated with running the steps and the results of running tests. “**Artifacts**” are the static files which cannot be changed. An artifact is a collection of files or packages published by a build. Artifacts are made available to subsequent tasks, such as distribution or deployment. After the creation of artifacts now the project is ready to be released and hence we get our desired output. A release is the term used to describe one execution of a release pipeline. It is made up of deployments to multiple stages. A “**Deployment**” target is a virtual machine, web app, container or any service that is used to host the application being developed. A pipeline might deploy the app to one or more deployment targets after build is completed and tests are run.

Survey:

SR. NO	Tools	Advantages	Dis-advantages
1.	GIT	Free open source tool, allows distributed development, enables faster release cycle, supports pull request.	Binary files are a big no, Steep learning curve.
2.	Jenkins	Free open source tool, provides multiple ways of communication: web based GUI, automate testing ,integrate all your DevOps stages, find and solve bugs in your code.	Unpredictable costs, Lack of governance, No collaboration features, Lack of analytics, Needs personnel
3.	CHEF	Free open source tool, supports multiple platforms like AIX, freeBSD, easy to integrate with cloud based platform, active smart and fast-growing community support.	Bringing up new instances is extremely slow, there are many moving parts and there is a high risk of failure.
4.	PUPPET	Free open source tool, Based on Master-Slave architecture, long commercial track record , platform independency.	Hard to configure. Time Consuming and need large group of people’s participation.
5.	SELENIUM	Free open source tool, Create robust browser based automation suites and tests, supports multi-platform for testing like ios and android.	Designed only to create prototypes of tests. No support for conditional and iteration operations.

ACKNOWLEDGEMENT:

Hence, DevOps is about strongly advocating automation and monitoring all the software engineering activities right from planning, requirement gathering construction, testing, and

deployment. DevOps is the integration of Development, Operations and Quality Assurance. It aims at delivering the product at a faster pace, thereby improving the quality of the product. It comes with the advantage of version control feature, where we can rollback to previous versions of the code without actually using a backup tool. The output will help companies who practice DevOps and quality engineering teams to take decision to improve testing practices.

Research findings have clearly indicated that culture, automation, measurement, and sharing have impacted on quality of the products therefore practicing DevOps will improve the software quality if they consider above facts correctly.

REFERENCES

- [1] Pulasthi Perera. Roshali Silva. Indika perera, Improve software quality through practicing DevOps, 2017 IEEE. 2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer).
- [2] P. Perera, I. Perera, M. Bandara, "Evaluating the Impact of DevOps Practice in Sri Lankan Software Development Organizations", *ICT for Emerging Regions*, pp. 281-287, 2016.
- [3] "Agile Manifesto", *Agile Manifesto*, 04 2017, Available: <http://www.agilemanifesto.org/>.
- [4] M. Huttermann, "Beginning DevOps for Developers", *DevOps for Developers*, pp. 3-13, 2012.
- [5] D. L. Farroha, B. S. Farroha, "A Framework for Managing Mission Needs Compliance and Trust in the DevOps Environment", *Military communicationsConference*, pp. 288-293, 2014.
- [6] "2016 State of DevOps", *Puppet and Dora*, 2016.
- [7] D. Farley, J. Humble, *Continuous Delivery: Reliable Software Releases through Build Test and Deployment Automation*, Pearson Education, 2010.
- [8] *diazhilterscheid*, 06 2016, [online] Available: hilterscheid.de/en/ready-for-cams-the-foundation-of-devops/.
- [9] "ISO/IEC 9126-1:2001 - Software engineering - Product quality - Part 1: Quality model", *ISO*, 03 2001, [online] Available: <https://www.iso.org/standard/22749.html>.
- [10] M Rajkumar. Anil Kumar Pole. Prabal anta. Vittalraya Shenoy Adige, DevOps culture and its impact on cloud delivery and software development, 2016 IEEE.
- [11] Manish Virmani, Understanding DevOps & bridging the gap from continuous integration to continuous delivery. 03 August 2015, IEEE.