iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** IX **Month of publication:** September 2021

DOI: https://doi.org/10.22214/ijraset.2021.38008

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue IX Sep 2021- Available at www.ijraset.com

Evaluation of the Feasibility of Parametric Estimation in Devops Continuous Planning

Dr. Diwakar Ramanuj Tripathi
Information Technology & Legal Consultant, Nagpur, Maharashtra (India)

Abstract: ***DevOps aims to shorten project schedules, boost productivity, and manage quick development-deployment cycles without compromising business or quality. It necessitates good sprint management. Continuous Testing detects integration issues considerably earlier in the development process. It reduces the cost of defect resolution and frees up the tester's time for exploratory testing and value-added activities. Continuous testing allows for more frequent, shorter, and more efficient releases. It ties people, technology, and processes together. Continuous Planning, particularly effort estimation, is closely linked to Continuous Testing. This paper examines the state of the art in DevOps parametric estimate in continuous planning, as well as the difficulties and best practises.***

Keywords: Project, Testing, Continuous, Planning.

I. INTRODUCTION

DevOps is a multidisciplinary methodology that arose from continuous development, testing, and deployment cycles. DevOps' purpose is to deliver the project on time and with great reliability. It introduces a variety of tools, methods, and ideas, as well as a shift in the project execution culture. DevOps fosters quick failures, which aids in quick recovery.

DevOps is a compound phrase that encompasses both development and operations. DevOps is a new wonder in the world of programming, automation, and virtualization, emphasising a collaborative effort and new tools to expand operations and development prospects. It focuses on increasing team communication and collaboration, as well as combining operations and team development. It is an effective approach to increase customer happiness while also improving the quality of software systems. The fundamental goal is to increase customer happiness, as it is the foundation of any programme that is spread globally. This could be accomplished by acquiring a quicker delivery of a product with the desired attributes. DevOps has the capacity to provide continuous improvement and development, thus it must provide continuous customer happiness. It assists the software industry in both a nontechnical and a technical manner in order to improve customer responsiveness by releasing software in periodic cycles and automating software releases.

Compilation of code is a crucial step in the software development process that converts source code into executables. Design patterns, refactoring of code, and other methods are used by large enterprises in Continuous Integration and Test Automation (TA) for improved quality and increased productivity. In the field of embedded systems, DevOps has yet to catch on. It is very adaptable in the web domain, as the concept of virtualization will greatly assist developers in extracting the software infrastructure. Configuration tools aid development, which is followed by testing, which is reflected in production. Configuration management software is used in conjunction with automatic verification and validation software. This will increase confidence in the first-time deployment of newer features for software that is being deployed in production. Every day, software is being deployed in the field of information technology. It is primarily dependent on the software's dependability and stability.

II. BENEFITS AND CHALLENGES OF ADOPTING DEVOPS

A. Benefits of Adopting DEVOPS

Adopting DevOps in the embedded sector, where agile development is used, has various advantages. A programme that is directly used or embedded into the hardware device is known as embedded software. It is mostly written for specific hardware due to the computing capabilities of the device, which typically includes processor and memory limits. Because software implementation in the embedded system area is more dependent on hardware or devices. The creation of embedded software may not be as quick as that of a web application domain. It offers a shorter delivery time, which is critical in this era of digitization and rapid technological progress for enhancing business and satisfying customers. The following are some of the most significant advantages:

1) When agile development is used, DevOps and Automation allow for a faster release and deployment cycle.
2) It enhances collaboration between application developers, business stakeholders, and the operations team.
3) It reduces the time it takes to implement new features or services from months to minutes.

4) It boosts business and operations teams' productivity.
5) It streamlines the distribution process by standardising the procedure for easy replication.
6) All system components' quality, reliability, and reusability are improved.
7) It increases the success percentage of digitization and transformation programmes.

B. Difficulties In Implementing DEVOPS

Another goal of this article is to identify the obstacles that will be encountered while using DevOps to develop embedded system software. When DevOps and automated CI/CD are combined in an embedded domain, the following issues may arise:

1) Because they do not follow a modular approach, there may be hardware and interface dependency, and all or some of the components may be dependent on one another. As a result, the flow between the components must be kept independent of the application and architecture, which is not always practicable.
2) Integrating new tool with current tools in the business
3) The software used should be compatible with the hardware version that is being utilised.
4) In the embedded systems domain, there is a lack of appropriate automation tools.
5) Software that needs to be cross-compiled for an embedded platform
6) In the embedded system sector, there is a lack of visibility into the production environment.
7) Inadequate resources for setting up habitats
8) Log timer culture and mindset in businesses
9) Hardware resource restrictions and scalability
10) Legal and regulatory restrictions

Adopting DevOps comes with a number of obstacles, some of which can be overcome by utilising new open-source tools such as Ansible for configuration management, deployment automation, and monitoring. C/C++ is used to write the majority of embedded software. To run binaries into targeted hardware or environments, cross compilation should be done and several tool chains should be employed. The build infrastructure is automatically generated and controlled by eclipse's C/C++ Development Tooling project (CDT), which consists of a collection of Makefile that is generated and put into software projects.

There are no tools available for deploying an application to a target environment. The essential scripts must be loaded via the network on a target platform that had previously been produced and utilised but never maintained. Manually uploading the code to the board via a USB stick is another option. The number of feedback cycles in software development rose as a result of this. Multiple engineers were required to work on the totally manual testing process over the course of a week, and builds were performed infrequently.

III. STUDY OF DEVOPS AND CT TOOLS

DevOps is a collection of best practises, philosophies, and individuals. Various release management and deployment solutions support it. Collaboration, automation, measurement, and monitoring are all key components of DevOps. It focuses primarily on four process steps: log monitoring, monitoring, build-test, and deployment-configuration. They are supported by a variety of tools and technologies, as shown in Figure 1.

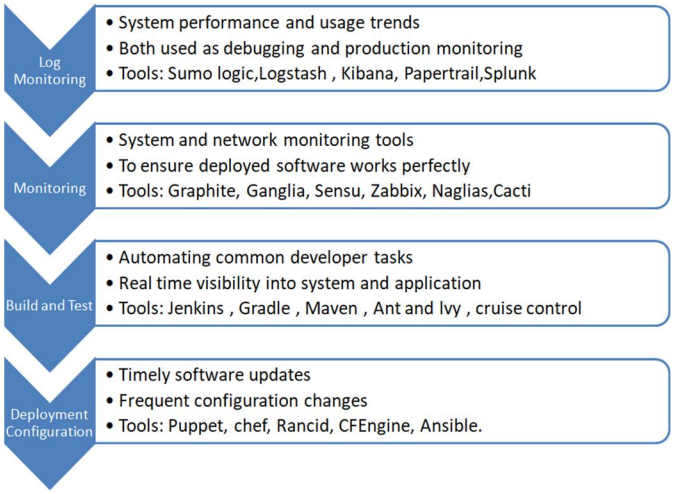


Figure 1: DevOps Key Areas-Tools

The DevOps lifecycle is all about putting continuous concept into practise and maintaining a persistent feedback loop. Continuous Testing is inextricably linked to the other parts of the continuous mindset. The same is depicted in the accompanying Figure, which is supported by the required toolset.

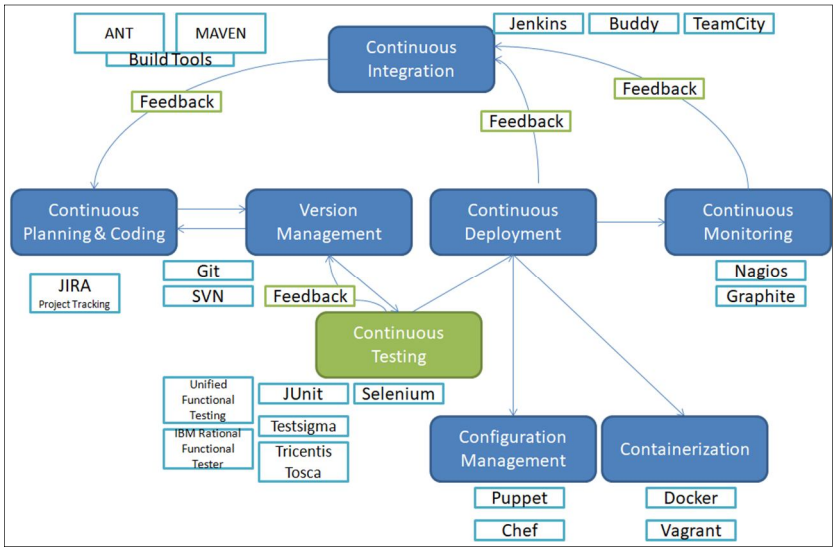


Figure 2: DevOps Lifecycle and Tools Support

IV. PROPOSED CONTINUOUS TESTING FRAMEWORK

The success of continuous testing is determined by the following four tactics and their methodical implementation: Strategy for Tools and Technology, Skills, Organizational Culture Adoption Strategy, Process, and Metrics Figure below shows a strategy.

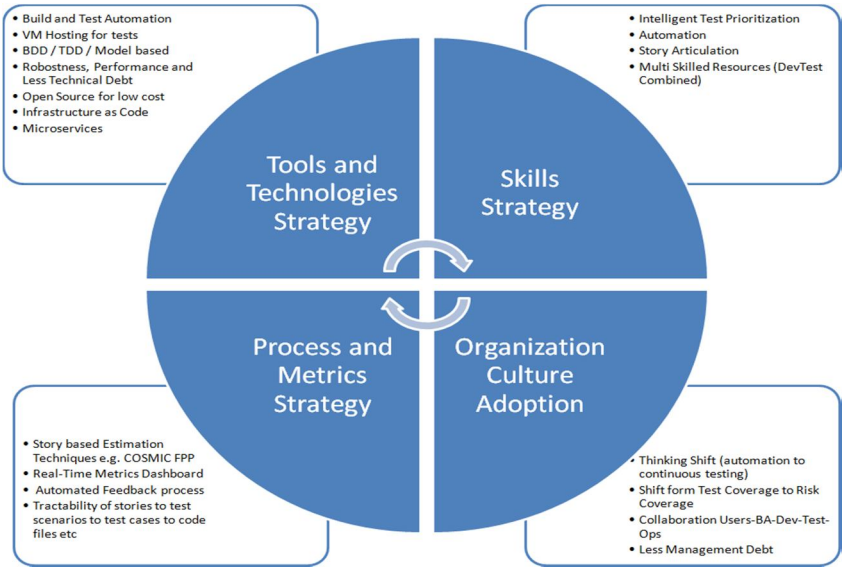


Figure 3: Strategies for Successful CT

The CLAMS performance model can be used to evaluate an organization's DevOps success (Maturity Model). It stands for "culture, lean automation, measurement, and sharing," which are the four main pillars of the DevOps business model. DevOps is more of a culture than a work. It's a method of thinking about things. It eliminates the blame game, gives continual feedback, and adds value to the user's business. It supports tool-driven project management and focuses on goals, ideas, metrics, and measurement. The suggested continuous testing methodology must be integrated with the CLAMS maturity model.

The suggested Continuous Testing (CT) framework is depicted in the diagram below. It encourages high test coverage, early feedback through a shift left method, efficient management of test defects, analytics-driven implementation, test data abundance, and service virtualization adoption. The CT framework significantly reduces the cost of problem fixing.

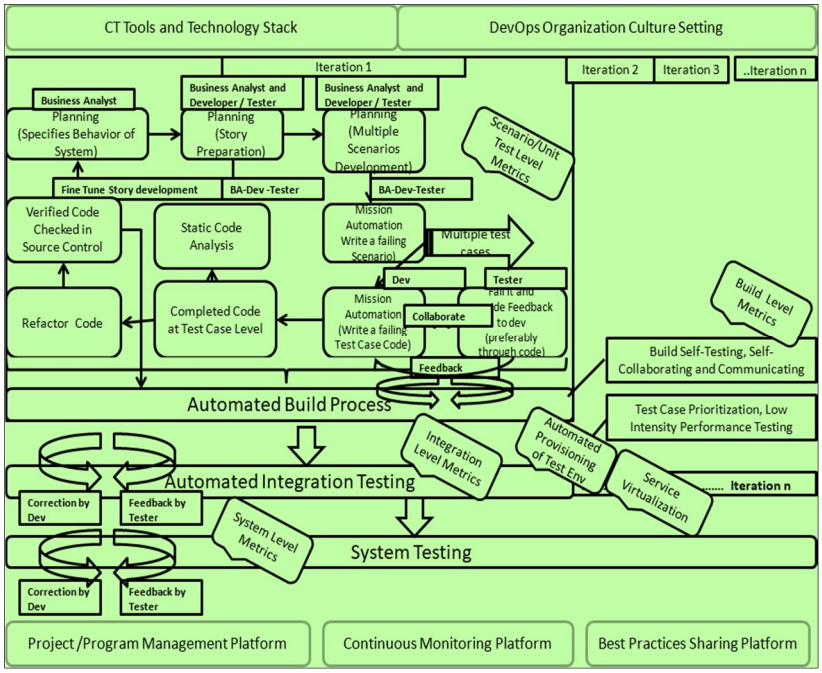


Figure 4: Proposed CT Framework

In the context of Continuous Testing, project management / programme management is crucial. The Project Manager must adequately define the scope of the current running test and balance the effort of the team. All conceivable dependencies and risks must be anticipated by the PM, and the project must be scheduled accordingly. PMs' responsibilities have evolved from examining Gantt charts and holding meetings to participating in actual release cycles. They must adopt a microservices approach. They should have access to the same tools that other developers, testers, and operations teams do. This is the new age competency that a project manager is expected to have.

V. CONCLUSION

In the framework of DevOps continuous philosophy, this study stresses the need of continuous testing. This research also claims that successfully implementing Continuous Testing (CT) in a company leads to enhanced efficiency, risk management, user experience, and quality. Continuous testing is closely linked to test case automation, auto-generation of test cases using model-driven frameworks, and metrics-driven processes. All integration difficulties are discovered much earlier in the development life cycle with CT. It facilitates defect resolution at a lower cost and in a shorter time frame. It allows testers to devote their valuable time to exploratory and value-added testing. In the future, It is possible to assess the impact of Continuous Test Driven Development (CTDD)/Pair Programming in the context of Continuous Testing and the tooling that supports it.

REFERENCES

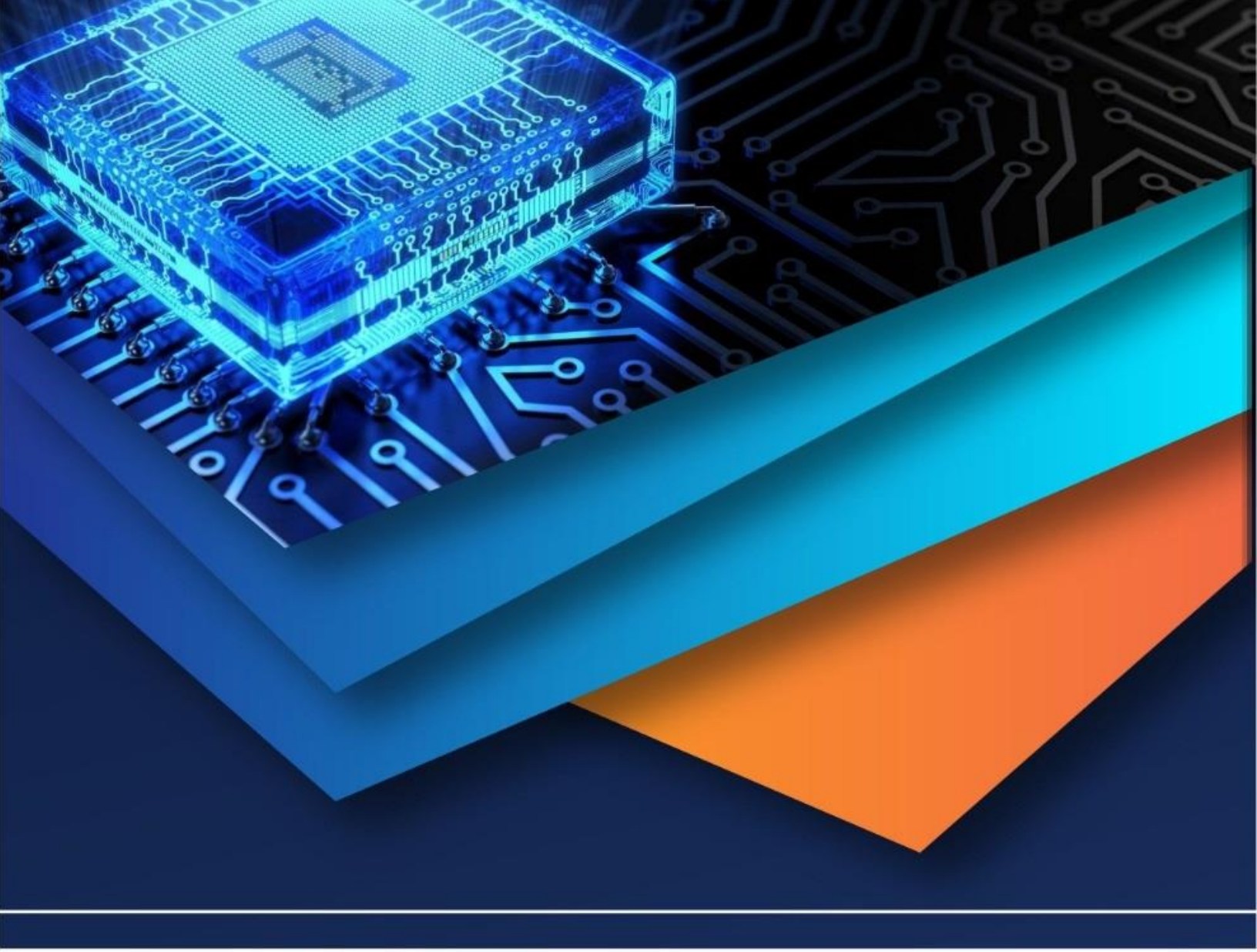
[1] Angara, Jayasri & Prasad, Srinivas & Gutta, Sridevi. (2020). DevOps Project Management Tools for Sprint Planning, Estimation and Execution Maturity. Cybernetics and Information Technologies. 20. 79-92. 10.2478/cait-2020-0018.
[2] Karamitsos, Ioannis &Albarhami, Saeed & Apostolopoulos, Charalampos. (2020). Applying DevOps Practices of Continuous Automation for Machine Learning. Information. 11. 363. 10.3390/info11070363.
[3] Gokarna, Mayank & Singh, Raju. (2020). DevOps A Historical Review and Future Works.
[4] N, Sushma. (2020). Automation of Software Development using DevOps and its Benefits. International Journal of Engineering Research and. V9. 10.17577/IJERTV9IS060369.
[5] Ska, Yasmine & Syed, Habeebullah Hussaini. (2019). A STUDY AND ANALYSIS OF CONTINUOUS DELIVERY, CONTINUOUS INTEGRATION IN SOFTWARE DEVELOPMENT ENVIRONMENT. SSRN Electronic Journal. 6. 96-107.
[6] Wiedemann, Anna &Wiesche, Manuel &Gewald, Heiko & Krcmar, Helmut. (2019). Implementing the Planning Process within DevOps Teams to Achieve Continuous Innovation. 10.24251/HICSS.2019.841.
[7] Mamatha, Chellamalla& S C V S L S, Ravi Kiran. (2018). Implementation of DevOps Architecture in the project development and deployment with help of tools. International Journal of Scientific Research in Computer Science and Engineering. 6. 87-95. 10.26438/ijsrcse/v6i2.8795.
[8] Senapathi, Mali & Buchan, Jim & Osman, Hady. (2018). DevOps Capabilities, Practices, and Challenges: Insights from a Case Study. 57-67. 10.1145/3210459.3210465.
[9] Shahin, Mojtaba. (2015). Architecting for DevOps and Continuous Deployment. 10.1145/2811681.2824996.

AUTHOR PROFILE

Dr. Diwakar Ramanuj Tripathi,

Received the graduation (B.Sc.) degree in Computer Science, Master degree (MCA) in computer Application and Doctor of Philosophy (Ph.D.) in Computer Science. He is a Microsoft Certified I.T. professional (MCITP), Microsoft Certified Technology Specialist (MCTS) and Microsoft Certified Trainer (MCT) with 12 + years' experience in Computer Science. He has awarded the Life Time Achievement Award & Outstanding Scientist Award in Computer Science. He has pro-actively associated with various professional bodies.

IEI; IACSIT, Singapore; CSI; ISTE; IAENG, Hong Kong; IEEE; ASDF UK; CSTA; ACM, SDIWC, USA; IFERP; AIMA; ISCA; He is currently working as an Information Technology & Legal Consultant at Nagpur.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY