# DevOps - A Comparative and Perspective Study

Overview, Practices, Capabilities, Challenges, Tool Support, Experience Reports and Case Studies

Sai Preetham Reddy Veerannagari (40162727)

Kumar Arunesh (40170830)
Manoj Tooprani (40162843)

Azimbhai Karimbhai Surani (40169849)
Mohammod Suhel Firdus (40163582)

***Abstract: The objective of DevOps is to work hand in hand with developers in order to deliver or maintain software as efficiently as possible without compromising quality.*** **Various studies are being performed to understand and find ways to make DevOps work effectively while overcoming its challenges as the trend of organizations moving towards DevOps grows. Studying DevOps, understanding definitions, and tools, as well as summarizing past research on the subject, is the purpose of this study. Furthermore, DevOps benefits, challenges, and tools that are available for supporting the process are discussed. Two case studies are presented in this Paper - one is adapted from our study materials and the other is from a real-life DevOps team. For the validation of concepts and conflicting statements, a comparative study of papers is undertaken.**

***Keywords—DevOps, Continuous Delivery, Continuous Integration, DevOps Challenges***

## I. Introduction

The DevOps process involves developers and operations teams working together to support each other's work. *It is a way of thinking and a way of working*.[5]. An organization thrives on people who can perform their jobs effectively and contribute value to both the organization and their colleagues. By ensuring a competitive release deadline of software, DevOps facilitates the development and release of software products efficiently. Bringing together operations and development teams is not the only definition of DevOps. The moment that a programmer commits their changes to a version control software (such as GitHub), the changes are installed on a test server, giving them peace of mind that their code is error-free. The automation engineer sets up scripts (such as Jenkin) that periodically pull bug fixes from the source control system, deploy them to the test servers (applications as well as databases), and then execute regression tests on them. Automation test software executes written test cases, as well as checking code coverage to detect any effects of the recent changes. These can all be done behind the scenes, with no or very little human intervention. Containers, such as docker, can be provided by the operation team with prebuilt configurations for faster development. Code execution in a production environment should be the concern of the programmer, not environmental issues. As a result, continuous integration and continuous delivery are integral parts of DevOps. We will cover best practices,challenges , advantages and disadvantages in this summary as well as some tools available which help perform the operation and well as development. Additionally, a case study will be adapted based on the individual's experiences.

The remainder of the paper is organized as follows: we start with a background of the topic DevOps in section II where a brief overview of the DevOps process is discussed, followed by section III where a summary of each paper studied is presented. Section III also gives an elaborated take away from all the papers studied and discusses the DevOps process with more details and a tabular comparison of all the papers. Section IV discusses controversial statements and discussions followed by chapter V which is the conclusion.

## II. Background

### a. DevOps Practice

*Devops could be a movement defined by stories and ideas of people, teams, and organizations. it's the continuing conversations and evolution of processes and concepts that lead to growth and change* [5]

The concepts that define Devops are similar to those of Agile and Lean. Developers in agile environments work in a fast-track environment called sprints- a continuous planning process that initiates at the beginning and continues throughout the process. Agile development has resulted in parallel cycles of development, release, and learning. Agile and DevOps are two processes that synergize well.

DevOps is not a full-fledged software development methodology like waterfall but too is influenced by agile and Extreme Programming. This practice brings together the Development team and the Operations team. With DevOps, the team's culture and cognition is enhanced.

Organizations can strengthen their Software Development Life Cycles with DevOps to improve efficiency and satisfy clients' changing needs. Some of the organizations use enhanced definitions of DevOps like DevSecOps or

SecDevOps(Development-Security-Operations), BizDevOps (Business-Development-Operations) and DevNetOps (Network-Development-Operations). Vital Organizations describes DevOps as the cognition between Software development and Operations. Varied organizations define DevOps differently as it ultimately focuses on agility, availability, stability, reliability, and security.

For digital transformation to happen rapidly, DevOps plays a critical role. A digital transformation focuses on customer cognition and faster operational changes. Seeing that technology was not just an asset, but also cultural and procedural practices, organizations started to use DevOps to reduce costs and improve performance and quality in software development.

DevOps practices emphasize Continuous Delivery, which decreases risks in software development by acquiring continuous feedback. When choosing aspects in deployment, collaboration and procedural practices are important as well as monitoring and security.

Project requirements are divided into user stories in DevOps collaboration culture, which are prioritized and assigned to different team members with deadlines. During scrum meetings, each member of the team will check the status of each task.

DevOps procedures are based on deliverables that are delivered by development teams. Feedback enables continuous delivery to reduce several risks.

### b. DevOps Capabilities:

DevOps capabilities can be categorized as follows:

***(a)** Continuous Integration: It* allows developers to instantly see the impact of their code changes and fix problems on the spot when they occur in the development environment, this further became a major point of interest for DevOps as smaller, more frequent changes reduce merge and integration issues.

(***b)**Continuous Deployment:* DevOps emphasizes continuous deployment, which means deploying several smaller changes at the same time, rather than waiting for a 'full package' of changes to be ready. This comes directly from the practice of frequent releases.

***(c)** Continuous Monitoring:* By collecting metrics and data from all stages of an application lifecycle, Continuous Monitoring allows users to react quickly, possibly improving functionality.

***(d)** Continuous Testing:* It means conducting tests as soon as possible and continuously during the development lifecycle, which reduces development costs and improves the quality of software. The importance of this capability is that it will eventually increase customer satisfaction, as the customer has a larger and more immediate impact on the product.

***(e)** Continuous Feedback:* Developers are expected to provide continuous feedback on coding, architecture, build failures, test status and file uploads.

***(f)** Infrastructure as code:* It involves an on demand scalability of infrastructure, handling the configuration code the same way as the application code. Development and Operations teams can work together to automate, manage and maintain systems and applications through a common set of tools and practices.

### c. Challenges:

The development of DevOps is still an evolving phenomenon. Several challenges arise due to its vague yet huge area of inclusion in terms of technologies, legacy infrastructure, culture, etc. It takes a team of specialists to successfully onboard a company on DevOps. Historically, Development and Operations were not linked, thus the area is too large to be covered by one factor or one technology, it needs a coordinated effort on all fronts. Studies are done to understand the various aspects of these challenges, from interviewing IT professionals to applying complex mathematical approaches in order to gain a deeper but still clearer understanding.

Culture and technology are two important factors that reverberate repeatedly. DevOps expects the culture of a company to change overnight, and for this to happen, a culture of trust, transparency, open communication, and collaboration is what's needed. It is crucial that the management as well as the professionals involved in decision making be given more freedom and power so as to avoid a simple breakdown destabilizing the entire process. HR (Human Resources) should be more proactively involved than in the past, since the system of checks and balances is not efficient enough because of so many players within one team.

It is also not in favor due to the fact that each project uses a unique stack of tools that were neither conceptualized nor designed considering DevOps. Complete renovation requires a lot of time, effort, and skill. To determine what is best, professionals at the bottom of organizational hierarchy must be aptly consulted. It is a major obstacle that key people leave, they do not stay with one organization or project for a long time, which means knowledge is lost as time passes, which makes DevOps Onboarding rather time consuming as well as expensive. The backward compatibility of modern technology is also a major concern.

There have been many studies done and are still being done to uncover these hidden factors, their solutions, and mitigation strategies. It has always been a continuous process.

The popularity of DevOps came from its collaboration between development and Operations *[11]*. It is important to support this collaboration both culturally and technologically. There are a lot of factors on which the machinery relies, and it tends to fail often at its weakest link. The issues organizations face are: ***(a)*** Construction and maintenance of services were arduous and time consuming **(b)** Slow deployments were costly and error prone **(c)** No shared ownership results in no feedback and inefficient practices **(d)** Configuration management was not properly managed.

Possible solutions: **(a)** *Creating a synergy between Developers and Ops Mindset* - Developers are happy with the setup, as the development part remains the same, but for Ops, it is troublesome, as they now have to coordinate with Developers. This needs a solution as frequent updates with few changes per cycle. ***(b)*** *Work disruption:* A completely new workflow is required and many tools must also be replaced or redesigned as a result. An in-depth, holistic study is required to sense out and plan such changes meticulously. ***(c)****Management's responsibility*-Management must empower every team member and professional to express their concerns. A proper trust and chain of command must be implemented as soon as possible in DevOps since a lot of management decisions become more collaborative in nature. ***(d)*** *A move towards modern infrastructure:* Legacy infrastructure must be modernized using available tools to provide a much better technological support. When using a VM snapshot for development, the on-premises server is not at risk of failure. By analyzing such vast legacy infrastructures, we can gain important insights on how to efficiently transform the old yet powerful on-premises hardware .

**d. Tools Support:**

The following tools are used by DevOps professionals today. The tools family is divided into the following categories: **(a)** Development/Build and Test **(b)** Deployment and Configuration **(c)** Monitoring **(d)** Project Management/Collaboration **(e)** Scripting Support (Language)

TABLE I shows how DevOps tools can be classified, despite the fact that a single tool may serve multiple purposes.

TABLE I. CLASSIFICATION OF TOOLS

DevOps Tools Classification

| Category | Sub-Category | Software/tool |
|---|---|---|
| Development/Build and Test | Source Code Management | GitHub, GitLab, BitBucket |
| | Build | Ant, Ivy,Gradle, Maven |
| | containers | Docker,Kubernetes,Node Package Manager OpenShift,Pypi. |
| | Infrastructure | Puppet, chef, Ansible |
| | Testing | SaltStack, Ansible,Chef, Puppet, Chaos Monkey, batfish |
| Deployment and configuration | CI/CD | Jenkins, BitBucket, GitHub, GitLab, CircleCI, Azure DevOps, Google Cloud |
| Monitoring | Log Monitoring | Loggly, Sumo logic, Logstash, Kibana (ELK stack), Papertrail, |
| | System Monitoring | Graphite, Ganglia, Sensu, |
| | Network Monitoring | Zabbix, Nagios, cacti, |
| Project/operation management tools/software | Collaboration | Slack |
| | Issue/Requirement Tracking | Jira, Zendesk, GitHub, ServiceNow |
| Automation Scripting | Language | YAML Python, Unix Shell, PowerShell, Groovy, |

*(a) DevOps Development Tools* : The development phase can be eased by many open-source innovations, whereas the build and test phases can be smoother with many automation tools.

People who work in collaborative environments work within a larger system. Users may work on a segment of a large source code, or develop new code that must be integrated with other developers' work. Collaboration with other developers' work is facilitated by a source code repository via version control tools. GitLab, an enhanced version of GitHub, is the go-to tool for anyone looking for source code or wanting to interpret a software.

GitHub is one of the version controlling tools of choice for DevOps teams. In addition to the command-line interface (Git) for advanced users, a web-based interface is available for ease of use. Several team collaboration features are included as well as access control. It serves as a central hub for developers. Build and deployment tools can be integrated with Git. Automating software build can be achieved by building a daily/weekly build trigger on the GitHub repository.

Bitbucket, SourceForge, etc., are other version control tools.Google and Amazon have also added version control capabilities to the cloud with the advancement of cloud technologies.As an alternative to manually compiling every package, build tools come in handy for creating changes or pieces of software. Currently, Java tools like Gradle and Maven can interoperate with version control systems, making DevOps life easier. To download dependent libraries, these tools connect to the web or a centralized repository. Combining build tools with code coverage tools (like Maven Jacoco) and unit test scripts can help improve code consistency and quality.
The choice of build tool for an enterprise depends on its software and array of products.

***(b)*** *Deployment, Configuration and Integration* ***:*** The days of manually configuring the server and deploying the software are pretty much over. Continuous integration is an essential part of continuous development. The process of manually building and deploying a large source code project with multiple changes in a short period of time will be very time-consuming with a large team.

The build tools (ant, maven, or gradle) prepare the executable of the software (.war/.ear/.jar/.exe, etc. ), so they should be deployed in an application server or database so that end users can download the latest version of the software. One can do that manually if the number of software it handles is a handful. But as the number of applications increases, or if it is a microservice-based application, the number of deployments increases. Also, with the increasing frequency of enhancement and change on the software product, the number of deployment requests increases.

Tools like Jenkin can perform the following: **(a)** Build automation, **(b)** Test Automation, **(c)** Deployment to the servers.

This enables CI/CD to process quicker. Other tools available are UrbanCode Deploy (Udeploy). Tools like AutoSys also help to deploy but is more useful for database related deployments

*(c) Monitoring* **:** The log is the primary source of investigation if anything fails in any software. A log monitoring process can identify any error the software has encountered during runtime and can be used to identify the root cause of that error. Maintainers can analyze and store logs with log monitoring tools. Splunk, paper trail, and sumo logic are a few tools that allow the user to search for exceptions in large log files. Open-Source software stacks like ELK (Elasticsearch, Logstash, Kibana) when used together bring log and performance monitoring to the next level. Also, system and network performance can be monitored using specific tools for measuring *predictive rejuvenation*.

*(d) Project management tools/collaboration:* As part of DevOps and continuous software maintenance, a tool like JIRA can be used to log problems, assign problems to experts, and track issues. All software or maintenance issues are logged in JIRA, or similar software like Zendesk. JIRA serves as a documentation tool as well in Agile. In order to fix a problem, these tools serve as a way for teams to communicate. Recently, Slack has gained popularity as a communication tool between teams and teammates, and is also considered a collaboration tool that aids in DevOps operations.

**(e)** *Automation scripting* **:** Many of the DevOps tools are backed by scripts which are then executed by the tools. If a developer wants to clean up logs of a server at a certain frequency, he or she can use a shell script which will be executed by cron job or scheduler. Many programming languages help in scripting like Python, Groovy etc.

## III. Comparative study of papers

### *a. Summary of studied papers*

***Paper: [1] A Basic Introduction to DevOps Tools by Akshaya H L, Nisarga Jagadish S, Vidya J, Veena K***

***i. Motivation:*** Discovering the different tools and techniques used in DevOps

***ii. Methodology used:*** After reviewing tools available to developers, authors categorized them according to how the tool is used in DevOps stag

***iii. Findings:*** *Akshaya H L et al* [1] spend a large portion of their paper studying the tools used in DevOps. An overview of a wide array of tools is presented and they have also categorized them by the kind of work they can complete with them. As described in this paper, DevOps tools can be categorized as follows: **(a)** *log monitoring* **(b)** *monitoring* (which includes system administration and network administration), **(c)** *build and test* and **(d)** *configuration and deployment* [1]. An overview of commonly used DevOps tools is provided in this study, including a description of their features.

***Paper: [2] A Survey of DevOps Concepts and Challenges -Leonardo Leite, Carla Rocha, Fabio Kon, Dejan Milojicic, Paulo Meirelles - Cornell University -Wiedemann, Anna & Wiesche, Manuel***

***i. Motivation:*** To explore DevOps tools and understand how they operate

***ii. Methodology used:*** *Leonardo Leite et al* [2] designed a study based on 198 found papers, which underwent several stages of selection before 50 core papers were selected. The purpose of this study is to discuss other researchers' research on DevOps.

***iii. Findings :*** *Leonardo Leite et al* [2] reviewed DevOps tools and classified them according to their usage and users. Project teams (the entire team) use the tools, but DevOps teams are solely responsible for their deployment and maintenance. An overview of these tools is given in the paper, and it discusses how these tools are useful for different types of systems (legacy vs. modern). They classify DevOps in six examples [2] viz- ***(a)*** *DevOps emphasizes collaboration* **(b)** *DevOps advocates collaboration* **(c)** *DevOps emphasizes collaboration and communication* **(d)** *DevOps shares knowledge and tools* **(e)** *DevOps has a culture of collaboration / Collaboration means sharing knowledge and tools* **(f)** *DevOps emphasizes working together / working together by sharing tools, processes, and practices* [2]. Their study focuses on the following research questions regarding DevOps - **(a)** *definition of DevOps* **(b)** *motivation for adoption* **(c)** *expected benefits* (**d**) *main challenges in DevOps* [2].

A tool can be classified by Leonardo Leite [2] et al according to its usability into **(a)** *tools which assist with collaboration* **(b)** *tools for Continuous Integration / Continuous Delivery* and **(c)** *tools that support software reliability*. Furthermore, they discussed the actors who are using these tools at different stages of DevOps.

***Paper: [3] DevOps in Practice: A Multiple Case study of Five Companies. Information and Software Technology - Lwakatare, Lucy Ellen & Kilamo, Terhi & Karvonen, Teemu & Sauvola, Tanja & Heikkilä, Ville & Itkonen, Juha & Kuvaja, Pasi & Mikkonen, Tommi & Oivo, Markku & Lassenius, Casper.***

***i. Motivation:*** To gain an understanding of how DevOps is practiced in various IT companies.

***ii. Methodology used:*** The researchers initially conducted workshops to identify open ended questions that were divided into 6 themes namely: - **(a)** *the background of the case and interviewee* **(b)** P*ractices involved in Development* **(c)** *Practices involved in Build and Integration* **(d)** *Practices involved in monitoring and infrastructure management* **(e)** *Recognized impacts* and **(f)** *Development culture*. After that, 3 researchers were assigned to each company for carrying out the study on a particular project about the questions identified above.

***iii.Findings:*** Researchers interviewed 5 small and medium IT firms (Company A through Company E) which collaborated on a project where the team implemented DevOps. Their effort led to success.

Researchers observed the interviewees after taking the initial interview following common things helping them in achieving DevOps: - **(a)** *Standup Meetings* (Company - A, C, D, E) **(b)** *Team Working Space* (Company - A, B, C, D, E) **(c)** *Development Technical Environment* (Company B, C) **(d)** *Ongoing deployment Process* (Company E)

TABLE II shows summary of each case study:

TABLE II. COMPARISONS OF CASE STUDIES

| *Characteristic* | *Company A* | *Company B* | *Company C* | *Company D* | *Company E* |
|---|---|---|---|---|---|
| Team Size /Roles | 1 Project Manager, 6 Full Stack Developers | 1 Project Manager, 5 Full Stack Developers, 1 UX Designer | 1 Product Owner, 1 UX Designer and 4 Developers | 1 Product Owner and 7 Developers | 8 Developers and 1 Team Lead |
| Product | Road Maintenance Reporting Tool | Health and Quality portal | Web-app for companies to order, monitor and manage purchased services | Rest API for providing media content | Security Cloud Service |
| Sprint Period | Fixed time of 2 years for whole project | 2 weeks | Undefined | Undefined | 1-2 Weeks |
| Output Image | JAR File | Docker Image | AWS (Amazon Web Services) machine image | Debian Package | Docker Container |
| Tools Used | Deveo, Jenkins, Selenium | GitHub, Jenkins, SonarCube, Ansible, Docker | BitBucket, Jenkins, Chef, New Relic | Puppet, Git, Jenkins, New Relic | Bitbucket, Jenkins, Docker, Amazon CloudFormation; Kinesis |
| DevOps Adoption | Developers were given freedom to improve their way of working and remove painstaking practices. | The customer willingness to adapt to AWS based deployment was helpful in practicing DevOps. | The company adopted AWS infrastructure and created an automatic deployment pipeline to shorten the time between releases. | The DevOps implementation is still under way in this company but currently it is just motivated by few practitioners. | The operations team and developers team collaborated to implement DevOps environment to avoid wastage of time for small tasks. |

The main DevOps practices that were seen are: - **(a)** *Automated Deployment Mechanism* - This is a mechanism designed to ensure that builds are automatically deployed if they pass all tests and criteria and alerts the developer if something goes wrong. (**b**)Infrastructure as a code - the act of repeatedly provisioning and maintaining. The configuration of environments is done by scripts instead of by manual intervention. (**c**) *Monitoring* - It is the process of continuously monitoring the production environment by using tools such as New Relic or Kinesis, as well as the Continuous Integration environment to test the quality of development builds.

The authors noted some of the benefits of DevOps such as - (**a**) *Better delivery speeds* (**b**) *Improved productivity in the operations department* (**c**) *Improved software quality* (**d**) *Organizational confidence in using DevOps*. Representatives of the team talked about challenges encountered during the development process: **(a)** There were challenges faced in infrastructure automation because scripts and legacy systems can be error-prone. **(b)** Few developers leave their company for personal gains, so training new employees or hiring new skilled workers would take up a large amount of their time. **(c)** Due to the limited project budget and deadline, the quality of the system was often compromised **(d)** Monitoring microservice based applications was difficult because developers did not know what matrices and graphs to use.

In the end the interviewers concluded with the following items for successful implementation of devops [3]:- ***(a)*** DevOps requires a software development team that is committed and has ownership of installing software changes in production. **(b)** The use of tools and support for pipeline operations accelerates the delivery of software changes, bug fixes, and production management. **(c)** The learning curve in the new skill area becomes more acute for software developers and operating personnel as both must deal with dealing with pressure when working under time constraints. **(d)** Project completion times affect product delivery speeds due to contextual issues including approval from the product owner.

***Paper: [4] State of Devops Report 2021 by Puppet - Nigel Kersten, Kate McCarthy, Michael Stahnke, Caitlyn O'Connell***

***i. Motivation:*** To understand the progress and barriers of DevOps adoption in the whole world.

***ii. Methodology:*** Information was collected from IT Developers and industry leaders with a working knowledge of their software operations and delivery processes. Participants were recruited by two methods, a snowball sampling method and a third-party panel, during the survey which was conducted online for a month (2021) around the world.

***iii. Findings:*** The authors of this white paper have conducted this type of survey for over 10 years, and have divided organizations into three categories: - (**a**) *High* - Organizations where DevOps is highly developed. (**b**) *Middle* - Organizations in which DevOps is not fully adopted. (**c**) *Low* - Organizations with little or no experience with Devops.The survey showed that 78% of organizations have been stuck in the Middle Group for the last four years, and only 2-3% managed to move to the High Group each year.

TABLE III. COMPARISON OF CHARACTERISTICS IN 3 TYPES OF ORGANIZATIONS BASED ON THEIR LEVEL OF DEVOPS ADOPTION

| ***Characteristic*** | ***High*** | ***Middle*** | ***Low*** |
|---|---|---|---|
| Deployment Frequency | Whenever Needed | Between 1 day to 7 days | Once in a month or two months |
| Max time for changes | < 60 mins | < 1 week | 1 week to 6 months |
| Mean Time to Recovery | < 1 hr | < 1 day | < 1 week |
| Change Failure Rate | < 5 % | < 15 % | < 15 % |

TABLE III suggests how DevOps adoption changes the performance by huge margin.

Researchers found the following major blockers to successful adoption of DevOps: - Support from executives - More than 90% of executives supported DevOps in the Highly Evolved group of organizations, while less than 45% supported it in the Low Support group. In the case of traditional authorities, it therefore makes sense. A particular organization would never evolve even though the team leads or team members are enthusiastic.

Separate Platform Team Support - Almost 50% of High Group Organizations had a separate platform team and their roles were clearly defined, whereas only 8% of Low Group Organizations had a separate platform team.

Several common myths about DevOps were also observed by the authors such as: (**a**) *DevOps and automation are the same* - Many companies believe that if they have automated the repeated processes, they have implemented DevOps. However, they do not understand that automation will not be helpful if people aren't using it. Among companies at the

Mid-Level, only 60% used automation to the fullest extent. One of the main reasons they were stuck at Mid Group was because of that. (**b**) *Adopting a public cloud is equivalent to implementing DevOps*, according to the majority of mid-level practitioners. However, they noted that it is still not utilizing the main benefits of cloud computing, such as automatically provisioning and releasing computing and network capabilities (mail, storage, network).

***Paper: [5] Book extract: Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale - Jennifer Davis***

***i. Motivation:*** The book explains the basics of DevOps and how it can be implemented effectively

***ii. Methodology used:*** Case studies are cited to explain DevOps concepts on the fly

***iii. Findings:*** The author of the book has presented a nice example of how a new programmer is onboarded and transitioned into the development environment where DevOps was successfully implemented.

From the book, *Effective Devops* [5], we learn the following before and after case study of implementation of devops at Etsy:-

*Before implementation of DevOps:-* The developers had their own blade servers to work on, which were not able to run automated test suites because of poor specifications. Tests that were run in a staging environment took more than 2hrs to complete, which were less useful because fixing and testing them would take away more time. Additionally, the developer team and operations were separate and had no communication between them. The merging in the branch used by the development team was done by using traditional svn, and then the Operations engineer would deploy that particular branch.This process would take more than 4hrs, so they did it once in 3 weeks only. Eventually these would delay the release of the product.

*After implementation of DevOps:-* Every employee is given their own local VM, whose settings are identical to the production environment and can run tests locally. This allows them to finish testing in an earlier phase only. Secondly, by the use of mock Jenkins Cluster of Continuous Integration, allows to merge the code with the master branch without committing any code to the master branch,which gives confidence to developers of not breaking the functionality at a later stage during the main integration phase in the Master's Branch. After that an employee pushes his/her changes to the Production using push queue ( allowing to resolve conflict between builds of employees). This build is then deployed to the QA server and all the test cases are run on the new build. If the QA test cases are passed the deployinator shows the status ready to be deployed else it logs the errors. This process is streamlined to be completed within 10-20 minutes and developers can do this process 60 times a day. This eventually helps to quicker deliveries with bug-free builds.

This before and after story tells us how the correct implementation of devops can increase the lifetime of a product by years.

***Paper: [6] Industrial DevOps - Wilhelm Hasselbring Soren Henning, Bjorn Latte, Armin Mobius, Thomas Richter§, Stefan Schalk, and Maik Wojcieszak***

***i. Motivation:*** To investigate whether DevOps concepts can be used in industries other than IT

***ii. Methodology used:*** comparing DevOps processes (such as ecommerce software) to industrial processes.

***iii. Findings:*** The authors explore and propose the methodology and approaches of DevOps in Industrial System Integration. The collaboration between an industry domain expert and a software expert will result in better industrial system design. DevOps adapted processes can be useful in factories that use software systems, rather than outsourcing the software development process to big software companies.

***Paper: [7] Exploring the Link Between Leadership and DevOps Practice and Principal Adoption - Krikor Maroukian, Stephen R. Gulliver, Stephen R. Gulliver***

***i. Objectives:*** (**a**) To identify and understand practices and principles of Agile, Lean, and DevOps for improving productivity. (**b**) To determine whether DevOps is more beneficial than traditional service management. (**c**) Determine whether Leadership affects DevOps adoption within an organization.

***iii. Methodology used:*** To gather relevant data, semi structured one-to-one interviews were conducted with practitioners who had already adopted DevOps in their organizations. A later step involves aggregating the data.

***iii. Findings:*** Most participants prefer structured, agile, and lean practices. Many organizations choose SCRUM because it allows them to address backlogs while delivering products. The specific practices and principles of Agile, Lean, and DevOps for improved productivity include: **(a)** *Organizational culture* (**b**) *Monitoring* and **(c)** *Automation*.

Moreover, 66.67% practitioners have argued these practices are extensions to structured traditional processes. Change Management was the most considered process compared to To achieve best productivity, make sure all parts of ITSM (IT Service Management Process) are completed. Whereas, Release and Deployment, Incident and Problem and Service Level Management are top four ITSM processes, which affects the efficiency in delivery of software development.

Among practitioners 53%, stated that DevOps leader role should be an individual professional who is not involved in the project. Whereas 33% of them trusted their role in the team. The leadership skills that were mentioned by 50% of practitioners are: (**a**) *Technical background* (**b**)

*Communication and Collaboration skills* (*c*) *Relevant experience*. Albeit Challenges identified in DevOps acquisition are - (**a**) *Poor cognition between cross teams*. (**b**) *Organization's traditional cultures which are hard to modify* (**c**) Operations teams being less involved in the requirements phase. One of the interviewed IT managers stated that *"Leadership skill set is the most important thing to adopt barrier breakdown."* DevOps practice should be adopted ubiquitously (87%), Consisting traditional Development (97%), Operations (97%), Quality Assurance (93%) and Information Security (80%) teams.

***Paper: [8] Communication Challenges and Strategies in Distributed DevOps -Elisa Diel, Sabrina Marczak, Daniela S. Cruzes***

***i. Motivation:*** (**a**) to discuss the strategies and the communication practices of a distributed agile team. (**b**) to focus on communication between developers and operations teams that are distant from each other in a multinational enterprise.

***ii. Methodology used :*** A survey is conducted by the authors based on their observations of the team members and face-to-face interviews. This interview involved twelve professionals and all data was collected using DevOps.

***iii. Findings:*** *Th*e conceptual mapping analysis approach was developed based on the maintenance performed by the operations team. Using the above method, all of the key points were named and then grouped by meaning.

*Mohr et al* [8] explain four types of communication strategies as (**a**) *Frequency*: refers to the frequency of contact between interlocutors. (**b**) *Direction*: refers to the vertical and horizontal movement of communication within the organizational hierarchy (**c**) *Modality: It* refers to the method used to convey information. (**d**) *Content*: It refers to the message being transmitted.

Study participants were multinational IT companies that develop software products to support software development. The company includes eight distinct locations and employs about 1,400 people. Most of the development team is located in Australia, but the team operates across different time zones. There were 13 members in Brazil who participated in the activities as well as 3 different development teams with 18 members (total of 31 people) in Australia. Product and infrastructure deployment is the responsibility of operations staff, and for collaborating on customer-reported issues. It is the development team's responsibility to design the product and fix any bugs.

***Paper: [9]: Challenges and Approaches for the Assessment of Micro-Service Architecture Deployment Alternatives in DevOps -Alberto Avritzer***

***i. Motivation:*** To provide an overview of architecture assessment challenges and approaches in the context of DevOps and microservices.

***ii. Methodology used:*** A model is presented using data from application performance management (APM) tools, access to operational workload profiles, and security advice.

***iii. Findings:*** By focusing on an approach for reliability assessment of microservices architectures, this article provides better recommendations than approaches that do not take production use into account.

An overview is given of (**a**) a*dvanced approaches to collecting operational data from production systems using APM tools* (**b**) *reliability challenges for DevOps and microservices* and (**c**) t*he approaches selected based on operational data for reliability assessment*. CI/CD pipelines for DevOps can fully incorporate these selected approaches.

***Paper: [10] Challenges in Adopting Continuous Delivery and DevOps in a Globally Distributed Product Team -A case study of a healthcare organization, Rajeev Kumar Gupta, Mekanathan Venkatachalapathy, Ferose Khan Jeberla***

***i. Goal:*** Embrace the challenges of DevOps Continuous Delivery.

***ii. Methodology used:*** The authors describe their experiences of timeboxed release strategies, value stream-based execution, operations, testing, and transforming traditional scrum teams into DevOps teams. They were able to stabilize processes and methods through these practices.

***iii. Findings:*** It provides an overview of the challenges and practices of DevOps, the formation of the DevOps team, enabling the DevOps team, and the impact of the newly formed DevOps team. Also, it covers the tools and technologies that the DevOps team uses and how they can be deployed.

All the members of the team from multiple locations, including the operations team, quality manager, and regulatory expert, attend a program increment workshop. During this two-day workshop, feedback from every team member is considered, and the backlog is adjusted. It is a team-wide event where all assumptions are cleared up and focus is focused on what is needed.

***Paper: [11] Survey on Challenges in DevOps -Mr. Veeranna Kotagi, Mrs. Swathi. K, Dr. Piyush Kumar Pareek***

***i. Motivation:*** To gain a greater understanding of DevOps, the challenges, and possible solutions.

***ii. Methodology used:*** A literature review is carried out based on six major factors - Code, Build, Check, Package, Release, Configuration, and Monitoring.

***iii. Findings:*** The article demonstrates how DevOps challenges are mitigated in a practical setting using agile software development. DevOps is considered a cultural phenomenon; therefore, it heavily emphasizes human relationships as a major factor for a successful deployment.

***Paper: [12] Prioritization Based Taxonomy of DevOps Challenges Using Fuzzy AHP Analysis, Muhammad Azeem Akbar, Wishal Naveed, Sajjad Mahmood, Abeer Abdulaziz Alsanad,(member, IEEE), Ahmed Alsanad, Abdu Gumaei and Ahmed Mateen***

***i. Motivation:*** Identify all major DevOps challenges and prioritize them.

***ii. Methodology used:*** The method used was systematic literature review (SLR) and to gather information

Surveying industry experts and prioritizing their insights via Fuzzy AHP Analysis.

***iii.Findings*** An extensive review of the Literature on CAMS (Culture, Automation, Measurement, Sharing) led to a list of 22 challenges. To determine the relative importance of challenges, a complex Fuzzy AHP analysis is performed. There was a lack of DevOps metrics, a communication gap, and a lack of trust among teammates.

***Paper: [13] Communication Challenges in DevOps& Mitigation Strategies, Snehitha Mandepudi***

***i. Motivation:*** To investigate the communication challenges developers face while implementing DevOps, and derive strategies to overcome them

***ii. Methodology used:*** Interviews with six experienced DevOps professionals along with 18 papers out of 538 to identify hidden communication challenges.

***iii. Findings:*** This paper relies heavily on existing literature reviews, as well as interviews with experienced subject matter experts, which provides mitigation strategies for various factors such as geographic and temporal distance as their literature counterparts aren't as complete and robust.

***Paper: [14] An Empirical Taxonomy of DevOps in Practice by Ruth W. Macarthy and Julian M. Bass***

***i. Motivation:*** **(a)** Propose a definition and description of DevOps that reflects practitioners' perceptions. **(b)** *To illustrate the implementation of DevOps*. **(c)** *To identify DevOps-specific functions that differ from IT Operations and development teams functions* **(d)** Demonstration of DevOps implementation in practice.

***ii. Methodology used:*** Interviews were conducted with industry practitioners with two years of DevOps experience, and Grounded Theory [16] was used to collect data efficiently. The audio of recorded interviews has been ejected into Nvivo, a qualitative analysis program. The data have been analyzed based on **(a)** *open coding*: The data have been coded line-by-line and clustered using concept classifications [17]. **(b)** *Memoing*: Memoing has been used to capture concepts and give relation among relevant concepts and concepts originating in stage open coding. **(c)** *constant comparison*: comparison between data collection and analysis. **(d)** *saturation*: When new categories no longer emerge, a saturation has been reached. According to [17] Glaser, saturation means that no further data are being discovered [17].

***iii. Findings:*** Participants described DevOps as better communication between development and operations teams, and others as the automation of software for better quality. DevOps has been divided into two categories: (a) *DevOps as a Culture* **(b)** *DevOps as a Job Description*. However, DevOps is not only a culture or description, but also separate teams from developers as well as IT Ops. Interviewees described four approaches to DevOps based on their descriptions of the concept and practice: developers' interaction with **(a)** *On-premises Ops* **(b)** *Outsourced Ops* **(c)** *DevOps teams* **(d)** *DevOps bridge teams*. DevOps teams provide physical infrastructure to IT Operations teams, and DevOps teams provide tooling and automated pipelines to developers; the latter provide the business value through applications and functionalities.

***Paper: [15] How DevOps Practices Support Digital Transformation -Sarah AL-Zahran Bahjat, FakiehBahjat Fakieh***

***i. Motivation:*** Determine the perceived benefits of DevOps practices that support digital transformation and common DevOps practices in digital transformation.

***ii. Methodology used:*** Two cases of development teams have been selected from **(a)** *Government Sector* and **(b)** *Private IT Company*. In addition to conducting interviews, open-ended questions regarding Development practices and DevOps impacts were emailed to interviewees before the interview. They have conducted online interviews for managers and face to face interviews for developers for about 1-2 hours. A process called cross-case synthesis was used to identify processes and draw conclusions for both cases. This method follows three main procedures: (**a**) *Decrease and summarize data* (**b**) *Present data* (**c**) *Illustrate and verify conclusions.*

***iii. Findings:*** DevOps implementation and the benefits that support digital transformation are: (**a**) *Incremental development with reduced delays in Software Deployments* (**b**) *More creativity and invention in employees as they are included in deliverables*. (**c**) I*ncreased cognition with customers with increased satisfaction through continuous feedback and continuous delivery*. (**d**) *Greater data insights Customer feedback*. (**e**) *Escalated skills and knowledge in team members*.

Common practices in digital transformation are categorized under two cases: (**a**) *Collaboration Culture* (**b**) *Procedural Practices*. Common practices in both Government and private organizations are Organized teams with common values and goals, with shared skills and knowledge in collaboration culture and Continuous delivery and integration with automated development and deployment in procedural practices.

However, Differences observed in both organizations are in the usage of automated dashboards and automated tests, which are only used in private organizations, whereas the Government Sector has used Excel instead of automated tools to update daily status..

***Experience Report/Case Study:***

We have interviewed a few consultants of a corporation who are deployed in a major USA bank for implementing BFSI(Banking, Financial,Securities and Insurance) software to understand how DevOps plays a role in day to day operations. The bank has multiple vendors and multiple softwares which are integrated for financial operations. DevOps plays a key role in that environment where mission critical operations are running and the DevOps team works behind the scene to support the same.

The integrated system undergoes multiple rounds of testing and fixing before being deployed to the production environment. With fast changing needs of the end users and change in the banking regulations( E.G. The payment network had changed in european market due to BREXIT) , the team adopted agile and took the approach of deploying recent changes every 24 hours. HP ALM and JIRA is being used as a project management and collaboration tool to log new defects or requirements. Support engineers and developers use GitHub for releasing recent changes. Jenkin builds the changes and is deployed in lower environments( which is not a production server) and informs the beta users about the new change. The change is tested using manual testing methods as well as automated softwares(proprietary). All the changes are then documented in JIRA against the original defect. Slack is used as an internal communication tool within teams. Tools play a major role in the entire life cycle.

The challenges faced by the implementation team was release management. As the software product was a monolith legacy software and was developed by using the waterfall model, adapting to agile took a lot of time. The legacy software was identified as modules and various new microservices were developed to entertain frequent change requests. The Ops team (who are the end users of the bank) are not dedicatedly assigned to the Dev team but a proper communication channel was set up using daily calls and communication tools.

### *b. Comparison of the reviewed papers*

TABLE IV. COMPARISON OF PAPERS STUDIES

| **Comparison of papers** | | | | |
|---|---|---|---|---|
| ***Paper*** | ***Criteria 1***<br>***Focus area***<br>***(DevOps Overview/DevOps Practice/Tools/Devops Challenges)*** | Criteria 2<br>**Method of Research**<br>(Systematic literature review/Interview/Case Study) | Criteria 3<br>Paper is based on (Research/Experiment/Comparison of Case Studies) | Criteria 4<br>Practical Example/Case Study |
| [1] A Basic Introduction to DevOps Tools by Akshaya H L, Nisarga Jagadish S, Vidya J, Veena K | Tools | Systematic literature review | Research | Practical Example |
| [2] A Survey of DevOps Concepts and Challenges -Leonardo Leite, Carla Rocha, Fabio Kon, Dejan Milojicic, Paulo Meirelles | Tools, DevOps practice | Systematic literature review | Experiment | Practical Example |
| [3] DevOps in Practice: A Multiple Case study of Five Companies. Information and Software Technology - Lwakatare, Lucy Ellen & Kilamo, Terhi & Karvonen, Teemu & Sauvola, Tanja & Heikkilä, Ville & Itkonen, Juha & Kuvaja, Pasi & Mikkonen, Tommi & Oivo, Markku & Lassenius, Casper. | Devops Tools, Challenges, Practices | Interview, Case Study | Comparison of Case Studies | Case Study |
| [4] State of Devops Report 2021 by Puppet - Nigel Kersten, Kate McCarthy, Michael Stahnke, Caitlyn O'Connell | DevOps Practices | Survey and Interview | Research | Statistics of practical examples |

|  |  |  |  |  |
|---|---|---|---|---|
| [5] Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale - Chapter 1 - Jennifer Davis | DevOps Overview | Systematic Literature Review | Case Study | Practical Example |
| [6] Industrial DevOps - Wilhelm Hasselbring Soren Henning , Bjorn Latte , Armin Mobius, Thomas Richter§, Stefan Schalk, and Maik Wojcieszak | DevOps practice | Systematic literature review | Research | Practical Example |
| [7] Exploring the Link Between Leadership and DevOps Practice and Principle Adoption-Krikor MaroukianKrikor MaroukianStephen R. GulliverStephen R. Gulliver | DevOps practice | Systematic literature review | Research, Interview | Practical Example |
| [8] Communication Challenges and Strategies in Distributed DevOps -Elisa Diel,Sabrina Marczak,Daniela S. Cruzes | DevOps Challenges | Systematic literature review | Research, Interview | -- |
| [9] Challenges and Approaches for the Assessment of Micro-Service Architecture Deployment Alternatives in DevOps -Alberto Avritzer | DevOps Challenges | Systematic literature review | Research, Interview | -- |
| [10] Challenges in Adopting Continuous Delivery and DevOps in a Globally Distributed<br>Product Team -A case study of a healthcare organization,Rajeev Kumar Gupta,Mekanathan Venkatachalapathy,Ferose Khan Jeberla | DevOps Challenges | Systematic literature review | Research, Interview | Case Study |
| [11] Survey on Challenges in DevOps -Mr.Veeranna Kotagi, Mrs. Swathi .K , Dr. Piyush Kumar Pareek International Journal of Innovative Research in Computer Science & Technology (IJIRCST) - ISSN: 2347-5552 | DevOps Challenges | Systematic literature review | Research | Practical Example |
| [12] Prioritization Based Taxonomy of DevOps Challenges Using Fuzzy AHP Analysis ,Muhammad Azeem Akbar,Wishal Naveed, Sajjad Mahmood ,Abeer Abdulaziz Alsanad,(member, ieee), Ahmed Alsanad,Abdu Gumaei and Ahmed Mateen | DevOps Challenges | Interviews, , Systematic literature review | Research. | Practical Example, Case Study |
| [13] Mandepudi, Snehitha. “Communication Challenges in DevOps a Mitigation Strategies.” (2019). | DevOps Challenges | Interviews, , Systematic literature review | Research, , Experiment | Practical Example , Case Study |
| [14] An Empirical Taxonomy of DevOps in Practice by Ruth W. Macarthy and Julian M. Bass | DevOps practice | Interviews, , Systematic literature review | Research, , Experiment | Practical Example |
| [15] How DevOps Practices Support Digital Transformation -Sarah AL-Zahran Bahjat, FakiehBahjat Fakieh | DevOps practice | Interviews, , Systematic literature review | Research, , Experiment | Case Study |

## IV. Controversial statement and discussion

*Leonardo Leite et al [2]* quotes previous researchers who advocate that engineers must have unrestricted access to production data, which can be controversial in specific environments, such as financial systems. This is controversial as developers are always provided role-based access and a full access to production system may not be possible because of the following reasons – **(a)** Organizations nowadays are multinational with teams working in various geographical locations. Many countries like the USA have their data privacy rules where the data must stay in the country. In this case, it is not possible to provide production data access to developers who are located outside of the United States. (**b)** production data is critical to business operations and susceptible to corruption by untrained developers.

*Leonardo Leite et al [2]* identifies for microservice management patterns versioned APIs are controversial. That means some practitioners recommend versioning of the API in development whereas some others do not. Versioning is required for (**a**) proper management of source code of the API (**b**) Serving all the clients who call an older version of API.

Whenever possible, we should not update the API version. Because we can never be sure when we may be forced to change it, we should probably support at least one version of our API. Adding versions is fine, but avoiding constant changes is better. A new API can often be introduced or an existing API can be extended without breaking the current version contract.

Furthermore, we never know when new technologies or design patterns will emerge that will allow more than one version of an API to coexist on the same machine in production in a very elegant manner. During the life of such a product, API's may see more version changes.

Despite their study concentrating only on medium and small-sized organizations, *Lwakatare et al* [3] did not consider large firms in their study. However if they had considered large organizations they could have found out that there are more challenges such as the traditional mindset of leaders, which was found out to be a major issue in a white paper study[4] by Puppet. As major DevOps players in the market (like Amazon,Google) are changing the perspective of the way we look at DevOps, a case study of larger organizations will definitely give us a deep understanding of DevOps practices.

In the whitepaper [4], the authors state that Automation is not equivalent to DevOps, but most software developers believe that automation is part of DevOps since faster delivery cannot be achieved without it, which is the main reason for using DevOps.

DevOps involves automation, but automation cannot be a synonym for DevOps, as is popular myth. Adopting a DevOps culture ensures faster and better delivery of software, and automation and tools play an important role in that.

*Krikor et* ***al [7] ,*** evaluates that more than half of interviewed software Developers stated that DevOps leader role should be an individual professional who is not involved in the project .Additionally one third believe that managers must be a part of the team. Devops need a much higher collaboration at all levels , so decision making should be a collective exercise .

Despite not being controversial, there are a few popular myths about DevOps that prevent people from truly understanding it: (**a**) DevOps revolves around automation. (**b**) DevOps demands that you adapt to the cloud. The study we conducted revealed that DevOps goes beyond just technology and tools, it's a cultural phenomenon.

## V. Conclusions

Creating and maintaining quality software with DevOps involves people, processes, and tools. The key to DevOps is automation, which can be achieved through the use of the right tools. Communication channels help teams collaborate, but setting them up can be challenging. DevOps gives all team members the opportunity to learn about the entire product development and release life cycle, allowing them to enhance their skills. Despite DevOps' compatibility with agile development, traditional software development is also capable of adapting to change, adapting SDLC processes, and providing training resources. In a traditional organization, the development team is responsible for different things than the operations team, but DevOps brings these two responsibilities together. Due to the availability of many tools for automating the development process, the process of developing quality software can be made significantly faster than traditional methods. The initial adaptation of DevOps can take a great deal of time and leadership experience.

## References

[1] A Basic Introduction to DevOps Tools by Akshaya H L, Nisarga Jagadish S, Vidya J, Veena K

[2] A Survey of DevOps Concepts and Challenges -Leonardo Leite, Carla Rocha, Fabio Kon, Dejan Milojicic, Paulo Meirelles - Cornell University Wiedemann, Anna & Wiesche, Manuel

[3] DevOps in Practice: A Multiple Case study of Five Companies. Information and Software Technology - Lwakatare, Lucy Ellen & Kilamo, Terhi & Karvonen, Teemu & Sauvola, Tanja & Heikkilä, Ville & Itkonen, Juha & Kuvaja, Pasi & Mikkonen, Tommi & Oivo, Markku & Lassenius, Casper.

[4] State of Devops Report 2021 by Puppet - Nigel Kersten, Kate McCarthy, Michael Stahnke, Caitlyn O'Connell

[5] Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale - Jennifer Davis and Ryan Daniels

[6] Industrial DevOps - Wilhelm Hasselbring Soren Henning, Bjorn Latte, Armin Mobius, Thomas Richter§, Stefan Schalk, and Maik Wojcieszak

[7] Exploring the Link Between Leadership and DevOps Practice and Principle Adoption-Krikor Maroukian, Krikor Maroukian, Stephen R. Gulliver, Stephen R. Gulliver

[8] Communication Challenges and Strategies in Distributed DevOps -Elisa Diel, Sabrina Marczak, Daniela S. Cruzes

[9] Challenges and Approaches for the Assessment of Micro-Service Architecture Deployment Alternatives in DevOps -Alberto Avritzer

[10] Challenges in Adopting Continuous Delivery and DevOps in a Globally Distributed Product Team -A case study of a healthcare organization, Rajeev Kumar Gupta, Mekanathan Venkatachalapathy, Ferose Khan Jeberla

[11] Survey on Challenges in DevOps -Mr.Veeranna Kotagi, Mrs. Swathi. K , Dr. Piyush Kumar Pareek International Journal of Innovative Research in Computer Science & Technology (IJIRCST) - ISSN: 2347-5552

[12] Prioritization Based Taxonomy of DevOps Challenges Using Fuzzy AHP Analysis, Muhammad Azeem Akbar,Wishal Naveed, Sajjad Mahmood , Abeer Abdulaziz Alsanad,(member, ieee), Ahmed Alsanad,Abdu Gumaei and Ahmed Mateen

[13] Mandepudi, Snehitha. "Communication Challenges in DevOps a Mitigation Strategies." (2019).

[14] An Empirical Taxonomy of DevOps in Practice by Ruth W. Macarthy and Julian M. Bass
[15] How DevOps Practices Support Digital Transformation -Sarah AL-Zahran Bahjat, FakiehBahjat Fakieh
[16] B. Glaser and A. Strauss, "Grounded theory: The discovery of grounded theory," Sociology the journal of the British sociological association, vol. 12, no. 1, 1967.
[17] B. Glaser, "Theoretical sensitivity," Advances in the methodology of grounded theory, 1978.
[18] DevOps for Developers -Michael Hüttermann ISBN: 978-1-4302-4569-8

| *ID* | *Name* | Contributions |
|---|---|---|
| 40162727 | Sai Preetham Reddy Veerannagari | Reviewed paper: [8] [9] [10].<br>Peer Review: [11] [12] [13]<br>Focus Area: DevOps Practice,DevOps |
| 40170830 | Kumar Arunesh | Reviewed paper: [11] [12] [13].<br>Peer Review : [3] [4] [5] [18]<br>Focus Area: DevOps challenges. |
| 40162843 | Manoj Tooprani | Reviewed paper: [7] [14] [15]<br>Peer Review : [1] [2] [6] [15] [16] [17]<br>Focus Area: DevOps OverView |
| 40169849 | Azimbhai Karimbhai Surani | Reviewed paper: [3] [4] [5].<br>Peer Review : [3] [4] [5][13]<br>Focus Area : Case Study and Experience Report |
| 40163582 | Mohammod Suhel Firdus | Reviewed paper: [1] [2] [6].<br>Peer Review :[3] [4] [5][1][5]<br>Focus Area: Case Study and Experience Report, DevOps Overview |