2022 International Arab Conference on Information Technology (ACIT) | 979-8-3503-2024-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/ACIT57182.2022.9994139

Effect of Using Continuous Integration (CI) and Continuous Delivery (CD) Deployment in DevOps to reduce the Gap between Developer and Operation

Abrar Mohammad Mowad
Zarqa University
Faculty of Information Technology
Software Engineering Dept
abrarmoed1994@gmail.com

Hamed Fawareh
Zarqa University
Faculty of Information Technology
Software Engineering Dept
e-mail: fawareh@zu.edu.jo
ORCID: 0000-0002-0853-3149

Mohammad A. Hassan
Zarqa University
Faculty of Information Technology
Computer Science Dept
email: mohdzita@zu.edu.jo

Abstract: **This paper focuses on how to use continuous integration (CI) and continuous Delivery (CD) methodology in DevOps to reduce the developer-operator gap. It also, shows how CI can be a CD bridge. The paper review DevOps and analyze strategies, methodologies, issues, and processes identified for the adoption and implementation of continuing practices. The result of our case studies shows the benefits, and advantages of using CI/CD in software development. Furthermore, this paper presents DevOps as a new model for reducing the gaps between development (Dev) and operations (ops). The Azure tool is used as DevOps CI/CD for the empowerment of continuous delivery of software to enable rapid and frequent releases, this enables rapid responses to changing customer requirements and thus it may be a decisive competitive advantage. This paper also measures the effectiveness of using CI/CD for reducing the time and effort in software development. We also, focus on the DevOps initiative to benefit of CI/CD and to effect of enhances flexibility in delivering the program with the expected quality on time to determine the areas that bridge the gap between Continuous Integration for Continuous Delivery. The methodology used in this paper by exploring and tracking a project developed by the company using the Azure software development tool. The experiment includes a project performance and evaluation.**

Keywords: DevOps (Development Operations), CI/CD CI (Continues Integration), CD (Continuous Delivery), Azure, IAAC (Infrastructure as a Code)

I. INTRODUCTION

Recently, during the development process, the relationship between developer and operation is a big issue in a software system, combined concept DevOps, used to change incentives and share technologies throughout the entire development process. Development and operations must work together more effectively to address critical issues during the software development process. Developers are more interested in introducing new features or amendments to consumers quickly. While operations want greater reliability and security, they advise them not to make regular product changes. Operations cannot tolerate frequent new release controversies that may hamper software progress to bridge the gap between development and operations [1].

DevOps is a culture where (development and operations) teams join together in collaboration to deliver results on an ongoing basis. DevOps is used to change incentives and share technologies throughout the entire development process. Development processes and operations must work together more effectively to address critical issues during the software development process, developers are more interested in introducing new features or modifications to consumers quickly, while operations want more reliability and security, advising them not to make regular changes to the product. Processes cannot tolerate frequent new releases. Disagreements may hinder the progress of programs to bridge the gap between development and operations [2-3].

DevOps emerged to bridge the gap between software development using Azure and the deployment of these programs in production within a large software company. An important component of DevOps is the use of continuous software development processes such as continuous delivery, continuous Integration, and services to support the agile software development lifecycle [4].

Developers and operators can communicate and work together using a set of techniques called DevOps to provide software and services more rapidly, consistently, and of higher quality. DevOps is the sharing of tasks and responsibilities within a team that assumes full responsibility for their services and their core technology suite, from development to deployment, and supports the accelerated deployment and the accelerated speed with which businesses can serve customers without compromising the quality of their deliveries. [5].

Developers tend to shift their focus to developing complex systems rather than evaluating recent gaps. Errors may appear unexpectedly, leading the team to stop its current work from tasks and look at failures. Developers are working for it longer hours to meet these demands and stay ahead of the competition. Paradigm the goal of DevOps is to bridge the gap between software development and operations by merging best practices from both fields, fostering team cooperation, and aligning incentives and methods for processes and tools. [7].

DevOps capabilities include planning, integration, continuous testing, release, deployment, continuous infrastructure monitoring, optimization, collaboration, delivery, and continuous monitoring of user behavior and feedback [8].

Continuous Delivery (CD) is Continuous practice designed to help organizations accelerate growth, delivery, and product improvement designed to deliver new programs with higher speed and frequency features, Continuous integration (CI), is a testing mechanism that uses automated testing, application integration, and testing to solve problems with branches of applications in development, and deployment is automated software capabilities to rapidly develop and deploy high-quality software (Continuous Integration and Continuous Delivery) is a set of software practices and technologies that It allows for the frequent release of small sets of code changes that force organizations to automate building, testing, and deployment of applications and to bridge the gaps between the operational team and the development team as well as their activities throughout the SDLC lifecycle with broad visibility and traceability [7].

The goal of Industrial DevOps is to make development, updates, and service into continuous processes that operate continuously in tandem with system operation.

Azure is to measure the effect of CI/CD on the software development process that we use for projects Azure tool. Azure platforms are currently widely used in the sector. The primary usage of the Azure platform is to host the backend, apps, and corresponding deployments. With the aid of industry-leading technologies, Azure DevOps enables enterprises to manage the whole software development and deployment process efficiently and effectively. Agile-based software as a service (SaaS) platform Azure DevOps was introduced by Microsoft. A viable alternative for building the DevOps toolchain is Azure DevOps. [9].

In order to evaluate the efficacy and efficiency of CI and CD in project development, this paper offers an empirical investigation. The two methodologies will be evaluated in this study based on the two elements listed below. In order to achieve the input result, the primary contributions of this work are: 1) using the Azure tool to create a testing project.; 2) carrying out an empirical investigation to look into the following collection of research queries: **RQ1**: What are the CI/CD work studies in Azure DevOps that are most relevant? **RQ2**: What CI/CD strategies might be used to close the gap with Azure DevOps? **RQ3**: What are the drawbacks and advantages of CI/CD with Azure DevOps? **RQ4** How CI/CD in Azure DevOps is validated? The main objective of this paper is to conduct an empirical study on several projects developed by several companies in Jordan. The empirical techniques are based on the following two factors to estimate reducing the gap between CI and CD: i. reducing the development cost. ii. Reduce the development time. A set of experiments will be conducted to measure the reducing the gap according to these two factors.

II. RESEARCH BACKGROUND

The timing of releasing a high-quality software product or software service is crucial, and integrating a new approach is crucial to reaching this quality [10]. Software developed using a variety of integration and testing technologies is continuously integrated and tested as part of the DevOps process. [11]. DevOps is a continuous process and automation from development to operations, which means providing automation for the steps in between. One of the main obstacles slowing down DevOps deployments is the data gap. This wastes up to dozens of cumbersome wait times for data each week, slowing company growth and reducing competitiveness consisting of multiple tools and systems.

Software development processes can be made more efficient by using a combination of integration and automation technologies and techniques known as DevOps. DevOps is divided into two processes namely development and operation: The task of operations is to assure the daily seamless operation of a complex and occasionally highly interdependent portfolio of software, whereas the task of development is to create unique software solutions that match customer or user requirements. Because of technology developments in areas like automatic testing, deployment, and integration tools as well as growing cooperation between IT development and IT operation teams, the lines between development, deployment, and operations have grown less clear. This may result in numerous gaps, a lack of attention during development to non-functional needs, or challenges while trying to remedy programming faults in operating systems. The increasing interest in development and operations, or DevOps, stems from a desire to break down these established barriers and enhance collaboration between the two areas of an IT business. By coordinating incentives and exchanging methods for processes and tools, the DevOps paradigm promises to close the gaps. The paradigm will expand the application of agile methods in operations to promote teamwork and completely streamline the software delivery process. [12].

DevOps, fail often to get quick feedback and make faster deliveries, (CD) is a framework for developers to release software more quickly and receive feedback more quickly, assisting with faster builds and releases, (CI) is software development the methodology that developers take responsibility for consolidating code updates into one common repository, helps ensure parallel development and rapid integration the testing mechanism uses automated testing, and release deployment automatically. The main goal is to find and debug early in the product development lifecycle, reducing software release time, this need for speed in DevOps requires continuous integration (CI) and continuous delivery (CD) pipelines [7, 13]. DevOps is a collection of best practices that integrates software development (Dev) and IT operations (Ops) with the goal of reducing the time it takes to develop new systems and ensuring continuous delivery of high-quality software. This is one of the standard deployment strategies using agile methods for software engineering. Agile methodologies are used to deploy software in a single deployment [14].
The DevOps process is whenever a commit is made to code is automatically pulled and built. That is unit testing, code review, and code validation are done in code. This process is called Continuous Integration (CI). Code is then sent to the test servers to perform user acceptance testing through the process of Continuous Delivery (CD). The code is then tested by the server. Continuous integration and continuous delivery (CI/CD) form the backbone of DevOps. The CI/CD pipeline will be built using Azure DevOps services, and Azure App Service will be used to deploy to development, staging, and production. [15].
A tool for implementing agile methods and a mindset is the CI/CD pipeline. Iterative procedures are used in modern software products and services to provide value to the user. The goal of every software development project is the same: to provide a high-quality, production-ready release. Projects might be simple or complicated and have changing needs. The upcoming development team can concentrate more on the development work and less on maintaining the Azure demo application. [16].
Causing the data gap, first, the supply of test data slows down the pipeline, leading to increased latency and further widening data gaps. DevOps teams are often limited by the slow process of copying and moving data from production systems to test environments. Time to provision a data environment varies from four business days to over a week. During this time your competitors may overtake you by releasing multiple products to the market in the same period often, some DevOps teams use synthetic data to speed up production and close data gaps.

III. RELATED WORK

There has been numerous research done to close the gap between development and operations.
Yarlagadda concentrated on DevOps concepts and how to transition from continuous integration to continuous delivery. The information to be covered will be used to apply various DevOps concepts to various SDLC stages according to customer needs, as well as how to switch from continuous integration to continuous delivery, including various advantages.
Testing is one of the top goals for the development of DevOps because it has been shown that there are advantages to bridging the gap between CI and CDE. This shows that it is acknowledged that testing cannot be successful if only one part is automated rely on DevOps to comprehend the need for them to be aware of the gaps between CI and CDE. Speedy delivery of the same or better content is the goal of DevOps. Automation and group operation are important components. According to the DevOps philosophy, there should be no more resource waiting, quick failure, quick input, and quick delivery. A tremendous accomplishment will be made by the CI/CD working together perfectly. Channels for continuous delivery (CD) and continuous integration (CI) are required. Continuous integration (CI) supports speedy integration while CD advances development. The continuous approach provides a number of benefits, including increased and quicker customer and application development process input; timely and regular updates that increase consumer happiness and quality standards; and a stronger connection between development and operations [17].
With a focus on business requirements, techniques from continuous integration to continuous delivery, and their cost-saving benefits, Virmani attempts to study the DevOps aspects that are relevant to each stage of the SDLC. For example, getting any test setup only takes a few minutes of user action as opposed to

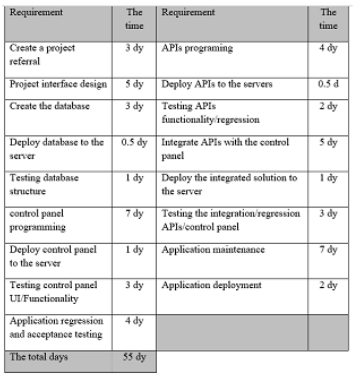
days without this automation. Cloud resources can be made available for usage by others after validation. Because resources are shared more effectively, dedicated hardware is no longer required. The complete automation may be applied to any setup (with small tweaks) because it is automated, and the test infrastructure provisioning is now always configured appropriately (no more weird unknown issues). Deployment turns into a reliable, repeatable process. The continuous deployment paradigm gives developers access to systems that are identical to those in use in production, allowing them to do validation in that environment. Resources don't need to be held in reserve because deployment can be automated in nearly no time. The morning can release the resources as soon as testing is complete. This article explains the continuous delivery revolution with a real-world case study of how to maintain infrastructure as code alone (IAAC), How to achieve convergence of the two primary components of business processing One being presented as a second automated tool is the issue. Being a cloud resource provider requires DevOps to transform not only processes but also culture. Companies that follow the DevOps principles will be at an advantage over those that don't. Processes must evolve over time as the market environment in which we operate continues to change. [8].

The possibilities to enhance continuous integration and continuous deployment in Azure DevOps for a controlled Microsoft NET environment utilizing various technologies and methods were introduced by Dileepkumar et al. We outline many approaches, methods, procedures, and procedures that are suitable for this investigation. Look for papers, tools, CI/CD techniques, and other resources. He thought about using several technologies and methodologies to enhance continuous integration and continuous deployment in Azure DevOps for a controlled Microsoft NET environment. Our suggested fixes primarily concentrate on cost-based management and configuration, architecture, development, technology, quality control, manual testing, test automation, and test coverage, which will assist in lowering costs, efforts, time, processes, and manual interventions throughout the SDLC application development and maintenance cycle. To be successful in Azure DevOps for Controlled Microsoft NET environment Continuous Integration and Continuous Deployment using various tools and methods. To establish the proper practices, it is crucial to concentrate on changing the current operational and corporate culture and securing support for procurement from the executive team. [18].
DevOps is described by Katal as an organizational and cultural shift that may be adopted into the environment. It can concentrate on the reasons why it is necessary, shifting accountability, putting trust in one another to work toward a common goal, locating bottlenecks, forming dynamic teams, altering one's way of thinking, rearranging more in the direction of services rather than pump and dump tactics, and delivering lasting values. Employee satisfaction increases with a DevOps approach, and systems have fewer flaws that can be corrected more quickly thanks to automation tools like source control, issue tracking, and management tools. DevOps As a result of the rise of cloud computing, the majority of IT business units are being outsourced by executives who don't see the benefit in investing in IT. The trust between the various departments of the company needs to be rebuilt, and executives need to understand that this will demand cultural and organizational transformation. DevOps must be included in the company culture in order for IT to create personalized software that can benefit other business divisions. [19].

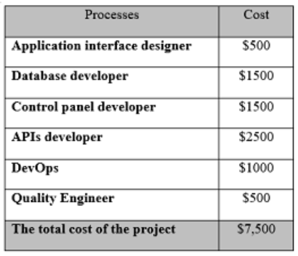
IV. CASE STUDIES.

The case study of the project describing the project for order delivery applications was taken from JFI express company in Jordan. The project taken is a custom order which contains several function requests from several departments. The requests are displayed to the delegates and the logistics company, and the representative is directed to deliver and update the request cases. Table 1 shows part of the requirement with time and cost of estimation.

TABLE I. FUNCTION TIME AND BUDGET ESTIMATION



Requirement	The time	Requirement	The time
Create a project referral	3 dy	APIs programing	4 dy
Project interface design	5 dy	Deploy APIs to the servers	0.5 d
Create the database	3 dy	Testing APIs functionality/regression	2 dy
Deploy database to the server	0.5 dy	Integrate APIs with the control panel	5 dy
Testing database structure	1 dy	Deploy the integrated solution to the server	1 dy
control panel programming	7 dy	Testing the integration/regression APIs/control panel	3 dy
Deploy control panel to the server	1 dy	Application maintenance	7 dy
Testing control panel UI/Functionality	3 dy	Application deployment	2 dy
Application regression and acceptance testing	4 dy		
The total days	55 dy		



Processes	Cost
Application interface designer	$500
Database developer	$1500
Control panel developer	$1500
APIs developer	$2500
DevOps	$1000
Quality Engineer	$500
The total cost of the project	$7,500

DevOps' primary objectives are to deliver more dependable and timely deliveries. Continuous Integration and Continuous Delivery have gained a lot of traction. Software engineers working on the project must integrate code into a shared repository as part of the Continuous Integration (CI) strategy. The build, unit tests, acceptance tests, and integration tests must all pass for the CI process to be successful. Prior to executing the CD process, it is crucial to ensure that the CI procedure was properly completed. A feature of continuous integration (CI), continuous delivery (CD) enables you to ship software more quickly. Continuous Testing leads to Continuous Delivery. The CI/CD pipeline incorporates automated testing.

Practices for continuous delivery your engineers have put in place pipelines and CI tooling that allow you to achieve this once you've determined that your code has to be continually delivered to reduce feedback loops. Continuous delivery is what we refer to as a collection of procedures that enables you to consistently and sustainably release new software changes of all kinds. Keeping the primary project in a deployable state at all times by ensuring that developer changes never break it. Before deploying your entire project to the production environment, you can use the process to ensure that it is in a stable, error-free state. All changes passing through the pipeline are validated thanks to the CI/CD pipeline. Here's the entire workflow (CI/CD) in a step-by-step manner:

1. The developer checks the source code and uploads it to the code repository
2. The Deploying build of the DevOps environment or code repository process the procedure (CI/CD) is Throw triggered by the Checks Configuration(Direct Automatically Deployment on DevOps stage-ready build)
3. All automated tests, including acceptance and unit tests, are executed.
4. If any of the tests are unsuccessful, the construction process abruptly ends.
5. The release procedure is only activated if the build process is successful.
6. The application is deployed if a release is successful and is used by stakeholders.

CI/CD you work automated. Come environment instead developer the environment work deployment run of the developer. Table 2 shows CI/CD process shortens the development time of the project and reduces the project cost. This process is an automated process done by the Azure tool.

Pipeline caching can assist reduce build time by allowing the outputs or downloaded dependencies from one run to be reused in later runs. We used to build time utilizing caching in the Azure pipeline. When the same dependencies must be downloaded again at the start of each run, caching is extremely helpful. The process frequently takes a long time and involves hundreds or thousands of network calls. Every time a new Pipelines Run starts, you are on an agent machine that has been created just for that run. Nothing is stored there apart from the binaries and services the CI/CD process needs. The spooler pipeline task is added to a pipeline. This task works like any other. Based on the inputs, the task will restore the cache. If no cache is located, the step is finished, and the job moves on to the following stage. A specific "save cache" step is automatically injected and run for each "restore cache" step that was not skipped once all stages in the job completed and the job has assumed a successful job status. The cache needs to be saved during this step.

TABLE II. REDUCED TIME AND BUDGET.

Requirement	The days	Requirement	The days
Create a project referral	3 days	APIs programing	4 days
Project interface design	5 days	Deploy APIs to the servers	Automated
Create the database	3 days	Testing APIs functionality/regression	Automated
Deploy database to the server	Automated	Integrate APIs with the control panel	5 days
Testing database structure	Automated	Deploy the integrated solution to the server	Automated
control panel programming	7 days	Testing the integration/ regression APIs/control panel	Automated
Deploy control panel to the server	Automated	Application maintenance	7 days
Testing control panel UI/Functionality	Automated	Application deployment	Automated
Application regression and acceptance testing	Automated		
The total days		34	

Processes	Cost
Application interface designer	$500
Database developer	$1500
Control panel devilloper	$1500
APIs developer	$2500
DevOps	$0
Quality Engineer	$0
The total cost of the project	$6,000

V. RESULT AND DISCUSSIONS

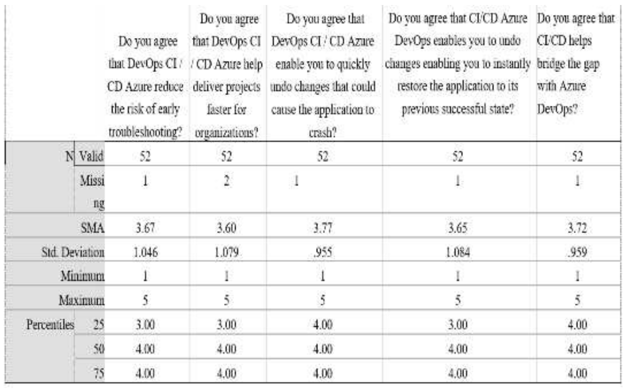
To answer the research questions, the procedure of the experiment is performed by three stakeholders namely, the developer, honor, and customer. The technique used includes a questionnaire and project performance evaluation. As shown in table 3, the developer's answer shows a clear gap reduction between Developers and Operation and Using DevOps Azure (CI/CD). Also the developer the result analyses highlight the uses, benefits, and challenges of DevOps.

TABLE III. THE DEVELOPER RESULT.

		Do you agree that CI/CD in DevOps is faster and easier for properly designed systems?	Do you agree that CI/CD error prevention in DevOps is failure prevention is the error released?	Do you agree that CI/CD DevOps facilitates system monitoring?	Do you agree that CI/CD DevOps integrate small code changes very quickly and code changes can be easily tested?	Do you agree that DevOps does a CI/CD that incorporates code changes?	Do you agree that DevOps easily performs CI/CD testing?	Do you agree that DevOps CI/CD detects errors as soon as they occur in the system?
N	Valid	52	52	52	52	52	52	52
	Missing	1	1	1	1	3	1	1
SMA		3.87	3.89	3.69	3.78	3.68	3.75	3.83
Std. Deviation		.979	.893	1.056	.955	1.084	.958	.971
Minimum		1	1	1	1	1	1	1
Maximum		5	5	5	5	5	5	5
Percentiles	25	4.00	4.00	3.00	4.00	3.00	4.00	4.00
	50	4.00	4.00	4.00	4.00	4.00	4.00	4.00
	75	4.00	4.00	4.00	4.00	4.00	4.00	4.00

According to the results in the above table the mean of the first question was high (3.87 out of 4) and question 3 as well (3.89 out of 4), followed by the rest of the questions the answers were high except for question 2 because it is related to the correction factor. Overall the developer results indicate a high rating. You can integrate tiny chunks of code at a time using continuous integration and continuous delivery. These code modifications have fewer problems that might need to be fixed later because they are simpler and easier to handle than large chunks. As soon as they are incorporated into the code source, small sections can be tested. This enables developers to identify a problem before too much further work is required to fix it. Failures are more quickly identified and remedied. Only if the code is developed in a constantly moving system are frequent releases feasible. After comprehensive testing, CI/CD constantly merges codes and releases them to production, keeping the code in a release-ready state. Testing the code is made much easier by containerization. It will only test the parts of the production environment that will be impacted by the release.

TABLE IV. THE HONOR RESULT



		Do you agree that DevOps CI / CD Azure reduce the risk of early troubleshooting?	Do you agree that DevOps CI / CD Azure help deliver projects faster for organizations?	Do you agree that DevOps CI / CD Azure enable you to quickly undo changes that could cause the application to crash?	Do you agree that CI/CD Azure DevOps enables you to undo changes enabling you to instantly restore the application to its previous successful state?	Do you agree that CI/CD helps bridge the gap with Azure DevOps?
N	Valid	52	52	52	52	52
	Missing	1	2	1	1	1
SMA		3.67	3.60	3.77	3.65	3.72
Std. Deviation		1.046	1.079	.955	1.084	.959
Minimum		1	1	1	1	1
Maximum		5	5	5	5	5
Percentiles	25	3.00	3.00	4.00	3.00	4.00
	50	4.00	4.00	4.00	4.00	4.00
	75	4.00	4.00	4.00	4.00	4.00

The results in table 4 indicate that the user's assessment of the right to access information is high, only the arithmetic averages of the three questions in this axis were high (3.67,3.60,3.77)**.**The number of mistakes that can happen during repetitive processes is decreased through automation in the CI/CD pipeline.

Additionally, by catching the problem as soon as possible, less code modifications will need to be made in the future to repair it, which frees up developer time that may be used for product development. Automation can lower code quality while increasing return on investment. Increasing the quality of your code is another excellent technique to raise ROI.

CI/CD is a fantastic technique to receive ongoing consumer feedback. This emphasizes responsible accountability and increases the transparency of any team issues. Since the development team is the primary target of CI, input from this section of the pipeline has an impact on build failures, merging issues, and architectural setbacks. To obtain crucial consumer input, CD places a greater emphasis on getting the product to end customers rapidly. You may improve your product consistently and continually by using both CI and CD, which both offer quick feedback. CI/CD Constant dependability. Due to system upgrades, testing reliability has improved, enabling more precise positive and negative tests to be performed. Stakeholders are ensured by the constant blending and launching of new features and products.

Transformations to DevOps don't take place instantly. Many businesses begin with a modest application or pilot project so that employees can experience new procedures and resources. Consider adopting DevOps on a broad scale incrementally. By requiring automation in the development, testing, and deployment of applications, CI/CD fills the gaps between development and operation activities and teams.

Continuous development, testing, integration, continuous deployment, and monitoring of software applications are all part of modern DevOps methods.

Today's activities are based on CI/CD pipeline. Prior to production, minor flaws are found and addressed before being made available to end consumers. The amount of non-critical faults is decreased in your organization's development process by implementing CI/CD.

TABLE V. THE CUSTOMER'S RESULT.

		Do you agree that Azure DevOps is able to adapt to the needs and scope of each team while facilitating uninterrupted collaboration?	Do you agree that Azure DevOps has met your needs?	Do you agree that Azure DevOps has reduced gaps in the system?
N	Valid	52	52	52
	Missing	2	1	1
SMA		3.85	3.60	3.64
Std. Deviation		.901	1.087	.994
Minimum		1	1	1
Maximum		5	5	5
Percentiles	25	4.00	3.00	3.00
	50	4.00	4.00	4.00
	75	4.00	4.00	4.00

According to table 3, all the arithmetic averages of the questions in this axis are high only; they were 3.85 for the first question, 3.60 for the second question, and 3.64 for the third and last question. The arithmetic mean of this axis is high, which means a high evaluation of customers Adoption issues often arises during the implementation stages of a project. The product owner, work stream leads, and stakeholders view deliverables after it has been built by the development team. Azure DevOps allows the project team to work together to provide a deliverable that meets the customer's needs and that is free of defects. Azure DevOps uses the below elements to improve stakeholder engagement .Delivery teams can improve the likelihood of the system being adopted. Product owners or stakeholders can state the priority of items and the order in which they wish for them to be developed. They can be created manually or via Azure DevOps's team integration with Microsoft Project. DevOps can adapt to team needs while facilitating collaboration and this demonstrates the power of collaboration between the development team and the process. Reduced system gaps resulted from customer satisfaction and adaptation.

The benefits of CI/CD are found not only in the technical area but also in the organizational scope. Using a CI/CD strategy maintains your product up to date with the newest technology and enables you to attract new clients who pick you over the competition due to positive product reviews and word-of-mouth advertising. Whether the feedback is favorable or unfavorable, it always results in improvements to usability and general client satisfaction. You can keep your current users by implementing updates to the CI/CD pipeline and adding new functionality. You may attract new customers based on how your current clients use the product. The primary users of the product are consumers. Consideration should be given to what they say. Whether the feedback is favorable or unfavorable, it always improves usability and raises consumer satisfaction levels. You'll be able to attract new users and keep your current ones if you update the CI/CD pipeline and add new features based on how people interact with the product. DevOps CI/CD has led to the following outcomes:

1. Benefits of DevOps: DevOps's goal is to get the best investment while ensuring the quality of programs and meeting customer demands, DevOps Eliminates organizational and cultural problems through the Integration of development and operations to allow Frequent and fast software releases. Making the DevOps process run more smoothly because when bugs are discovered early on in software development, they are simpler to correct and have less of an influence on the final result [1]. When CD and CI work together to sustain continuous development and quick integration, the result is a reduction in the average lead time. CI/CD Establishing suitable quality gateways in the development and testing process will allow for faster fault identification and resolution. Bugs are fixed early in the development cycle thanks to quality assurance, deployable applications, and quick feedback loops to the engineers.
2. Present a challenge to DevOps: A fundamental shift in organizational culture and structure is needed for DevOps. Many organizations, especially those that are bigger and more conventional in their approach, may struggle with this. It might be challenging to accomplish the strong communication and coordination that DevOps calls for between the development and operations teams. DevOps prioritizes speed above security when building software. Keep track of the security breaches during the integration of your cloud services in your business organization.
3. Technical Challenges: There are various technological difficulties with DevOps. The requirement to standardize and automate tools and processes throughout the entire organization is one of the biggest. This can be challenging, especially in companies with a lot of platforms and apps. Adding DevOps to a current system without sacrificing security or performance is another hurdle.
4. Overcoming the dev versus ops attitude in relation to operational challenges and challenges facing developers. Many businesses still use the tired trope of developers throwing code over a fictitious wall to a central operations team. The operation team is working hard to maintain excellent service levels while developers are attempting to innovate and make changes as rapidly as feasible. The goals of these two groups frequently conflict with one another, creating friction points that lead to handovers, higher expenses, and longer feedback loops. Integrating teams and dismantling silos inside IT businesses are the two main tenets of DevOps. Setting out a vision for how this will function for your organization is the first step in this journey. Any organization would do well to begin by understanding the current division of labor between development and operations, as well as how these two functions may be most effectively combined. It frequently faces this challenge as it begins to implement DevOps techniques.

VI. CONCLUSION

Enterprises can plan more intelligently, collaborate more quickly, and deliver better software thanks to DevOps. Together, continuous integration and continuous delivery enable your company to produce high-quality software with shorter time-to-market cycles. The software delivery process can be automated with the help of both CI and CD. Additionally, the automatic Pipeline setup makes it simple to test, create, and deploy the application. The software delivery process can be automated with the help of both CI and CD. The frequency of source code commits determines everything. Using the Azure tool to analyze the impact study of continuous integration (CI) and continuous delivery (CD) deployment in DevOps The combination of continuous integration and continuous delivery can help your company create excellent software with quicker time-to-market releases. The result of our experiment show CI/CD reduces the gap between the developer and the operator. The results indicated that there are some hurdles to overcome, and the gaps between CI and CD, but it is worth following the DevOps methodology when adopting DevOps is faster and less costly than the application and customer development process.

References

[1]. Battina, D.S., 2021. The Challenges and Mitigation Strategies of Using DevOps during Software Development. *International Journal of Creative Research Thoughts (IJCRT), ISSN*, pp.2320-2882.

[2]. Ferdian, S., Kandaga, T., Widjaja, A., Toba, H., Joshua, R. and Narabel, J., 2021.Continuous Integration and Continuous Delivery Platform Development of Software Engineering and Software Project Management in Higher Education. Jurnal Teknik Informatika dan Sistem Informasi, 7(1).

[3]. Battina, D.S., 2021. Improving La Redoute's CI/CD Pipeline and DevOps Processes by Applying Machine Learning Techniques. International Journal of Emerging Technologies and Innovative Research pp.2349-5162.

[5]. Mali Senapathi, Jim Buchan, Hady Osman."DevOps Capabilities, Practices, and Challenges: Insights from a Case Study." Conference: the 22nd International Conference,2018 ,pp 1-11 https://dl.acm.org/doi/abs/10.1145/3210459.3210465

[6]. Alok Mishra a,b,∗ , Ziadoon Otaiwi b.(2020, Nov)."DevOps and software quality: A systematic mapping."ScienceDirect. [Online].v(38),pp.1-14.Available: https://www.sciencedirect.com/science/article/pii/S1574013720304081?via%3Dihub [Nov.2020]

[7]. Ugarte Querejeta, M., Etxeberria, L. and Sagardui, G., 2020, September. Towards a DevOps approach in cyber-physical production systems using digital twins. In *International Conference on Computer Safety, Reliability, and Security* (pp. 205-216). Springer, Cham.

[8]. Virmani, M., 2015, May. Understanding DevOps & bridging the gap from continuous integration to continuous delivery. In *Fifth international conference on innovative computing technology (intech 2015)* (pp. 78-82). IEEE. https://ieeexplore.ieee.org/abstract/document/7173368/

[9]. van Son, J., Visser, J. and Poll, E., 2020. Security by design in Azure DevOps pipelines, a case study at SpendLab technology (Doctoral dissertation, Bachelor's thesis, Radboud University).

[10]. Fawareh H., Al-Bardeen, A. "An Investigation Of E-learning Websites Usability Issues during the COVID-19 Pandemic: Jordan case study", 2021 22nd International Arab Conference on Information Technology, ACIT 2021, 2021

[11]. Gawade, S. and Ramteke, V., 2021, July. Assessment of the Impact of the Hybrid Software Development Approach. In Journal of Physics: Conference Series (Vol. 1964, No. 4, p. 042009). IOP Publishing.

[12]. Nielsen, P.A., Winkler, T.J. and Nørbjerg, J., 2017. Closing the IT development-operations gap: The DevOps knowledge sharing framework. In *BIR Workshops*.

[13]. Khan, M.O., Jumani, A.K. and Farhan, W.A., 2020. Fast delivery, continuous builds, testing, and deployment with DevOps pipeline techniques on the cloud. *Indian Journal of Science and Technology*, *13*(05), pp.552-575.

[14]. Sethi, F., 2020. Automating software code deployment using continuous integration and continuous delivery pipeline for business intelligence solutions. *Authorea Preprints*.

[15]. Satapathi, A. and Mishra, A., 2021. Deploying Your Azure Functions Using a CI/CD Pipeline with Azure DevOps. In *Hands-on Azure Functions with C#* (pp. 373-399). Apress, Berkeley, CA.

[16]. Saarenpää, J., 2020. Creating an Azure CI/CD pipeline for a React web application.

[17]. Yarlagadda, R.T., 2018. Understanding DevOps & bridging the gap from continuous integration to continuous delivery. *Understanding DevOps & Bridging the Gap from Continuous Integration to Continuous Delivery, International Journal of Emerging Technologies and Innovative Research (www. jetir. org), ISSN*, pp.2349-5162. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3807611

[18]. Dileepkumar, S.R., and Mathew, J., 2021, February. Optimize Continuous Integration and Continuous Deployment in Azure DevOps for a controlled Microsoft. NET environment using different techniques and practices. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1085, No. 1, p. 012027). IOP Publishing.

[19]. Katal, A., Bajoria, V. and Dahiya, S., 2019, March. DevOps: Bridging the gap between Development and Operations. In *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 1-7). IEEE.

[20]. Fawareh, H. "Software quality model for maintenance software purposes" International Journal of Engineering Research and Technology, 2020, 13(1), pp. 158–162

[21] Ibrahim Assi, Rami Tailakh, and Abdelsalam Sayyad, "Survey on Software Changes: Reasons and Remedies" International Arab Journal of Information Technology, IAJIT, Vol. 18, No. 2, 2021.