APPLYING AUTOMATION TECHNIQUES IN AGILE SOFTWARE DEVELOPMENT

SUBMITTED BY

QURRAT UL AIN BROHI (SP21-MSSE-0015)

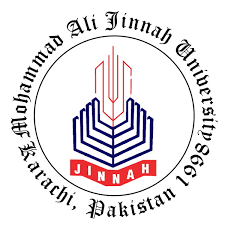
SUPERVISED BY

SYED SARMAD ALI



RESEARCH PROJECT SUBMITTED TO THE FACULTY OF COMPUTING, MOHAMMAD ALI JINNAH UNIVERSITY, IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE/ SOFTWARE ENGINEERING

SPRING 2022



**RESEARCH CERTIFICATE**

*This is to certify that the research titled, “\_\_Applying automation techniques in agile software development ”, is submitted to the Department of Computer Science, Fall 2021, by Qurrat ul Ain Brohi for the award of the degree of Master of Science in the discipline of Software Engineering. The research has been carried out under my supervision. I certify that the work submitted is original and not plagiarized from any other source, except as specified in the references. Neither the research nor the work contained therein has been previously submitted to any other institution for a degree.*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Supervisor’s name and signature

**MOHAMMAD ALI JINNAH UNIVERSITY**

# ORIGINAL LITERARY WORK DECLARATION

|  |  |  |
| --- | --- | --- |
| **PROGRAM** | MS software Engineering | |
| Student Name: Qurrat ul Ain Brohi | | Reg. No:SP21-MSSE-0015 |
| Email: sp21msse0015@maju.edu.pk | | Mobile No: |

Research Title: *Applying automation techniques in agile software development*

|  |
| --- |
| I do solemnly and sincerely declare that:  1. I am the author of this work;  2. This work is original;  3. Use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work; I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;  4. I hereby assign all and every rights in the copyright to this work to Mohammad Ali Jinnah University (MAJU), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of MAJU having been first had and obtained;  5. I am fully aware that if in the course of making this work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by MAJU. |

Student’s Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Place: Karachi-Pakistan.

Acknowledgments

First, praise to Almighty ALLAH for all his support and help in every stage of concluding this project. I would like to take my sincere and heartiest gratitude to Sir Syed Sarmad Ali for providing me this opportunity for my research project. I am highly obliged for his continued support and encouragement. I would not be able to complete it without sharing expertise, and knowledge. Besides, expressing my pure appreciation to my research advisor, I would like to pay a huge thanks to my parents, my siblings for being an endless support and guide for me because without their assistance I would have never been at this stage of life. After them there are many other people to whom I wish to dedicate my work who are always there for me to resolve my queries with regards to any work of mine and they also helped me in completing the project successfully.

Executive Summary

In this paper, I research how using automation techniques in agile software development benefits the software business. A growing number of businesses are failing to meet these growing challenges and are looking for improvements in software development. In today's software business, agile is the most popular software development strategy. Agile offers a variety of methods from which organizations can choose the most appropriate method for implementing agility. So I analyze that agile practices combined with a continuous integration and continuous delivery (CICD) pipeline approach have increased project efficiency by building the solution and running a set of automated tests each time a change is committed. A ci/cd pipeline also provides rapid feedback to developers about their changes. The CICD pipeline eliminates the need for manual process execution and automates the entire process. Using the CICD pipeline approach, I've also looked at how automation works better than manual process execution. I also go over the three automation phases of the CICD pipeline: benchmark, load test, and scaling. By including this three-phase strategy as a CICD automation enhanced feature, I can lessen the risk of application deployment. Due to simulated traffic patterns that are greatly diverged from production, typical load testing approaches are unable to identify production performance behavior. To address these challenges, our technique has added three automated phases to the CICD pipeline: benchmark, load test, and scaling. It uses the production traffic for load testing, which delivers more accurate results, and it avoids system interruption by using the test bench approach while system benchmarking. System scaling can be reviewed after the benchmark and load test phases are done, and the research provides an effective technique to manage Agile-based CICD projects.

The goal is to avoid laying weak foundations and keep the code in a releasable form at all times. Not only is it more efficient to handle problems as soon as they arise, but it also allows for a speedy correction if something goes wrong in production.

Given its growing importance, I've also discussed DevOps. The terms DevOps and Agile are much a part of the world of software development that they are often confused by people in the industry. . Although both refer to software development methods, they are two different words that end up coming together when putting them into practice together. This study’s major goal is to cover CICD methodologies (automation techniques) with new feature enhancement as well as agile methodology.

Keywords— continuous integration; continuous delivery; virtual Agent agile project management; Automation, software engineering

# Table of Contents

[ORIGINAL LITERARY WORK DECLARATION](#_heading=h.30j0zll) **3**

[Table of Contents](#_heading=h.1fob9te) **3**

[List of Tables](#_heading=h.3znysh7) **4**

[List of Figures](#_heading=h.2et92p0) **5**

[**CHAPTER 1**](#_heading=h.5bgijtajmyhk) **7**

[Introduction And Motivation](#_heading=h.tyjcwt) **7**

[1.1 Values of the Agile Development Cycle](#_heading=h.3dy6vkm) [9](#_heading=h.3dy6vkm)

[1.1.1 Individuals and interactions over processes and tools](#_heading=h.1t3h5sf) [9](#_heading=h.1t3h5sf)

[1.1.2 Working product over comprehensive documentation](#_heading=h.4d34og8)  [9](#_heading=h.4d34og8)

[1.1.3 Customer collaboration over contract negotiation](#_heading=h.2s8eyo1) *[9](#_heading=h.2s8eyo1)*

[1.1.4 Responding to change over following a plan](#_heading=h.ek0bqq4k81gb) [9](#_heading=h.ek0bqq4k81gb)

[1.2 Continuous Integration](#_heading=h.l1d4kfk49ox8) [11](#_heading=h.l1d4kfk49ox8)

[1.3 Continuous Delivery](#_heading=h.yjqhk9qey0gn) [12](#_heading=h.yjqhk9qey0gn)

[1.4 Continuous deployment](#_heading=h.gbxiifikxvew) [12](#_heading=h.gbxiifikxvew)

[**CHAPTER 2**](#_heading=h.8nqkiaigy1qh) **13**

[**LITERATURE REVIEW**](#_heading=h.1pyr3l27a8ft) **13**

[2.1 LITERATURE REVIEW](#_heading=h.w7qqctj6iimp) [13](#_heading=h.w7qqctj6iimp)

[**CHAPTER 3**](#_heading=h.3wsx2wsv6s58) **19**

[METHODOLOGY](#_heading=h.mmr460z7prn2) [19](#_heading=h.mmr460z7prn2)

[3.1 PROBLEM STATEMENT](#_heading=h.6o2hk2t9g4je) [19](#_heading=h.6o2hk2t9g4je)

[3.2 RESEARCH QUESTION](#_heading=h.e5ihfzcuqhzi) [19](#_heading=h.e5ihfzcuqhzi)

[3.2.1 RQ 1: How we can implement CI CD process in agile software development.](#_heading=h.zglbi712g6fk) [19](#_heading=h.zglbi712g6fk)

[3.2.2 RQ 2: What is the Big challenge that Occur in CICD pipeline automation process.](#_heading=h.kwsk4rmgur3) [20](#_heading=h.kwsk4rmgur3)

[3.3 GAP IDENTIFICATION](#_heading=h.5nqt1duuthhz) [20](#_heading=h.5nqt1duuthhz)

[3.4 PROPOSED SOLUTION](#_heading=h.tvjna2vxgnk3) [21](#_heading=h.tvjna2vxgnk3)

[3.4.1 CI CD METHODOLOGY DIAGRAM](#_heading=h.bdh8ta2jgo2i) [22](#_heading=h.bdh8ta2jgo2i)

[3.4.2 VIRTUAL AGENT](#_heading=h.wmmd6irr59q2) [23](#_heading=h.wmmd6irr59q2)

[3.4.3 Integration of virtual agent in CICD pipeline](#_heading=h.sptgc9r9gikh) [24](#_heading=h.sptgc9r9gikh)

[**CHAPTER 4**](#_heading=h.jgkrsgpiu28r) [26](#_heading=h.jgkrsgpiu28r)

[RESULT AND DISCUSSION](#_heading=h.il4vibn8gc0m) [26](#_heading=h.il4vibn8gc0m)

[**CHAPTER 5**](#_heading=h.c53mbavkhggv) **29**

[Conclusion](#_heading=h.rmarfo5vkn60) [29](#_heading=h.rmarfo5vkn60)

[**References**](#_heading=h.1iqs3jnxwks3) **30**

[PLAGIARISM REPORT](#_heading=h.35nkun2) **33**

# List of Tables

[Table 1 COMPARISON OF CICD TOOLS 28](#_Toc104853755)

# List of Figures

[Figure 1. Agile software development life cycle 10](#_Toc104853470)

[Figure 2.Continuous integration 11](#_Toc104853471)

[Figure 3.Continuous Deployment 12](#_Toc104853472)

[Figure 4.The deployment management cycle with CICD. 22](#_Toc104853473)

[Figure 5. Integration of virtual agent in the deployment management cycle with CICD 24](#_Toc104853474)

APPLYING AUTOMATION TECHNIQUES IN AGILE PROJECT DEVELOPMENT

QURRAT UL AIN BROHI

Mohammad Ali Jinnah University

Author Note

Department of Computer Science, 22-E, Block-6, P.E.C.H.S., Lal Kothi Stop, Main Shahrah-e-Faisal, Karachi, 75400, Sindh, Pakistan. Email ID: sp21msse0015@maju.edu.pk

# CHAPTER 1

# Introduction And Motivation

Many software development models and approaches have arisen as a result of the progressive growth of software engineering disciplines, including agile software development methods, which are now widely utilized. On the basis of the benefits of agile development approaches, agile software development provides software engineers with increased confidence because it leads to efficient software development, high-quality deliverable software. Agile software development ensures that software engineers remain committed to the most valuable software features while adapting to turbulent requirements changes, and it also provides a high return on investment for the business.

Agile refers to software development methods that emphasise incremental delivery, team communication, continuous planning, and continuous learning rather than trying to release everything at once near the end.Agile emphasizes keeping the process lean and developing minimum viable products (MVPs) that go through multiple iterations before being released. Feedback is constantly received and applied, resulting in a much more dynamic process in which everyone is working toward a common goal.

In this paper, we research how using automation techniques in agile software development benefits the software business. A growing number of businesses are failing to meet these growing challenges and are looking for improvements in software development.The improved product issue necessitates faster coding, which necessitates expedited testing. When a new app ( or an update to an existing app) is released to the market, it must be fully functional for all users and simple to use for all users. Clients will simply switch to a competitor and write a barrage of negative reviews, resulting in the programme losing credibility. The only way to guarantee flawless operation is to properly test and improve each feature. It’s hard to accomplish it manually for complex, feature-rich software under tight deadlines.Continuous integration and deployment (CI/CD) in modern agile software development environments necessitates multiple tests in short sprints [6]. Traditional software development approaches are no longer sufficient to meet today’s business needs. Software development firms are attracted to Agile approaches because they provide flexibility, efficiency, and speed in the Software Development Life Cycle (SDLC). Given its growing importance, I have discussed how to use CICD automation pipeline methodology for deployment process, as well as the relationship between DevOps and agile and continuous software development delivery methodologies in this research study. Having a clear division between QA and Dev teams proved wasteful as agile developers became more successful at producing and deploying code. The DevOps technique was created to address this issue. Implementing the CICD pipeline on agile has resulted in faster software delivery and increased productivity.[1] The key advantages of CI approaches are that they reduce risk and provide bug-free and dependable software, lowering the barrier to frequent delivery.DevOps may be thought of as an extension of Agile approach .[2]Agile emphasizes cross functional teams, which often include designers, testers, and developers. According to Drury et al., DevOps goes a step further by providing an operator who can help with the transition from development to implementation. [6] DevOps can help to automate procedures and promote transparency for all teams because of the natural communication between DevOps and other teams.Having a clear division between QA and Dev teams proved wasteful as agile developers became more successful at producing and deploying code. [3] The DevOps technique was created to address this issue

## 1.1 Values of the Agile Development Cycle

The iterative approach of the agile method is ultimately about a set of values, which were popularized in The Agile Manifesto. All agile practices are guided by these values.

### 1.1.1 Individuals and interactions over processes and tools (Put People First)

People come first in the Agile lifecycle, not processes or tools. This means that collaboration takes precedence over a long document that no one will read.

### 1.1.2 Working product over comprehensive documentation (Working product trumps thorough documentation.)

Although this may seem self-evident, how frequently does the software development lifecycle result in excessive documentation?

Yes, having some documentation to explain how the product was produced and how to use it is beneficial, but the ultimate goal is to deliver working software.

This can be accomplished through excellent design and a focus on technical excellence.

#### 

### *1.1.3* Customer collaboration over contract negotiation (Collaboration with customers is more important than a contract.)

Customers must be included at all phases of the project according to agile principles. Customers can only negotiate before and after the project with the waterfall technique or traditional methodologies. This used to result in both time and resource waste. Team members may ensure that the final product satisfies all of the client's needs if consumers are kept informed throughout the development process.

### *1.1.4* Responding to change over following a plan (Responding to change with a strategy)

An Agile approach is fluid rather than linear. Its adaptability allows it to change course as needed without having to undo weeks of work.

It also implies that team members must be willing to adjust requirements at any point during the process.

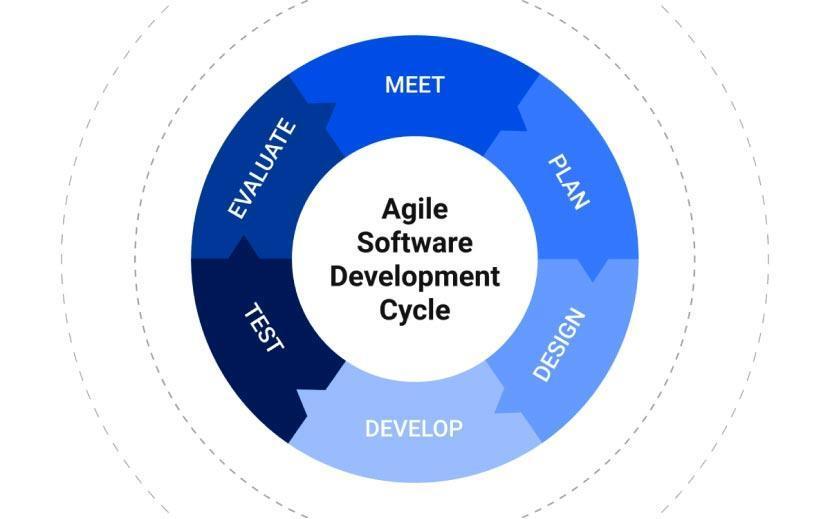


Figure . Agile software development life cycle

## 1.2 Continuous Integration

The creation and distribution of Internet software has resulted in a set of standard procedures, the most important of which is continuous integration (CI).

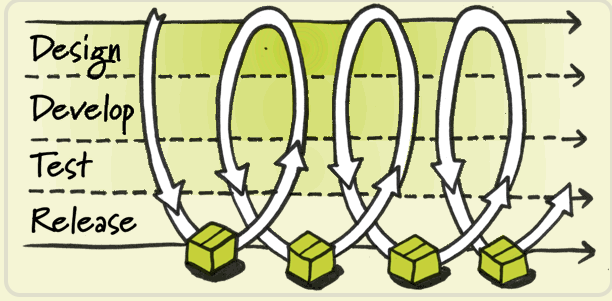


Figure .Continuous integration

Continuous integration is the process of frequently integrating code into the trunk (multiple times a day).

Its benefits are mainly two.

* Find faults quickly. Every time a minor update is done, it is merged into the trunk, allowing for faster error detection and easier error location.
* Keep branches from diverging too far from the stem. It will become more difficult to integrate in the future, or possibly impossible to integrate, if it is not integrated frequently and the backbone is constantly updated..

Continuous integration allows products to iterate quickly while retaining a high level of quality. Its primary criterion is that code must pass automated tests before being included into the trunk. It is impossible to integrate a test case that fails.

"Continuous integration doesn't remove bugs; it makes them very easy to discover and cure," observed Martin Fowler.

## 1.3 Continuous Delivery

Continuous delivery is the process of regularly distributing new software versions to quality teams or users for approval. If the review succeeds, the code is moved to production.

After Continuous Integration, Continuous Delivery is the next natural step. It emphasizes the fact that software can be distributed at any time, from any location, and in any format.

## 1.4 Continuous deployment

The next stage after continuous delivery is continuous deployment, which refers to the automatic deployment of code to the production environment when it has passed review. Continuous deployment aims to ensure that code can be deployed and put into production at any time. The testing, building, and deploying operations are all automated in continuous deployment. The difference between it and continuous delivery is depicted in the diagram below.

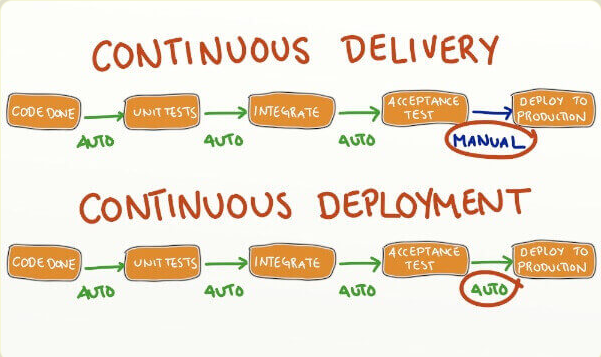


Figure .Continuous Deployment

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 LITERATURE REVIEW

Ongoing integration, according to Childs et al., necessitates constant testing because the purpose is to offer consumers with reliable software and code. Continuous testing is used to implement the automatic regression, output, and other tests that are done in the CI pipeline. A significant principle in the first phase of Agile software development models’ innovation is to iterate configuration changes more quickly and decide on the best course of action through experimentation – in theory, to ”fail fast” and optimise consistency as a core project goal. [3]A developer’s inability to anticipate long-term programme demands and adequately specify long-term project requirements at the start is due to a lack of understanding and failing to predict a client’s shifting needs .Agile approaches also enabled full-time cooperation with client partners’ production teams, providing in-house, real-time insight into consumer preferences and needs in order to develop an iterative, failfast workflow .Agile approaches have resulted in a real-time continuous loop between customer subject matter experts and software development teams. DevOps builds on the real-time feedback loop notion and extends it to various stages in the SDLC development process, minimizing the risks of disconnects between developers, QA, operators, and developer-to software disconnects.[2]. According to author Naeem Ahmed khan in 2017, the old automation techniques are not enough to produce quality software products so look for the better result which is AI based testing techniques [4]. CICD has improved the system’s productivity and implemented a more efficient agile delivery process. [5]. S.A.I.B.S. Arachchi, Indika Perera describe the deployment management cycle with CICD. An qualitative analysis was done by Sikender Mohsienuddin Mohammad, Agile approach was created to overcome the challenges that the traditional model had and to provide project teams with a variety of options during the development process. Agile and DevOps approaches are changing software development methodologies, and as a result, their application development is handled in short time (sprints) and other involving team members are adjusted.[2] . The combination of Agile and DevOps is always the best strategy to effect change in a team, department, or organization. Accepting these theories necessitates being adaptable to how they change over time and understanding that no single solution can address all organizational needs. Julen khales, present the framework for combining combining expert knowledge into machine learning algorithms. author look at how to build a feature extraction procedure, how to define ground-truth categories, and how to evaluate machine learning algorithms.[7]. In 2015 Lianping Chen attempt a work to identify such a CD environment, explain why we need to plan for CD, detail the implications of architecting for CD, and examine the issues that this new context presents.[8]. Ravi Teja Yarlagad da describe the benefits of devops and he also mentioned that how devOps are playing important role in the IT industry.[9] The term “continuous delivery” (CD) refers to the end result. CD capability, as well as studies on the several advantages and there are several difficulties to overcome. This information can be beneficial. Practitioners should think about how they’re going to implement CDs and how they’re going to do it the research agendas of researchers are created.[10]

According to Mojtaba Shahina , Muhammad Ali Babara , Liming Zhub in their research paper which title CI, CD and CD: A Systematic” The goal of this study was to comprehensively analyse thestate-of-the-art of continuous techniques in order to classify approaches and tools, identify difficulties and techniques, and indicate research needs in the future

Between 2004 and June 1st, peer-reviewed publications on continuous practices were published. 2016 were reviewed using the systematic literature review (SLR) technique by the author of this work. We used the theme analysis method to analyse the data gathered from analysing 69 papers that were chosen based on preset criteria.

The findings are based on a combination of data taken directly from the peer-reviewed studies and our limited interpretations.

There are various issues and gaps that will necessitate further research: enhancing the capture and reporting of contextual data in studies that report on various aspects of continuous practices;

According to Sikender Mohsienuddin Mohammad in 2017 on his research `DevOps automation and Agile methodology” he analyzed that This study will be focused on a qualitative investigation of DevOps and agile methodology replies. The primary goal will be to undertake a literature assessment of what other researchers have discovered about DevOps automation and Agile methodology, as well as their impact on the present information technology industry. With this in mind, DevOps contributes to a better knowledge of information system efficiency.

In this paper the author describes the difference between devops and agile culture .The findings are intended to demonstrate the differences between agile and DevOps methodologies.

In future,there has been relatively little documentation of the Organizational Process Improvement Program in the literature on how Agile techniques interact with progress. Corporate business models, concurrent design, multisectoral management, and proactive development, for example, can affect future technological innovation.

According to Yasmine SKA, Janani P in year 2019, on his research “A study and analysis of continuous Delivery, continuous integration In software development A study of environmental science” he describes that The advantages and features of employing continuous delivery and continuous integration (CI/CD) are discussed in this paper. Also looks into the effects of (CD/CI). Continuous delivery (CD) is a collection of methods aimed at streamlining the transition from version control to production or release to manufacturing. build and test process are all important components.Comprehensive version control, automation of the test and deployment processes, and the use of continuous integration to quickly evaluate the accuracy of every change by executing the automated

This study's findings are based on an examination of CI and CD in development of software .In future we need to work more in CI/DC to grape the attention n market and try to cope up with their requirements

According to NAEEM AHMED KHAN in year 2017 , the topic of his research is Old Testing Automation techniques are lagging: Artificial Intelligence has the pace where The author of this paper describe that old automation techniques are not enough to produce quality software

The author of this paper addressed that the high Script Maintenance Requirement and high Skill Requirements are the problem with automation techniques

Author has present the automation techniques based on AI.they also address the issue with traditional techniques. and they try to prove that artificial intelligence has enough potential to overcome the issues which occurs in traditional techniques

We've started applying basic AI techniques in software testing, but we need to push it to its maximum capacity to satisfy future software quality and scheduling needs, which we plan to achieve in the near future.

According to S.A.I.B.S. Arachchi, Indika Perera in 2018, the research topic is “Continuous Integration and Continuous Delivery Pipeline Automation for Agile Software Project Management” he describe

The authors of this paper describe how to understand the CICD pipeline by analyzing existing methods, providing a method for performing incremental load tests with minimum system impact utilising production traffic, defining a scalability factor to keep the system running smoothly consistency for the most recent production release, automating the By incorporating the process of benchmark and recommended load test method further into CICD pipeline, and measuring system performance with monitoring tools, system performance may be evaluated.

According to Sikender Mohsienuddin Mohammad in year 2016 on research “continous Integration and automation” his paper examines the concept of Continuous Integration and Automation, as well as how it affects various platforms and technologies.

According to a study published in 2019 by Julien Khales, Juha Torronen, Timo Huuhtanen, and Alexander Jung titled "Automating Root Cause Analysis through Machine Learning in Agile Software Testing Environments,The author of this publication describes For RCA 1, we present a framework for combining expert knowledge into machine learning algorithms. We look at how to build a feature extraction procedure, how to define ground-truth categories, and how to evaluate machine learning algorithms.

According to Ravi Teja Yarlagadda in year 2021, on the topic “DevOps and Its Practices”

The author of this paper describe the benefits of devops and he also mentioned that how devOps are playing important role in the IT industry

According to Lianping Chen in the year 2015 on the topic “Continuous delivery huge benefits and challenges too “ This article discusses why Paddy Power chose to use this strategy.

The term "continuous delivery" (CD) refers to the end result. CD capability, as well as studies on the several advantages and There are several difficulties to overcome. This information can be beneficial. Practitioners should think about how they're going to implement CDs and how they're going to do it The research agendas of researchers are created.

According to Robin Bolscher and Maya Daneva in 2019 on his research “Designing Software Architecture to Support Continuous Delivery and DevOps: A Systematic Literature Review”

his study gives a thorough review of software architectural techniques that facilitate Continuous Delivery (CD) and DevOps implementation. Its purpose is to give both researchers and practitioners with a grasp of the state-of-the-art on the subject.

When applying CD and DevOps approaches to an existing software system, this SLR has discovered ten unique architectural difficulties, the most serious of which is strongly connected monolithic systems.

# 

# CHAPTER 3

## METHODOLOGY

## 3.1 PROBLEM STATEMENT

In today’s world Traditional software development methods are no longer adequate to fulfil the demands of today’s businesses. Agile techniques are attractive to software development organizations because they enable flexibility, efficiency, and speed throughout the Software Development Life Cycle (SDLC). Given its increasing significance, a growing number of businesses are failing to meet these growing challenges and are looking for improvements in software development. In today’s software business, agile is the most popular software development strategy

## 3.2 RESEARCH QUESTION

Before proceeding further i would like to mention research question of the study are:

### 3.2.1 RQ 1: How we can implement CI CD process in agile software development.

A CI/CD pipeline is a set of procedures that must be completed before a new version of software can be released. CI/CD pipelines are a practise that aims to improve software delivery by automating the software development life cycle.

Organizations may generate higher quality code quicker by automating CI/CD throughout the development, testing, production, and monitoring phases of the software development lifecycle. Although each stage of a CI/CD pipeline can be completed manually, the true value of CI/CD pipelines is realised through automation.

### 3.2.2 RQ 2: What is the Big challenge that Occur in CICD pipeline automation process.

If we're using a CI/CD pipeline, we'll almost certainly be collaborating with a large group of people. We might even be divided into teams with various duties. Human communication is sometimes the most difficult aspect with a CI/CD. If something goes wrong during a software deployment, for example, communication is critical to resolving the problem quickly.

CI/CD pipelines will rely largely on automation throughout their entire lifecycle. Human communication, collaboration, and teamwork are three characteristics. Without these three characteristics, success will be incredibly difficult. If you want your CI/CD pipeline workflow to succeed, you must improve communication and transparency

## 3.3 GAP IDENTIFICATION

This study seeks to identify tools and methodologies, identify problems and practises, and identify research gaps by conducting a reliable estimation of the technology in consistent practises. Software engineering research and practice has shifted its focus to continuous practices.While the techniques, tools, and practises described here cover a wide range of issue There are a number of issues and limitations that will necessitate additional research: enhancing the collection and analysis of contextual data in research examining various elements of continuous practises; having a thorough understanding of how software-intensive systems should indeed be (re-)architected in order to enable continuous practises; solving the knowledge gap about continuous practices. As well as the major challenge we have analyzed throughout the study that is “human communication”If we're using a CI/CD pipeline, we'll almost certainly be collaborating with a large group of people. We might even be divided into teams with various duties. Human communication is sometimes the most difficult aspect with a CI/CD. If something goes wrong during a software deployment, for example, communication is critical to resolving the problem quickly.

If you want your CI/CD pipeline workflow to succeed, you must improve communication and transparency

Number of people are working remotely from different countries as we all know that software deployment process are getting advance day by day if we have CICD a smooth process of integration and deployment so we must have any technique to remove the issue that occur in real world while the implementation of CICD process

## 3.4 PROPOSED SOLUTION

By constructing the solution and executing a set of automated tests each time a change is submitted, I examine how agile methods paired with a continuous integration and continuous delivery (CICD) pipeline strategy have boosted project efficiency. A ci/cd pipeline also offers developers with immediate feedback on their changes. The CICD pipeline automates the entire process, eliminating the need for manual execution. I’ve also looked at how automation works better than human process execution using the CICD pipeline technique. Because of the better product issue, faster coding is required, which needs faster testing. When a new app (or an update to an existing app) is introduced to the market, it must be completely functioning and user-friendly for all users. Customers will easily switch to a competitor and leave a slew of negative evaluations, causing the programme to lose credibility. The best way to ensure that everything runs smoothly is to thoroughly test and improve each function. Manually completing it for complicated, feature-rich applications under tight deadlines is difficult. In modern agile software development settings, continuous integration and deployment (CI/CD) needs several tests in short sprints deployment pipelines.

### 3.4.1 CI CD METHODOLOGY DIAGRAM

The CI/CD pipeline automates and continuously monitors software development throughout its lifecycle. It covers everything from integration and testing to delivery and deployment. The CI/CD pipeline refers to these interconnected activities. The deployment scaling approach using CICD presented by this research work is shown in the image below.

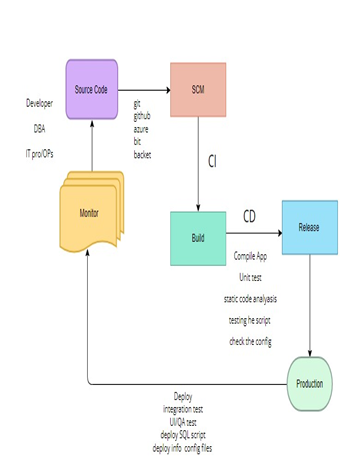


Figure .The deployment management cycle with CICD.

A CI/CD pipeline is a set of procedures that must be completed before a new version of software can be released. CI/CD pipelines are a practise that aims to improve software delivery by automating the software development life cycle.

Organizations may generate higher quality code quicker by automating CI/CD throughout the development, testing, production, and monitoring phases of the software development lifecycle. Although each stage of a CI/CD pipeline can be completed manually, the true value of CI/CD pipelines is realised through automation.

### 3.4.2 VIRTUAL AGENT

A **virtual agent** (also known as an intelligent virtual agent (IVA), virtual rep, or chatbot) is a software programme that provides automated service or instruction to humans using predefined rules and, increasingly, artificial intelligence applications.

A virtual agent is not the same as a virtual assistant, so don't get them mixed up. The main distinction is that virtual assistants are human agents, whereas virtual agents are computer programmes.

A virtual assistant is a remote employee who performs duties for their boss. Some of their responsibilities could include:

* Appointment scheduling and calendar management
* arranging transportation
* Calling and transferring calls

### 3.4.3 Integration of virtual agent in CICD pipeline

If we're using a CI/CD pipeline, we'll almost certainly be collaborating with a large group of people. We might even be divided into teams with various duties. Human communication is sometimes the most difficult aspect with a CI/CD. If something goes wrong during a software deployment, for example, communication is critical to resolving the problem quickly.

So we have integrated Virtual agent if in the condition of failure in the deployment process or any misshape done during deployment process. A virtual agent will read the issue and inform the other team members that due to some reasons “Build is in pending state”. virtual agent also work on run time.

As we all know, software deployment processes are progressing day by day. If we have CICD a smooth process of integration and deployment, we must have any approach to remove the issues that arise in the real world while implementing the CICD process.

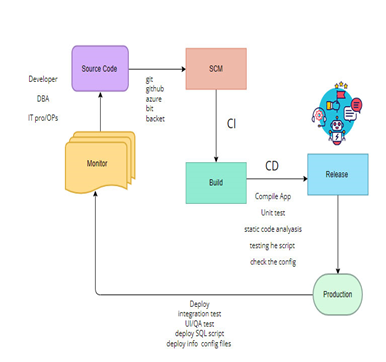


Figure . Integration of virtual agent in the deployment management cycle with CICD

## 

# CHAPTER 4

## RESULT AND DISCUSSION

We have three different tools to help you build your practise, so keep that in mind. A house cannot be built with just one tool. You can't use one to enable your development practise. As a result, Agile, DevOps, and CI/CD are three distinct tools, each with its own importance. The outcomes are revolutionary when a development organisation uses all three for their intended goals. Many people mistakenly believe that Agile approach and DevOps are the same thing, which is entirely incorrect. Agile was a response to waterfall techniques' approaches, while DevOps was not a response to Agile. It's critical to understand the distinction between agile and DevOps. The only way out of this misunderstanding is to examine the differences between DevOps and agile methodologies.

Agile emphasises practises that emphasise change while speeding up delivery.

• CI/CD emphasises tools that prioritise automation and focuses on software-defined life cycles.

• DevOps emphasises roles that emphasise responsiveness and focuses on culture.

• While Agile methodology focuses on project management, DevOps focuses on pipeline optimization.

• While Agile methodology focuses on project management, DevOps focuses on pipeline optimization.

• Agile focuses on requirements adaptability and function advancement, whereas DevOps stresses continuous integration and installation of software development.

**RQ.1:**What are the advantages of working in a CI/CD environment?

* More flexible development. Traditional app development slows down many firms who aspire to provide and operate at real-time speeds. Developers can quickly prioritise and address each organisational need with CI/CD. This also necessitates a good working relationship with the business side in order to prioritise areas with the most potential business benefit, rather than what developers may regard as more technical.
* Improved code quality. Smaller changes require less code, and the quality of the code can be examined more thoroughly before it is sent to a comprehensive testing environment. Early in the DevOps process, developers can more readily identify and solve problems. However, code quality is crucial; CI/CD does not imply that you can sacrifice quality for speed.
* Testing cycles are shorter. Because there are fewer complex and smaller quantities of code to examine, there is less need to test functionality periodically during the CI/CD process. New functionality produced by the developer community is deployed much faster than thorough retro-testing of large changes. Again, this does not imply that you should skimp on testing; rather, a well-run CI/CD system entails fewer interactions between what has changed and what has not.
* Changes in the operational environment can be tracked more easily. If something goes wrong during a rollout, identifying the root cause is usually much easier when only one or two changes are involved. However, the correct instruments must be in place to provide adequate and effective monitoring.

**RQ 2:**What are the drawbacks of CI/CD?

There are many factors that go into making a Continuous Integration and Delivery pipeline work. A long chain of successful occurrences must occur in order for the automated processes to function successfully. Unfortunately, when attempting to set up and operate with a CI/CD pipeline, you are likely to encounter at least a few frequent challenges.

* CI/CD pipelines offer numerous benefits for automating the deployment of software builds. Unfortunately, many cyber attackers will target CI/CD pipelines since they often have security flaws that aren't addressed. A CI/CD pipeline can include a lot of sensitive data, which is why hackers would try to obtain access to the pipeline via extreme tactics.
* The goal of any CI/CD pipeline would most likely be to automate the delivery of software and code upgrades as quickly as feasible. The difficulty is that if things aren't done correctly, a lot of performance difficulties might creep into the software.
* If you're working in a CI/CD pipeline, you'll almost certainly be collaborating with others. You might even be divided into teams with various duties. Human communication is sometimes the most difficult aspect with a CI/CD. If something goes wrong during a software deployment, for example, communication is critical to resolving the problem quickly.

**COMPARISON OF CICD TOOLS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | jenkins | Circle CI | TeamCity | GitLab |
| Open source | YES | NO | NO | NO |
| Easy to use | Medium | Medium | Medium | Medium |
| Supported OSss | windows,linux,macOS,unix like OS | linux,macOS | windows,linux,macOs,unix like Os,freeBSD and more | linux,distributions,ubuntu,debian,centOS, Oracle linux |
| Hosting | on premises or cloud | on premises or cloud | on premises | on premises or cloud |
| Free version | YES | YES | YES | YES |

Table COMPARISON OF CICD TOOLS

# 

# 

# 

# 

# 

# 

# CHAPTER 5

## Conclusion

In this research we have discuss how to implement CI/CD automation pipeline methodology for deployment process and the association of DevOps Given its growing importance, agile and CSDD methodologies are becoming more popular. CD makes deployment management more efficient and more automated. so that developers deliver their software earlier, more often and on a regular base.We have analyze that there is also some pros and cons of CICD pipeline we have highlighted one big issue which is is facing by many teams working from different country and using same platform for sharing their code ,software program and projects but if any hazard occur between the releasing build process we lost the **Human communication** so there should be any feature that which integrate and eliminate this issue . the ratio of this issues highlighted at the level where people are working remotely although CDCI pipeline working perfectly Based on our experience we have analyzed that for the betterment of the software industry it is necessary that we should work more in the area of automation where CI and CD pipelines play a great role for the betterment of the software company. Many researchers have done a lot of the work in this area but we would like to add our contribution too so people will have a better understanding of CI/CD , Agile and DevOPs.

# 

# References

[1] A. Phillips, M. Sens, A. de Jonge, and M. van Holsteijn, The IT Manager’s Guide to Continuous Delivery: Delivering business value in hours, not months: XebiaLabs, 2015.

[2] Sr. Associate, KPMG LLP, Department of Information Technology, Wilmington University 419 V street, Apt D, Sacramento, CA 95818

[3] Dingsøyr, T., Nerur, S., Balijepally, V., and Moe, N., 2012, ”A decade of agile methodologies: Towards explaining agile software development”, Journal of Systems and Software, 85(6), pp. 1213-1221.

[4] Yasmine SKA, ,Janani P “A STUDY AND ANALYSIS OF CONTINUOUS DELIVERY, CONTINUOUS INTEGRATION IN SOFTWARE DEVELOPM.

[5] S.A.I.B.S. Arachchi, Indika Perera,“Continuous Integration and Continuous Delivery Pipeline Automation for Agile Software Project Management’.

[6] W. Cunningham, ”Principles behind the Agile Manifesto”,agile manifesto.org, 2017. [Online] Available

http://agilemanifesto.org/principles.html. [Accessed: Sept. 2017]

[7] Julen Kahles, Timo Huuhtanen, Alexander Jung, ’Automating Root Cause Analysis via Machine Learning in Agile Software Testing Environments’

[8] Childs, H., Brugger,E.S, Bonnell,K.S., Meredith,J.S., Miller,M., Whitlock,B.J., Max,N.,2005, “A contract-based system for large data visualization. In: Proceedings of IEEE Visualization, pp 190–198.

[9] Berg, A. M., 2015, “Jenkins continuous integration cookbook: Over 90 recipes to produce great results from Jenkins using pro-level practices, techniques, and solutions,” Birmingham, UK: Packt Publishing.

[10] Elbaum, S., Rothermel, G., and Penix, J.,2014, “Techniques for improving regression testing in continuous integration development environments. In FSE

[11] Lianping Chen, ’Towards Architecting for Continuous Delivery ’,’015 12th Working IEEE/IFIP Conference on Software Architecture’

PLAGIARISM VERIFICATION (for Final Submission)

(Note: *This form is filled at the final submission of Thesis/Research Project Reports for graduation. Library MUST submit his/her/their work in turnitin repository*)

Title of Research Project: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Total Pages \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Registration ID \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Supervisor:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Program: Faculty: Faculty of Computing

This is to report that the above document was scanned for similarity detection and has been submitted in turnitin repository. Process and outcome is given below:

Software used **turnitin** Date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Similarity Index\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Total word count\_\_\_\_\_\_\_\_\_\_\_\_\_\_

File Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Digital Receipt No \_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Checked by Library Staff Head Librarian

(Name & Signature) (Signature & Stamp)

*\*Please attach plagiarism report mentioning similarity index duly signed and stamped by library at the end of the thesis/ Research Project.*

# PLAGIARISM REPORT

Attach the title page and summary of plagiarism report provided by coordinator (checked by library). Plagiarism report from any source other than library is not acceptable.