MIT | School of Distance Education

A

PROJECT REPORT

ON

"Importance of Quality in Project Management with the help of CI/CD and Agile methodologies"

UNDERTAKEN AT

"MIT School of Distance Education"

IN PARTIAL FULFILLMENT OF

"PGDM in Project Management"

MIT SCHOOL OF DISTANCE EDUCATION, PUNE.

GUIDED BY

" Dr. Jayant Panigrahi "

SUBMITTED BY

"Ms Preetika P Ravkhande"

STUDENT REGISTRATION NO.: MIT202102268

MIT SCHOOL OF DISTANCE EDUCATION PUNE - 411038

MIT | School of Distance Education

Exempt Certificate - If you're not able to provide the Project Executed Certificate

To

The Director

MIT School of Distance Education, Respected Sir,

This is to request you to kindly exempt me from submitting the certificate for Project Work due to the reason mentioned below:

Tick the right option

1. As per the Rules of the Organisation
2. ✓ Self Employed
3. Working in Public Sector
4. Full-time Student

Thanking you in anticipation of your approval of my request. Regards

Student Sign: -

Student Name: - Ms Preetika Ravkhande

Student ID: - MIT202102268

DECLARATION

I hereby declare that this project report entitled **"Importance of Quality in Project Management with the help of CI/CD and Agile methodologies"** bonafide record of the project work carried out by me during the academic year **2023-2024**, in fulfillment of the requirements for the award of **"PGDM in Project Management"** of MIT School of Distance Education.

This work has not been undertaken or submitted elsewhere in connection with any other academic course.

Sign:-

Name:- Ms Preetika Ravkhande

Student ID: MIT202102268

ACKNOWLEDGEMENT

I would like to take this opportunity to express my sincere thanks and gratitude to **"Dr. Jayant Panigrahi"**, Faculty of MIT School of Distance Education, for allowing me to do my project work in your esteemed organization. It has been a great learning and enjoyable experience.

I would like to express my deep sense of gratitude and profound thanks to all staff members of MIT School of Distance Education for their kind support and cooperation which helped me in gaining lots of knowledge and experience to do my project work successfully.

At last but not least, I am thankful to my Family and Friends for their moral support, endurance and encouragement during the course of the project.

Sign:-

Name:- Ms Preetika Ravkhande

Student ID: MIT202102268

ABSTRACT

Since software is created by people like you and me, errors are bound to occur at various development stages. Some errors might be minor while others if detected at a later development stage, can have dire consequences on the final product. To ensure that such issues are minimized, frequent and rigorous testing is required.

There are many reasons why testing is beneficial to software companies, but here are the main ones:

- To detect possible issues or defects at an early stage.
- To improve the quality of the MVP and final product.
- To identify and tackle flaws in a component or the whole system.

Besides the main benefits of testing mentioned above, there are many other reasons for testing a software product. For instance, to comply with industry-specific standards (e.g. products for the healthcare industry) Many industries such as motor or pharmaceutical industries require software to comply with their rules and standards. Testing allows developers and prod uct managers to structure the product development process around those rules and standards to avoid possible issues. The ultimate goal of testing is to produce software that is both user-friendly and reliable in the long run. From the perspective of a software company, testing is essential as bugs can be detected before the final product is presented to the client, thus guaranteeing the quality of the product delivered.

TABLE OF CONTENTS

CHAPTER 1

Introduction

Proper Project management makes use of different policies, procedures and principles to plan, implement and finish a project. To guarantee a satisfactory result, projects must start with specified parameters that are designed to produce the desired outcome. Each project that undergoes the project management process needs to follow a project life cycle that basically guides the project from start to end. This is why companies invest heavily in capable PM positions to oversee some of their most prized projects

Project management in software testing walks a similar line but due to the difference in the industry, it has its own unique challenges and quirks. Software development projects are often quite complex and multi-faceted. In order to finish a project on time and within budget, software developers must utilize various efficient planning, organization and monitoring techniques.

CHAPTER 2

Objective and scope

Testing is carried out at different stages as developers verify the efficiency and performance of an application before releasing it into the real-world. Such practices in automated testing are proving to enhance the Quality Assurance process and resulting in better outcomes based on early bug detection, executing repeatable tasks, and benefits from constant feedback.The main goal of this is to identify the strengths and weaknesses of the implementation of quality in projects management in the stable and well established project management company.

The project focuses on using CI/CD principles to ensure rapid testing and deployment. The main feature of DevOps and agile in software testing is that various testing is performed to verify the efficiency of the software application. Agile methodologies & DevOps are foundational principles of effective software testing around the world.

CHAPTER 3

Benefits of Quality Software Testing in Project Management

Software testing is a method of determining whether the actual software product meets the expected requirements and ensuring that the software product is free of defects. It entails running software/system components through their paces using manual or automated tools to evaluate one or more properties of interest.
The goal of software testing is to find errors, gaps, or missing requirements in comparison to the actual requirements.

When a software development project is going on, you need to know that errors may appear in any phase of the life cycle. Few of them are known to be undiscovered. Thus, the importance of **Quality Assurance** cannot be ignored.

There are high chances that the final code has errors of functionality and design. For the identification of the issues before the occurrence in the critical environment, it is a prerequisite to performing the testing of software.

Software testing is important because if there are any bugs or errors in the software, they can be identified early and fixed before the software product is delivered. A properly tested software product ensures dependability, security, and high performance, which leads to time savings, cost effectiveness, and customer satisfaction.

It happens to be an integral part of the process. However, it involves a huge cut off from the pocket.

Nevertheless, you need to keep in mind that the price owing to the failure of the software can be really high.

As with most techniques used in the software testing world, utilizing proper project management methods has its fair share of benefits too. Let’s go through a few of them:

- Time management: By creating a Work Breakdown Structure (WBS) for the project, you can easily manage your efforts across activities and define timelines for the project.
- Scope management: Through project management, you essentially define the scope of testing which is approved by the stakeholders. This eliminates any factor of ambiguity in the scope of testing. This ensures that there is no confusion among the stakeholders of the project.
- Quality management: The acceptance criteria tabled at the start of the testing can help you to manage the quality of the software product.
- Cost management: The WBS of the project will allow you to keep track of the project cost and make any necessary changes in case of deviations.
- Communication management: A thought-out communication process can help you in managing communication among stakeholders.
- Team management: The project management tools help to manage resources better and keep track of the performance of the team members. At the same time, you can utilize the team to the optimum.
- Focused approach: Project management can utilize specific tools to aid the testing process. Test case management tools, for instance, can help the team to maintain a focused approach towards the goals of the test items while keeping track of the tests that have been completed.

CHAPTER 4

SOFTWARE TESTING: A Key Aspect of Quality

The delivery of an optimal quality software product that has unique and innovative features has always been the priority of the software industry worldwide. However, without evaluating software components under various expected and unexpected conditions, the team cannot guarantee these aspects. Therefore, testing is performed to test every software component large and small.

To understand Software testing importance, let's look into the below points:-

1) The software test is essential. So, do not start over from scratch again: Sometimes, we test a fully developed software product against the user requirement and find that some basic functionality was missing. It may happen because of a mistake in the requirement gathering or the coding phase. Then to fix such types of errors, we may have to start the development again from scratch. Fixing such kinds of mistakes becomes very tedious, time-consuming, and expensive. Therefore, it is always desirable to test the software in its development phase.

2) Evaluating the ease of use of the software: Ease of use is a simple concept; it specifies how easily the intended users can use the final product. The software testing ensures the construction of the software product in a way that meets the user's expectations regarding compliance with the requirements in a comfortable, satisfactory, and simplistic manner.

3) Verification of all aspects of the software- You can verify all the aspects of the software in software testing, such as checking the basic functionalities as well as testing a system for unexpected conditions. Unexpected conditions can be from an incorrect data type or due to a piracy attack. Therefore, testing makes sure that the system can handle these situations very well. Thus, if we find an error in advance, we have the option to correct them. It can prevent complaints once the software or application has reached customers.

4) Software tests help accelerate development- Software tests help developers find errors and scenarios to reproduce the error, which in turn helps them to fix it quickly. Besides, software testers can work in parallel with the development team, thus understanding the design, risk areas, etc. in detail. This knowledge exchange between testers and developers accelerates the entire development process.

Types of Testing:

Testing is both manual and automated and can be divided into functional and non-functional testing.

Manual testing is testing completed by an actual person who will click through the application and interact with the software. Automated testing is completed by a machine that executes a test script. Automated testing is a crucial element of continuous delivery.

Functional testing focuses on the requirements and features of your product. Essentially, functional testing looks at the elements users interact with and may include:

- Unit testing. Unit tests are automated tests designed to catch errors early by checking the functionality of a single unit of code.

- Integration testing. Integration tests look at how different units of code or systems interact and are therefore helpful for catching issues arising from integrating different components.
- Acceptance testing. Acceptance tests make sure that the entire system is working how a user would expect it to.
- Regression testing. Regression tests checks whether new features will break or degrade functionality. Sanity testing is a simplified form of regression testing that examines functions, commands and menus.
- Smoke testing. Smoke tests usually take place after a new build or deployment and are designed to verify the functionality of basic features.
- End-to-end testing. End-to-end testing works by mimicking end users' behaviour within the application to ensure everything is working as expected. It includes things like logging in and out, loading a page, and making online payments.

Non-functional testing is focused on the behaviour of the application and the things that users cannot see such as security, reliability and performance. Non-functional tests include:

- Performance testing. Performance testing examines how the software performs under different workloads. Load testing is a similar test that is performed under actual load conditions.
- Stress testing. Stress testing is designed to see how much strain the system can handle before crashing.
- Usability testing. Usability Testing looks at how well a customer can use a system or application to complete a task and if they have any problems with the UI.
- Security testing. Security testing allows you to verify that the software is safe and free from vulnerabilities and flaws that could compromise data safety.
- Compatibility testing. Compatibility testing ensures that your software will work across different browsers, hardware, devices and operating systems.
- Reliability testing. Reliability testing determines how your software performs in different conditions.

CHAPTER 5

Achieving Efficient Project Management in Testing

Just like in any other software development endeavor, software testing also has a fixed start date and an end date. Because of this, software testing can be seen as a project too and project management principles and tools can be used to manage it effectively.

Once the software is given to the software testing team, the initiation phase of the project begins. If you're overseeing the project as its manager, your job is to review and analyse the business requirements of the software. After this, you identify the procedures and processes that need to be followed, followed by collecting historical data, if any exist, that can be utilised as a reference when testing the software.

After you've understood the project, you'll begin to define the preliminary requirements for it and the risks involved. Then, you'll define measurable objectives for the testing project and communicate these to the stakeholders. Stakeholders could include the management and the software development teams.

To further develop a successful and detailed roadmap for the software testing project, you can further break down the planning phase into the following sub-phases:

- Defining the scope of Testing: A well-defined and approved scope of the projects makes sure that there is no misunderstanding across teams and the scope does not alter frequently.
- Defining a Test Strategy: The next phase is to define a test strategy, based on the scope you've defined. The test strategy is a high-level document that states the approach you intend to take in order to fulfill the objectives of the software testing. This document can generally consist of the scope and approach of testing, the testing tools you'll be needing, the metrics you'll utilize, the roles and responsibilities of testing teams and tracking and reporting.

- Finalizing Requirements: For proper testing, you must identify tools and techniques that you'll be using for software testing. These tools include any automation based CLI or GUI based tools or test case management tools.
- Building a Work Breakdown Structure: By creating a work breakdown structure (WBS), you can manage the project to the smallest level. This is done by breaking test projects into small deliverable units of work known as activities.
- Estimating Test Efforts and Defining the Team: After the WBS is made, you should move on to estimating the efforts needed to complete each activity of the WBS.
- Build a Test Schedule: After the previous step, you can define and communicate the test schedule to the stakeholders and get it approved.
- Defining Test Metrics: Clear metrics should be defined and based on these metrics, you'll be able to define the quality of the software.
- Approve a Software Test Plan: Lastly, you should define a software test plan and have it approved by the stakeholders. This document should cover the answers to these basic questions:
- What to test?
- Who will test?
- When to test?
- And how to test?

With the solid groundwork in place, you're set to begin your software testing project. With the steps highlighted, we hope we have provided an in-depth understanding of the importance of project management and the advantages it can have for any software development business if leveraged correctly. A solid project management framework can allow the most demanding and difficult projects to be run at many improved and effective levels, which are guaranteed to fail without it.

CHAPTER 6

Importance of Agile Software Testing in project management

In the early days of software development, teams were following the Waterfall software development methodology. Software testing came second to last on the list, only before the delivery of the finished product. Testers were kept in the dark, validating the features outlined in the requirements document. The methodology was ineffective. Proof of the ineffectiveness were the overtime hours and the costs that exceeded the budget.

The best quality is understood as meeting the needs of the customers and stakeholders. In general, quality in projects is something what will satisfy the needs for which it was undertaken. In the connection with the project management, quality should be decided by end-user rather than by company, which realizes the project. The agile testing era includes software testers at the beginning of the project. There's less documentation and more room for adaptation. Instead of testing the software right before deployment, agile testing happens during each two-week sprint. Minds change and the product needs to adapt. In agile testing, testers are involved directly in the development process so they can detect bugs as early as possible.

The agile resolves issues at every stage of the development process, helping the product to be released on time. Agile testing has three main benefits: increased interaction, a high-quality product, and faster delivery.

1. **Increased interaction**. The highlight of agile testing are teams, people and interactions. Team members closely communicate about any setbacks, preferences for specific tools, and methodologies.
2. **High-quality product**. In an agile setting, testers are in close communication with developers. Testers and developers are equally involved in the process and their skills are put to good use. There's continuous feedback and any bugs can easily be removed. What's more, testers and developers are in communication

with the customer who can give their input to help them develop a high-quality product.

3. **Faster delivery**. When teams use agile in testing, there's continuous feedback and communication. The development process consists of separate sprints and testers can fix errors in the middle of the project. The end result? Faster and timely delivery of a high-quality product.

Most Common Agile Testing Methodologies:

1. Exploratory Testing

Testers explore the application to discover any edge cases and learn what needs to be changed. They put themselves in the shoes of the personas who will use the application to find out what needs fixing and updating.

Exploratory testing is a cyclical practice that starts from test design and progresses to test execution, analysis, learning, and then the process starts all over again. There are no scrips, no predefined set of instructions. The agile tester relies on their skills to explore and update the product.

2. Acceptance Test-Driven Development

Acceptance-driven development is collaborative testing that brings together team members with a different perspective. Customers, testers, and developers collaborate to write acceptance tests that represent the user's point of view.

This gives them insight into what customers expect from the product and how the product will be used. It's the best way to make sure that everyone on the team has the same shared understanding of what they're actually building.

3. Behavior-Driven Development

Behavior-driven development refines the process of test-driven development (TDD) and acceptance test-driven development (ATDD). It augments TDD and ATDD by following these five basic steps:

1. Begin with user stories.
2. Automate your BDD scenarios.
3. Implement the features.
4. Run the automated BDD scenarios to show the feature is completed.
5. Repeat.

4. Integration Testing

A software project involves several software modules that are coded by different developers. Although each software module is tested, defects can still exist for reasons like inadequate exception handling or developer's error.

The aim of this type of testing is to expose errors in the interaction between integrated modules.

CHAPTER 7

Quality Improvement using CI/CD

The main goal of DevOps is to increase collaboration. This is not achievable without having *Continuous Testing, Continuous Integration, and Continuous Delivery*. Testers play a vital part in each of these processes.

CI/CD allows organizations to ship software quickly and efficiently. CI/CD facilitates an effective process for getting products to market faster than ever before, continuously delivering code into production, and ensuring an ongoing flow of new features and bug fixes via the most efficient delivery method.

Continuous integration (CI) refers to the practice of automatically and frequently integrating code changes into a shared source code repository. Continuous delivery and/or deployment (CD) is a 2 part process that refers to the integration, testing, and delivery of code changes. Continuous delivery stops short of automatic production deployment, while continuous deployment automatically releases the updates into the production environment.

CI/CD And DevOps

CI/CD tools can help a team automate their development, deployment, and testing. Some tools specifically handle the integration (CI) side, some manage development and deployment (CD), while others specialize in continuous testing or related functions.

CI/CD is an essential part of DevOps methodology, which aims to foster collaboration between development and operations teams. Both CI/CD and DevOps focus on automating processes of code integration, thereby speeding up the processes by which an

idea (like a new feature, a request for enhancement, or a bug fix) goes from development to deployment in a production environment where it can provide value to the user.

In the collaborative framework of DevOps, security is a shared responsibility integrated from end to end. It's a mindset that is so important, it led some to coin the term "DevSecOps" to emphasize the need to build a security foundation into DevOps initiatives. DevSecOps (development, security, and operations) is an approach to culture, automation, and platform design that integrates security as a shared responsibility throughout the entire IT lifecycle. A key component of DevSecOps is the introduction of a secure CI/CD pipeline.

How Software testing help organizations in CI/CD quality improvement

Software testing plays a crucial role in enabling Continuous Integration and Continuous Deployment (CI/CD) processes. CI/CD is a software development methodology that emphasizes frequent and automated code integration, testing, and deployment. The goal of CI/CD is to deliver software faster, with higher quality, and at a lower cost. Testing is an essential part of this process, as it ensures that changes made to the codebase don't introduce errors or bugs that could impact the software's functionality.

Here are some ways software testing can help in CI/CD:

1. Early detection of issues: Automated testing can detect issues early in the development process, allowing developers to address them before they become more significant problems. This can save time and resources in the long run.
2. Improved code quality: Testing helps ensure that code changes are of high quality and meet the necessary requirements. This can help prevent issues from arising in production and ensure that the software is functioning as expected.
3. Faster feedback: Automated testing provides quick feedback on code changes and allows developers to catch issues before they are merged into the codebase. This helps eliminate the need for manual testing and speeds up the development process.
4. Continuous testing: Automated testing can be integrated into the CI/CD pipeline, allowing for continuous testing as code changes are made. This ensures that

changes are tested thoroughly and quickly, reducing the risk of issues in production.

5. Better collaboration: Testing can help facilitate better collaboration between developers and testers. By sharing test results and feedback, developers can work more closely with testers to improve the quality of the software.
6. Support continuous improvement: By testing code changes continuously, organizations can identify areas for improvement and implement changes to address them. This supports a culture of continuous improvement and helps organizations to deliver better software over time.
7. Ensure compliance: Testing can help organizations ensure that their software meets regulatory and compliance requirements. This is especially important in industries such as healthcare, finance, and government.
8. Increase confidence in deployments: By testing code changes thoroughly, organizations can increase their confidence in deployments. This helps to minimize the risk of errors and reduce the time required for debugging and fixing issues after deployment.
9. Ensure reliability: Testing can ensure that the software is reliable and performs as expected. This helps to minimize downtime and ensure that the software meets the needs of users.

CHAPTER 10

Conclusions / Findings

As you can tell based on the above, testing is critical to project management and organizations overall. Without sufficient testing and defect resolution steps, it can lead to a disaster for an organization and cause consumers to lose trust in the company's products and services. A thorough testing approach is necessary to make sure that the solution is extensively tested to meet the consumer's requirements and provide value to the consumer at the end of the day.

With the solid groundwork in the right place, you're set to start your software system testing project. With the steps highlighted, we tend to hope we've provided a detailed understanding of the importance of project management and therefore the blessings it will have for any software system development business if leveraged properly. A solid project management framework will permit the foremost difficult and tough tasks to be run at several improved and effective levels, which square measure is bound to fail in the absence of it. Therefore, software testing tools not only make the task easier for the testers but also assist in creating flawless products.