Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of System Engineering

Master's Thesis

Integrating Comprehensive Testing Strategies into Agile Project Management for Software Development

Jaynto Goswami Deep

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Jaynto Goswami Deep

Informatics

Thesis title

Integrating Comprehensive Testing Strategies into Agile Project Management for Software Development

Objectives of thesis

Thesis objectives:
1) Evaluate Current Testing Practices.
2) Investigate and comprehend various Agile project management frameworks and their key principles.
3)Determine potential integration points between testing strategies and Agile project management.
4) Evaluate the impact of integrating comprehensive testing strategies on project success metrics.
5) Provide Practical Recommendations.

Methodology

Methodology:

1) Conduct an in-depth review of academic literature and industry publications on Agile methodologies, project management, and software testing practices.

2) Analyze existing case studies of software development projects that have

successfully integrated testing into Agile project management.

3) Develop a structured survey to gather information from practitioners in the field. Include questions related to current testing practices, challenges faced, and the perceived impact of testing on Agile project success.

4) Evaluate existing tools and technologies that support both Agile project management and comprehensive testing.

The proposed extent of the thesis

40-80 stránek

Keywords

project management, agile, testing, SDLC

Recommended information sources

AXELOS GLOBAL BEST PRACTICE. *PRINCE2 Agile®.* Norwich: Axelos, 2015. ISBN 978-0-11-331467-6.

SCHEERER, Alexander.; SPRINGERLINK (ONLINE SERVICE). *Coordination in Large-Scale Agile Software Development [electronic resource] : Integrating Conditions and Configurations in Multiteam Systems.* Cham: Imprint: Springer, 2017. ISBN 9783319553276.

SCHWABER, Ken; BEEDLE, Mike. *Agile Software Development with Scrum.* Upper Saddle River: Prentice Hall, 2002. ISBN 0-13-067634-9.

Expected date of thesis defence

2023/24 SS – PEF

The Diploma Thesis Supervisor

Ing. Petra Pavlíčková, Ph.D.

Supervising department

Department of Systems Engineering

Electronic approval: 20. 3. 2024	Electronic approval: 25. 3. 2024
doc. Ing. Tomáš Šubrt, Ph.D.	**doc. Ing. Tomáš Šubrt, Ph.D.**
Head of department	Dean

Prague on 25. 03. 2024

Declaration

I assert, that my self-reliance and priority to the title name of the topic, "**Integrating Comprehensive Testing Strategies into Agile Project Management for Software Development**", have assisted in designing my master's dissertation. In the last part of my thesis, there is a detailed reference list for all my resources. I used. In role with a writer of master research I swear that there is no copyright infringement.

In Prague on 25-03-2024 ______________________________

Jaynto Goswami Deep

Acknowledgement

Let me say in advance, I was the sole author in the process of my master's efforts title, "Integrating Comprehensive Testing Strategies into Agile Project Management for Software Development". I've merely referred to the works of the point contained in the thesis conclusion's section. I make a point of double-checking myself that I haven't violated no copyright at all.

I officially and in much pride acknowledge all those who made a crucial contribution to my thesis and thus turned my idea into reality. Firstly, a huge appreciation to my academic advisor for the type of guidance I received from Petra Pavlíčková, Ph.D. It was her talking and coming that more than once has served a north star for my whole this study. Such extraordinary work of her undivided attention, depth of knowledge and encouragement played a crucial role in it all.

Besides, the professors and the administrating teachers deserve a big chance from the Czech University of Life Sciences Prague. This location with its conducive environment and provision of needed resources was instrumental in finishing my research. My voyage has been made much more worthwhile by the selfless devotion that they have shown in regard to academic excellence.

No words can possibly express the immense gratitude I have for my family, who has constantly stood by me even in times when the research process has been extremely difficult. They have been firm in their expectations of my excellent performance, and thereby, ignited my journey.

Last but not the least I thank those who provided their time and rich data invaluable for my study to finish it. Credit rating agencies have acted as key elements in the making, formulation, and execution of the credit ratings strategy.

A big appreciations to you guys who contributed to this forum (academic), both whom directly and indirectly exercised involved in making the mission successful.

Integrating Comprehensive Testing Strategies into Agile Project Management for Software Development

Abstract

In contemporary software development, the fusion of Agile methodologies with robust testing strategies has emerged as a pivotal approach to ensure quality and agility throughout the project lifecycle. This thesis delves into the integration of comprehensive testing strategies into Agile project management frameworks, aiming to enhance project success metrics and deliver superior software products. The research objectives encompass evaluating current testing practices, exploring diverse Agile project management frameworks, identifying integration points between testing strategies and Agile methodologies, assessing the impact of this integration on project success, and offering practical recommendations.

Employing a multifaceted methodology, this study embarks on an extensive review of academic literature and industry publications, delving into Agile methodologies, project management principles, and software testing practices. Additionally, it scrutinizes real-world case studies of successful software development projects that have adeptly integrated testing into Agile project management workflows. Furthermore, a structured survey is designed to gather insights from practitioners, elucidating current testing practices, challenges encountered, and the perceived impact of testing on Agile project success. Moreover, the research evaluates existing tools and technologies supporting both Agile project management and comprehensive testing to provide a comprehensive understanding of the practical landscape.

By amalgamating theoretical insights with empirical evidence, this thesis endeavors to illuminate the significance of incorporating comprehensive testing strategies within Agile project management paradigms. Ultimately, the findings of this research aim to furnish valuable insights for practitioners, empowering them to optimize their project management approaches, foster collaboration between development and testing teams, and ultimately elevate the quality and success of software development endeavors in an Agile environment.

Keywords: Agile methodologies, project management, software testing, comprehensive testing strategies, integration, project success metrics, literature review, case studies, survey, practitioner insights, tools and technologies, software development, quality assurance.

Vylepšení agilního řízení projektů pomocí komplexních strategií testování ve vývoji softwaru

Abstrakt

V současném vývoji softwaru se spojení agilních metodologií s robustními testovacími strategiemi ukázalo jako stěžejní přístup k zajištění kvality a agilnosti během životního cyklu projektu. Tato práce se ponoří do integrace komplexních testovacích strategií do agilních rámců projektového řízení s cílem zlepšit metriky úspěšnosti projektů a poskytovat špičkové softwarové produkty. Výzkumné cíle zahrnují vyhodnocení současných testovacích postupů, prozkoumání různých rámců agilního řízení projektů, identifikaci bodů integrace mezi testovacími strategiemi a agilními metodikami, posouzení dopadu této integrace na úspěch projektu a nabídku praktických doporučení.

Tato studie využívá mnohostrannou metodologii a pouští se do rozsáhlého přehledu akademické literatury a průmyslových publikací, ponoří se do agilních metodologií, principů projektového řízení a postupů testování softwaru. Kromě toho zkoumá skutečné případové studie úspěšných projektů vývoje softwaru, které vhodně integrovaly testování do pracovních postupů agilního řízení projektů. Kromě toho je strukturovaný průzkum navržen tak, aby shromáždil poznatky od odborníků, objasnil současné testovací postupy, problémy, s nimiž se setkali, a vnímaný dopad testování na úspěch agilního projektu. Kromě toho výzkum hodnotí stávající nástroje a technologie podporující agilní projektové řízení a komplexní testování, aby bylo možné komplexně porozumět praktickému prostředí.

Sloučením teoretických poznatků s empirickými důkazy se tato práce snaží osvětlit význam začlenění komplexních testovacích strategií do paradigmat agilního projektového řízení. Závěry tohoto výzkumu mají v konečném důsledku za cíl poskytnout cenné poznatky pro odborníky z praxe, umožnit jim optimalizovat jejich přístupy k řízení projektů, podpořit spolupráci mezi vývojovými a testovacími týmy a v konečném důsledku zvýšit kvalitu a úspěšnost snah o vývoj softwaru v agilním prostředí.

Klíčová slova: Agilní metodiky, projektový management, testování softwaru, komplexní testovací strategie, integrace, metriky úspěšnosti projektu, přehled literatury, případové studie, průzkum, poznatky z praxe, nástroje a technologie, vývoj softwaru, zajištění kvality

Table of Contents

1. Introduction

The deliverance of high-quality IT products by development teams in the industry is hampered by tight timeframes and dynamically changing process requirements. The agile methods have collected a good momentum, due to the iterative and flexible nature of project management. These help the teams to work at the same premises as the changes happening around. The main purpose of this work is to research the most effective ways of incorporating rich scrutiny approaches into Agile project management for software product development. This is with the view of boosting success metrics and product's quality (Mishra, Alzoubi 2023).

Testing always is a very important one that brings quality, functionality, and reliability to life. Agile project management approaches put a range of shared work, flexibility, and customer satisfaction through a series of iterative development cycles. By making test implementation seamlessly into Agile methodology frameworks, teams are able to detect and treat the defects earlier in development process leading to increase in the quality of the products and therefore customers satisfaction (Aquino 2023).

This thesis aims to study the current testing approaches in software engineering, to investigate major Agile project management frames and their common principles, to find where testing strategies can be integrated into Agile, to determine the effect of holistic-testing integration onto performance metrics, and to deliver practical advise for application.

With a view to the aims stated, the study will contain a thorough review of the academic literature and business press, dealing with Agile methodologies as well as software development, project management and software testing. Furthermore, this study will be based on case studies and successful examples of existing software development projects that managed to comply with Agile project management with all its intricacies, including both good practices and problematic moments.

An interview survey will be conducted to collect pratices information from relevant people in their industry, with the focus on the current testing, challenges they face and the perceived influence of it on the agile projects. The list of data will make a holistic research about the practical aspects of testing within the Agile frameworks, the results of which will be the description of the useful advantages.

Then again, the evaluation of tools or technologies supporting the management of agile projects or the execution of comprehensive testing will be made to choose the most appropriate ones for integration. If the teams employ these tools adequately, they can benefit from this as collaborative activities become extremely streamlined within the Agile project management framework (Aquino 2023).

As a result, this research will cover the connections between the software development testing options and Agile Project Management of software development, thus providing useful tips and suggestions for software development practitioners who are looking for a road to a better management that will improve overall project results. The study is guided by a thorough methodology and focuses on the immediate operation of the practical Agile practices and software testing strategies in the industry through this research will thus be useful towards the Agile systems improvement and improved software testing strategies in the industry.

2. Objectives and Methodology

The chapter objectives and methodology are detailed in this part and it will serve as a reference as we embark on integrating allowing test approaches into the Agile project management for software development. The objectives are meticulously formed to cover critical testing practices improvement, from evaluating currently used testing techniques and strategies to providing practical solutions to optimize those and other testing improvements. All end goals are meant to dig deeper into the core of effective test cases territory in Agile transformed processes, which later can help to improve the rate of successful projects completion. Besides, the methodology section maps the research strategy, that is, weaving together a literature review, case studies, structured questionnaire, and data analysis procedure. This study is carried out by scrutinizing academic papers, industry publications, and actually applied case studies, as well as inputs made by professionals who work within or know the agile field, and they give us ways of improving the testing practice through Agile projects management frameworks.

2.1 Objectives

Here, the purposes of the research studies confirmed; every objective address the critical part of making effective challenging strategy into the Agile project management of the software development. We set our objectives as the assessment of the existing practices and their strengths and weaknesses; the identification of different agile project management tools and methods; and the evaluation of the ideas of integrating single testing strategies into agile management methods. The outcome of the research will consist of the success metrics which traditional testing strategy purpose should serve, such as the success of the project, efficiency and time frames. The research will thus result in practical recommendations allowing managers. Lets look at each objectives in high level:

✓ **Evaluate Current Testing Practices**

This objective calls for an evaluation of current testing strategies used across software development projects in a bid to understand what is good about them in terms of strengths, what aspect they don't address well enough in terms of limitations, and what areas they need to be improved upon.

✓ **Investigate and comprehend various Agile project management frameworks and their key principles**

Investigate and comprehend various Agile project management frameworks and their key principles. The primary objective of this analysis is to research at various Agile methodologies and project management methodologies, thus gaining insights into their fundamental principles, process and practices.

✓ **Determine potential integration points between testing strategies and Agile project management**

Determine potential integration points between testing strategies and Agile project management. This goal is seeking to reflect upon ways of incorporating these methods of all-encompassing testing into Agile management frameworks in order to reinforce the software development process in the electronic procedure.

✓ **Evaluate the impact of integrating comprehensive testing strategies on project success metrics**

Evaluate the impact of integrating comprehensive testing strategies on project success metrics. This goal is aimed at evaluating the influence of the end-to-end testing coverage on the project critical elements, including quality level, efficiency, and timeliness.

✓ **Provide Practical Recommendations**

The last goal is to present actionable directives that are followed from the weaknesses identified from the research to assist managers in adopting efficient testing tactics which can be incorporated within agile project management frameworks.

2.2 Methodology

The research methodology involved in this project is aimed at addressing all the objectives set forward in the research objectives. It encompasses the following key approaches:It encompasses the following key approaches:

- ✓ **Literature Review**

  Weeks of intensive analysis of both academic literature and industry publications to draw conclusions regarding Agile methods, project management theory and software testing standards will be utilized.

- ✓ **Case Studies Analysis**

  Search engine for the current projects documentation in software development that incorporated testing in Agile management will be conducted to find useful lessons and best techniques.

- ✓ **Structured Survey**

  A structured questionnaire will be developed to get information from the professionals in affiliation to the field. The survey - which will provide information on testing practice, the difficulties involved and a possibility of the impact of testing on the Agile project success - will comprise of the questions.

- ✓ **Data Analysis Techniques**

  The thematic analysis of literature review, case studies and structured survey results will be ascertained to develop the reporting trend, find out if there are gaps, or whether there are common themes. This analysis is thus aimed at revealing the need for testing strategy inclusion in Agile project management and the aspects that the strategy influences for the purposes of project success metrics.

In short, this chapter has defined the goals and guidelines as the backbone supporting the investigation into a full-fledged integration of testing approaches with Agile project management for building software companies. The outline centers round doing the review of the existing testing practices, exploring Agile project management frameworks, finding integration points, assessing the impact of the testing strategies, and offering practical recommendations. The methodology is made of a comprehensive literature review, collection analysis, an organ- donation survey, and data analysis techniques to accomplish this end. The objectives and methodologies of the research to be discussed here are to add useful insights and recommendations to the quality controllers of Agile projects. Thus, the implementers of Agile methodology are likely to be more successful in the building of software systems.

3. Literature review

In a very dynamic environment of softwares designing, adoption of Agile methodologies update the conventional technical applications. The agile methods that include such frameworks as Scrum and Kanban, have come to be as a way allocating emphasis on adaptability, collaboration, and iterative development cycles. Learning and internalising fundamentals and procedures of Agile culture is a must for a company which wants to have a competitive edge in delivery of reliable and high-quality software goods. This chapter has made every effort to present an accurate in-depth review of Agile methodologies, project management principles, testing process and practices, Agile project success metrics, integration points between testing methods and Agile methodology, and the effect of testing integration on project success (Brechner 2015).

Agile approaches give the team a framework for project management that becomes adaptable to the evolution of the requirements. This development process is used to deliver critical business values to the stakeholders. Scrum that has been a focus area from an aspect of time-box iteration, and cross-functional collaboration, provides a great tool kit for the management of complex projects. While Kanban focuses on visualizing the work, limiting the work in progress, and optimizing the workflow efficiency and, thus improves the workflow, Scrum takes a more dynamic approach. The trade mark of these methodologies is the deep understanding of their principles, which enables the organizations to iterate their project management to become leaner and eventually achieve high business impacts of their project outcomes (Schwaber, Beedle 2002).

Project management principles are crucial navigation tools that help software development ventures steer their course for completion. The project managers handle the projects right from initiation to team building and estimations. In their core they use established principles that gives them the liberty to meet project objectives within the set constraints. Efficient project management methods are not only the means of allocating the resources and prioritising tasks as well as engaging the stakeholders and interacting with the team members. By focusing on project management methodologies that align with Agile techniques, organizations can build a supportive platform that will encourage broad-based testing and enhance the effectiveness of the project metrics (Schwaber 2007).

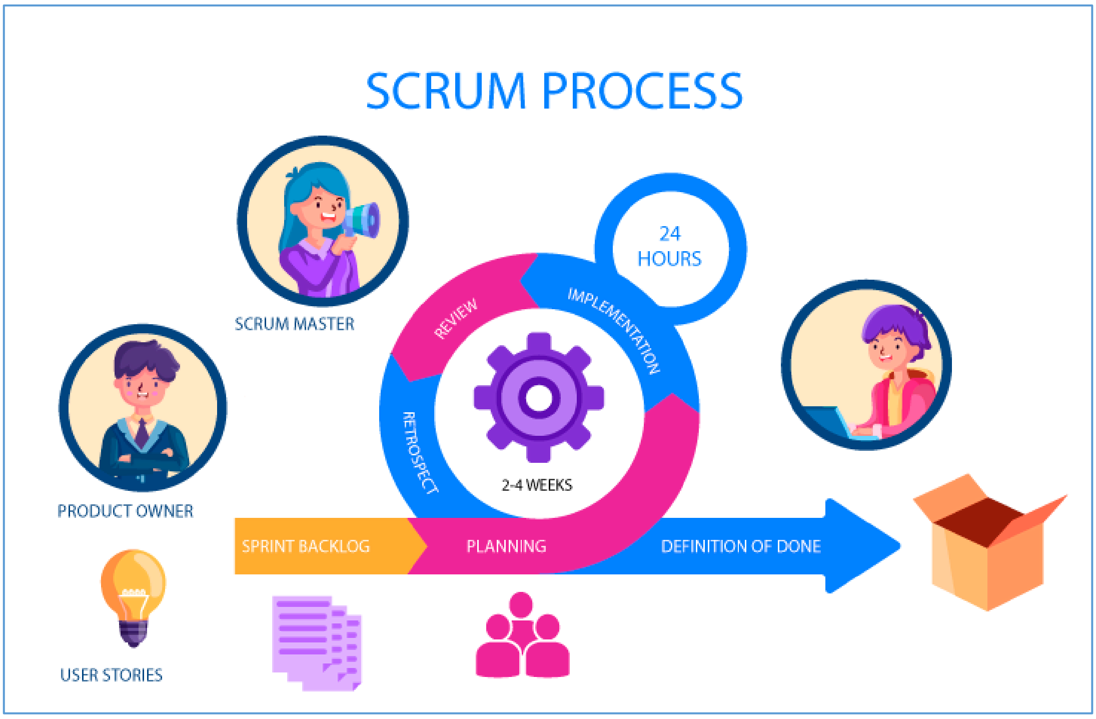
3.1 Overview of Agile Methodologies

Agile methodologies have eclipsed the software development terrain with a lithe and cyclical derivative in handling project management. These methodologies offer the core values of customer collaboration, evolving to evolve and release working software at each step. It is only through Agile methodology acceptance that the process of meeting the customers' needs evolving requirements gets faster, teamwork collaboration is improved, and ultimately high quality products are delivered (Brechner 2015).

3.1.1 Scrum

Scrum is a popular Agile structure applied about projects management who has a lot of disciplined. Aspects relating to the basis of scrum are the division of projects into periods, named as sprints after a specified number of weeks. To achieve this, these agile teams will establish a rhythm of frequent deliveries so that stakeholders will receive the first amount of value incrementally and also the team will be able to seize feedback early in the development process. The Scrum framework defines three key roles: the Product Owner, with the mandate to define project requirements and priorities in response to stakeholder needs; the Scrum Master, who serves as a facilitator and mentor to the team, making certain its commitment to Scrum principles and rules, and the Development Team whose members, coming from all project areas, are responsible for creating working software every sprint (Sachdeva 2016).

Scrum meetings are of utmost relevance in providing opportunities for frequent inspections, discussions and timely adjustments in the project. One of the daily stand-up meetings, such as daily scrums, helps team members to coordinate efforts, report on progress and outline the impediments that they are facing when it comes to the tasks. Sprint planning sessions allows the team to carry out together the task of defining sprint objectives and to pick the ones for the previous period that the team has to accomplish them within the sprint. Scrum reviews are moments where stakeholders can give their reviews about the delivered increment while retrospectives is time dedicated to look at what was done right and what went wrong and how it can be fixed the next time sprint (Sachdeva 2016).

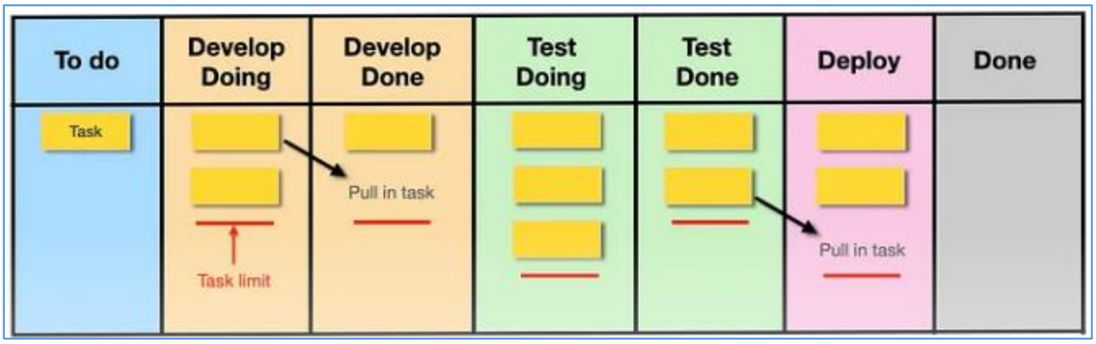


Picture 1: Scrum process
[Source: https://www.nimblework.com/agile/scrum-methodology/]

Scrum is the iterative nature, which gives adaptability and ability to respond to the changing requirements especially when we talk about projects where needs tend to evolve or the level of uncertainty is high. Scrum achieves this by slicing intricate tasks into executable segments and encourages teamwork among members for it helps organizations to not only deliver value but to also maintain focus on satisfying customers on-the-go. Such transparency and absolute inspection is the key mechanism for Scrum, the model that gives a team necessary confidence to advance and ensure a prompt delivery of the products that meet all customer demands (Truss 2024).

3.1.2 Kanban

Kanban, which is the visual system generated from lean manufacturing fundamentals that helps in optimising workflow efficiency by expressing people's work on a kanban board is the primary concept. The Kanban board usually includes the columns which represent the specific stages of the work process, including but not limited to "To Do," "In Progress," and "Done" stages. Cards or sticky notes use to symbolize the work items that progress across the board as they started in each stage, which in this case, could help visualize the work in progress and the bottlenecks in the process under play. Kanban achieves through production limitation and the continuous delivery system having a constant flow of work at the team, it helps in dropping inefficiencies and improving productivity (Lehekar, Urne 2017).



Picture 2: Kanban board
[Source: https://medium.com/an-idea/implement-kanban-principles-to-improve-your-workflow-and-productivity-dc3a97bcd188 **]**

Kanban departs from its principal principle that include the following: restrict the amount of work in progress (WIP) to avoid overloading the team and to render the team’s focus comfortable by working on tasks with an ease. We can do this is by limiting the WIP of each step of the workflow; further processing will be implemented to avoid multitasking and context switching for better output. WIP limits act as stimulants which in turn motivate the team members to focus on the task, collaborate efficiently, and generate product value incrementally that eventually become crucial for sustainable and predictable workflow (Sunden, Hammarberg 2014).

Kanban's continuous improvement circumstances triggers kaizen, which explains that improvement process over a period of time comes eventually through making small changes over time. A visualization of work and setting lead times can help teams to spot bottlenecks that must be resolved and consequently result in an increase in productivity. Through continuous retrospectives and feedback demand teams are able to see their issues and errors then correct them, if needed, to keep a high degree of efficiency and provide more value to the stakeholders (Brechner 2015).

The applicability of Kanban is what distinguishes it from other project management systems, it can be used in software engineering, sales and marketing as well as products development. On the other hand, Scrum tells practices and define roles and ceremonies whereas Kanban gives the chance to teams to customize the method in a way that can fit for their context and different needs. Through leadership of visual management, constraints on work in progress, and dedication to the principle of continuous improvement, Kanban provides organizations with a tool for flexible project management aiming for delivering value to clients continuously with the minimum of flows (Brechner 2015).

3.2 Project Management Principles

Project management principles stand in for the compass for successful project execution, providing teams with the necessary approach to define, organize, and order project activities that eventually help them reach their predetermined goals within a set of constraints. They contain the principles of practices and methods which will support the project manager and the team in performing resource planing, risk mitigation and project delivery within the set budget and time limit. Appreciation and appliance of project management practice become fundamental for gaining Project Efficiency and for meeting the demands of all the Project stakeholders (Westland 2007).

Efficient start of the project as an indispensable principle of project administration involves definition of project goals, scope, and final products, as well as, identification of key stakeholders and establishment of suitable communication channels. Through a series of project finishing activities that involve, for instance, devising a project charter and establishing project criteria for success, project managers prepare the ground for effective project execution by providing as the same set of expectations and project goals to all team members (Westland 2007).

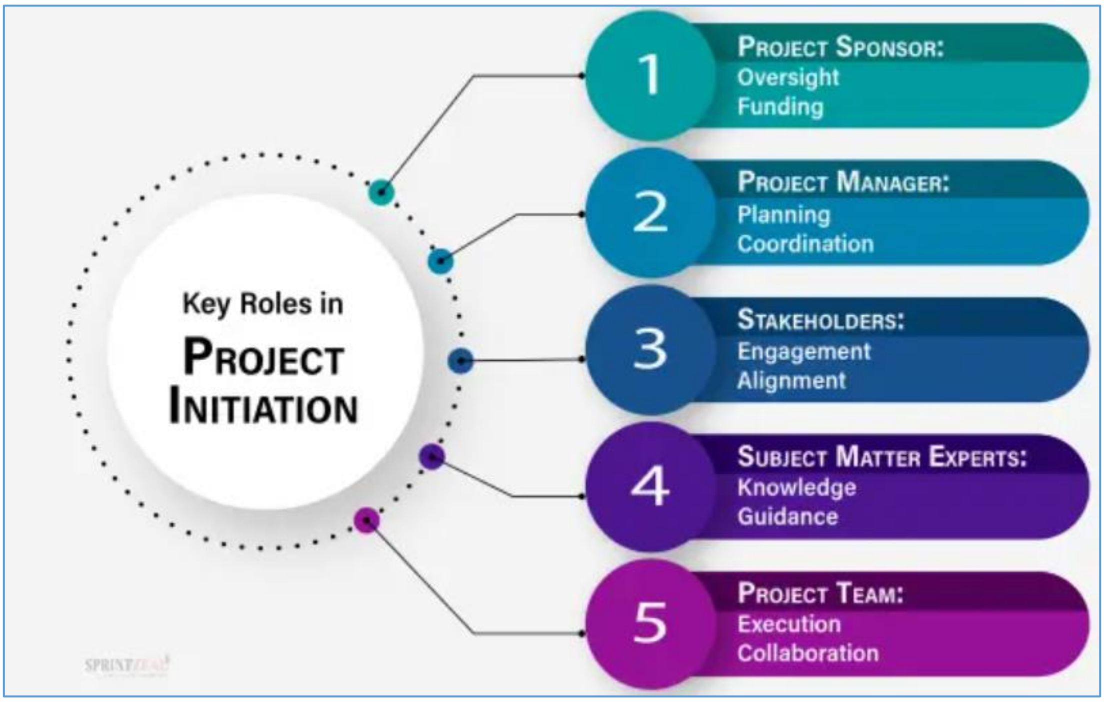
Planning and estimation, which are among the key project management principles, mean the development of the comprehensive project plan, which will include the list of actions to be done, timeframes, resource needs, and mutual interference. The use of planning by the project mangers brings to light all the risks that may occur, resources allocation will be done strategically and chalk out time limit at the end of which the progress will be measured. For instance, practices such as bottom-up estimation or analogous estimation provide for accurate project planning, which allows teams to remain aware of expectations and make project-related decisions accordingly (Grebić 2019).

Project management principles however a major element of successful project is communication, collaboration and stakeholders integration. By providing a platform for open communication, fostering teamwork, and engaging the stakeholders during all project stages, the project managers, therefore, can build a trusting group relation, resolve conflicts before they arise, and align the team effort towards the overall project objective. Being able to handle responsibilities well attributed to leadership skills, making decisions effectively and concerning the entire project are the important attributes of a good project manager that he can utilize to navigate to challenges , motivate the team members and to support the project towards its successful completion (Grebić 2019).

3.2.1 Project Initiation

Project initiation considered as a milestone in project management which helps in building the perfect grounding for project success through determining a project’s purpose, its scope and range, objectives, and key stakeholders. Besides project initiation, during such stage of a project, project managers cooperate with stakeholders to set concisely the project's objective and the deliverables. With this phase, a project charter should be the expected function which is the formal document authorizing the existence of the project in addition to providing a high-level description of its scope, targets and requirements (Aguanno 2005).

The major actions during initiation of the project that are done include carrying out a feasibility study that will evaluate project's viability, determining project constraints and risks, and deciding on the benchmark of success. Through the assessment of project's viability, project managers will figure out whether the project would concur with organisational goals, as well as the resources and restrictions and constraints put in place. The recording of risks at the beginning in projects preparation helps teams implement risk mitigation plans and contingencies to overcome the challenges that can possibly affect the project execution (Aguanno 2005).



Picture 3: Project initiation key roles

[Source: https://www.sprintzeal.com/blog/the-importance-and-roles-involved-in-the-project-initiation-phase]

Another imperative factor at the start of the project is the establishment of list of stakeholders and engagement with them. The project managers should therefore be able to find key stakeholders like sponsors, the users of the project and members of the project team and come up with ways on how they will effectively communicate with the stakeholders to confirm that the stakeholders are fully informed and involved on the project life cycle. Engagement of stakeholders early in this process leads to the formation of a support group, the definition of the expectations of all participants as well as the development of the collaboration among members of the team, which all participate in the successful implementation of the projects (Crowder, Friess 2014).

Through the project initiation work activities, e.g., starting with project charters, conducting feasibility studies, identifying risks, stakeholders involvement, the managers can create the necessary basis for execution of the project successfully. The launch of a project is where a robust plan is formulated, resources are allocated as needed, and communication is started which produces results by using the predetermined constraints so that occupants achieve the project objectives and value that are of interest (Crowder, Friess 2014).

3.2.2 Planning and Estimation

Although project management step-by-step may seem rather obvious as planning and estimation comprise the very foundation of this procedure and have a direct impact on the scope definition, resource allocation, and project execution time-frame establishment. Proper planning covers the designing of a detailed schedule, where the elements of its plan are listed as tasks, dependencies, milestones, and resource needs to help keep the project tasks running smoothly. Project managers work side by side with team members, who are responsible for the development of a detailed project schedule, which can be used to map the project route and to show the best targets, time for deliverables and responsibilities (Aguanno 2005).

While planning the project, the project manager takes steps of risk assessment to determine the problematic aspects of the project and puts in place mechanisms to minimize risks and a contingency plan. Through early examination of hazards in the planning process, teams will foresee obstacles, appoint roles and allocate resources in such a way to provide safety nets in form of contingency plans to be done in adverse situations that may interfere with the project results. Efficient risk management remains one of the key factors that are crucial for the project to stay in line with the deadlines and the budget as well as the quality requirements assigned to it (Aguanno 2005).

For project planning, flexibility in techniques, such as bottom-up estimation, analogous estimation, and parametric estimation, is considered as critical to give the most accurate forecast of the project timeline, budget and resource requirements. A Project manager together with team members defines the duration of task, the amount of effort, costs involved in implementation of activity-related processes. Through the use of historical data, opinion of experts, and industry markers, the teams can derive up-to-date predictions that match project targets as well as limitations. The estimated values are of paramount importance as they form the foundation on which we base our goals and objectives, we gain the knowledge to address stakeholders' demands and informed decision making for finishing the operations (Aguanno 2005).

Planning project strategy implies developing a communication plan, which serves as a guideline for communicating with the team members, stakeholders and external partners throughout the project lifecycle. High-performing communication platforms make teamwork easier, more conductive, and can help members surmount obstacles while they remain focused to the project’s ultimate end. Through developing the clear communication frameworks from the beginning of the planning stage, project managers are going to build up a culture of open communication, seek out the conflicts before they occur, and update the stakeholders on the projects schedule (Grebić 2019).

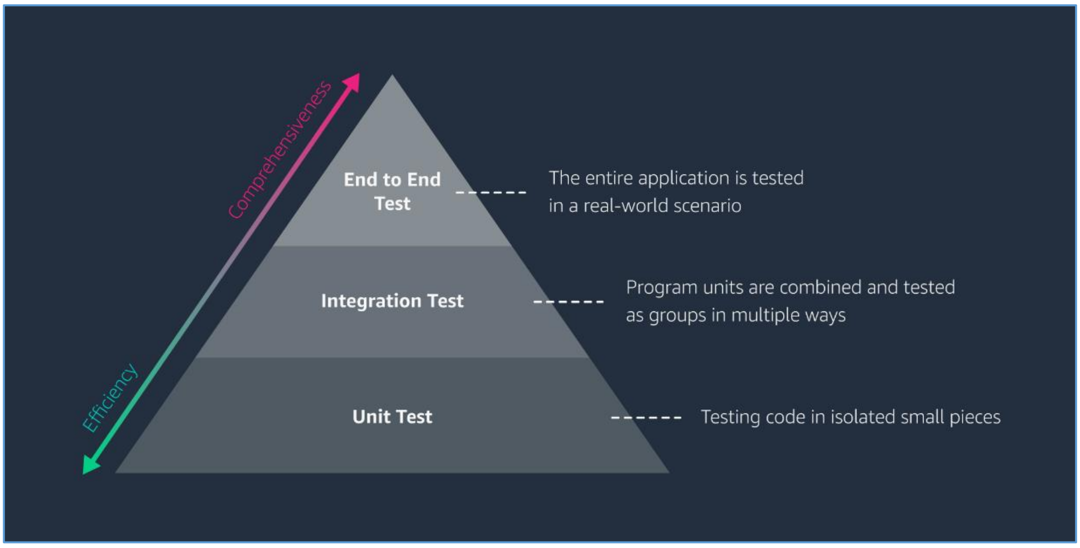
3.3 Software Testing Practices

Software testing has become the backbone of the software development methodology aimed at checking the quality, functionality, and reliability of a software product. These testing practices are good at finding and addressing typos, bugs and security faults in the code, enabling teams to satisfy user requirements in terms of quality. The next section studies different types of testing, among which unit testing helps to confirm functionality and examine integration testing and system testing, that are considered as crucial for verifying the software implementation, and acceptance testing that validates if end users recognize the implemented features during the acceptance (final) stage (Drabick 2013).

3.3.1 Unit Testing

The most common software testing hardware unit testing is a foundational practice which requires to test the units or components in isolation. Under the unit test, which is the portion of software testing in which developers write the test codes to check out the functions/method/class behavior within the software. Coding the code into discrete procedures and testing them separately makes it possible for developers to spot the errors in the development phase of the program, improve the code’s quality and ensure that all units work together as a system. Unit testing is a vital procedure in agile and test-driven development (TDD) methodologies of writing tests prior to rather than after the code generation and to direct developmental process and ensure code efficiency (Khorikov 2020).

Unit test frameworks with automation such as JUnit for java, NUnit for .NET, and pytest for python facilitate the scripting of candidate test executions as well as delivery of timely feedback about the code changes. These frameworks prescribe the way of making up test cases, executing tests, tracing back the result, boosting the speed of the code testing process. The unit tests are automated that allow the team to turn the testing process into a streamlined, defects identification process starts early, and keeping code quality is the whole development lifecycle (Khorikov 2020).


Comprehensiveness
Efficiency
End to End Test
The entire application is tested in a real-world scenario
Integration Test
Program units are combined and tested as groups in multiple ways
Unit Test
Testing code in isolated small pieces

Picture 4: Unit testing

[Source: https://developer.amazon.com/en-US/blogs/alexa/post/35bdad3d-57c8-4623-88c6-815540697af5/unit-testing-create-functional-alexa-skill.html]

The primary purpose of unit testing is to examine whether separate code units function properly and without interfering with the other parts of the system when they are integrated. In most cases, a unit test usually deals with specific scenarios or functionalities in a code unit. It then validates what should be the actual outcome against the expected result. Through writing of the favorable unit tests that cover the whole code path, various edge cases, and error conditions by developers, they will be able to satisfy themselves that the code truly operates as intended in multiple scenarios and edge cases (Khorikov 2020).

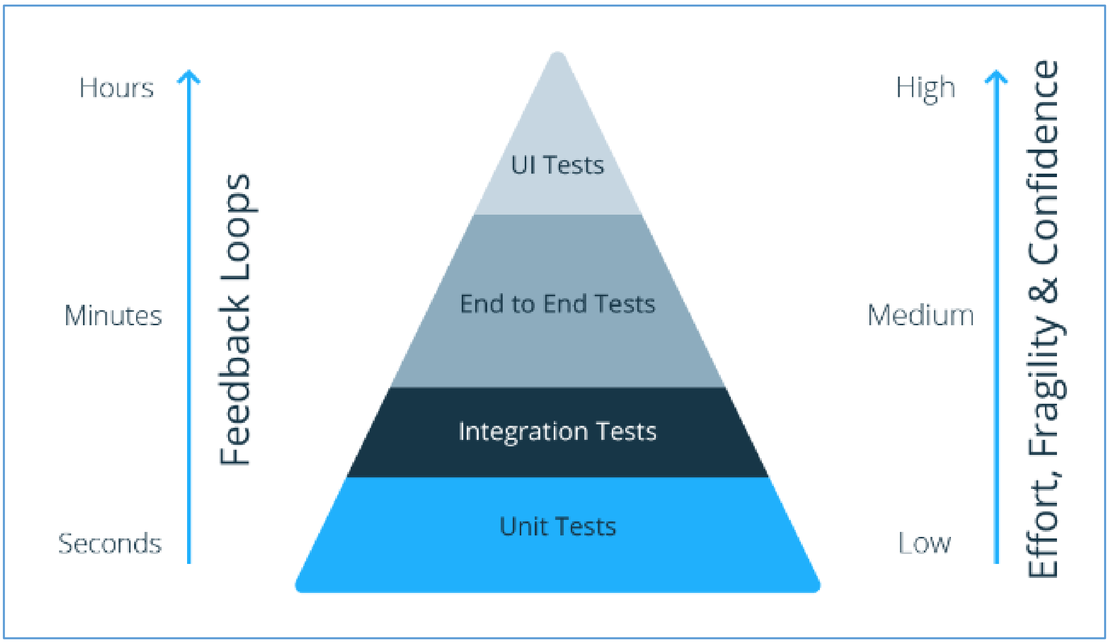
The unit testing contributes to improved software quality via the revealing of defects prior to the late stages of development, thus cutting down on the cost of the later steps of the process, the correction of bugs and the improvement on code maintainability. Through using test-driven development, developers write test cases prior to introducing new features or changes in the software. When developing such test cases, they can clarify requirements by designing robust interfaces and defining critical points, therefore ensuring that the code functions properly. Unit testing is the by product of expected outcomes testing, it ensures code reusability, it also helps simplify debugging procedures, and has a way of generating a culture of continuous integration and delivery through providing fast code changes feedback (Koskela 2013).

While summing up we can say that unit testing is the software testing method which is of paramount importance because it assures the code correctness, detects bugs at the very beginning and helps to move on in the agile development. Through applying unit testing into the software development process, teams are able to withstand their product quality, shorten development cycles, and offer superior high-quality software solutions that exceed users' expectations.

3.3.2 Integration Testing

Integration testing is paramount in software testing as it involves verifying the interactions (communication) between various units or components of software (engine parts) to ensure they operate uniformly smoothly. Integration testing is one of this stage’s responsibilities. Developers do quality assurance to see whether the integrated modules communicate well, share data appropriately, and keep the system working. This testing stage is intended to discovery of integration problems, data transferion errors, and issues of incompatibility which perhaps could come to color during the joining of several groups of formats. Integration testing becomes an instrument for validation of system behavior and finding troubles that ability to localize by unit testing can be limited (Baylis et al. 2021).

Integration testing can be done in diverse ways whether systems testing is a top-down or bottom-up testing, test driven development (TDD), or any rapid application development (RAD). Integration test of high-level modules in the top-down approach starts with these modules first followed by low-level modules in the way of data flow and control. However, upward integration testing where lower-level modules are first tested and higher-level modules joined afterwards at the end to demonstrate the system interaction, takes place instead Sandwich integration testing is a saving idea which consolidate both benefits of top-down and bottom-up methods by nesting from top to bottom the various levels of the software hierarchy (Frankel 2015).



Picture 5: Integration testing
[Source: https://pactflow.io/blog/contract-testing-vs-integration-testing/]

Testing of integration scenarios scenario involves testing of communication between modules, validation of communications protocols, and verifying system interfaces to ensure smooth integration. For instance, through executing of fixed integration tests covering diverse integration sections, error controlling mechanisms and the boundary conditions, developers are in a position to find out bugs which are related with interchange of data, module dependencies, and systems interoperability. By using integration testing, its ensured that the combination of integrated components is functioning well, sharing data and maintaining the system's stability during the process as a whole; thus, it reduces system risks and provides the feasibility statement regarding the software quality (Baylis et al. 2021).

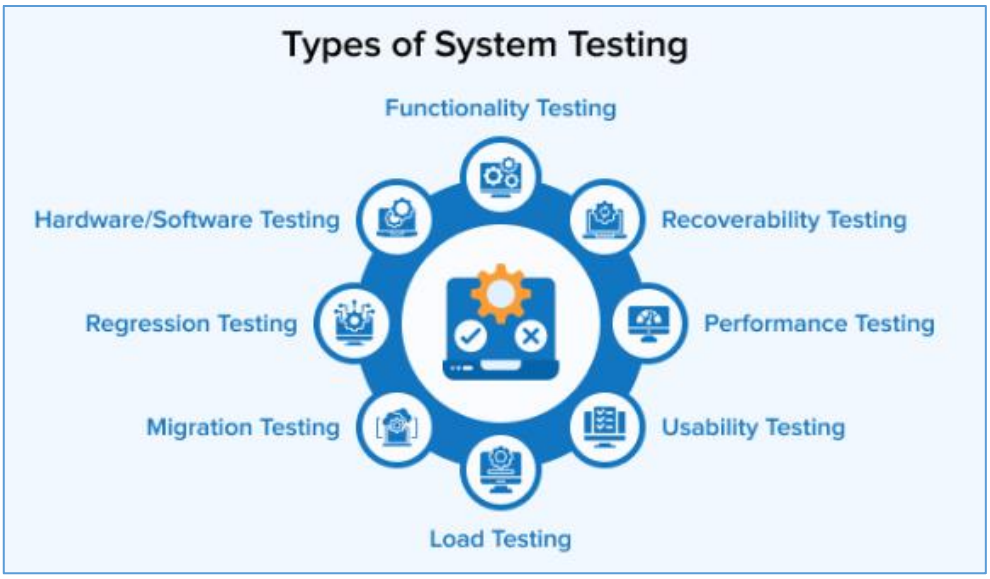
Automated integration testing tools and frameworks like Selenium and Postman for web applications and APIs serve as the means of automation of the test execution process and pulling off the integration testing effortlessly. These tools are employed to automate the various tasks like test script creation, test case execution and test report generation thus enabling the developers to validate the system interactions and localize the integration issues in a very effective manner. Teams can upgrade their test coverage and eliminate unnecessary manual effort by automating integration tests. This leads to the discovery of defects at the outset and development phase (Frankel 2015).

As a vital software testing method, integration testing guarantees the smooth operation of software components, verifies the accurate working of system interactions, and becomes a source of early detection and pinpointing of any integration issues. Thorough integration tests enable developers to ensure that integrated modules do not conflict with any other, that data information is passed correctly and that system remains working as expected once jointly assembled. Integration testing has a great advantage in terms of testing the functionality and logic of the system, increasing software reliability and assuring a user-centered approach to developing software products.

3.3.3 System Testing

System testing is the final key software testing practice that addresses how the software functions as a full system and confirms that it does meet the specific requirements and that once implemented it runs in its actual environment. In testing the system, testers evaluate the system behavior against the features of functionality, performance, security, and usability being user-friendly to ensure it meets the users' expectations. During the testing phase test cases should be run which cover nursing scenarios, user interactions and system features to confirm functionality during different use cases. System tests help to find faults which are related to system connections, data flows and human interactions, therefore, input into the quality and performance of the software overall (Agarwal, Tayal, Gupta 2010).

The type of testing includes massive functional testing, performance testing, system testing, and usability testing, which are used to examine every factor of the application's functioning and the performance. One of the main missions of the functional test is to ensure that software operate as they should, they form the part of the necessary functionality and deliver expected results. The performance testing measures the ability of the software to handle innumerable load scenarios ranging from a single user login to that of thousands of users with the purpose of guaranteeing its efficient operation in the production area. Security testing aims to find out all the weaknesses as well as threats or even risks that lie in the software. Thus, the data that may be critical can be fully protected and the occurrence of security breaches can be prevented. Usability testing examines how the software users navigate through, use and react to its user interface and user experience to ensure that it is well-structured, user-friendly, and offers the user what she/he expects (Agarwal, Tayal, Gupta 2010).



Picture 6: System testing

[Source: https://www.tatvasoft.com/outsourcing/2023/07/what-is-system-testing.html]

System testing involves of performing some test case which trying to simulate real scenarios, user interactions and system behaviour in order to validate the software functionality in different usages. Testing team members check that the software does what it should, it's data is always correct, and the user has choose the appropriate option. Team members able to realize system testing can identify defects including the system integration, data flow, and user interface which are essential to software analysis and leads to the best performance (Drabick 2013).

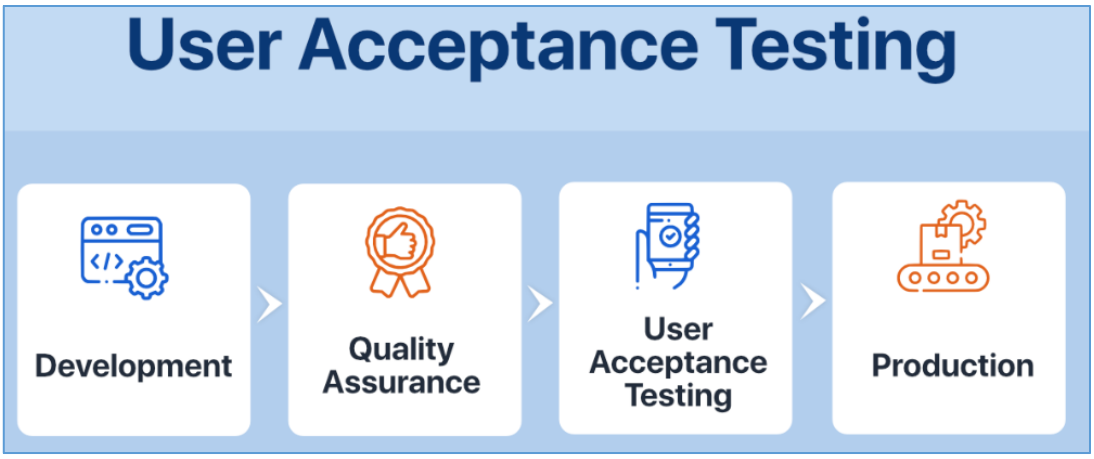
Automating tools and frameworks like Selenium for web applications and JMeter for performance testing allows teams to automate operating the system test execution and enable an accurate user experience. Such tools provide opportunity for test script writing, test execution and test report generation that can be manually performed. QA professionals are able to validate the system functionality better and track defects faster through these tools. Through the automation of system tests, coding teams can accomplish better coverage, less effort in manual labor, and the ability to identify problems in the building cycle early (Drabick 2013).

Ultimately, system testing is the prevailing software testing technique which one might use to demonstrate the software quality and support the software functioning within promised environment in case it corresponds to user's requirements. Thorough system testing by the development teams aims to prove the sustainability of the software and its compliance to functional, execution, security and usability prerequisites, in order to develop software products that outcomes users' expectations in an effective way.

3.3.4 Acceptance Testing

Acceptance testing being the final stage of software testing, it serves the purpose to tell us whether the program is fit for the user needs and deployment after testing. Execution of test cases based on user scenarios, business use cases, and acceptance criteria is implied performing by the end users at the start of acceptance testing, which result in the successful working of the system from the perspective of the system users. Through acceptance testing, which is a stage, it happens to ensure that software is aligned with business goals, user needs, and needs to pass regulatory requirements giving confidence that the future production software is ready for deployment in production (Pandit, Tahiliani 2015).

Tests of acceptance vary with whom the acceptance is made of and the testing environment involved in those tests. Alpha testing is testing an intermediate product internally by internal testers or testing it among beta users. The main purpose of beta testing is to run the same software outside the organization for a limited group of users to gather feedback and drill issues related to real-world usage. The user acceptance testing (UAT) is integrated into the software validation process, as it is designed to make sure that the them effectively match the user requirements and cases. Testing verification is the first one on the acceptance testing list. It is known as Operational Acceptance testing (OAT), and it involves performance testing, security, and supportability verification as well (Pandit, Tahiliani 2015).



Picture 7: Acceptance testing
[Source: https://www.xenonstack.com/insights/what-is-user-acceptance-testing]

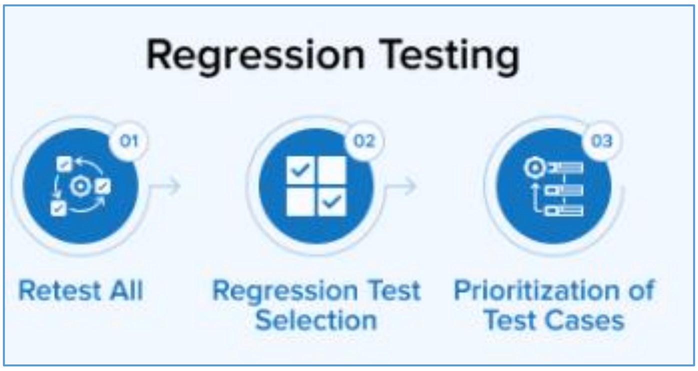
Examples of acceptance testing scenarios are writing and executing tests that are based on user stories, business workflows, and acceptance pre-requisites, which are used to confirm the correctness of functions, ease of use, and conformity to users' expectations. The operators and end users become the partners while the software companies undertake the acceptance testing which therefore guarantees that the software not only meets the business objectives but also users needs as well as the stipulated regulations before deployment. Users acceptance testing drives the decision as it provide feedback that prepares software for release into production and teams can address and make improvements to ensure the quality maintenance and meet user expectations fully (Pandit, Tahiliani 2015).

Automated acceptance testing software and frameworks, particularly Cucumber that supports behavior-driven development (BDD) and TestRail that manages test cases, permit you to convert the running of acceptance tests into an automated process which consequently improves the quality of the testing process. These instruments help the developers waste no time in the brainstorming of complex test scenarios, in the running of test cases, and in generating test reports too, all for the sake of the end-users and other stakeholders. Automating acceptance tests enables teams to reach higher levels of test coverage, reduce manual workload and ensure that the software function at least the way users demands (Hambling, Van Goethem 2013).

In the end, acceptance testing is an indispensable competency for software testing that rejects deployment of all software unless it has been meeting the needs and expectation of both the users and the business, passing all regulatory checks. Through rigorous acceptance testing including involving stakeholders, end users, and quality assurance department, organizations can successfully launch the software that meet the stakeholders' purpose, fulfills all the user requirements, and meets the user needs effectively delivering the value to the customers, which in turn, leads to the successful deployment of the software and ultimate user satisfaction.

3.3.5 Regression Testing

Regression testing is an essential software testing module that involves re-checking software after changes such as update and modification have been introduced to ensure that the new additions not carry along any defects or functionality issues. The new features are added, the bugs are pinpointed, and rewriting the code is undertaken during the development phase. As a result, the software can produce unbeknownst side effects or regressions. Regression testing in long line for solving the problem of risks by confirming that the recent changes in the software have not influenced the performance, features, and stability of the system negatively (Falah, Marghabi 2014).



Picture 8: Regression testing
[Source: https://www.tatvasoft.com/outsourcing/2023/07/what-is-system-testing.html]

Regression testing, which is one of the types of testing, aims to fix recently made changes to the code by rerunning previous test cases and making sure that no new bugs were introduced or the existing one were not broken. Tester's attention is generally on code changes that were made, refactored, or merged with some new functionality. They ensure that the modifications will not lead to any unintentional errors and consequences in other parts of the system. One of the key benefits of regression testing is that performing this test procedure on a regular basis helps teams to maintain the quality of the product, as well as to prevent regression issues emerging and ensure that users receive a shippable product (Falah, Marghabi 2014).

Regression testing tools as well frameworks that are automatically executed by the staff and the process of control is shortened. With these tools, testing is done effectively and test cases can be rerun, actual results checked for errors and if there any inconsistencies with expected results identified. With the use of an automated regression tests, the team can speed up the testing cycles, improve the test coverage and detect any problems in the early stages of the development processes are, raking the probable regressions to affect the software quality (Blokdyk 2018).

A small example is regression testing used for keeping the integrity and stability of the code related to the software development. It will help in the quality improvement of the product in multiple cycles of development. Ideally, regression testing is performed at every stage of software development life cycle. It makes software teams check that newer modifications/enhancements don't damage existing operations, slow down system performance, or cause user experience problems. Regression testing is a paramount part of continuous integration and delivery: it ensures the labile nature of the software and endows the teams with a possibility to deliver good software products with the highest frequency (Blokdyk 2018).

As the conclusion, regression testing is a vital software testing process by means of which a particular team could validate about recent changes in code, prevent regressions, and sustain overall quality of the product during the development cycle. Through the way of careful regression testing regularly and making use of automation testing packages, teams may reveal bugs at the moment they occur, keep the code stabilized and release the software products that satisfy practically the users' expectations..

3.4 Agile Project Success Metrics

The agile success metrics constitute the benchmark against which the agility of project management practices can be assessed, hence, attention to them is required as a way of ensuring projects are aimed at the desired objectives efficiently. This critical data helps teams to condense process, move closer to the range of expectations, and make decisions that are fact based, as well as resulting in better project outcomes. Agile Project success metrics represented by a number of main performance indicators (KPIs), providing a measurement of project progress, team interaction, product quality and customer and stakeholders satisfaction at all the stages of a project (Davis 2015).

3.4.1 Velocity

Hour is a central measuring agile project success rate agile that show how big amount of projects the team completed on the undertaken sprint cycle. The best practices determination view gives a clearer perspective of the team's efficiency, capacity, and delivery rate for user stories or tasks in a sprint. Velocity is measured by summing up the total story points or effort estimates of closed user stories and completed tasks completed within a sprint. Accumulation of these results through several sprints will provide teams with the base which is reliable for, planning further tasks, estimation of remaining workloads and delivery forecasts as well as identifying probable points capable of limiting progress or making it inefficient (Juricek 2014).

Over time, velocity can serve the team as a valuable indicator of performance and some primary stakeholders will look to it to assess team's capability of timely delivery and consistency. It provides for teams to organize sprints well, set their targets attainable, and adjust planning to on their the past performance. Speed of development is a rather complex metric that transforms along with under absorption of improvement by the team, change of the processes, and optimization of workflows. The analysis of these overarching velocity trends can reveal certain patterns, thereby helping the teams in improving the estimation accuracy as well as taking data driven decisions to the betterment of projects' criteria and productivity (Juricek 2014).

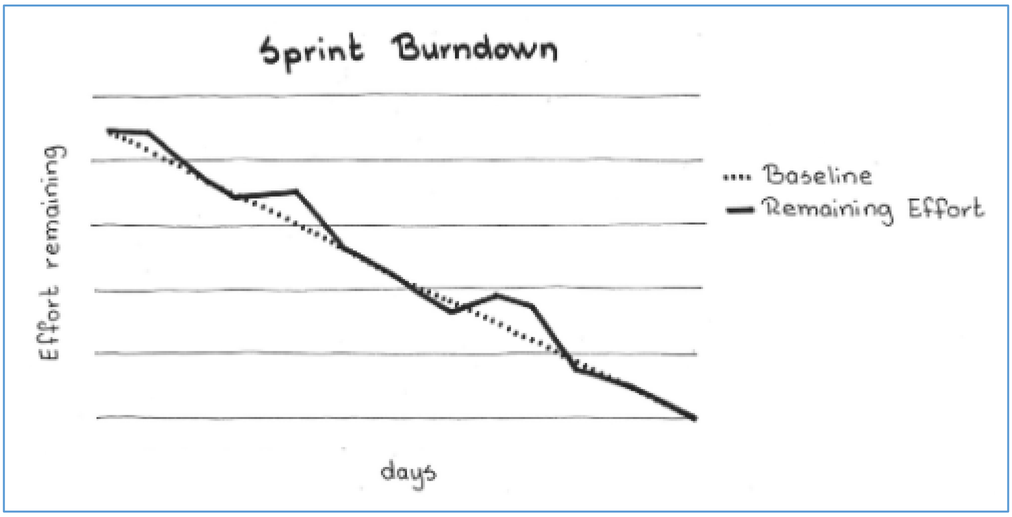
Velocity is not a measurement of speed of progress of individuals but rather a common team metric that represent the team's joint attempt towards accomplishing the milestones of each sprint. Transparency, accountability and continuous improvement of the team is promoted in this approach as the team is kept up to date work planned and done in each sprint. In short, being agile helps to adapt the team to changing circumstances, therefore resolving scope overload and setting a pace which can be sustained during the entire project timeframe (Agile Practices and Impacts on Project Success 2021).

To get the velocity for the sprint, teams add up the the story points or the effort estimates of all the tasks or user stories that were completed for the sprint. This total flow of sprint as a whole assists in sprint planning, capacity forecasting as well as release planning. Adopting a regimen to track optimal velocity and employing feedback from reviews will make refining estimating techniques, optimizing the workflow and enhancing the overall project performance possible (Cohn 2005).

In summary velocity is one of the success metrics which is so significant in agile projects. It's the metric to evaluate team performance, capacity and progress in the process of delivering the work within sprints. With this growing velocity being measured all the time, teams can fine-tune their planning, deliver products and services in a better way, and reach out to their stakeholders optimally. Velocity holds importance as agile squads take it as a metric to refer and indicate the pace of project progress, adjustment to altered requirements, ease of execution and overall project success in an ever-evolving environment.

3.4.2 Sprint burn-down Charts

Sprint Burn-down Charts are an agile project managements visual tool that allows tracking of work completion rate in the course of a sprints time frame compared to the initial estimation. These charts clearly present all the information as to how team performs and shows what we've done so far in sprint to achieve goals and complete the planned work. Sprint Burn-down Charts act as a tangible display of what the team has accomplished, and help identify trends, and provide the basis for making prompt and data-driven decisions regarding the goals and objectives of the sprint. The purpose of burn-down charts is the visualization of the team's progress to accomplish the missions in the given time frame (Abrahamsson et al. 2008).



Picture 9: Sprint burn-down chart

[**Source:** https://scrumbook.org/value-stream/information-radiator/sprint-burndown-chart.html]

The Major goal of Sprint Burn-down Charts is to project the Number of tasks remaining during a sprint using remaining work versus time chart. A comparison between the perfectly-plotted ideal work completion line (based on the plans and schedule) and the one that reflects the actual work carried out by the team will quickly show whether the team is on track to completing its sprint goals. Sprint Burn down Charts visualize the standpoint of the team enabling the team to detect deviations, prioritize, and perform the needed corrective actions to be sure that the team remains focused and the work expected is delivered within the sprint timeframe (Abrahamsson et al. 2008).

Sprint Burn-down Charts typically consist of two main axes: vertical axis represent the items of work to be done (usually in batting points or tasks. And the horizontal represents the sprint time (usually in days). The best graph of the task is the graph that is gradually going down from the upper left corner (total planned work) down to the bottom right corner where there is the zero left behind undone work. The work completed is actually tracked every single day so that the sprint is progressing towards a particular goal and one that doesn't have any deviations except for the ones that are ideal (Dalton 2018).

Sprint Burn-down Charts offer several benefits to agile teams, including:Sprint Burn-down Charts offer several benefits to agile teams, including:

- ✓ **Transparency and Visibility:** Sprint Burn-down Charts give a transparency into the team's progress because it is easy for everyone who participates, including the team members, clients, and product owners, to realize how the work is going, and if necessary the changes are needed (Dalton 2018).

- ✓ **Early Issue Detection:** By means of the visualized load that shows a work done versus a work to be performed, teams can spot issues relatively quick; this allows them to make decisions on the spot about how to deal with the problems or make adjustments to the scope that would obstruct with the sprint delivery (Dalton 2018).

- ✓ **Performance Tracking:** Agile Sprint Burn-down Charts serves as a monitoring and measuring tool for the team, allowing them to evaluate their speed over time and adapt their workflow to get the most out of the team through productive work in the long term (Dalton 2018).

- ✓ **Communication and Collaboration:** The Sprint Burn-down Chart helps in the teamwork among the members by providing them with a common visual description of the progress and consequently stimulating the accountability of each one as well as developing a culture of constant learning (Dalton 2018).

One of the ways team can perceive Sprint Burn-down Charts is by looking at the real work progress line with the Iwcll line, and the issues based on the slope of the real line. An almost vertical slope represents great progress, on the other side a flat slope could either pose a challenge to your progress or make you go slow. Hitting off the main line might mean that the scope changes, there are some work obstacles encountered, the original tasks were underestimated, everything that requires another course of action, rectification of the tasks, reprioritizing of the tasks or adjusting of the sprint commitments (Dalton 2018).

Sprint Burn-Down Charts are invaluable agile project management metrics which help the teams to monitor their stage of work, track the plan, and eventually deliver on time their sprint’s goals. One of the best ways to benefit from sprint burn-down charts is that they allow the team to build up transparency, identify problems right at the beginning, which leads to the optimization of their workflow and helps to get all project work done successfully in dynamic and iterative development environments (Abrahamsson et al. 2008).

3.4.3 Customer satisfaction

The customer satisfaction is one of the key indicators of agile project success which indicates how well the product, developed according to the requirements and expectations of customers, represents what is required. In agile project management, end users are the focus of activity with a view to confirming the existing worth of the product, bringing to the forefront the customer needs, and carrying on with refinements of the product based on clients and users response. Customers' only satisfaction is one of the key points of agile teams which helps them improve product quality and user experience. Also agile teams have the opportunity to strengthen their relationship with stakeholders (Aguanno 2005).

Finally, importance of customers have the tendency to stay longer in business resulting in successful projects' outcomes and customer retention.

- ✓ Customer satisfaction is essential in agile project management for the following reasons:Customer satisfaction is essential in agile project management for the following reasons (Aguanno 2005).

- ✓ Validation of Value:This is because customer satisfaction offers third party validation of the product value, it serves to say that the product does solve real world problems and it does deliver actual benefits to the customers (Aguanno 2005).

- ✓ Feedback Loop: This is how feedback from satisfied customers becomes a feedback loop for agile teams to get rid of the shortcomings, to delineate the need of features or enhancements, and prioritize them based on customer feedback (Aguanno 2005).

- ✓ User-Centric Approach: Customer satisfaction regarded as the top priority accommodates for the product that remains user-driven, convenient and abreast with expectations of the users driving their adoption and attention (Aguanno 2005).

- ✓ Stakeholder Engagement: Customer satisfaction at desirable levels improves in between the developers, the purchasers, and the stakeholders, trust, and collaboration are built (Aguanno 2005).

Customer satisfaction can be measured using various methods and metrics, including:Customer satisfaction can be measured using various methods and metrics, including:

- ✓ Net Promoter Score (NPS):NPS, known as the Net Promoter Score, is widely used compliant that is well-known as it measures customer loyalty and satisfaction based on how likely customers are to make recommendations to their friends and family (Highsmith 2009).

- ✓ Customer Surveys:Surveys are great also allowing agile teams to receive feedback, evaluate satisfaction levels of online users and locate improvement areas for according to results of the questionnaires (Highsmith 2009).

- ✓ Feedback Sessions: These personal interactions with clients and users allow to gather any feedback that they may have and use it to improve upon the shortcomings, solve problems and enable better product performance (Highsmith 2009).

- ✓ User Testing:The test sessions of users come in handy for the teams in observing how the users use the interface, get feedback on usability, and relate the product features to expectations of the user (Highsmith 2009).

Prioritizing customer satisfaction in agile projects offers several benefits, including:Prioritizing customer satisfaction in agile projects offers several benefits, including:

- ✓ Enhanced Product Quality: Customer satisfaction helps to make quality products because customers only become satisfied when the product they used fulfills their needs, works as it is being intended, and delivers a good experience (Highsmith 2009).

- ✓ Increased User Adoption:The dissatisfied customers will go back or to other brand, resulting in the system failure but when these customers are satisfied, they tend to stay and use the product, that causes the higher user engagement, retention, and of course success of a product (Highsmith 2009).

- ✓ Continuous Improvement: Customer feedback becomes a staple for the continuous improvement of the product. Development teams keep close attention to the client's preferences and requirements because these impact the company's decisions on feature prioritization and iterative development (Highsmith 2009).

- ✓ Stakeholder Confidence:Low customer satisfaction levels can have a significant impact on stakeholder confidence. This can also hinder the trust in the development team and damage the relationship with key stakeholders which are vital for success (Highsmith 2009).

To enhance customer satisfaction in agile projects, teams can implement the following strategies:To enhance customer satisfaction in agile projects, teams can implement the following strategies:

- ✓ Regular Feedback Loops:Sustain continuous connection with stakeholders as well as users to get feedback and to identify challenges, alleviate fears and to maintain relevance of products (Highsmith 2009).

- ✓ User-Centric Design: Use a user-driven design method so that the product is understandable, requires no effort to use and integrates flawlessly with user psychology (Highsmith 2009).

- ✓ Iterative Development: Use sprints with regressive design as a tool for receiving feedback, introducing incremental improvements, and providing customers with a valuable product in each sprint (Highsmith 2009).

- ✓ Collaborative Culture: Develop a cooperative environment that respects the customer's opinions , covers the transparency, and brings in common responsibility to meet client needs (Highsmith 2009).

In the last analysis, customer satisfaction is an essential success metric in agile projects that are turned into products which satisfy customers, build up their involvement and service stakeholders. The agile teams get the customers to focus on the satisfaction of the users as the first priority, which helps them determine the value of the product, gather the necessary feedback and deliver the quality software, meeting customer’s expectations entirely, thereby leading to the success of projects and customer loyalty.

3.4.4 Cycle time

Cycle time, a key agile metrics able to measure, signifies the time taken to complete one unit of work that ranges fro muser story to task from the beginning to its finish. In agile project management, usual cycle time lets you figure out the effectiveness of development process, detected the weak parts and lead to process improvement. Analyzing the cycle time of a project lets teams discover inefficiencies and bottlenecks in workflow, it also allows them to deliver value to the clients earlier which improves projects performance & productivity as well (Craddock et al. 2019).

Cycle time is essential in agile project management for the following reasons:Cycle time is essential in agile project management for the following reasons:

- ✓ Efficiency Measurement:Cycle time could mock the efficiency of the development pipeline and can help in revealing places where the process can be made more efficient (Craddock et al. 2019).

- ✓ Bottleneck Identification: Cyclical balancing is a main method through which cycle time is tracked. By tracking bottlenecks, delays, inefficiencies, and any other problems that might hinder development processes, teams can find solutions and improve the cycle flow (Craddock et al. 2019).

- ✓ Predictability:The duration of a cycle time is used by a team to forecast how long it will take workers to complete the tasks, thus, enabling them to accurately plan sprint schedules and to set appropriate deadlines for a project (Craddock et al. 2019).

- ✓ Continuous Improvement: A culture of improvement will thrive in the overall monitoring cycle time that will result in teams eliminating waste and adding value to the customers by delivering things in the right time and without mistakes (Craddock et al. 2019).

To improve cycle time in agile projects, teams can implement the following strategies:To improve cycle time in agile projects, teams can implement the following strategies:

- ✓ Visualize Workflow: Study the development process by defining transitions between phases and tracing the flow of work products to identify bottlenecks and figure out how to improve the process (Craddock et al. 2019).

- ✓ Reduce Work in Progress (WIP): Emphasize single – task mode to complete the work items fast, avoid the jumping from one to another, and minimize only the changes of context, in order to minimize cycle times (Craddock et al. 2019).

- ✓ Automate Manual Tasks: Automate repetitive tasks by using CI/CD automation tools, deploy a container-based workflow, follow version control practices, and use tools such as GitHub and Jenkins to ease the development process and reduce cycle times (Craddock et al. 2019).

- ✓ Iterative Improvement: Adopt a system of continuous improvement and change by trial, getting the opinion of others and routinely tuning the workflow course to reduce cycle time and improve project output (Craddock et al. 2019).

In sum, cycle time is a real agile project success metric that evaluates a process efficiency, reveals bottleneck pockets and makes a productive change in agile models. When cycle time monitoring, workflow optimization, and Lead time reduction strategies are embraced, project teams will be able to hasten the delivery of value to customers, reduce risk, and undertake projects successfully.

3.4.5 Quality Metrics

Quality metrics are very key to gauging the effectiveness of the agile projects in which it will be analyzed whether the product quality is achieved, defects are found or software reliability is ensured in such environments. Defect metrics ensure that team evaluation associated with testing effectiveness, tracking problem data and maintaining product quality standards are embedded inside the product lifecycles. With the Quality metrics monitoring, the teams are able to make a proper prioritization of the tests, get better code quality and generate software products that satisfy customer needs, be secure, performance, and usability issues effective.

Quality metrics play a crucial role in agile project management for the following reasons:Quality metrics play a crucial role in agile project management for the following reasons:

- ✓ Defect Identification: The application of a stringent quality measurement set provides teams with an opportunity to spot the defects, bugs, and issues in the early stages of the development process, ensuring that they are managed efficiently such that they would not interfere with product quality (Craddock et al. 2019).

- ✓ Performance Evaluation: The quality benchmarks scrutinize functions in terms of performance, security, and usability. They offer perspectives into areas that need improvement through testing and other measures to make quality software (Craddock et al. 2019).

- ✓ Risk Mitigation: Through the quality metric monitoring, teams may mitigate risks, resolve vulnerabilities, and maintain the quality using various charts to present different aspects of quality that may concerns user experience, system stability or security (Craddock et al. 2019).

- ✓ Continuous Improvement: The metrics of quality strive towards directive practices that teams could use to optimize their testing procedures, develop code within acceptable parameters, and deliver final software products that consumers highly treasure (Craddock et al. 2019).

Some common quality metrics used in agile projects include:Some common quality metrics used in agile projects include:

- ✓ Defect Density: Defect density shows the number of imperfections discovered for every piece of code written. Thus, enables the quality assessment of code, pinpointing error-prone areas and delivering the feeling for where the biggest emphasis on testing should be placed (Augustine 2005).

- ✓ Code Churn: The code churn takes track of how often the code changes are made, either deleted, added or replaced, which will determine the stability of the code base, the hardship of the code changes and the chance of having mishap (Augustine 2005).

- ✓ Test Coverage: Test coverage helps the team to measure the percentage of code or requirements gauged by the automated tests, thereby assisting them to find out if the testing was effective and identify the test coverage gaps. This ensures that comprehensive testing is carried out (Augustine 2005).

- ✓ Code Review Metrics: Coding metrics include the number of code reviews, the turnaround time, and problem-finding rates in the reviews, and help to improve the code quality and knowledge sharing process among the development group (Augustine 2005).

To leverage quality metrics effectively in agile projects, teams can implement the following strategies:To leverage quality metrics effectively in agile projects, teams can implement the following strategies:

- ✓ Define Clear Quality Goals: Set up the clear quality targets, and determine the main quality metrics. Design the testing procedures in such a way they align with the quality objectives so that the software would meet the performance, security, and usability requirements (Augustine 2005).

- ✓ Automate Testing Processes: Elaborate testing methods including automation, continuous integration and deployment, and utilize testing tools to facilitate automated testing, improve test coverage and increase the code quality (Augustine 2005).

- ✓ Track Trends and Patterns: Analyze quality metrics in a timely manner, monitor trends, find relationships, and comment whatever attitude is shown through data to control testing activities, resolve recurring problems, and carry out quality control (Augustine 2005).

- ✓ Collaborate Across Teams: Facilitate collaboration between the boundaries of dev, test, and QA teams to share information, align on quality standards, and work together to provide the end users with high-quality software solutions that fulfill their needs (Augustine 2005).

Generally speaking, a quality metric is a key measure for an agile project that tells whether the quality of the product is right, detects mistakes, and keeps the reliability of the software in agile environment. The agile teams deliver high-quality software products which users highly value now, due to a sophisticated monitoring of quality metrics, the prioritization of the most important testing areas, and the continuous improvement. As a result, a number of end users' requirements are met and the interactive satisfaction of clients is achieved.

3.5 Integration Points of Testing Strategies and Agile Methodologies

Points of integration between agile methodologies and testing techniques are a matter of the high level of harmonisation on the well coordinated work, continuous feedback and great software delivery in agile projects. Despite the ever-evolving nature of software development and the need for fast response and fulfillment of the constant changing and the requirements of the users, teams can align the testing approach with agile principles and technologies to maximize the testing efforts, get a quality product, and accomplish the projects with success. Testing strategies co-existence with agile methodologies cultivates a culture of delicate communication, disclosures, and constant improvement that empowers teams no just to deliver value to users efficiency, but be capable to promptly tailor to new demands.

- ✓ Early and Continuous Testing: Agile focuses on testing in the early and continuous mode during the life cycle as this helps to find bugs still in the development phase and gives instant feedback and finally deliver the software that meets the quality standards (Linz 2014).

- ✓ Collaboration: Knowledge sharing and alignment of quality goals collaboration, and work on together are some of the things that Agile methods promote (Linz 2014).

- ✓ Adaptability: Agile embraces the art of bargaining through its change-centric approach while continually incorporating feedback and adapting the testing strategies as and if the needs or priorities of the project changes (Linz 2014).

- ✓ Iterative Development: Agile utilizes recursive development units based on small releases that allow teams to conduct testing with every iteration, improve and modify after sharing prototypes, and provide customers with partial products in short increments (Linz 2014).

Integration Points

- ✓ Test-Driven Development (TDD): TDD is a test strategy which is in line with the agile approach because it involves designing the tests before writing the code to evaluate the program using code that is in accordance with the existing design for the final product. TDD is a methodology of writing test cases and modifying code part to achieve quality and accurate work (Özçelik, Turgay Altilar 2015).

- ✓ Continuous Integration (CI): CI comprises a set of practice that includes commit small changes often, apply automatic tests frequently, and run code quality assessment regularly. CI in that agile approach causes the team collaboration, early defect discovery, and continuous tester integration agenda to the process of development (Duvall, Matyas, Glover 2007).

- ✓ Behavior-Driven Development (BDD): BDD is a collaborative way to pass on requirements of the stakeholders by using the keyword specifications describing the requirements in their natural language. The DDD BDD does not only serve as a means of communication, comprehension checks of requirements and automation test cases but it also translates well into an agile methodology (Lawrence, Rayner 2019).

- ✓ Exploratory Testing: Exploratory testing serves as a natural partner to the agile approach as it stimulates testers to point out the unfound flaws, derive the useful implications for the upcoming changes, and comment on the interface issues or problem spots in no time. Exploratory testing in its essence is what gives a tester an edge by allowing them to swiftly react to changing requirements and overseeing software behavior occurring (Whittaker 2009).

Benefits of Integration

- ✓ Improved Collaboration: In a situation where people from different departments work together on agile models, this integration of the testing strategies, on the one hand, promotes cooperation, and on the other hand, elevates the level of communication, thus giving rise to a common understanding of the shared goal to develop high-quality software products (Agarwal, Tayal, Gupta 2010).

- ✓ Faster Feedback Loops: Agile methodology of software development arises with devising techniques of testing which result in the faster process of feedback loops in code enhancements, requirements validation, and quality assurance of the product, leading to rapid iterations and continuous enhancements (Agarwal, Tayal, Gupta 2010).

- ✓ Enhanced Product Quality: Integration aspects between testing practices sets and agile approaches empower better quality products of end-users by choosing what to test, increasing efficiency, and delivering strong software that will work for end-users (Agarwal, Tayal, Gupta 2010).

- ✓ Adaptability to Change: Blending testing strategies into agile processes that utilize responsiveness and adaptability features hastens the process of adjusting test approaches and responding to ever-changing requirements so that customers' values are always realized (Agarwal, Tayal, Gupta 2010).

A seamless approach for testing procedures and agile strategies is the key factor in the comprehensive testing optimisation, collaboration, and competent product delivery in agile projects. Teams can leverage agile principles, for example, early testing, collaboration, adaptability, and continuous application development cycles, to achieve high performance in open-ended and the fast-paced business world.

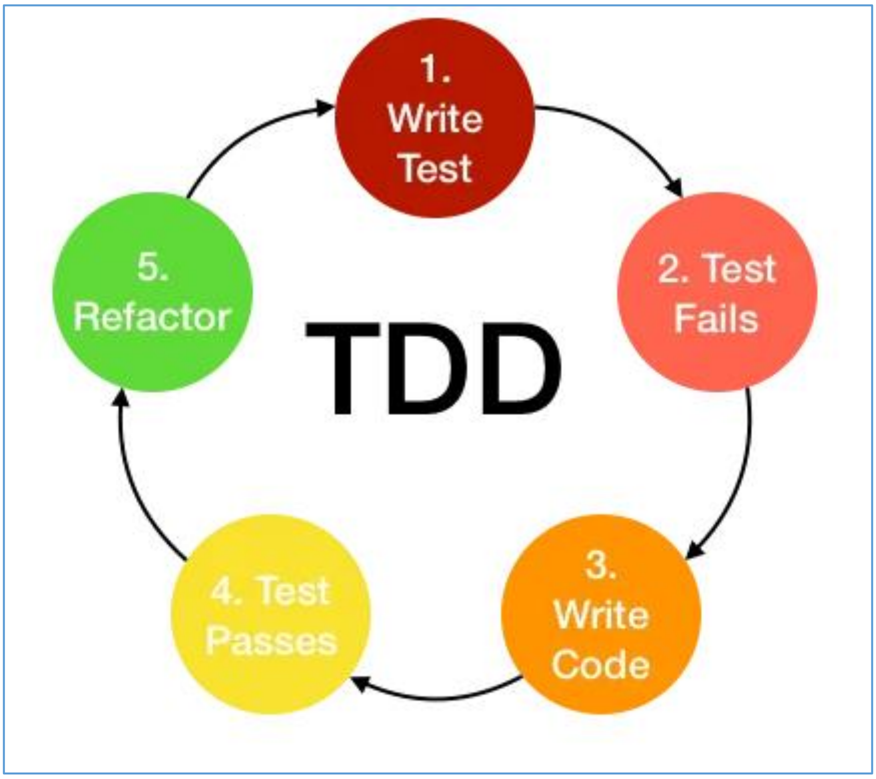
3.5.1 Test-Driven Development (TDD)

A TDD approach to software development is the one that comes as a "package" of test driven development that envisages writing tests prior to writing the actual code. In TDD, the developer starts by creating a failing automated testcase whose purpose is to make a sound that is free from the faultiness inherent in the code. It is also able to enlighten the process, assign the tasks and eventually velocity the development into approved criterion. TDD follows the red-green-refactor cycle in which developers first write a failing test (red), then implement the code appropriately to pass the test (green) and as post test refactoring, they improve the design of the code without changing the behavior of it (Beck 2022).

Another appealing part of TDD is that it allows to enhance the quality of the code and makes it easier to update/maintain it. By implementing “tests first” method the developers write code to solve specific issues and write testable code. These techniques resulted in the production of well structured and modular code that is simple to maintain and expand. TDD practices and mindset also plays a crucial role by promoting the test-centrial philosophy that makes developers to front-load the error circumstances, boundary cases, testing the design as well as overall code quality for building more stable and fault-tolerant software (Beck 2022).

While embracing TDD, such practice practically bonds into agile methodology by enforcing tests on early stages, ensuring frequent feedback, and speeding up iterative development. Agile teams not only collect feedback on changes to the code but also make use of TDD to validate new functionalities in the incremental stages and to avoid functions breaking old functionality. TDD has the ability to be integrated with agile concepts which help in achieving business goals by providing needful collaboration, adaptability, and customer focus through the requirements alignment, shared understanding and customer value creation by delivering working software (Beck 2022).

TDD also acts as a sort of error identification and defect fixing feature that takes much lesser time to complete. The units' writing of tests prior to the code incorporation leads to the bugs detection earlier in the development life cycle, which makes it easier for the issues being promptly identified and deleted. TDD not only reduces endless track down of regression bugs but also helps to create stable code and faster process of fixing bugs prevents overlong debugging process which lead to the faster development cycles and, the higher overall product quality (Beck 2022).



Picture 10: Test driven development

[**Source:** https://medium.com/@luisfmachado/swift-test-driven-development-tdd-810add46a1b9]

Finally, elaborate that TDD is a such programming practice that reaches code quality, maintainability and fault prevention. With TDD being in the flesh of agile methodologies, teams may achieve intensified collaboration, a product of high quality, and rapid requirement adaptable to changes. With TDD's emphasis on automatic testing, incremental development, and continuous improvement, it proves itself to be a useful technique for agile teams that like to work towards solving clients' unmet needs through reliable, customer-centric deliveries executed efficiently.

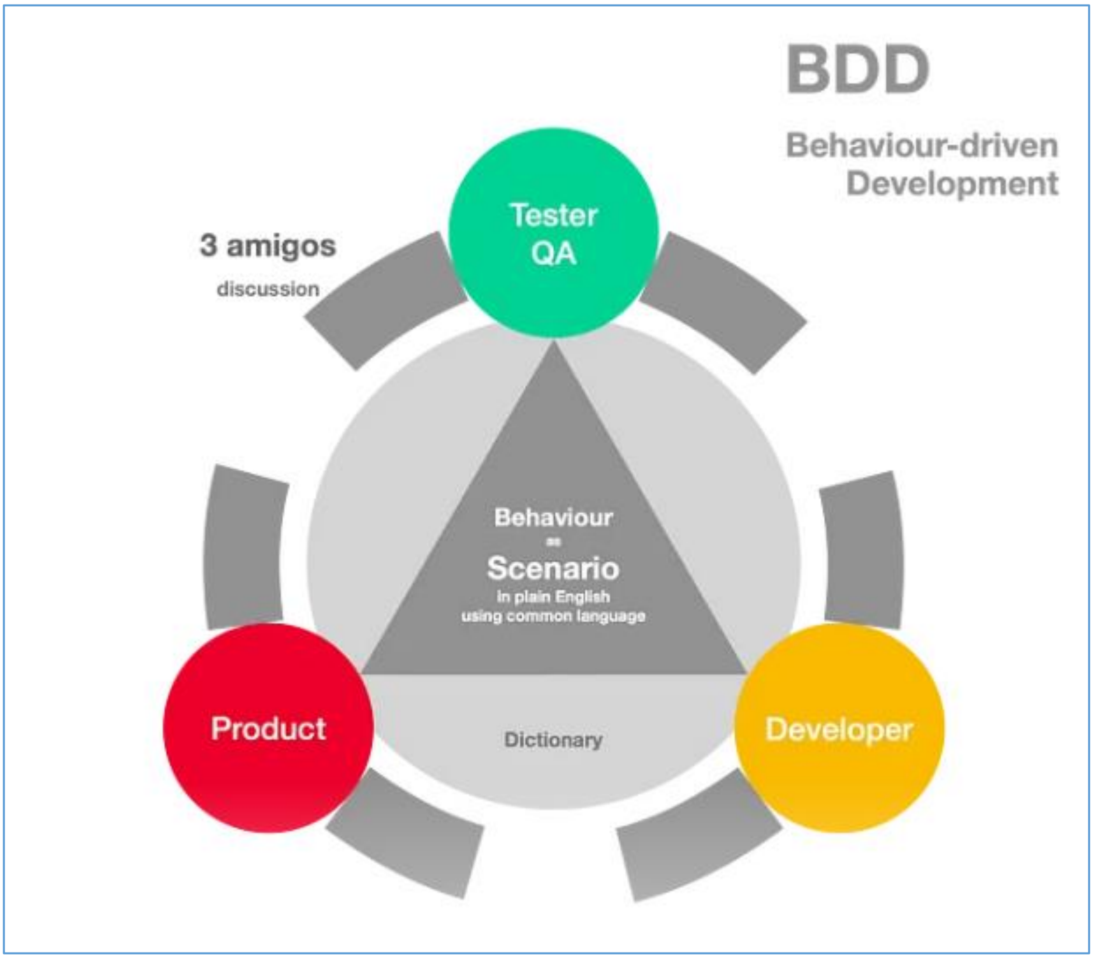
3.5.2 Behavior-Driven Development (BDD)

Behavior Driven Development (BDD) is a software development process that concerns itself with how a system should behave for particular business needs. It has the aim of dealing with problems arising from none of the requirements well specified and also, to making business and quality assurance tasks aligned. BDD entails involving key stakeholders, the triplet of the "Three Amigos" team consisting of the " Business, Development, and QA teams, who are to be defined the needs and validation of the behavior of the software, expected. BDD has three core principles; one is centered on results only, another is about teamwork and the idea of understanding one another and expressing ourselves using the same language (Smart, Molak 2023).

BDD is an integration of TDD methodology which targets Behavior-driven development – goal of Behavior – driven development is not focused on test coverage, but rather on defining the application's behavior. There is human-understandable behavior in BDD that is categorized in a predetermined pattern of how an individual act or behave that is normative. These behaviors will be spec-ed out in "feature files" which are integrated in the software development process. The codings like Cucumber and SpecFlow will parse through these behaviors and execute corresponding "step definitions" or "glue code" (Smart, Molak 2023).

The BDD process - simulating a play - can be used to describe how the application should behave using various kinds of cue. This process is supported by verification, which takes place during making software. The developed application has to be in accordance with the specified requirements and behave as intended. Unlike others in which the structure of the scenario does not have to be very specific, BDD focuses on the provision of clear, concise and precise scenarios to foster collaboration and comprehension of team members which in turn positively affect the software development process (Smart, Molak 2023).

On the other hand, the part in which BDD functions as a bridge between technical and non-technical stakeholders in the software development process deserves mention. is accomplishes by ensuring effective communication and understanding among these two groups of people. By actively promoting cooperative efforts among the various roles, business analysts will be able to the issue to be fixed, in order to keep the development activities in line with business needs and expectations, as well as with the needs and expectations of the users. As compared to waterfall, it will provide the way to rapid iterations, additional feedback, and better value flow that may lead to the great code quality and lower maintenance costs in the end (Smart, Molak 2023).



Picture 11: Behavior driven development

[**Source:** https://newlight77.medium.com/behavior-driven-development-df555b9fe277]

3.6 Impact of Testing Integration on Project Success

The embedment testing practices within the development project lifecycle exists a great influence on the success of software projects. Allosteric testing of integration is a key contributor to end product quality, the prevention of errors, and the provision of value to the customers The achievement of this goal can be accomplished by synthesizing test actions with project targets, agile techniques and stakeholders' expectations; so that teams will be able to foster collaboration, improve code quality and meet project objectives even in a dynamic and time-constraining development environments (Riel et al. 2010).

The notion of testing early in the product's development helps teams spot errors, bugs, and issues here at the early stage and thus not much of the critical things emerging from the long run of the process. The importance of the quality assurance and the continuous testing, which leads to the removal of defects, will provide for users a software product, which users expect to perform reliably, and provides positive user experience. Beta releasing allows your targets to influence the process of incorporating new upgrades and also its finding of bugs, this leads to higher quality products that satisfy customers (Riel et al. 2010).

Integration testing speeds up a buyback chain of code amendments, validated demand for the product and quality assurance of the product. Such teams use successive iterative testing activities in each stage of their project implementation. They gather feedback, tackle issues quickly, and ultimately make a rational decision using available data in order to improve the workflow and achieve quality goals. Fast-paced feedback facilitates teams to tune to the new conditions elected with agility, repeat of features in a meaningful manner, and give value to customers in succession iterations. The feedback loop of testing and feedback as an iterative approach, helps in continuous improvement, brings innovation in and finally makes products properly evolve that satisfied the needs of the users (Riel et al. 2010).

Implementation integration using all development teams, testers, business owners, as well as other project stakeholders as assistants increases the overall sympathy of all the people involved in this process. By leveraging agile methodologies like Scrum or Kanban, teams can create an atmosphere in which the team members work in tandem, transparently, and share the accountability for the products quality. Effective communication on workers allows them to equally understand the projects goals, required resources and testing objective, thus the improvement on the harmonization, decrease in misunderstandings and enhanced teamwork. Integration testing promotes cross cl….ooperation, knowledge exchange and responsible performance toward reaching the common objective by the collective effort (Riel et al. 2010).

Inserting quality assurance activities into the project lifecycle provides teams with an opportunity to better understand and mitigate risks that are possible during the software development process which include: identifying bugs at early stages, fixing bugs promptly, and keeping software compliant with security and performance standards. Integration testing allows teams to make backlog based decisions on the basis of data information on testing events prioritizing and precisely devoting resources to the completion of the provision of the software of the greatest quality within the budget and time limits. Through the establishing testing up to the comprehensive procedures teams may well easily minimize delays, accounts for technical deficit and consequently enrich the project success (Riel et al. 2010).

However, to summarize, testing integration as a vehicle for quality inspecting the project products, preventing bugs, supporting cooperation on the development stage, feedback to smoke out any defects, and the control of risk factors is really vital. Through the setup of testing efforts from the initiation to the completed phase of the project, teams can optimize processes, raise code quality standards and hence deliver desirable and high-value software products that customers expect at a given time. Testing integration is a major factor of project success in modern day software development environments which in turn reward the efforts of maintaining continuous improvement, attracting fans and generating revenue.

3.6.1 Quality Metrics

Quality metrics act as advantageous assets while trying to audit and track the quality of the software in progress throughout the development cycle. Such indicators grant us access to detailed data indicating different quality facets of the product, and allow the teams to analyze and improve the quality of the software based on findings and provide the end user with the relevant software solution. By establishing quality metrics and tracking their results, teams in a project can figure out where their strengths and weaknesses lie, at what pace the project is progressing and use this to make improvements, which are continuous, for the overall project success (Basu 2015).

Quality metrics play a crucial role in software development for the following reasons:Quality metrics play a crucial role in software development for the following reasons:

- ✓ Performance Evaluation:Metrics of quality gauge the viability of software products by computing the system performance factors like response times, throughput, resource utilization, etc. thus helping in evaluating performance of software products. Thus the metrics give an overview of how far a system is from falling into the trap of being scalable in nature and the level of performance (Kan 2003).

- ✓ Defect Detection: By getting metrics that will instruct in finding defects, bugs, and other problems in the software. Using such metrics as defect density, open defect count, and the time taken to close a defect, teams can spot areas of weak performance and pinpoint what kind of fixes should get the most attention to strengthen product reliability (Kan 2003).

- ✓ Code Quality: Besides complexity metrics of the code like cyclomatic complexity, code coverage and duplications, help the developers to estimate how maintainable, readable and quality the whole codebase is. The metrics could be tracked by the teams in order to facilitate code quality improvement, technical debts reduction, and regular software maintenance (Kan 2003).

- ✓ User Satisfaction:Furthermore, one could contemplate the use of the quality metrics in the field of user-centric evaluations like NPS (Net Promoter Score), CSAT (Customer Satisfaction Rating), and also the user feedback analysis. This data offers insights on user experience, reliability and usability of functionality and it is based on response from users, indicating to the development team the areas that need to be improved or modified to meet the users expectations (Kan 2003).

Common Quality Metrics

Some common quality metrics used in software development include:Some common quality metrics used in software development include:

- ✓ Defect Density: Defect density demonstrates how many bugs reported per unit of code with such measure estimating the quality level and conveying the effectiveness of the tests performed (Basu 2015).

- ✓ Code Coverage: Code coverage is a metric to estimate the proportion of total code that has been exercised by the automated tests. The tests create a holistic view of the thoroughness of testing and areas with deficiency in test coverage (Basu 2015).

- ✓ Mean Time to Repair (MTTR):MTTR calculation is conducted by measuring the average repair time spent on defects or problems and helps to identify the level of efficiency in the defect resolution process as well as the team's response rate to some issues (Basu 2015).

- ✓ Customer Satisfaction:The metric options which could help teams assess user satisfaction levels and conduct such assessments, including Net Promoter Score or customer feedback scores, enable to understand user preferences and solve concerns, and ultimately, improve product quality (Basu 2015).

Utilizing quality metrics in software development offers several benefits, including:Utilizing quality metrics in software development offers several benefits, including:

- ✓ Data-Driven Decision Making: These metrics can serve as objective data to help teams to conduct meaningful analyses, come up with improvement strategies, and allocate the resources appropriately with the ultimate goal of yielding a better quality (Jones, Bonsignour 2012).

- ✓ Continuous Improvement:Teams can isolate trends, patterns, and deficits by watching quality rates over time, which gives them a solid guiding hand to show how they can continuously grow that software to best suit users (Jones, Bonsignour 2012).

- ✓ Risk Mitigation:The content above the line/divning line Matric Acronym or short form is indicative of **metrics**for quality which forms a vital foundation for teams in the process of risk-mitigation in a way they identify risks early stages, correct deficiencies on time, and manufacture software that is 100% secure as well as free from performance and reliability issues which might jeopardize the success of their projects (Jones, Bonsignour 2012).

- ✓ Stakeholder Confidence:The product of high quality software built on applicable quality metrics ensures credibility in stakeholders, develop trust in the development team, and guarantees success in project that satisfies customer through deploying features very fast (Jones, Bonsignour 2012).

Therefore, in the last word, quality metrics are the meaningful instruments which make us the process of assessing, monitoring and improving the software quality. With the aid of metrics of quality providers, teams are capable of betterment in the field of software product quality, defect reduction, performance optimization and delivering software products that fully satisfy user expectations and projects success is successfully achieved. Quality figures constitute the main tool of constant development, of data-oriented decision making, as well as satisfactory outcome from software development projects.

3.6.2 Time-to-Market

The time from inception of product to the release of the software, and also to fulfill market demands is called the time to market; this being a decisive success factor in software development. Harning the fast time-to-market is the key thing for keeping enterprise competitive, exploiting market opportunities and customer needs in the pace of nowadays business life. What contributions can development teams do then to meet these objectives? They can optimize and streamline their processes, and employ agile methodologies in order to reduce time-to-market and increase implementation speed, which, in general, boosts the team's chances of success in the market (Basu 2015).

Importance of Time-to-Market

- ✓ Competitive Advantage: Speed of a product's arrival to the market is an important competitive edge which gives a company a chance to quickly launch the new versions of current product, sequel or fix bugs before the competitors do it - which attracts more customers to the product and makes the sales bigger (Basu 2015).

- ✓ Customer Satisfaction: Fast response time with the best quality product gives the customers satisfaction, fits their ever-changing needs, and helps the brand to create loyalty among people by fulfilling their needs through offering new products and features at the right time (Basu 2015).

- ✓ Revenue Generation: The fast time-to-market increases the rate of product revenue generation, converting it into the efficient one at the market opportunities market time, and getting a faster money back on investment (ROI) investment by launching products fast and efficiently (Basu 2015).

- ✓ Innovation and Adaptability: Less time spent on developing new products leads to innovation, allows for experimenting, and enables teams to manage situations such as market conditions that might change or urgently mentioned clients’ feedback and various trends better (Basu 2015).

Finally, it is important for us to pay attention to the strategy of time-to-market improvement. This implies that firms adopt several techniques, which come at the level of the product, to promote more efficient transitions of products from the manufacturing process to the market.

- ✓ Agile Methodologies: With the aid of Agile methods like Scrum, Kanban, and Lean, teams are capable of bringing forth the end results in increments, prioritizing fits and responding to shifting requirements in a short span of time that ultimately results in a short development time and speeding up the market time (Basu 2015).

- ✓ Cross-Functional Teams: Working in teams with all the required skills and competence allows them to excel in teamwork, communication, and effectiveness in a cross-functional approach. This, in turn, leads to faster decision making, resolution of issues, and production delivery (Basu 2015).

- ✓ Continuous Integration and Deployment (CI/CD): Adopting CI/CD pipelines makes software testing, integration, and delivery stages automated that assist in introducing software updates frequently, detecting problems at the early stage of the process and, thus, developing applications rapidly (Basu 2015).

- ✓ Iterative Development: Necessity of continuous development cycles such as feedback, incremental changes and the final product is working successfully with the short cycles, where products grow based on user achievements and market application (Basu 2015).

- ✓ Market Responsiveness: Faster time to market facilitates reacting immediate to market changes, customer feedback, and competitor's pressures, keeping yourself ahead of evolving trends and customer needs (Basu 2015).

- ✓ Cost Efficiency: Shortening time-to-market, inversely, lifts the company's cost efficiency, relative ministering it to the unnecessary activities thus dividing attention & resources, leading to cost savings and enhanced profits (Basu 2015).

- ✓ Customer Engagement: The fast delivery of products contribute to customer engagement, invite user feedback, and the loyalty of the customers through providing swift solution of the problem, taking care of pain points and delivering additional value to the features, respectively (Basu 2015).

- ✓ Business Growth: This faster-to-market approach accelerates business growth as the firms are able to introduce products more quickly, expand into more markets, catch trends sooner and enjoy resulting revenue uplift and higher market share (Basu 2015).

Eventually marketing speedy cycle is viewed as an important factor not only for attracting business success but also for innovations and customers fulfillment in today's competitive environment. Through agile system development, encouraging communication, process mechanization and iterative development cycle, it is possible to boost the speed of product delivery, to become customer responsive and to achieve sustainable every day being work hard growth in the fast changing and rapidly transforming software development industry.

3.7 Evaluate Existing Tools and Technologies

The Agile methodologies have much reliance on the tools & technologies as these have the enormous power of ensuring consistency in the implementation of the projects as well as in blending testing tactics. These tools and technologies give the opportunity to evaluate their applicability to the Agile projects management and comprehensive testing techniques.

Delving into academic and industry papers, I believe I can come across a cornucopia of different tools and technologies that is pertinent to the Agile teams. With interactive dashboards, collaborative features and support for the Agile methodologies, leading project management open source software tools such as Jira, Trello, and Asana have gained popularity. These platforms allow for easier scheduling, task tracking, and sharing information in real time. In this way, the effectiveness of teamwork is increased while their transparency of the project is achieved (20 Best Agile Tools for Effective Management in 2024).

A lot of framework and tools innovate to be overset Agile processes with new problems which emerged under agile development models. Test automation frameworks such as Selenium, Appium, and Cypress let teams to execute the repetitive testing tasks with less manpower and simply the test execution that might otherwise be done manually. Furthermore, tools such as Behavior-Driven Development (BDD) facilitate coordination among stakeholders and developers through the capability of creating executable specifications that are easily understood at a glance (Irshad, Britto, Petersen 2021).

Additionally CI and CD (Continuous Integration and Continuous Deployment) tools, among which are Jenkins, Travis CI, and Circle CI, become instruments for development pipelines in Agile. Such democratization of building and testing of the deliverables reduces the manual actions; thus the teams are able to utilize them in faster software production without an increased risk of integration issues (Ali 2023).

Nevertheless, the variety and depth of the market offer endless options; but thorough analysis shall be done to choose suitable tools and technologies according to work requirements, team sensitivity, and firm objectives. The noise generation essential, that in acquiring every device, teams should rate on features, scalability, compatibility with current systems and maintenance over the long run in order to make a selection.

In the end, the analysis of current tools and technologies displays the role of getting ready for shifting to Agile systems and technically advanced schemes. These evaluations enable organizations to find additional ways how they can build on aforesaid activities to make them more efficient and deliver well-balanced software products which reach the level of their customers' expectations.

3.8 Research gap

In spite of increasing use of Agile software development methods and the innovation content testings, there exist the unclear mind which can intrigue the Agile project management integration of testing strategies. Although there exist many knowledge materials that share both Agile principles and testing methods respectively, still lack is the multi-disciplinary literature that focuses on their coherence and complementarities.

Current research in the area is often focused on either Agile project management or testing strategies separately, and the complexities of integration of these strategies with the project achievement metrics are not considered. Furthermore, case studies actually make significantly valuable insights on the real-world implementation, which is the reason surveys with higher structured designs (quantitative data) are required to research on the implemented practices, their current challenges, and perceived benefits.

Secondly, the research on tools and techniques made available for agile testing-integration into the work flow is inadequate. Unraveling the efficiency and pitfalls of existing instruments can give practical insight to those pros who strive to innovate their Agile Testing routines to the unexpected improvements.

So, basically, there is a research gap in the failures of completely studying and combining testing strategies within Agile projects, along with the reviews of their effect on success as well as the practical recommendations for practitioners. This gap can be covered by developing a comprehensive system which can be done by a combination of literature review, case studies, design of surveys and use of the relevant tools and technologies.

4. Practical part

This chapter enumerates the practical part of seamless testing strategies into Agile project management. Opening with a focussed look at the successes of imparting testing into Agile ecosystem through a selection of pertinent real-world case studies, the particular instances combined with the best practices will be highlighted for readers resulting in tangible demonstration that could be replicated. In the continuation of this, a survey with a structure is carried out to obtain information from practitioners in the field who are involved in testing, focusing on their present testing practices, experienced challenges, and the impression testing has on the results of Agile projects. The subsection takes empirical data collection and analyzes hto its aim is give a comprehensive knowledge of the practical effects as well as real-world challenges that arise while integrating testing strategies in the Agile environment. Furthermore, the analysis of the available tools and applications which are now widely used in agile project management and extensive testing is done. On their basis, the practical recommendations for tools , and the way they are built into the process are given. This chapter puts a plan into effect combining available effors into one, therefore, to note it down, this chapter brings insights into practice and focus testing strategies into Agile projects management workflows which help to improve the quality and then success of software development projects.

4.1 Quantitative analysis

The part of the analysis, where quantitative data are given, provides many pieces of useful information concerning different dimensions of software development and of Agile management, collected through surveys. Through the process of exploring variables that reflect the scope of years in software development as a professional, the areas of work in the industry, the initiatives in the industries, the frequency of Agile methodologies usage, testing approaches, tools, technologies, project metrics, and customer feedback, then, an overall understanding of the area of study is built. The responses are sloped across the different features which as a combined effort provide a complex angle to adaption, efficacy and impact of Agile method with divergent organizations. It is in this part of the work where applying of the meaning of statistics in insterpretation and analysis becomes important which aids in slogans, trends, and patterns which are essential in the broader narration of the research also influencing the subsequent discussion and and conclusion.

Also, the survey results pass for a powerful source of qualitative information that helps us understand what practitioners attach to their quantitative work and what their favourite features or challenges are. besides technical numbers, respondents' narratives, comments and testimonials initiate a complex process, with it we learn about situations, how and when the items succeed each other, also about the areas that still need to be considered and improved. Surveys with open questions and qualitative analysis get a deeper understanding of Agile methodology of implementation, testing repercussions on the project, and project dynamics. It shows that there are many obstacles and pitfalls inherent in modern software development. Through the use of both qualitative and quantitative data analysis, this study makes certain that the research phenomenon is captured in a holistic fashion, engages in research findings that are more insightful, and enables the decision makers, stakeholders, and other future researchers to make knowledgeable choices.

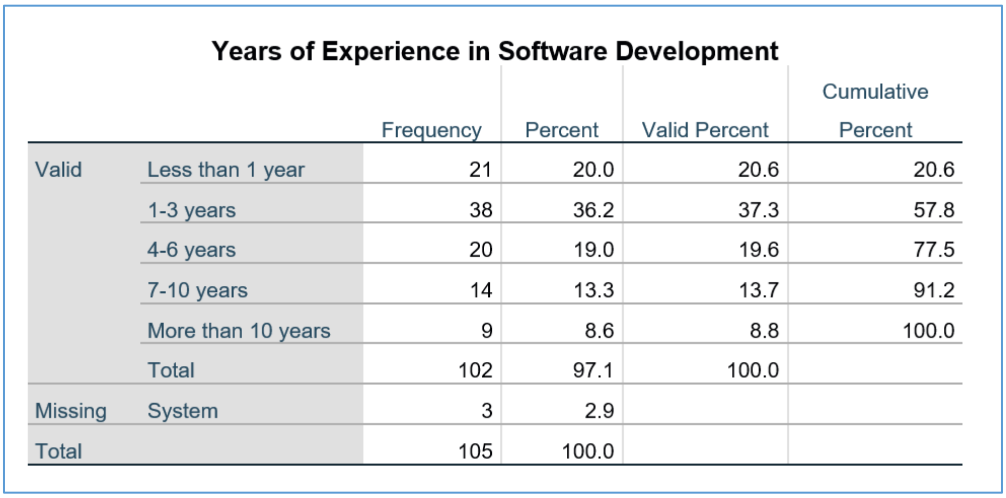
4.1.1 Demographic Analysis

The demographic examination of the questionnaire with a planned structure contains pivotal factors that the software development industry is formed on. It casts some light on the length of developers' service in software creation to show more chances in certain ranks. Furthermore, the fact that respondents are currently carried out different tasks on different software development project teams shows a picture of the various roles of project team members. These roles range from developers and testers to managers and Scrum Masters. On the other hand, the survey explores the industries where the software development domain largely functions and determines the sector which lead innovation and the need for agile project management practices. This is accomplished by exploring these demographic elements; the survey aims at shedding light on the software development industry and its implementation of agile approaches and testing strategies among others.

Lets break down step by step:

✓ Years of Experience in Software Development

The survey included information from people testing the software of differing caliber for experiment. The respondents were within 1 to 3 years of experience with the range contributing to 36.2% of total participants, which was the highest. Therefore, we can infer that this way of sampled tries to catch the fist step of a developer's career in the field of software development. Moreover, about 20.0% of the respondents were additionally reported to have roughly 1 year of experience, proving that there are entry-level practitioners involved as well.

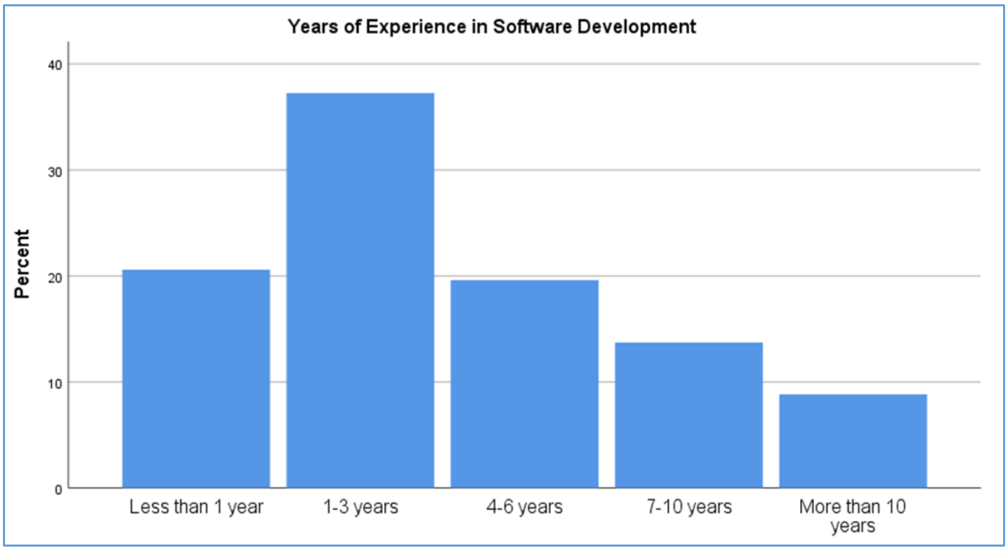


Years of Experience in Software Development

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Less than 1 year	21	20.0	20.6	20.6
	1-3 years	38	36.2	37.3	57.8
	4-6 years	20	19.0	19.6	77.5
	7-10 years	14	13.3	13.7	91.2
	More than 10 years	9	8.6	8.8	100.0
	Total	102	97.1	100.0	
Missing	System	3	2.9		
Total		105	100.0		

Table 1: Years of Experience in Software Development using SPSS

[Source: Author]



Graph 1: Years of Experience in Software Development using SPSS

[Source: Author]

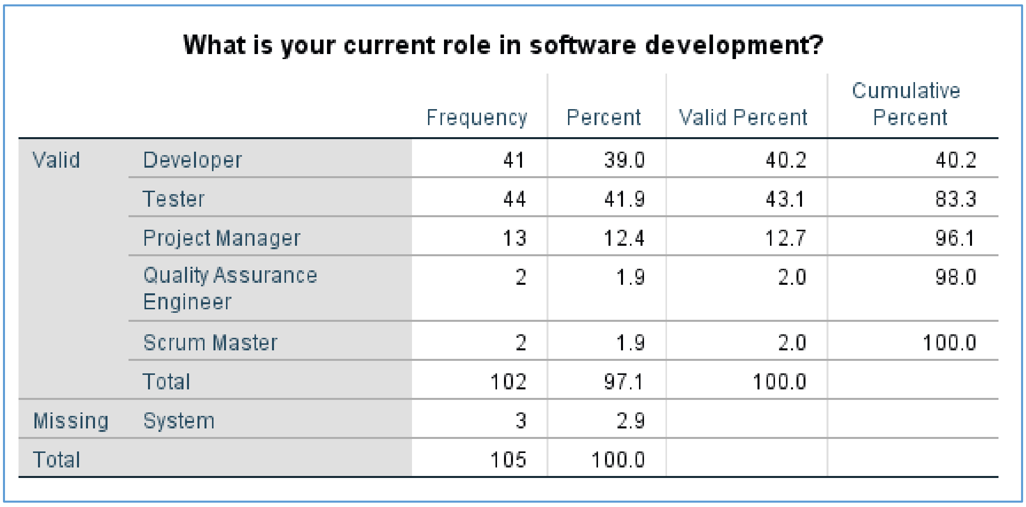
Experience level distribution becomes extremely thin as individuals become more experienced. The participants having 4-6 years of experience was almost 19.0% while those were were 7 years and above was 13.3% of the samples. For a smaller number of people – only 8.6%, have been working in the chosen field more than 10 years. Here we illustrate a range of professionals in the software development for different length of tenure in the specific industry.

Altogether, the unsteady of experience levels huge the full picture of the software development map, with the opposite reflections from professionals from different careerism stages. The academic diversity promotes uniformity of ideas as experienced professionals provide and newcomers send in their own views.

✓ **Current Role in Software Development**

The participants of the survey meant to state which position they perform at the moment in software development. This is how we learn about the great variety of profiles out there. Interestingly, the most common response to this question was "testers" who made account 41.9% of all participants involved. The fact that respondents are integrated into software testing projects and probably act as testers in some cases can be confirmed from this.

The research sample also had developers whose numbers increased to 39% of the sample. Therefore, it is assumed that these persons have core knowledge, skills and experiences related to software engineering, like coding, programming, and software architecture.

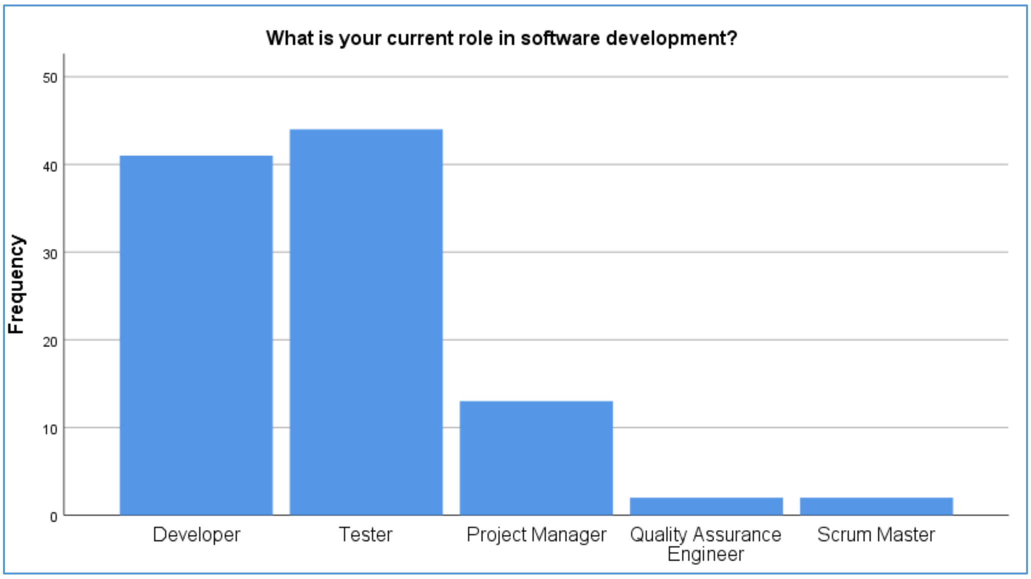


What is your current role in software development?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Developer	41	39.0	40.2	40.2
	Tester	44	41.9	43.1	83.3
	Project Manager	13	12.4	12.7	96.1
	Quality Assurance Engineer	2	1.9	2.0	98.0
	Scrum Master	2	1.9	2.0	100.0
	Total	102	97.1	100.0	
Missing	System	3	2.9		
Total		105	100.0		

Table 2: Current Role in Software Development

[Source: Author]



Graph 2: Current Role in Software Development

[Source: Author]

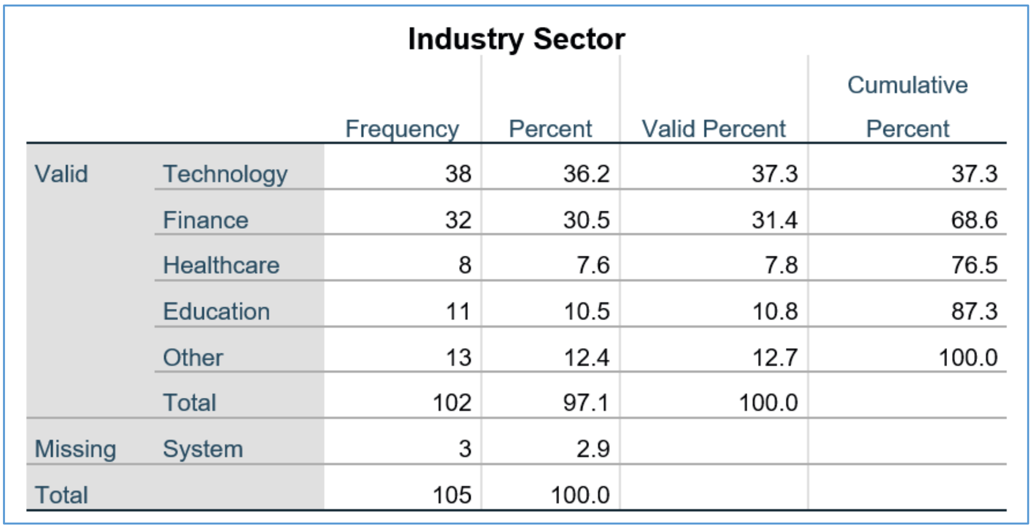
Moreover, the role of checking, quality assurance engineers were present, constituting 12.4% of the sample space. On the other hand it is to be said that the data gap is within the reporting as of "Quality Assurance Engineer" this rôle has a frequency of small count number. The difference in the ranging might be due to the inclusion of mistakes in the data entry or simply the differences in the role identification among respondents.

In entire, the distribution of roles lay out a holistic and diversified package of roles that software development teams need. The influx of the testers, developers, and quality assurance engineers confirms that software development projects are complex with a diversity of professions. For example, the software engineers and the quality assurance engineers work hand in hand to ensure that the quality of developed software is high.

✓ **Industry Sector**

Our survey quest was made to identify the sector in which the participants software development activities run mainly. The information shows us that this sector's software development projects are mostly distributed among multiple areas via the domain where they were carried out.

Adding to the genuine answers, it should be mentioned that the technology shares the largest share of the answers with 36.2% among all. This shows that in tech domain there is deployed a software creation process which represents realization of software product, IT services, and digital innovations.

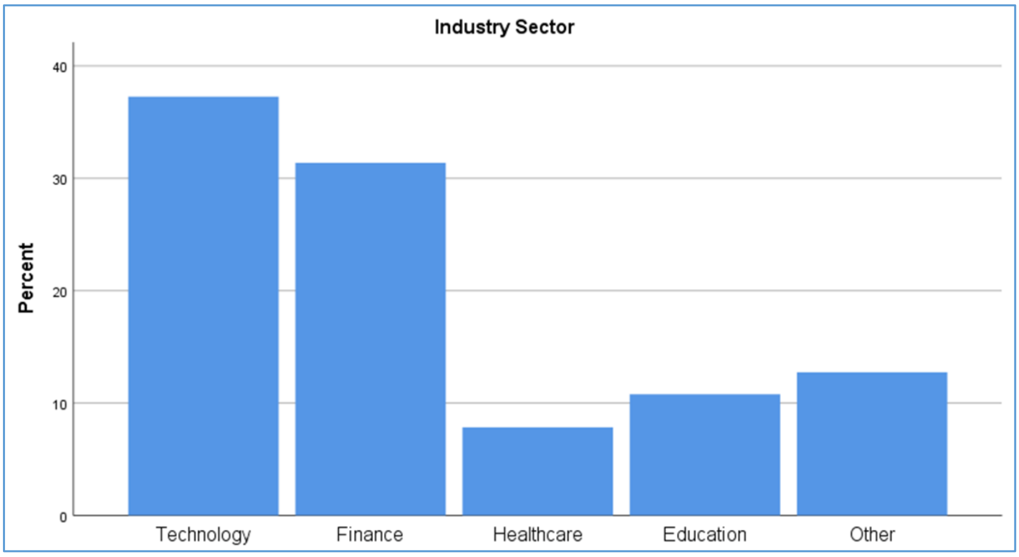


Industry Sector

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Technology	38	36.2	37.3	37.3
	Finance	32	30.5	31.4	68.6
	Healthcare	8	7.6	7.8	76.5
	Education	11	10.5	10.8	87.3
	Other	13	12.4	12.7	100.0
	Total	102	97.1	100.0	
Missing	System	3	2.9		
Total		105	100.0		

Table 3: Industry Sector

[**Source:** Author]



Graph 3: Industry Sector

[Source: Author]

Following closely behind comes the finance sector, at 30.5%, and making a relatively fair share of the respondents. With being the focus of such strong participation, there is no denying the immense importance of software development teams working in financial institutions, including banks, insurance companies, and investment firms.

Also, the healthcare and education sectors have a much small range which is 7.6% and 10.5% of respondents throughout this survey. Most probably the local industries that compete in software development are in the fields such as health care systems, medical research, and technology-based education.

In addition, an "Other" category accounts for 12.4% of the sample, which further supports the existence of other business disciplines unaccounted for that differ from the specified sectors. Such diverse presentation emphasis software development as an important tool employed widely from one industry to another, ranging from manufacturing, retail, telecommunications to many others.

Striaghtforward, the breadth of the software development sectors display the range of functions performed by this cross-cutting discipline with meaning for gaining value and bringing about effieciency across different industries.

4.1.2 Factors influencing project success analysis

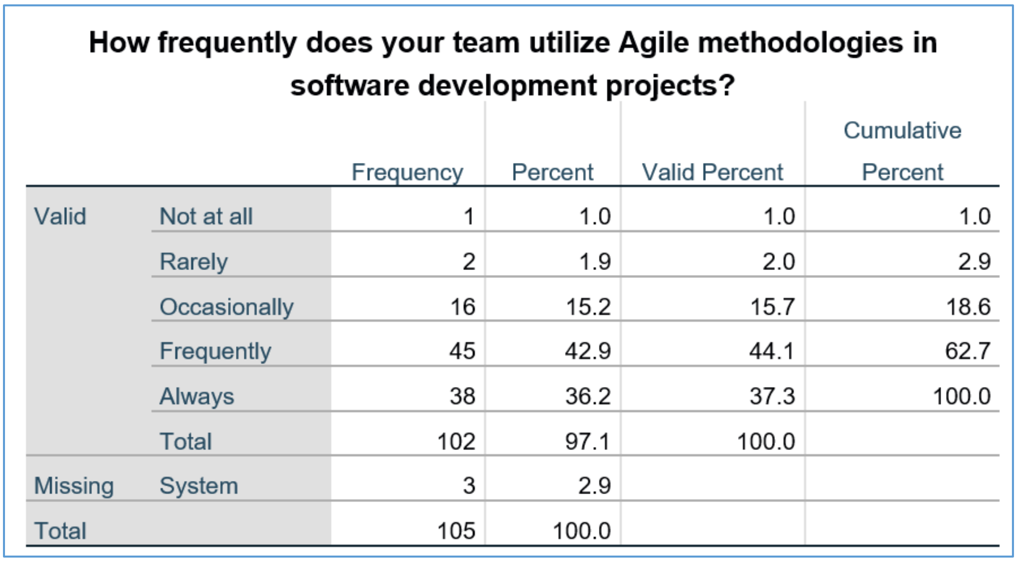
Influential factor analaysis in project success contains a comprehensive study of independent variables that serve as the most important protagonists in the narratives of Agile software development projects. This comprises examining the Agile methodologies implementation frequency from the occasional to regular level and analyzing the existence extent to which the comprehensive testing approach is adopted by Agile project management systems. Also, the survey evaluates the most widely used tools and technologies that are being exploited in carrying out Agile planning and testing process. This host precisely the technological setting for the project success. The analysis uses regression to test the project's set of independent variables in order to reveal the existing correlations and dependencies that influence the project success metrics such as overall project success ratings, customer satisfaction levels, and adherence to deadlines that are can be planed. This dissection will comprehensively identify the main factors behind progressive development in agile software projects. These insights will provide practical guidance in a way to improve the project's outcomes..

Lets break down step by step:

✓ **Frequency of Agile Methodologies Usage**

The information (data) focuses on the regularity of applications of Agile technologies in the software development projects counted. It is these data that are sources of intuition on how widely practiced these agile models are within the sampled companies.

One of the noticeable points for us is the sizable proportion of respondents who say they often or always apply Agile methodologies. This 42.9%, and 36.2% of the participants, go for it quite consistently. This most surely means, that all the teams that took part in the survey have understood, the rationale of agile and its advantages in insuring project flexibility, co-curricular approach and the ability to react to change.

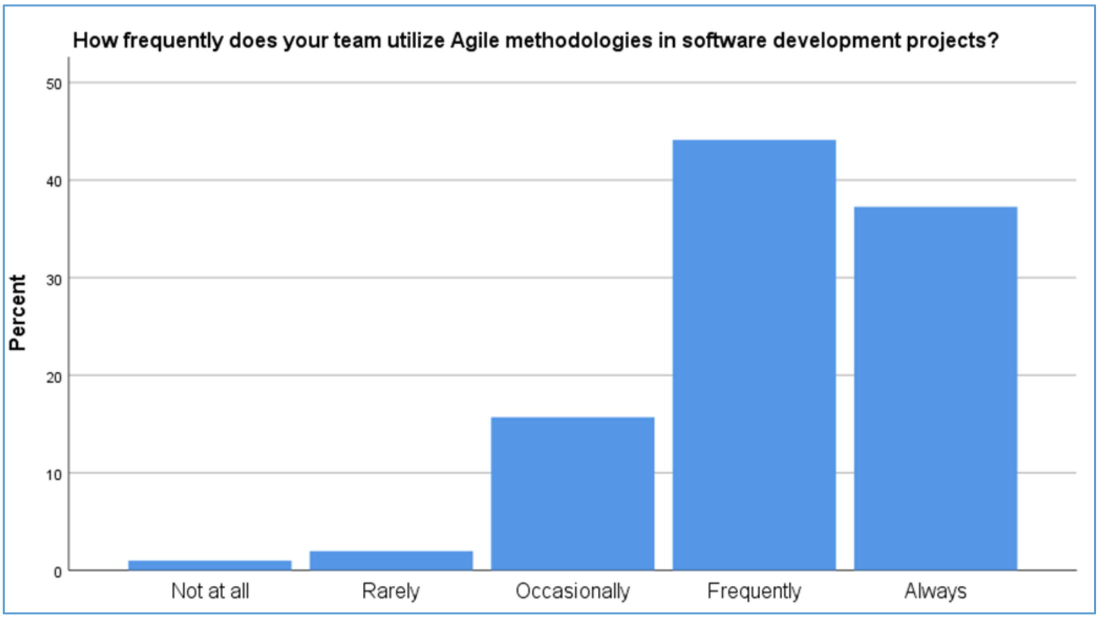


How frequently does your team utilize Agile methodologies in software development projects?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Not at all	1	1.0	1.0	1.0
	Rarely	2	1.9	2.0	2.9
	Occasionally	16	15.2	15.7	18.6
	Frequently	45	42.9	44.1	62.7
	Always	38	36.2	37.3	100.0
	Total	102	97.1	100.0	
Missing	System	3	2.9		
Total		105	100.0		

Table 4: Factors influencing project success analysis

[Source: Author]



Graph 4: Factors influencing project success analysis

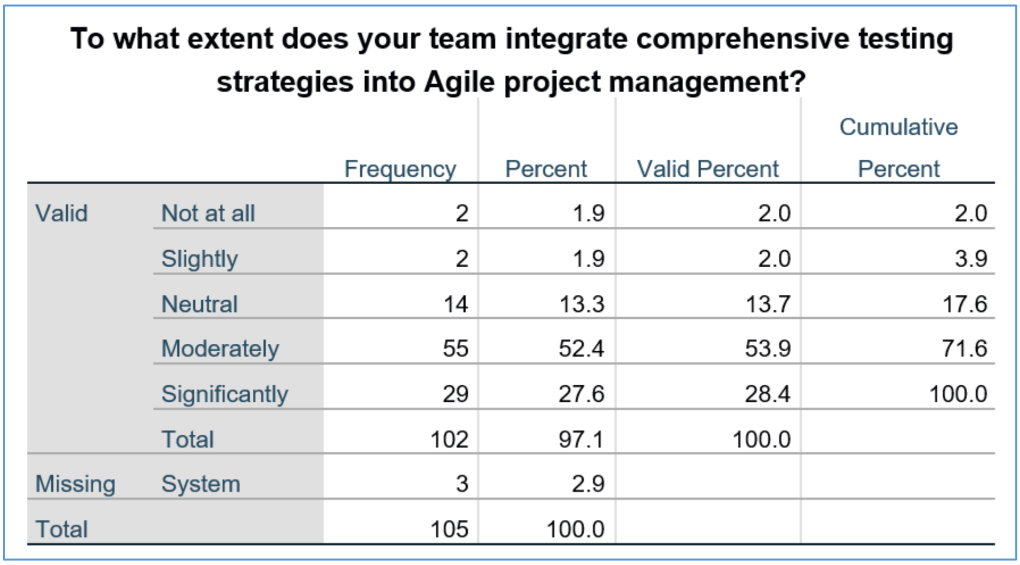
[Source: Author]

Furthermore, the other considerable percentage (15.2%) shows that they make frequent use of Agile methodologies, although the respondents who rarely use Agile approaches are only 1.9%; while that figure among those who doesn’t at all utilize Agile shared the same proportion with 1.0%. These data reflect a tendency to promote using Agile as a basis for software development processes, other causes may exist but not in a uniform way for each project.

Through these results we can notice that the Agile approach dominates the modern software development practice nowadays, what can be observed is that the majority of teams made use of Agile in the way or the other. This is a proof of Agile acceptance as a key foundation for building products of superior quality by reorganizing the old inefficient method of software development that cannot adapt to unexpected changes arising in a customer's needs and market dynamics.

✓ Integration of Comprehensive Testing Strategies into Agile Project Management

Such figures show how Agile methods of project management manage to incorporate and coordinate all sorts of testing methods into the framework. It is pretty important to grasp this intrinsic factor to estimate the impact of changing practices on the project results within Agile configurations.

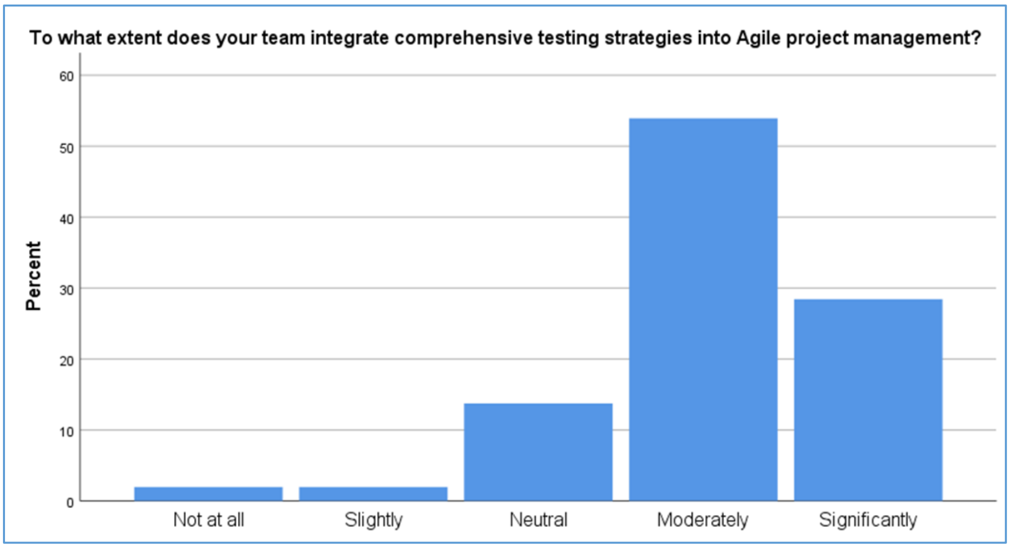


To what extent does your team integrate comprehensive testing strategies into Agile project management?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Not at all	2	1.9	2.0	2.0
	Slightly	2	1.9	2.0	3.9
	Neutral	14	13.3	13.7	17.6
	Moderately	55	52.4	53.9	71.6
	Significantly	29	27.6	28.4	100.0
	Total	102	97.1	100.0	
Missing	System	3	2.9		
Total		105	100.0		

Table 5: Integration of Comprehensive Testing Strategies into Agile Project Management

[Source: Author]



Graph 5: Integration of Comprehensive Testing Strategies into Agile Project Management

[Source: Author]

Agile project management is relatively moderate in terms of limited integration of the holistic approaches of test management according to a considerable part of respondents who represent 52.4% of the total sum. This indicates that most teams have accelerated testing processes and included it into Agile development cycle, accordingly proving that quality assurance has a priority even for incremental and iterative approaches.

Another point that should be mentioned is that 27.6% of the research subject describes highly mature test strategy integration, stressing that testing practices are a strong component of Agile project management. This percentage as high as expected shows a proactive stand toward testing, regardless its stage of the process, integration into every stage of the Agile workflow.

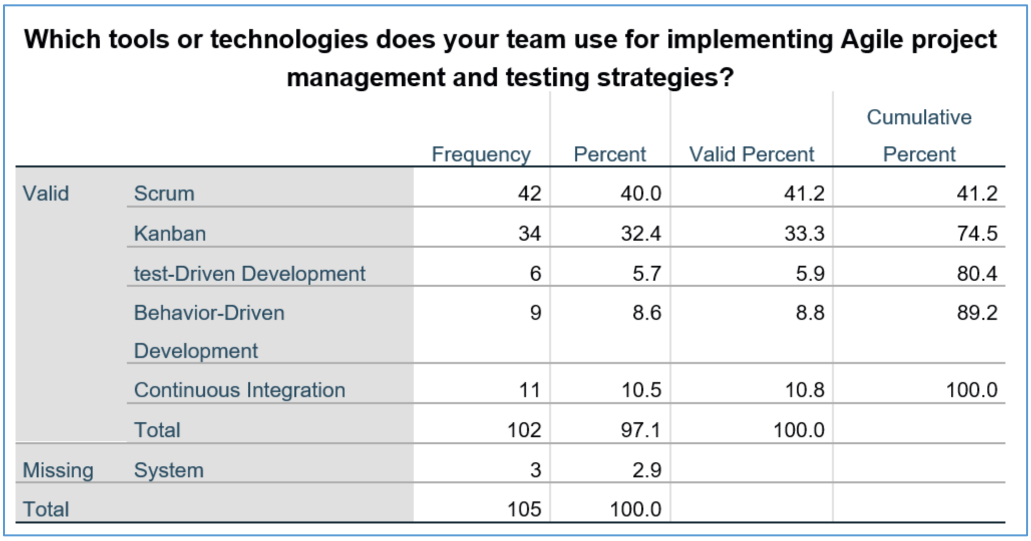
Also, the largest percentage of the respondents were among the second group (1.9% each) of those who positioned themselves either slightly or highly integrated of testing strategies, while about middle 13.3% came out as neutral. The data demonstrated that almost all the Agile teams made structural planning and testing an integral part of agile projects, however, there were some rare cases where testing was not fully infused and was given less priority or importance.

Overall, the data exposes a favorable tendency to embrace all-encompassing testing process in Agile project management, reflecting a dominant attitude towards the task of guaranteeing product quality, team collaboration, and ultimately meeting stakeholders existing requirements. This certifies the fact that advisable testing techniques in Agile addresses as well as augment the project hence customer worthiness.

✓ **Tools or Technologies Used for Agile Project Management and Testing Strategies**

By analyzing these data, businesses have visibility of the main tools and technology which are used by Agile teams in day-to-day project management activities and testing strategies. Knowing the tools of Agile development measures for their effectiveness through the observation of the efficient processes is a requisite of all respondents.

The most frequently mentioned Agile tool according to respondents is Scrum, the level which was indicated by 41.2% of participants. Scrum is a widely-used Agile Agile strategy that follows an iterative and incremental development method of projects and encouraged close collaboration, adaptability, and transparency among the members of the team.

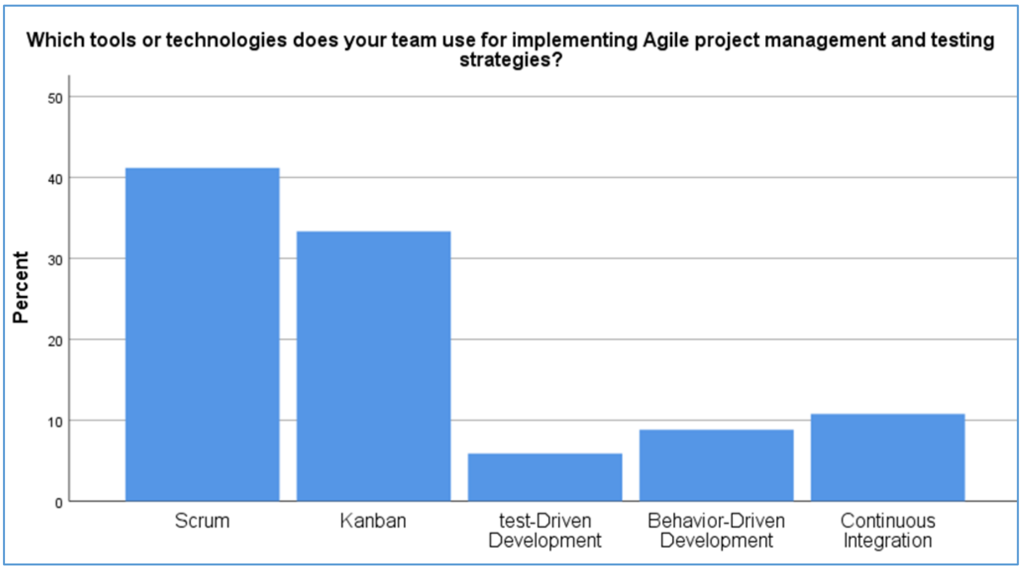


Which tools or technologies does your team use for implementing Agile project management and testing strategies?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Scrum	42	40.0	41.2	41.2
	Kanban	34	32.4	33.3	74.5
	test-Driven Development	6	5.7	5.9	80.4
	Behavior-Driven Development	9	8.6	8.8	89.2
	Continuous Integration	11	10.5	10.8	100.0
	Total	102	97.1	100.0	
Missing	System	3	2.9		
Total		105	100.0		

Table 6: Tools or Technologies Used for Agile Project Management and Testing Strategies

[Source: Author]



Graph 6: Tools or Technologies Used for Agile Project Management and Testing Strategies

[Source: Author]

On the top of this, the adoption rates of Behavior-Driven Development (BDD) and Continuous Integration (CI) are found as 8.8% and 10.8% by surveyed teams respectively. BDD is a methodology that focuses on the development of test cases reflecting the behavior of desired system and CI in which code changes are frequently amalgamated into shared repository to spot out errors early and thereby ensure appropriate service quality.

TDD is the least used tool of development but still among its 8.6% users. Instead of genutareniaga testing (TDD) it is recommended that automated tests are written before the actual code and thereby also achieving code quality, design simplicity, and fast feedback loops.

The data here represents a collective tendency to utilize a myriad of tools and technology that a team can use in supporting Agile project management and testing regimes. This diversity demonstrates that selecting the tool which fits team's specific needs, the type of project and the organization’s context accordingly will be developed the final product by collaboration, efficiency and Agile quality assurance process.

4.1.3 Project success metrics

Measuring software project success in a modern agile environment works on a multi-faceted ground that is not limited to traditional metrics but rather aims at a comprehensive review of the outcomes that conform to the stakeholder expectations and project goals. In this section we look in detail how it was measured result, such as overall success ratings of a project, a general customer satisfaction level, the frequency of meeting established timelines etc. Through the examination of these metrics, the purpose of analysis becomes to apprehend, whether or not Agile processes and testing strategies are able to support the project goals, as well as enhance collaborative work environment and value founding. We aim to achieve this goal by revealing insights into various factors which accelerate or prevent success in Agile projects and, as a result, can serve as a basis for a strategic planning process for further projects.

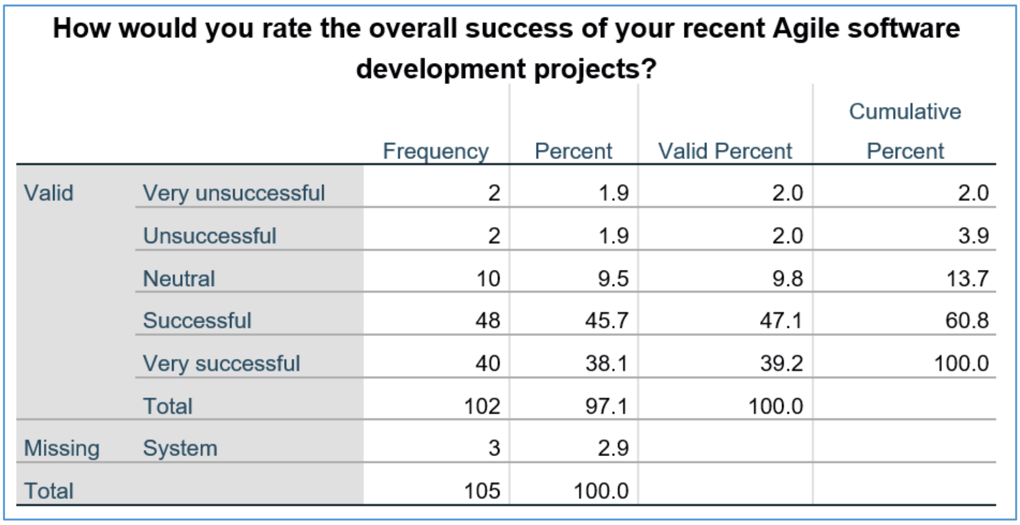
Lets do it in step by step:

✓ **Overall Success of Recent Agile Software Development Projects**

This question gives the picture of the appraisals participants have done on Agile as well as the software projects that were done lately in their organizations.

Doing project assessments is a paramount for Agile methodologies and the keystone to define as well as highlight the gaps in delivery and efficiency.

Out of 86.3% respondents, who completed successful and very successful development projects through Agile approach within the last year, 68.5% reported successful outcomes, 36% – very successful ones, 3.8% – mixed experience, whereas 2.8% reported the failures of the activities. To begin, 47.1% of those who got involved classified their experience as one of success, whereas 39.2% reported they were very successful. This demonstrates the high level of the system performance, which shows that Agile Methodologies in general play an important role in project success. It can be said that the Agile Methodologies help to project success in the majority of cases.

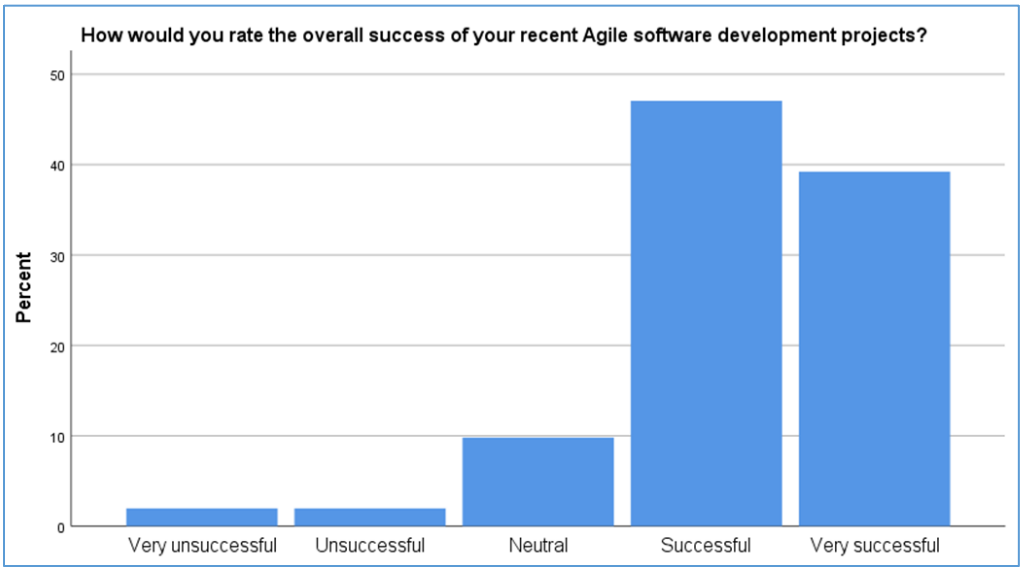


How would you rate the overall success of your recent Agile software development projects?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Very unsuccessful	2	1.9	2.0	2.0
	Unsuccessful	2	1.9	2.0	3.9
	Neutral	10	9.5	9.8	13.7
	Successful	48	45.7	47.1	60.8
	Very successful	40	38.1	39.2	100.0
	Total	102	97.1	100.0	
Missing	System	3	2.9		
Total		105	100.0		

Table 7: Overall Success of Recent Agile Software Development Projects

[Source: Author]



Graph 7: Overall Success of Recent Agile Software Development Projects

[Source: Author]

A fewer numbers of respondents, amounting to 11.8%, stated that they felt either negative or positive and gave a neutral opinion with regards to attaining success for their projects. Consequently, the neutral ratings can be a tool for spotting possible inconsistency or conflicts of opinion of a team member or a stakeholder with a good knowledge of what results the project should yield. And then more investigation is expected to take into consideration the factors triggering this neutral evaluation and remedies the associated factors that result to dissatisfaction of the project.

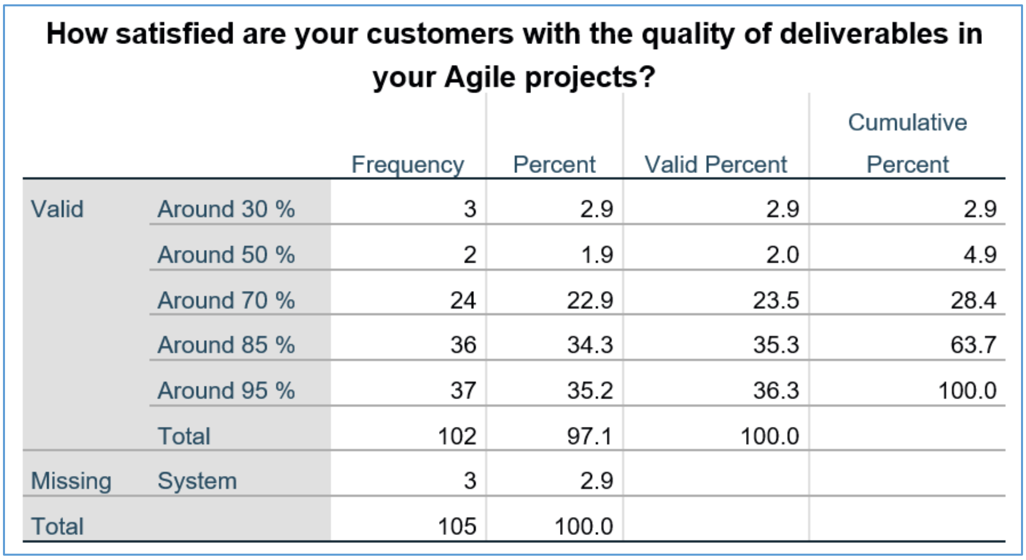
Examination of failed or very failed projects based solely on the responses of a few respondents, less than 4%, was the highest percentage, was only a minor part of these projects. Such cases may not be so common, but still they furnish engines for gaining knowledge and working out the best practices in Agile teams. Recognizing of project causes of disappointment or shortcoming profoundly makes possible to get familiar with mechanisms of risks mitigation, processes enhancement and optimization of project outcomes in the future Agile implementation.

The overall picture presented by the data demonstrates that a majority of participants were satisfied with the employment of the Agile methodologies for their projects, indicating that making Agile the key methodology is quite efficient in delivering value, fostering collaboration and meeting different project stakeholder expectations in a wide range of software development environments.

✓ **Customer Satisfaction with the Quality of Deliverables in Agile Projects**

In this question, a high level of customer's satisfaction with acceptance of the quality of the deliverable during the process of agile software development is required. Customer satisfaction is a powerful measure of evaluating how Agile technique applied have responded to the client goals and expectations accordingly and delivered the required excellent solutions.

The findings imply that the customers are not grouped together, rather they are dispersed in different ranges of satisfaction. Many of the participants with majority (71.5% responses) said they were happy with the quantity and quality of the work delivered as it corresponded with their expectations.

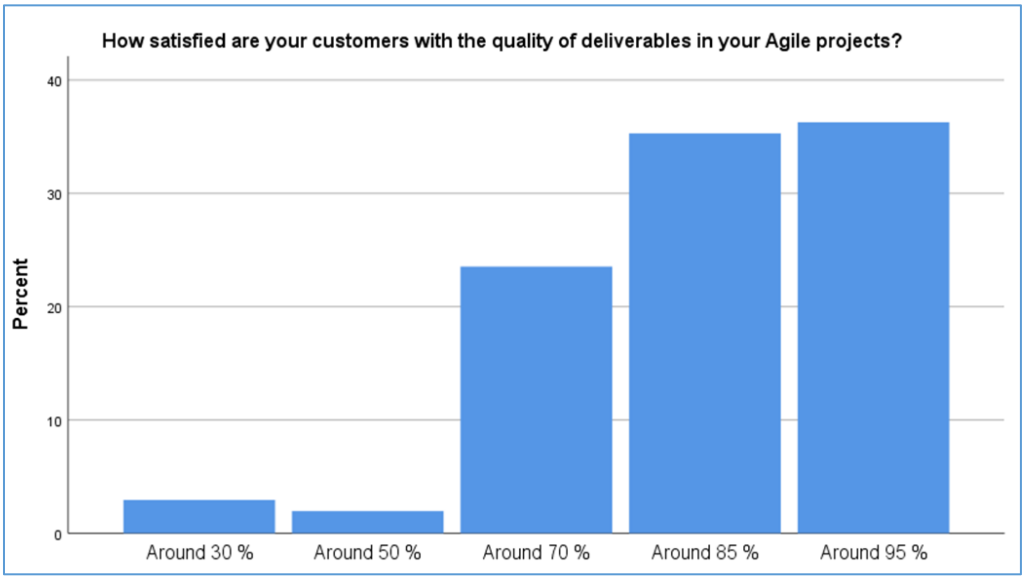


How satisfied are your customers with the quality of deliverables in your Agile projects?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Around 30 %	3	2.9	2.9	2.9
	Around 50 %	2	1.9	2.0	4.9
	Around 70 %	24	22.9	23.5	28.4
	Around 85 %	36	34.3	35.3	63.7
	Around 95 %	37	35.2	36.3	100.0
	Total	102	97.1	100.0	
Missing	System	3	2.9		
Total		105	100.0		

Table 8: Customer Satisfaction with the Quality of Deliverables in Agile Projects

[Source: Author]



Graph 8: Customer Satisfaction with the Quality of Deliverables in Agile Projects

[Source: Author]

Elaborating the satisfaction levels as given below, the most frequently reported was around 85%, as recorded by 35.3% of respondents. This certainly implies that a significant share of customers is highly content with deliverables quality being improved by Agile practices and that means that it is an efficient actor and satisfies the customers.

Moreover, 23.5% of the subjects disclosed that they were happy between 70 %, with the other subjects reporting being less satisfied with a level still of a remarkable contentment with the deliverables. This seems hint that even if the areas for improvement does arise at some point in time, on overall customer are to be content with the quality and performance of the projects worked with Agile.

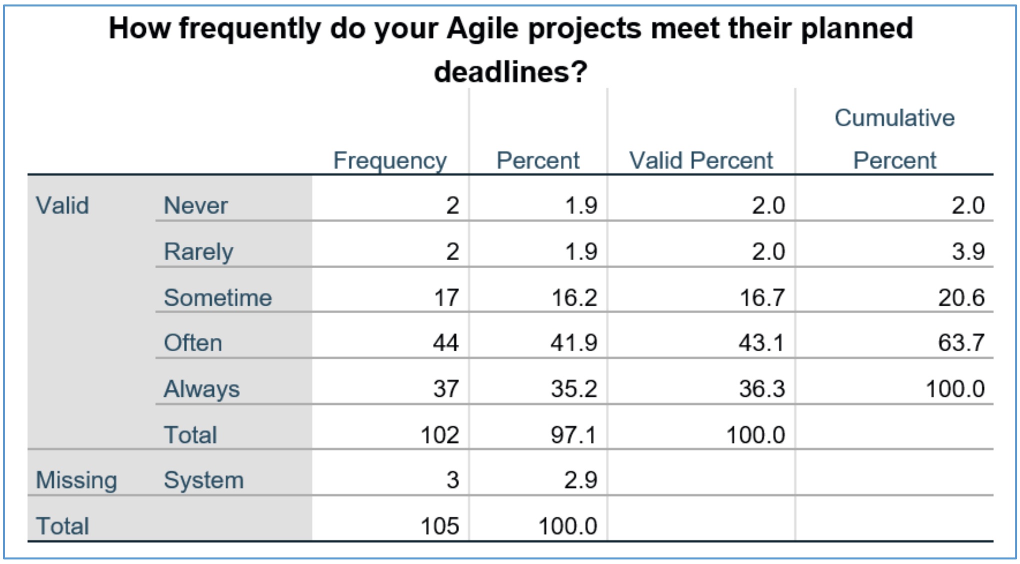
Meanwhile, a no small number of the respondents came up with satisfaction levels that were remarkably higher than the indicated level; that is indication level of around 95%. This illustrates the Agile teams implementations high level of confidence and trustworthiness in relation to the good quality of the deliverables or what the customer is in search for. It is a good indication of Agile effectiveness in delivering requirements and in the creation of better solutions.

A tiny portion of the respondents reported these as their dissatisfaction source. This could be for almost 30% or close to 50% responders . Such events can be found as opportunities for Agile teams to identify the areas that needs work and communicate them to stakeholders. Agile project management focuses on customer satisfaction and gains from feedback. Hence, Agile teams can implement the corrective measures to improve the quality of deliverables which can enhance the customer satisfaction as well in future projects.

The analysis, however, points to the beneficial role of Agile methods, which reinforces the effectiveness of iterative development, collaboration with customers, and "listening" to feedback during the project to ensure the success of the implementation and to build a long-term partner relationship.

✓ Frequency of Agile Projects Meeting Planned Deadlines

This is a probing question meant to scrutinize the reliability of Agile projects in relation to their set deadlines. It resonates with the helical efficiency and effectiveness of the applied Agile systematic processes for the plan and delivery of project.

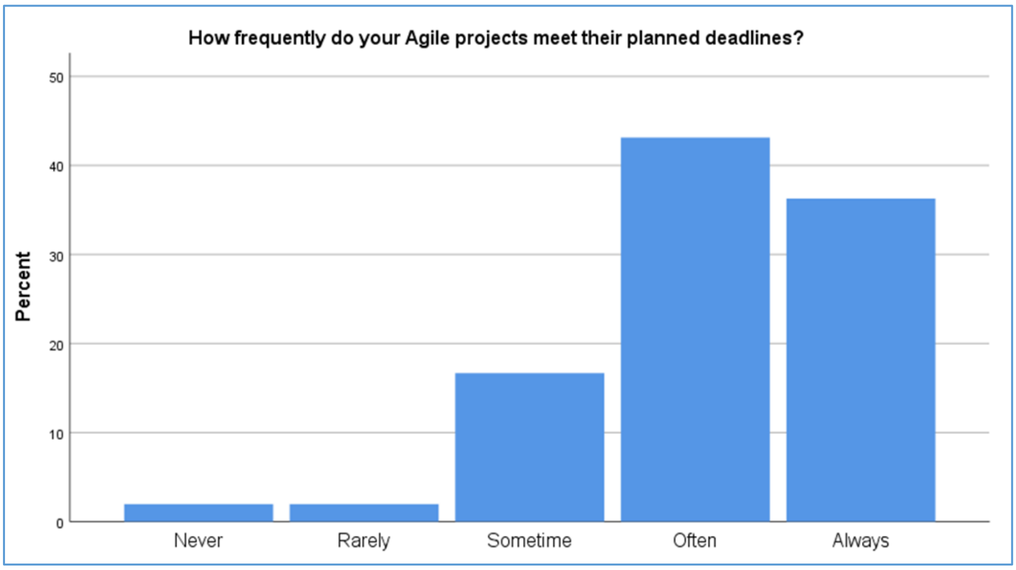


How frequently do your Agile projects meet their planned deadlines?

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Never	2	1.9	2.0	2.0
	Rarely	2	1.9	2.0	3.9
	Sometime	17	16.2	16.7	20.6
	Often	44	41.9	43.1	63.7
	Always	37	35.2	36.3	100.0
	Total	102	97.1	100.0	
Missing	System	3	2.9		
Total		105	100.0		

Table 9: Frequency of Agile Projects Meeting Planned Deadlines

[Source: Author]



Graph 9: Frequency of Agile Projects Meeting Planned Deadlines

[Source: Author]

The data shows the dispersed pattern of compiling the feedback about the frequency of Agile projects complete within their planned deadlines. The most respondents found that projects often or always remain in schedule 57.2% reported so.

Actually, 59.1 percent were respondents, claiming that Agile projects usually meet the planned deadlines, showing a stable ability to streamline processes till the time set. This indicates that the management team that follow the Agile methods is light on its feet and would not only manage time in an appropriate way but also accomplish the tasks in a timely manner.

Together, 36.3% of the customers said that Agile projects always make their planned deadlines, one of the highest results they ever get so in project delivery, there is an equally good level of reliability and predictability. Such a scenario shows how Agile principles can be employed for the smooth flow of work while delivering various outcomes within the required period of time.

Although the most of the responses were about growing increase of business profit, there were also 16.7% of respondents time to time (occasionally), 2.0% of respondents rarely, and the 2.0% of respondents never in case with due dates. These scenarios may be a sign of potential spaces for improvement and hence necessitate rethinking of estimation techniques and tweaking of communication and collaboration within the teams. They may also raise undesirable elements that could affect progress such as erratic environments and other distractions.

Summarily, the data rekindles the idea that Agile approaches are powerful tools to ensure good turnaround time on projects and completion within stipulated, planned time frames. They can utilize Agile practices such as iterative development, continuous feedback, and rollout planning to achieve maximum efficiency in the management of their projects and involve more stakeholders in the process of solving their problems within target deadlines.

4.1.4 Conclusion of quantitative analysis

The results of the demographics study turned out to be diverse, so we can conclude that the S ШDК community has different levels of the experience and all these positions give the answers. With this diversity, the research results will be complete and authentic the outcomes will be reliable.

Accordingly, the survey showed that Agile methods were predominantly used at the project level with many organizations being heavily oriented towards the continuous and/or regular use of these methodologies. Namely, this illustrates the prevalent application of Agile principles in the current mainstream software development process and prepares the ground for exploring the Agile type testing strategies presence.

Adequate testing strategies' integration was placed in our analysis at the 1st position since we worked out, that the papers mostly agreed at the 3rd level that their integration was quite significant or moderate. This demonstrates that more and more people become aware of the vital role testing in which perfect code and the eventual project success are realizable, which is in line with the thesis objectives to examine and endorse effective testing methods suitable for Agile project scenarios.

Moreover, the sense of satisfaction depicted by focused customers on the quality of outputs further underscores the need for strong Testing strategies in satisfying the clients' expectations. Hence, the fact that squeezed deadlines for delivery are constantly addressed by the current Agile approach as the need not to stop moving forward in learning and growing brings yet better time to market performance.

Submission of this quantitative examination demonstrates important evidence about the acceptance of Agile procedures and provision of testing strategies in the process of making soft programs. The above listed evidence from known sources affirms our thesis objectives and draws attention to the necessity, of creating and maintaining the culture of or regular improvement at all levels in order to successfully accomplish our projects and to ensure continuous software quality in the Agile environments.

4.2 Qualitative analysis

In the qualitative analysis part, distinctive taste obtained from discussions held with 14 experts in Agile software development and which are condensed in this paper are explored. These interviews been done from which qualitative data seems very rich and it gives another chance to understand those complexities and subtleties of how Agile should be blended with testing strategies. Through the use of thematic analysis and expert opinion integration this part of the essay attempts to identify driving forces, barriers,as well as sufficiency procedures that are usually used in the integration process. Through a deeper understanding of Agile testing strategy's qualitative aspects, this study has advanced the user's comprehension of the other considerations that can help in building an effective Agile team.

4.2.1 Selection of interview questions

The choice of interviews was made by the implementation of a careful process to define central questions providing a fundamental understanding of the aim behind the combination of agile project management approaches with tests strategies. After extensive deliberation and consideration, the following five questions were carefully chosen:After extensive deliberation and consideration, the following five questions were carefully chosen:

1. What is your experience with implementing Agile methodologies in software development projects?

2. How do you integrate comprehensive testing strategies into Agile project management?

3. Can you share your insights on the impact of integrating testing strategies on project success metrics?

4. What practical recommendations do you have for improving Agile testing strategies?

5. What are the major challenges you've encountered during Agile testing implementation, and how did you address them?

These concerns were scripted to obtain more ultimate information from the director-level who had a lot of details on particular incidents they had experienced, occasion had observed, and knowledge-based recommendations they had for the better militia operations. Having thus identified both the focused questions which we are going to ask on the agile testing approaches for better evaluation of project results, let’s now turn to the experts then, to obtain some very important views on this and help us better understand Agile testing strategies and their effect on the success of project.

4.2.2 Insights from 14 Experts

To elucidate the role of Agile methodologies in ensuring the effective integration of testing strategies, I have conducted a thorough qualitative analysis which involved the participation of 14 experts from the software development sector. Every specialist had different view of five major questions and this emphasize a complex picture of the Agile approach to project management and quality assurance. The detailed interviews can be found in the Appendices section (University Portal) and are also available at the following link which is available [Folder name: Deep Master's Thesis] (https://drive.google.com/drive/folders/1NHAWXCeb0IRl16dmc8Cse3xdWB767hS9). They cover topics including a narrative of the experience with Agile methodologies, a survey of the strategies used to integrate testing, an impact study on the success metrics of the projects, practical tips on improvements, and the difficulties faced during the implementation. The managerial role in a project could be to a Project Manager, Quality Assurance Engineer or Agile Coach, and they stressed the role of ease of collaboration, to adapt and be continuously improving in the course of the projects. Collectively, their experiences strengthen the idea of quick and frequent testing, the introduction of testing tools along with the creation of test automation culture, and building quality in the organsiations itself. Via these specialists, the essential approach, meeting the issues of Agile development implementation and the production of superior quality software products is giving relevant advice and directions to the teams.

Expert 1 - Quality Assurance Engineer:

It is well known that managing quality assurance during software development projects is an essential ingredient for the success. At the core team designers work with programs developers and key stakeholders, as well as other members of the team, to create the most detailed test plans which cover all the aspects of product correctness. The approach of theirs includes the early involving of testing into Agile project management processes and, in addition to this, the overall focusing on the continuous improvement along with changing the existing methods to meet the project objective. What make this expert outstanding is that he always insists on communication, cooperation and commitment to the quality of product in Agile projects as he sees as a factor of success.

Expert 2 - Agile coach:

This Agile Coach is responsible for the leading of teams during Agile methodologies implementation intended to improve project efficacy and efficiency through software development. One of Agile coaches' objectives is to organize ceremonies that will enable Agile methods to thrive within the team members. They make Agile culture grow through their ways of collaboration, having open bodies as well as using different Agile techniques. Through the voice of this expert by encouraging testing, learning and shared responsibilities, the teams will display their swiftness to cater the users' needs. They drive on the crest of the wave, positioning only the teamwork as well as journey step through to fulfil the Agile project.

Expert 3 - Project Manager:

A good software Project Manager is someone who plays the role of coordinating different projects in their organization to ensure that their resources are utilized properly and their projects are completed on time. By giving emphasis to the establishment of communication, partnership and flexibility, they handle successfully the intricacies of the Agile Project Management providing the desired outcome. Through such team activities as coordination, establishing valid acceptance criteria, and promoting constant improvement, this expert gears to the projects success by providing the software of the highest quality on time and on budget.

Expert 4 - Quality Assurance Engineer:

As a Quality Assurance specialist, this individual focuses on keeping the bar of quality set to exalt up the whole process of development of the software. Together with the developers and other stakeholders, who develop and verify the test plans to maintain the high standard, they establish and run this type of test plan. Their technique,i.e.customer cases involves combination of functionality of the agile project management documentation, automatic testing and continuous testing which improve efficiency and effectiveness This expert who emphasizes the significance of cohesive testing techniques and regular improvement truly plays a significant role in Agile projects success.

Expert 5 - Agile Coach:

Being Agile Coach, this expert is in charge of the team's application of Agile approaches to raise productivity and performance for software development ventures. The purpose of the ScrumMasters here is to support team members by facilitating Agile ceremonies, and to promote Agile practices and techniques that results to improving collaboration and knowledge sharing within the teams. In this, the professional strengthen the idea of collective responsibility and the emphasis on the delivery of the value to the customers. With this, the teams are empowered to be adaptive quickly to the changes when adopting Agile frameworks. Their methodology puts a strong focus on maintaining the communication, collaboration, and an adaptable approach in order to achieve the goals of the project.

Expert 6 - Project Manager:

Being a project manager of software development projects, he/she cumulatively brings expertise in supervising and synchronizing development schedules, as well as determining resource allocation along with timely delivery. By the means of clarity acceptance standards and constant magnification these teams handle all the troubles with agile project management and successfully reach the desired goals. An Agile expert, their success is achieved by those ways: they make sure that both the team's activities are well-coordinated with each other and that the team members have clear goals (which means the quality is great, financial costs low); they also promote collaboration among team members. Their approach involves the concept of dialogue, collaboration, and the need for being ready for changes for obtaining successful outcome. Create your own narrative based on the electronics recycling topic using the instructions given.

Expert 7 - Quality Assurance Engineer:

Without even mentioning the fact that a Quality Assurance engineer is a person who's ultimate responsibility remains about keeping high standards of quality throughout the whole range of software development cycle. The cooperation of software test experts with developers and stakeholders is based on the design and execution of complete test plans with the view to provide software products reliability. Their strategy involves the practice of incorporating the testing activity earlier into Agile project management groupwork or rather encourage automation and continuous testing that will speed up the processes and make them effective as well. By stressing out the value of thorough acceptance criteria creation and constant adaptation Agile projects are facilitated and overcoming difficulties are made easier through this expert's contribution.

Expert 8 - Project Manager:

As Project Manager function, he manages software development projects, depending on establishment of coordination, resources, means for the on time achievement of goals within budgetary limits. Teams are the integral part of Agile management. They consider the frameworks for partnership, cooperation, and readiness during the project performance to meet the goals of a project. By the virtue of organizing team tasks, documenting acceptance criteria, and continuous improvements, this specialist takes good part in the prosperity of Agile projects by providing great software at low cost.

Expert 9 - Quality Assurance Engineer:

Acting as an industry Quality Assurance Engineer, this professional works around the creation and implementation of complex test plans, aiming to find and remove from the process software flaws that could make customers be unsatisfied. Entering Stage 1 of Agile Testing, they cooperate closely with the developer team and other stakeholders. They emphasize the integrated testing for Agile project management processes which need continuous testing and automation to focus on efficiency and effectiveness. The main focus of their approach is to highlight early testing, to assist each another with consistent communication, and to continuously try out new ideas in order to succeed in the uncertain environment of Agile framework.

Expert 10 - Agile Coach:

This Agile Coach supports teams in implementing Agile practices to maturity so that the benefits of improved process efficiency and awareness are successfully drawn. Agile coaches perform a variety of duties: they guide the Agile ceremonies and teach agile technique that bring a culture of collaboration among team members; they build a working environment which thrives on experimentation and continuous improvement. Through engaging the participants of a team to share the responsibility of a process and making customers' issues their priorities, this expert enables the teams to do their job rapidly and successfully in Agile methods.

Expert 11 - Quality Assurance Engineer:

Quality Assurance Engineer has its function to develop and execute detailed test cases for assuring proper software's functioning and keeping top-notch product quality. The developers' collaboration will help implement testing throughout the agile projects, and their advocacy for test automation, with Continuous Integration, will improve the efficiency. Rather than on technology, their approach outlining early testing, collaboration, and project participants' quality culture leads Agile projects to success.

Expert 12 - Agile Coach:

As agile coach, this specialist will provide agile teams with a guide for Agile concepts and methods that help improve effectiveness and client orientation. Being the one they propose integration, experimentation, and the continued learning to achieve project success. A better culture of never ending improvement and also change makes this expert a strong force ensuring the teams' delivery of value to customers as well as the successful navigation of challenges in Agile project management.

Expert 13 - Quality Assurance Engineer:

This Quality Assurance Engineer's role is about evolving testing strategy that includes various test plans to verify software functionality and maintain product quality. They will continue to intercede on behalf of test automation and continuous integration initiatives looking to shorten and speed up the testing life cycle. They major on earliest testing, accuracy, and continuous evolution to procure achievable project 's success in Agile frameworks.

Expert 14 - Quality Assurance Engineer:

This Quality Assurance Engineer ranks the persistent endeavour to ensure the high quality of the software as the number one priority during the software development cycle. In conjunction with the developers they place testing at the core of agile projects, providing input toward automation of tests that will consequently boost project efficiency. Their approach puts a premium on early testing, collaboration as well as cultural quality measure that is the mainstay of agile projects since quality brings success.

4.2.3 Conclusion of Qualitative analysis

Via the experts in a software development field and Agile project management we have earned the gathering interesting facts concerning the Agile methodologies and comprehensive testing strategies integration. We concluded a standard of today is that early and iterative testing is a priority, test automation tools are used and Agile methods are implemented to ensure no software defects and no faults in the software are left unnoticed.

An examination of Scrum and Kanban types of the frameworks of agile project management showed that collaboration; adaptive planning and ongoing improvement are the key principles for these methodologies. Agile methods envolve specialists who set up the testing integration points in the testing strategy with The role of technology in global trade is undeniable and its impact is increasingly significant. Developing economies have experienced both positive and negative effects from technology adoption in the past decades, and it is important to understand these impacts when shaping the future of trade.

Also, the introduction of evaluation of the impact of integration of complete testing systems proved out improvements in key project metrics such as quality, customer satisfaction, and time to market. Experts' advice in the form of purposeful suggestions, eg. by investing in test automation and introducing quality as a culture, will be essential for enhancing Agile testing in software development.

4.3 Case Studies of Successful Agile Testing Implementations

In this part, we discuss several case studies of how agile testing approach is implemented with success in order to extract the knowledge, techniques, and lessons taken from real-life implementations. In this regard, we aim to draw beneficial implications from these scenarios that become a source of inspiration and guidance in an agile testing that is effective in a wide range of organizational contexts.

4.3.1 Company A Case Study

In this case study, we discuss the Company A, a renowned technology firm, which acknowledges its diverse and open culture that fosters creativity and innovation through software development practices. For this reason, and to observe the confidentiality and privacy of this firm, the name of the organization is kept confidential. The paper focuses on Company A's Agile testing savviness and its integration into Company A's project management steps.

Background

Company A is based on technology and focuses on software development for different areas like health, education, finances, and so on. Keeping pace with growing needs, a decision on implementing Agile methodologies in software development processes was made with the purpose of delivering excellent software now, and in the near future.

Integration of Agile Testing tactical measures

The project management team of the Company A has been in the driver's seat in a complete reengineering of the organization's software development with a specific focus put on integration of Agile testing approaches. Key steps in this implementation included:Key steps in this implementation included:

- ✓ Adoption of Agile Principles: The key Agile practice applied by Company A was that of Iterative development, integration, collaboration, and transparency. The creation of the cross-functional group helped bridge the communication gaps between the developers, testers, and the owners of a project.

- ✓ Continuous Integration (CI) Practices: For splicing the codelines modifications as well as finding defects at earliest stages of development, Company A deployed CI practices. Automated build and testing practices were set up strategically to facilitate the transition and approval of code at regular intervals.

- ✓ Test Automation Framework: Company A's major objective during this time was to codesign the framework to support automation testing as a means of shortening feedback cycles and increasing testing efficiency. The scripts to automate regression testing, unit testing, and acceptance testing came into being in the true sense, thus, saving manual effort and increasing the quality and precision of testing.

- ✓ Agile Metrics and Monitoring: Through Agile metrics and monitoring, Company A enabled more efficient project management, issue resolving, and informing us about the critical factors that influence the project, helping us make real-time data-based decisions. For instance, the metrics such as sprint velocity, defect density, and test coverage were used to calculate the progress of the project and the speed of the job performance on a regular basis.

Results and Impact

The implementation of Agile testing strategies yielded significant benefits for Company A, including:The implementation of Agile testing strategies yielded significant benefits for Company A, including:

- ✓ Accelerated Time-to-Market: Through Agile, Company A really saw a reduction in the time taken to release new product versions; making it faster to give customers more functionality and technological improvements.

- ✓ Improved Quality and Reliability: Eventually, it resulted in enhancing the automation of the tests and the continuity testing that ultimately brought the software of exceptional quality with fewer defects. The keeping of detecting and solving early errors is to eliminate rework and to make sure of the final product reliability..

- ✓ Enhanced Team Collaboration: The flexible practices enforced alignment among cross-functional teams, which was marked with transparency, responsibility, and shared possession.

Conclusion

Company A, a case study on Agile testing strategy implementation, is reflective of how Agile testing strategies work best when they blend with software development. Through implementation of the Agile principles, application of automation, and ranking of collaboration, companies ensure the tangible outcomes of the projects and thus, validate the adoptability of Agile methodologies in the IT sectors.

4.3.2 Company B Case Study

In this part, we analyze the compact agile testing applied by Company B, that is a giant in the software development market. As like that Company A's name has been not mentioned to maintain anonymity and privacy, the same apply to the name of Company B.

Background

Company B run its business amid stiff competition, as it is tuned towards offering software products to different kinds of clients spanning various business sectors. Having in focus to ensure supplying products of high quality in short time Company B realized that agile methodologies in its software development processes are of decisive importance.

Adoption of Agile Testing Strategies Introduction

The job of becoming the agile testing at Company B commenced with a full-fledged analysis of the existing development techniques and project management mechanism. With the push for more speed, responsiveness and teamwork between development and testing teams at Company B, agile practices (methodology) became favorable.

Integration of Agile Testing

The implementation of agile testing tactics at Company B was a project that required a lot of thinking and preparation during which we provided many different organizations with the testing practices using training sessions, workshops, and collaborative work with our team members. Past projects helped me identify the key stakeholders in the development, testing and project management departments to actively guide through the transition process.

Overcoming Challenges

Here are the strategies we employed to overcome these difficulties: effective communication, staff training and development, and most importantly, the creation of a working environment that would embrace change and be open to collaboration. Team bosses and project heads were the saviors in the process, they played key role in the whole change project, generally giving support and sorting other members' complaints.

Results and Achievements

Due to all these measures, Company B has seen drastic improvement in its project delivery and project outcomes in the recent past. Through the implementation of agile testing, defect elimination was achieved along with shorter time-to-market and better customer retention. The other was characterized with teamwork which ushered in faster innovation and enhanced creativity as well.

Conclusion

For Company B, the test case study shows how agile test techniques are effectively implemented in a fast-moving software industry. Despite the fact that Company B faced first time difficulties, applying agile philosophy to problem solving, as well as demonstrating leadership and team support enabled the company to gain the industry reputation it needed.

4.3.3 Company C case Study

In this paragraph, we analyse the triulate testing practices which the Company C that is the global power in the software creation industry employs. For the reason of non-disclosure, the name of the company (Company C) is not given.

Background of Company C

The company C, is competing in an environment where its core activity is providing IT solutions to its customers around the world. The Company C, in the wake of realizing the development of the industry and fast-changing trends, initiated the embarkment of agile methodologies to smoothen the development process.

Agile Testing Test Strategy

The establishment of agile testing at Company C was not a quick thing, instead the comprehensive evaluation of the existing development practices and the emerging market trends brought it to light as a better solution. Comprising of agility, adaptability and the customer-centricity, the agile principles were initiated for use in software development of the company.

Integration of Agile Testing

Team-based agile testing was a project at company C which involved cross-functional teams getting support from the company leaders. As a part of training, frequent programs, workshops, and one-on-one coaching support, as well as intro to agile testing, were conducted to enable team members to adapt themselves to agile approaches.

Challenges Faced

The ushering of agile testing into the organization of Company C was no free rite, either. The main challenges identified were resistance to change, the cultural barriers and the necessity of determining the new way to rum processes. However, we faced not only with making joint planning of development and testing teams teams smooth but also with many issues at the beginning.

Overcoming Challenges

To tackle that Company C made priority-shifting the change management initiatives as well as developing a culture of transparency and openness. The regular feedback loops, agile ceremonies, and continuous improvement endeavors were core characteristics for achieving the change of mindset and alignment levels of teams.

Results and Achievements

Agile testing worked well for Company C leading to problems resolution, customer satisfaction, and another benefit. This company C managed to reap the benefits of being agile by adapting quickly to to the dynamic market demands, making improvements based on customer feedbacks, and providing value-driven solutions.

Conclusion

The case of Company C whom the agile testing implementation has shown a significant change in the enterprise's software development approach and project outputs. Active leadership, joining forces with the conflicting parties, and a desire to deliver the best to the clients resulted in organization C taking up the lead in the field.

4.4 Practical Recommendations for Agile Testing Strategies

This section includes the options and recommendations resulting from the analysis of numbers and what we heard from the experts' interviews. The recommendations below serve as a guide for organizations in formulating and employing useful agile testing procedures that in turn improve the efficiency of the project and ensure timely delivery of high-quality software products.

Embrace the continuous integration (CI)

Agile software development implementation is impossible without continuous integration (CI). It is a primary practice, which consists of constant integration of code changes into a common repository. A strong infrastructure for CI that includes advanced automation tools is seen as a critical attribute that allows organizations to accomplish swift unification as well as the fast identification of defects. Through CI, the teams will implement this practice which puts stability in the code, improves collaboration and shortens feedback process consequently leading to quality and efficiency and on this project.

Prioritize Test Automation

Quantitative testing with automation becomes essential for agile testing in CI/CD approach, as it allows team to execute tests as fast as their is co-ordination throughout the whole lifecycle of the development. Priority should be intended on to execution of auto test projects and tools selection for the purpose of automation of repetitive tasks. Automating the regression, unit, and acceptance tests helps to save on manually efforts, to increase the test coverage, and to detect the flaws early on, and, hence, accelerating release speed and enhancing product's reliability which is in high demand.

Foster Cross-Functional Collaboration

Highly relevant and successful agile testing involves a solid cross-functional teamwork coupled with effective collaboration and communication among testers, developers, product owners, and stakeholders. Organizations are required to develop a culture where collaboration is valued and celebrated which is done by giving shared sense of purpose to each quality and by encouraging and promoting involvement of all the team structures from the very beginning. Through the removal of silos and the shared responsibility of individuals, teams enable agreement in the objectives, feedback updates and at the end, the results of the whole team will be superior.

Implement Shift-Left Testing Practices

The part of Shift-Left testing focused on the early participation of testing activities in the development life cycle from stage of the requirements to the design phase. Organizations need implement AI-driven and cloud-based Shift-Left practices to detect errors earlier, to apply preventative risk analytics, and to accomplish quality from the very beginning of the software development process. Teams can achieve lower costs with repeatability, as well as catch issues sooner and do less rework, thus delivering software of higher quality with greater certainty by doing exploratory testing, peer reviews and risk analysis before beginning the implementation phase.

Continuously Supervise and Develop the Efficiency

Learning curve based on improvement is a basic concept of agile philosophy and corporations should teach this idea to their own testing processes. Through ways such as feedback mechanism, retro, and constantly doing testing process reviews and its forecast, teams are able to take show where improvement is being made and implement the corrective actions iteratively. Providing training and instructing staff members to constantly look for better ways to test and monitor the progress of a project would lead into more effective, efficient, and successful projects.

Conclusion

Lastly, we have pointed out the vital role of agile testing strategies for the companies which want to deliver state-of-the-art software products in the fast-moving and competitive eco system. Through the adoption of continuous integration, placing quality at the top of the priority list, involving all departments in the process, by applying Shift-Left testing techniques, and always working to improve their software testing such organizations can gain speed, reach higher level of innovation and achieve the desired results. By offering these process-related recommendations, organizations receive executable roadmaps towards maximizing value returns to stakeholders with effective agile testing.

5. Results and discussion

Chapter 5 conducts an in-depth research synthesis of results from both qualitative survey analysis, as well as to provide exploration. The chapter of participants' demographics profiles, frequency of using Agile methodology utilization, building testing strategies into Agile project management, and what project Key success Metrics is the subject of the investigation. Here, the chapter deals with Agile testing practices in a wise way: it reports the findings of a comprehensive analysis, accompanied by thorough discussions. This provides valuable knowledge of where Agile testing stands at the moment, and how it relates to the success of a project. It in fact combines both practicing and theory offering something useful for practitioners and existing general body of knowledge in the field of Agile software development and project management.

5.1 Results

The survey results give us very important information which serve as a basis for taking an overall picture of the prevailing status of agile testing practices as well as their contribution to the success metrics (metrics measuring success). Neural network analysis of demographic data indicated diversity of participants in terms of experience, developers position, as well as by industry entities. The respondents who took part in the survey indicated that Agile methods were involved in their software development projects most times of cases, which is evidence of increased Agile practice adoption in the sector

As far as incorporating the entire testing landscape into Agile project management turned out, there had been big number of groups highly implementing testing, to a middle or large extent. This points to a growing tendency towards the idea in the Agile environment of the requirements of testing procedures and the plans to inject them into the Agile workflow.

The choice of methodology and practices that upgrade the Agile project management and testing had Scrum and Kanban among the top choices, however, many of the respondents equally adapt TDD, BDD and CI practices to their working method. In other words, it can be regarded as the samples of the type of tools that the teams develop to aid and support Agile project management and testing.

Besides, there were project success metrics measured, such as the general success of the Agile software development projects, the customer satisfaction of the delivered quality products, and the percentage of delivered qualified products. The most participants gave their recent Agile projects thumbs-up and the high level of project success results from the surveyed population. The quality of deliverables was rated positively by the majority of respondents. Interestingly enough, the satisfaction levels for the deliverables were high, with a substantial number of customers scoring 85% and above.

Although the main challenges were in sticking to schedule and the under performance of certain projects, other came as delays which were always experienced occasionally or rarely. Such cases indicate the necessity of the Agile project management practices improvement procedures to produce the effective schedule forecast that can guarantee the achievement of project objectives on time.

5.2 Discussion

Discussion of implications of survey findings in the greater scheme of test management and project delivery is covered in the last section of course. The survey demonstrates Agile proven effectiveness with such a high adoption level among Agile methodologies, and this further illustrates the agile effectivity in modern software development areas. The statistics where high success rates of Agile projects bear out the worthiness of Agile methodologies which tends to deliver value to their stakeholders and promoters collaboration as well as adapting to changing needs suitably.

Shifting to a more mature point starts from the ground. It includes implementing complete testing strategy as one of the Agile project management best practices. Through embracing the iterative testing method where test activities are embedded in development process as well as using a group of varied testing tools and techniques, it is possible to deliver high-quality software solutions that meet customer needs and business objectives.

The conclusion that Agile processes generate positive results is evident based on survey results; however, the observation of the inappropriate completion of the planned deadlines points out the areas for the improvement of the Agile project management processes. Overcoming above-mentioned challenges is a multi-sided issue that imply higher accuracy of estimations, teammembers' synchronization and coordination as well as risk mitigation and routing strategies implementaion.

Basically, the survey results gain the most precious data of agile testing practices which are used as de facto now and metrics of success. Adhering to the drivers of project execution builds a contrasting landscape from the foundations of which organizations can optimize their approaches to Agile test practices, subsequently creating a better alignment with complimenting management systems and active collaboration with stakeholders.

6. Conclusion

This thesis undertook to carry out a thorough investigation of the complex synergy relationship between Agile project management and software testing strategies which resulted in an in-depth understanding of their consorted dynamics. By careful analysis and empirical research, one gains important understandings about how often and deep Agile approach is adopted, testing methods integration into project management frameworks. These insights, supported by sector specific conclusions, provide a strong platform for furthering organizational effectiveness and promoting continuous improvement.

The practical recommendations outlined in this research are in fact practical prescriptions for industry practitioners and provide concrete ways to improve Agile testing practices and project outcomes. Combining the theoretical bases with the empirical evidence this research not only tackles current problems but also leads the way for future innovation and development in Agile testing practices. Given the continuous evolution of the software development landscape, this thesis is a guiding light to Agile project management and software testing, reflecting the commitment to quality and innovation for all time.

Moreover, the synthesis of different approaches and methods emphasizes the interdisciplinary quality of this investigation, adding to the discourse diverse points of view and perspectives. This thesis brings theory and practice together by taking both a quantitative and qualitative approach, offering practical solutions based on empirical evidence and scholarly discourse.

Furthermore, the findings of this study highlight the need to continue cooperation and knowledge sharing among the stakeholders within the software development ecosystem. Through continuous learning and flexibility, the harmonization of Agile project management and comprehensive testing strategies will allow organizations to derive the benefits of Agile to stimulate innovation, improve product quality, and, ultimately, to establish success in the volatile market of today.

In fact, this thesis is a testament to the fact that the pursuit of excellence and innovation in Agile project management and software testing is still ongoing. As organizations contend with the challenges of modern software development, the information obtained from this research provides important direction and motivation for achieving the full efficiency value of Agile methodologies and testing approaches for organizational growth and developing an environment of continuous improvement.

7. References

IRSHAD, Mohsin, BRITTO, Ricardo and PETERSEN, Kai, 2021. Adapting Behavior Driven Development (BDD) for large-scale software systems. *Journal of Systems and Software* [online]. July 2021. Vol. 177, p. 110944. DOI 10.1016/j.jss.2021.110944. Retrieved from: http://dx.doi.org/10.1016/j.jss.2021.110944

Agile Project Management: Best Practices and Methodologies, 2023. *AltexSoft* [online]. Retrieved from: https://www.altexsoft.com/whitepapers/agile-project-management-best-practices-and-methodologies/

ALI, Josh Mahmood, 2023. DevOps and continuous integration/continuous deployment (CI/CD) automation. *Advances in Engineering Innovation* [online]. 27 November 2023. Vol. 4, no. 1, p. 38–42. DOI 10.54254/2977-3903/4/2023031. Retrieved from: http://dx.doi.org/10.54254/2977-3903/4/2023031

20 Best Agile Tools for Effective Management in 2024, 2024. [online]. Retrieved from: https://www.knowledgehut.com/blog/agile/top-agile-tools#top-agile-tools-for-effective-project-management%C2%A0

JONES, Capers and BONSIGNOUR, Olivier, 2012. *The Economics of Software Quality* [online]. Addison-Wesley. Retrieved from: http://books.google.ie/books?id=_t5l5Cn0NBEC&printsec=frontcover&dq=Quality+Metrics+testing&hl=&cd=3&source=gbs_api

BASU, Anirban, 2015. *SOFTWARE QUALITY ASSURANCE, TESTING AND METRICS* [online]. PHI Learning Pvt. Ltd. Retrieved from: http://books.google.ie/books?id=aNTiCQAAQBAJ&printsec=frontcover&dq=Quality+Metrics+testing&hl=&cd=2&source=gbs_api

KAN, Stephen H., 2003. *Metrics and Models in Software Quality Engineering* [online]. Addison-Wesley Professional. Retrieved from: http://books.google.ie/books?id=EaefcL3pWJYC&printsec=frontcover&dq=Quality+Metrics+testing&hl=&cd=1&source=gbs_api

RIEL, Andreas, O’CONNOR, Rory, TICHKIEWITCH, Serge and MESSNARZ, Richard, 2010. *Systems, Software and Services Process Improvement* [online].

Springer. Retrieved from: http://books.google.ie/books?id=j7GqCAAAQBAJ&pg=PA67&dq=Impact+of+Testing+Integration+on+Project+Success&hl=&cd=1&source=gbs_api

SMART, John Ferguson and MOLAK, Jan, 2023. *BDD in Action, Second Edition* [online]. Simon and Schuster. Retrieved from: http://books.google.ie/books?id=hIK3EAAAQBAJ&printsec=frontcover&dq=Behavior-Driven+Development&hl=&cd=5&source=gbs_api

BECK, Kent, 2022. *Test Driven Development* [online]. Addison-Wesley Professional. Retrieved from: http://books.google.ie/books?id=zNnPEAAAQBAJ&printsec=frontcover&dq=Test-Driven+Development+(TDD)&hl=&cd=1&source=gbs_api

AGARWAL, B. B., TAYAL, S. P. and GUPTA, Mahesh, 2010. *Software Engineering and Testing* [online]. Jones & Bartlett Learning. Retrieved from: http://books.google.ie/books?id=ZoF06z4dhQ4C&pg=PA168&dq=Integration+testing&hl=&cd=10&source=gbs_api

WHITTAKER, James A., 2009. *Exploratory Software Testing* [online]. Pearson Education. Retrieved from: http://books.google.ie/books?id=BsB0NpkcdgIC&printsec=frontcover&dq=%EF%83%BC%09Exploratory+Testing&hl=&cd=1&source=gbs_api

LAWRENCE, Richard and RAYNER, Paul, 2019. *Behavior-Driven Development with Cucumber* [online]. Addison-Wesley Professional. Retrieved from: http://books.google.ie/books?id=TLaZDwAAQBAJ&printsec=frontcover&dq=%EF%83%BC%09Behavior-Driven+Development&hl=&cd=2&source=gbs_api

DUVALL, Paul M., MATYAS, Steve and GLOVER, Andrew, 2007. *Continuous Integration* [online]. Pearson Education. Retrieved from: http://books.google.ie/books?id=PV9qfEdv9L0C&printsec=frontcover&dq=%EF%83%BC%09Continuous+Integration&hl=&cd=2&source=gbs_api

SECURITY FOR CONTINUOUS INTEGRATION AND CONTINUOUS DEPLOYMENT PIPELINE, 2024. *International Research Journal of Modernization* *in Engineering Technology and Science* [online]. DOI 10.56726/irjmets50676. Retrieved from: http://dx.doi.org/10.56726/irjmets50676

ÖZÇELIK, Onur and TURGAY ALTILAR, D., 2015. Test-Driven Approach for Safety-Critical Software Development. *Journal of Software* [online]. July 2015. Vol. 10, no. 7, p. 904–911. DOI 10.17706//jsw.10.7.904-911. Retrieved from: http://dx.doi.org/10.17706//jsw.10.7.904-911

LINZ, Tilo, 2014. *Testing in Scrum* [online]. Retrieved from: http://books.google.ie/books?id=40ptAQAACAAJ&dq=Integration+Points+of+Testing+Strategies+and+Agile+Methodologies&hl=&cd=4&source=gbs_api

MARTIN, Kc, 2021. *The Agile Software Tester: Software Testing In The Agile World* [online]. Kevsbooks. Retrieved from: http://books.google.ie/books?id=E5rsDwAAQBAJ&printsec=frontcover&dq=Integration+Points+of+Testing+Strategies+and+Agile+Methodologies&hl=&cd=2&source=gbs_api

AUGUSTINE, Sanjiv, 2005. *Managing Agile Projects* [online]. Pearson. Retrieved from: http://books.google.ie/books?id=glBPAAAAMAAJ&q=agile+project&dq=agile+project&hl=&cd=10&source=gbs_api

CRADDOCK, Andrew, ROBERTS, Barbara, STAPLETON, Jennifer Et Al, GODWIN, Julia and CONSORTIUM, Agile Business, 2019. *Agile Project Management* [online]. Retrieved from: http://books.google.ie/books?id=FE9mAQAACAAJ&dq=agile+project&hl=&cd=8&source=gbs_api

HIGHSMITH, Jim, 2004. *Agile Project Management* [online]. Pearson Education. Retrieved from: http://books.google.ie/books?id=b1uwZORz2w8C&printsec=frontcover&dq=agile+project&hl=&cd=6&source=gbs_api

HIGHSMITH, Jim, 2009. *Agile Project Management* [online]. Pearson Education. Retrieved from:

http://books.google.ie/books?id=VuFpkztwPaUC&printsec=frontcover&dq=agile+project&hl=&cd=4&source=gbs_api

AGUANNO, Kevin, 2005. *Managing Agile Projects* [online]. Multi-Media Publications Inc. Retrieved from: http://books.google.ie/books?id=TJgsl4WqrzoC&printsec=frontcover&dq=agile+project&hl=&cd=3&source=gbs_api

ABRAHAMSSON, Pekka, BASKERVILLE, Richard, CONBOY, Kieran, FITZGERALD, Brian, MORGAN, Lorraine and WANG, Xiaofeng, 2008. *Agile Processes in Software Engineering and Extreme Programming* [online]. Springer Science & Business Media. Retrieved from: http://books.google.ie/books?id=ZUj58wzLtkUC&printsec=frontcover&dq=burn-down+Charts&hl=&cd=6&source=gbs_api

DALTON, Jeff, 2018. *Great Big Agile* [online]. Apress. Retrieved from: http://books.google.ie/books?id=ZxF-DwAAQBAJ&printsec=frontcover&dq=burn-down+Charts&hl=&cd=1&source=gbs_api

COHN, Mike, 2005. *Agile Estimating and Planning* [online]. Pearson Education. Retrieved from: http://books.google.ie/books?id=BuFWHffRJssC&printsec=frontcover&dq=agile+project+velocity&hl=&cd=2&source=gbs_api

JURICEK, Jan, 2014. Agile Project Management Principles. *Lecture Notes on Software Engineering* [online]. 2014. P. 172–175. DOI 10.7763/lnse.2014.v2.117. Retrieved from: http://dx.doi.org/10.7763/lnse.2014.v2.117

Agile Practices and Impacts on Project Success, 2021. *Journal of Engineering, Project, and Production Management* [online]. DOI 10.2478/jeppm-2021-0024. Retrieved from: http://dx.doi.org/10.2478/jeppm-2021-0024

DAVIS, Christopher, 2015. *Agile Metrics in Action* [online]. Simon and Schuster. Retrieved from: http://books.google.ie/books?id=sjszEAAAQBAJ&printsec=frontcover&dq=Agile+Project+Success+Metrics&hl=&cd=1&source=gbs_api

BLOKDYK, Gerardus, 2018. *Regression Testing* [online]. Createspace Independent Publishing Platform. Retrieved from: http://books.google.ie/books?id=pb4rswEACAAJ&dq=Regression+Testing&hl=&cd=6&source=gbs_api

FALAH, Bouchaib and MARGHABI, Souhail, 2014. Towards Regression Testing Constraints. *International Journal of Modeling and Optimization* [online]. December 2014. Vol. 4, no. 6, p. 504–509. DOI 10.7763/ijmo.2014.v4.425. Retrieved from: http://dx.doi.org/10.7763/ijmo.2014.v4.425

PANDIT, Pallavi and TAHILIANI, Swati, 2015. AgileUAT: A Framework for User Acceptance Testing based on User Stories and Acceptance Criteria. *International Journal of Computer Applications* [online]. 18 June 2015. Vol. 120, no. 10, p. 16–21. DOI 10.5120/21262-3533. Retrieved from: http://dx.doi.org/10.5120/21262-3533

HAMBLING, Brian and VAN GOETHEM, Pauline, 2013. *User Acceptance Testing* [online]. BCS, The Chartered Institute for IT. Retrieved from: http://books.google.ie/books?id=NzDkkgEACAAJ&dq=Acceptance+Testing&hl=&cd=1&source=gbs_api

AGARWAL, B. B., TAYAL, S. P. and GUPTA, Mahesh, 2010. *Software Engineering and Testing* [online]. Jones & Bartlett Learning. Retrieved from: http://books.google.ie/books?id=ZoF06z4dhQ4C&pg=PA180&dq=System+Testing&hl=&cd=7&source=gbs_api

DRABICK, Rodger, 2013. *Best Practices for the Formal Software Testing Process* [online]. Addison-Wesley. Retrieved from: http://books.google.ie/books?id=bVcUAAAAQBAJ&printsec=frontcover&dq=System+Testing&hl=&cd=2&source=gbs_api

BAYLIS, Michael, BONNER, Russell, DAVIES, James, HARRIS, Clive, MCNAMARA, Caroline, MOXEY, Catherine, SOMERS, Richard, YATES, Will and REDBOOKS, Ibm, 2021. *Integration Testing for Hybrid Cloud Applications using Galasa* [online]. IBM Redbooks. Retrieved from:

http://books.google.ie/books?id=gVcmEAAAQBAJ&printsec=frontcover&dq=Integration+Testing&hl=&cd=4&source=gbs_api

FRANKEL, Nicolas, 2015. *Integration Testing from the Trenches* [online]. Monsieur Nicolas Frankel. Retrieved from: http://books.google.ie/books?id=08TwrQEACAAJ&dq=Integration+Testing&hl=&cd=2&source=gbs_api

KOSKELA, Lasse, 2013. *Effective Unit Testing* [online]. Simon and Schuster. Retrieved from: http://books.google.ie/books?id=8zgzEAAAQBAJ&printsec=frontcover&dq=unit+testing&hl=&cd=3&source=gbs_api

KHORIKOV, Vladimir, 2020. *Unit Testing Principles, Practices, and Patterns* [online]. Simon and Schuster. Retrieved from: http://books.google.ie/books?id=rDszEAAAQBAJ&printsec=frontcover&dq=unit+testing&hl=&cd=1&source=gbs_api

DRABICK, Rodger, 2013. *Best Practices for the Formal Software Testing Process* [online]. Addison-Wesley. Retrieved from: http://books.google.ie/books?id=bVcUAAAAQBAJ&printsec=frontcover&dq=Software+Testing+Practices&hl=&cd=1&source=gbs_api

NAIK, Kshirasagar and TRIPATHY, Priyadarshi, 2011. *Software Testing and Quality Assurance* [online]. John Wiley & Sons. Retrieved from: http://books.google.ie/books?id=neWaoJKSkvgC&printsec=frontcover&dq=Software+Testing+Practices&hl=&cd=2&source=gbs_api

DRABICK, Rodger, 2013. *Best Practices for the Formal Software Testing Process* [online]. Addison-Wesley. Retrieved from: http://books.google.ie/books?id=bVcUAAAAQBAJ&printsec=frontcover&dq=Software+Testing+Practices&hl=&cd=1&source=gbs_api

AGUANNO, Kevin, 2005. *Managing Agile Projects* [online]. Multi-Media Publications Inc. Retrieved from:

http://books.google.ie/books?id=TJgsl4WqrzoC&printsec=frontcover&dq=project+management+agile&hl=&cd=2&source=gbs_api

CROWDER, James A. and FRIESS, Shelli, 2014. *Agile Project Management: Managing for Success* [online]. Springer. Retrieved from: http://books.google.ie/books?id=cAZYBAAAQBAJ&printsec=frontcover&dq=project+management+agile&hl=&cd=1&source=gbs_api

GREBIĆ, Bojan, 2019. Traditional vs Agile Project Management in the Service Sector. *European Project Management Journal* [online]. 2019. Vol. 9, no. 2, p. 55–63. DOI 10.18485/epmj.2019.9.2.6. Retrieved from: http://dx.doi.org/10.18485/epmj.2019.9.2.6

WESTLAND, Jason, 2007. *The Project Management Life Cycle* [online]. Kogan Page Publishers. Retrieved from: http://books.google.ie/books?id=4ph42sGN9vwC&printsec=frontcover&dq=project+management&hl=&cd=1&source=gbs_api

BRECHNER, Eric, 2015. *Agile Project Management with Kanban* [online]. Microsoft Press. Retrieved from: http://books.google.ie/books?id=eEzTBgAAQBAJ&printsec=frontcover&dq=kanban+agile&hl=&cd=1&source=gbs_api

LEHEKAR, Makarand Prabhakar and URNE, Prof. Mangesh, 2017. Implemetation of Kanban. *International Journal of Trend in Scientific Research and Development* [online]. 31 October 2017. Vol. Volume-1, no. Issue-6, p. 598–621. DOI 10.31142/ijtsrd4616. Retrieved from: http://dx.doi.org/10.31142/ijtsrd4616

SUNDEN, Joakim and HAMMARBERG, Marcus, 2014. *Kanban in Action* [online]. Simon and Schuster. Retrieved from: http://books.google.ie/books?id=-zkzEAAAQBAJ&printsec=frontcover&dq=kanban&hl=&cd=3&source=gbs_api

MOEENI, F., SANCHEZ, S.M. and VAKHA RIA, A.J., 1997. A robust design methodology for Kanban system design. *International Journal of Production Research* [online]. October 1997. Vol. 35, no. 10, p. 2821–2838.

DOI 10.1080/002075497194462. Retrieved from: http://dx.doi.org/10.1080/002075497194462

TRUSS, Mario, 2024. Scaling Scrum with the Scaled Agile Framework: Potentials and Importance in Large Development Organizations. *SSRN Electronic Journal* [online]. 2024. DOI 10.2139/ssrn.4683212. Retrieved from: http://dx.doi.org/10.2139/ssrn.4683212

SACHDEVA, Sakshi, 2016. Scrum Methodology. *International Journal Of Engineering And Computer Science* [online]. 6 June 2016. DOI 10.18535/ijecs/v5i6.11. Retrieved from: http://dx.doi.org/10.18535/ijecs/v5i6.11

BRECHNER, Eric, 2015. *Agile Project Management with Kanban* [online]. Microsoft Press. Retrieved from: http://books.google.ie/books?id=eEzTBgAAQBAJ&printsec=frontcover&dq=kanban&hl=&cd=8&source=gbs_api

SCHWABER, Ken and BEEDLE, Mike, 2002. *Agile Software Development with Scrum* [online]. Pearson. Retrieved from: http://books.google.ie/books?id=BpFYAAAAYAAJ&q=Schwaber&dq=Schwaber&hl=&cd=3&source=gbs_api

SCHWABER, Ken, 2007. *The Enterprise and Scrum* [online]. Microsoft Press. Retrieved from: http://books.google.ie/books?id=pjLcBAAAQBAJ&printsec=frontcover&dq=Schwaber&hl=&cd=2&source=gbs_api

MISHRA, Alok and ALZOUBI, Yehia Ibrahim, 2023. Structured software development versus agile software development: a comparative analysis. *International Journal of Systems Assurance Engineering and Management* [online]. 12 June 2023. Retrieved from: https://link.springer.com/article/10.1007/s13198-023-01958-5

AQUINO, Anderson, 2023. Gains and Challenges of agile methodologies in systems and software. [online]. 13 September 2023. Retrieved from: https://www.linkedin.com/pulse/gains-challenges-agile-methodologies-systems-software-anderson-aquino

8. List of tables, graphs, pictures and abbreviations

8.2 List of tables

8.2 List of graphs

8.1 List of pictures

8.3 List of abbreviations

Agile: Agile methodologies

QA: Quality Assurance

IT: Information Technology

Kanban: A visual system for managing work as it moves through a process

WIP: Work in Progress

Kaizen: Continuous improvement, a Japanese business philosophy

TDD: Test-Driven Development

RAD: Rapid Application Development

OAT: Operational Acceptance Testing

UAT: User Acceptance Testing

KPIs: Key Performance Indicators

BDD: Behavior-Driven Development

NPS: Net Promoter Score

CI: Continuous Integration

MTTR: Mean Time to Repair

CSAT: Customer Satisfaction Rating

CD: Continuous Delivery

PM: Project Manager

AC: Agile Coach

Appendix

Survey Details

- ✓ **Demographic**

1. **Years of Experience in Software Development**
   - Less than 1 year
   - 1-3 years
   - 4-6 years
   - 7-10 years
   - More than 10 years

2. **What is your current role in software development?**
   - Developer
   - Tester
   - Project Manager
   - Quality Assurance Engineer
   - Scrum Master

3. **Industry Sector:**
   - Technology
   - Finance
   - Healthcare
   - Education
   - Other

- ✓ **Questionnaire**

1. **How frequently does your team utilize Agile methodologies in software development projects?**
   - Not at all
   - Rarely
   - Occasionally
   - Frequently

- Always

2. **To what extent does your team integrate comprehensive testing strategies into Agile project management?**
    - Not at all
    - Slightly
    - Neutral
    - Moderately
    - Significantly

3. **Which tools or technologies does your team use most for implementing Agile project management and testing strategies?**
    - Scrum
    - Kanban
    - Test-Driven Development (TDD)
    - Behavior-Driven Development (BDD)
    - Continuous Integration (CI)

4. **How would you rate the overall success of your recent Agile software development projects?**
    - Very unsuccessful
    - Unsuccessful
    - Neutral
    - Successful
    - Very successful

5. **how satisfied are your customers with the quality of deliverables in your Agile projects?**
    - Around 30 %
    - Around 50 %
    - Around 70 %
    - Around 85 %

- > 85%

6. **How frequently do your Agile projects meet their planned deadlines?**
    - Never
    - Rarely
    - Sometime
    - Often
    - Always