



SEEK WISDOM, ELEVATE YOUR INTELLECT AND SERVE HUMANITY!

Addis Ababa University  
አዲስ አበባ ዩኒቨርሲቲ



# **Assessment of Agile Project Management Practices: The Case of Excellerent Technology Solutions**

Amanuel Degu

A project work report submitted to Addis Ababa University School of Commerce  
in partial fulfillment of the requirements for a Master of Art Degree in Project  
Management.

Advisor: Abraraw Chane (Ph.D.)

School of Commerce, College of Business and Economics

Addis Ababa University

September 2024

## **Statement of Declaration**

I, Amanuel Degu, hereby declare that the thesis entitled “Assessment of Agile Project Management Practices: The Case of Excellerent Technology Solutions” has been conducted by me under the guidance and supervision of Abraraw Chane (Ph.D.). I also declare that all materials and sources used for this project research have been accredited appropriately. I am also declaring that this work has not been submitted for the award of any academic Degree or Diploma Program in this or any other institution.

**Amanuel Degu**

Signature: \_\_\_\_\_

Date: \_\_\_\_\_

## **Statement of Certification**

This is to certify that Amanuel Degu has carried out this project work on the topic entitled “Assessment of Agile Project Management Practices: The Case of Excellent Technology Solutions” under my supervision. This project work is original and suitable for submission in partial fulfillment of the requirement for the award of a Master of Arts Degree in Project Management.

**Abraraw Chane (Ph. D.)**

Signature: \_\_\_\_\_

Date: \_\_\_\_\_

## Statement of Approval

This project report, entitled “Assessment of Agile Project Management Practices: The Case of Excellerent Technology Solutions,” prepared and submitted by Amanuel Degu in partial fulfillment of the requirements for the degree of Master of Art in Project Management, is hereby approved and accepted.

### Advisor

<u>Abraraw Chane (Ph. D.)</u>	_____	_____
Name	Signature	Date

### Internal Examiner

<u>Mengistu Bogale (Ph. D.)</u>	_____	_____
Name	Signature	Date

### External Examiner

<u>Tesfa Nega (Ph. D.)</u>	_____	_____
Name	Signature	Date

## **Acknowledgment**

Throughout the preparation of this project work report, I have been supported by several people. I am grateful to have such kind and endearing people around me. This report would have not been realized without the kind and relentless support and guidance from my advisor, Abraraw Chane (Ph. D.). I would like to thank him for everything he did. I would like to thank all, who have persistently motivated and encouraged me to do it.

Thank you.

Amanuel Degu

## **Abstract**

*This case study, titled “Assessment of Agile Project Management Practices: The Case of Excellent Technology Solutions”, focuses on the implementation and effectiveness of agile project management methodologies within Excellent Technology Solutions, a technology company operating in Ethiopia. Agile methodologies have gained prominence in the software industry due to their flexibility and adaptability to changing requirements and technological advancements. The study highlights the suitability of Agile approaches for software development due to the inherent characteristics of such projects, including their iterative nature, evolving requirements, and the need for flexibility. Data was collected through a census survey administered to all employees to analyze the overall process involving agile project management on project timelines, quality, and overall satisfaction within the software development teams. From the result, it is found that the framework used is the Scrum framework. Key findings also include the positive influence of Scrum in fostering collaboration, adaptability, and continuous improvement. The study also identifies challenges encountered during Scrum implementation and provides recommendations for optimizing its usage within Excellent Technology Solutions. The results offer valuable insights for organizations considering Agile methodologies, particularly Scrum, in their software development processes, and lay the groundwork for future comparative studies on the effectiveness of different Agile frameworks in various organizational contexts.*

**Keywords:** *Agile project management, Scrum framework, Software development, Excellent Technology Solutions*

# Contents

Chapter 1 Introduction .....	1
1.1 Background of the Study.....	1
1.2 Statement of the Problem .....	4
1.3 Research Objectives .....	5
1.4 Research Questions .....	6
1.5 Significance of Study .....	6
1.6 Scope of the Study.....	7
1.7 Limitations of the Study.....	7
1.8 Operational Definition of Terms .....	7
1.9 Organization of the Study .....	8
Chapter 2 Related Literature Review.....	10
2.1 Overview .....	10
2.2 Software Projects.....	12
2.3 Challenges of Software Projects .....	14
2.4 Project Management.....	14
2.4.1 Traditional Project Management.....	16
2.4.2 Agile Project Management .....	19
2.5 Agile Project Management Frameworks.....	21
2.5.1 Scrum .....	21
2.5.2 Extreme Programming (XP) .....	23
2.5.3 Feature-Driven Development.....	25
2.5.4 Lean Software Development.....	27
2.5.5 Kanban .....	28
2.6 Agile Project Management Issues and Challenges .....	30

2.7	Empirical Review .....	30
Chapter 3 Research Methodology .....		32
3.1	Research Design .....	32
3.2	Research Approach .....	32
3.3	Target Population and Sampling .....	32
3.4	Data Collection Techniques .....	33
3.5	Data Analysis .....	33
3.6	Reliability and Validity .....	34
3.7	Ethical Considerations.....	34
Chapter 4 Data Analysis and Presentation .....		35
4.1	Overview .....	35
4.2	Participants' Demographic Profile.....	35
4.3	Experience of Team Members .....	38
4.4	Agile Methodology Practices .....	41
4.5	Agile Software Development Methodology Challenges.....	46
Chapter 5 Conclusions and Recommendations.....		48
5.1	Overview .....	48
5.2	Summary of Major Findings .....	48
5.3	Conclusions .....	49
5.4	Recommendations .....	51
References.....		53

## List of Figures

Figure 2.1 Tasks in Software Projects (Ahmed, 2012).....	13
Figure 2.2 Project management in the waterfall model (Ahmed, 2012).....	18
Figure 2.3 Incremental and Iterative nature of Agile methods (Stanley & Gross, 2020).....	20
Figure 2.4 The Agile Manifesto (The Agile Manifesto Authors, 2001).....	21
Figure 2.5 The Scrum Approach (Layton et al., 2020).....	22
Figure 2.6 The FDD Lifecycle (AgileModeling.com, 2022).....	26
Figure 2.7 Kanban System (Zayat & Senvar, 2020).....	29
Figure 4.1 Participants' experience in the IT industry .....	37
Figure 4.2 Participants' Field of Specialization .....	38
Figure 4.3 Agile Methodology Experience.....	41
Figure 4.4 Have you worked on projects with non-agile software development methodology? .	42
Figure 4.5 Familiarization with Agile Methodologies.....	42
Figure 4.6 Participants practicing Agile Events .....	45
Figure 4.7 Sprint Planning with Velocity .....	45
Figure 4.8 Standup meetings frequency.....	46
Figure 4.9 Challenges in Agile Software Development .....	47

## List of Tables

Table 4.1 Demographic Profile of Participants.....	35
Table 4.2 Participants' Experience .....	39
Table 4.3 Prior Agile Software Development (ASD) Experience of Participants.....	40
Table 4.4 Project Management Training .....	41
Table 4.5 Preferred Communication Means .....	43
Table 4.6 Frequency of Discussion with Product Owner .....	44
Table 4.7 Sprint retrospective meetings frequency.....	46

# **Chapter 1 Introduction**

Project management engages in carrying out well-planned activities involving management skills to achieve project objectives. Depending on the context of the project, one project management methodology could become more effective than another. Different methodologies, designed to overcome project management challenges, have been practiced over time. However, the agile project management methodology is gaining more acceptance for projects with highly volatile specifications. This study aims to assess agile project management practices in an offshore software development company.

This chapter provides the introduction of the study discussing the background and context, followed by the problem statement, the research questions, the significance of the study and the limitations.

## **1.1 Background of the Study**

A project is defined as a sequence of unique, complex, and connected activities that have one goal or purpose and that must be completed by a specific time, within budget, and according to specification (Wysocki, 2019). A project is a unique undertaking, which is temporary and has a defined duration, to create a unique product or service.

The product of a project could be different based on the objective set and the activities carried out. It could be a new product or service, a new promotional campaign, a new software, a new or improved process, etc. The outcome of the project may be tangible or intangible. Although repetitive elements may be present in some project deliverables and activities, this repetition does not change the fundamental, unique characteristics of the project work (PMI, 2013a). A project has set of goals and objectives that must be achieved within a certain duration and within set of limited resources. Albeit these set of constraints, the quality of the outcome of the project should not be compromised. Based on the context and these conditions, a project possesses unique characteristics; and depending on the scope and complexity of the product, its management task could also be complex.

There could be various as to why projects are initiated. A new market demand driven by change in taste preferences, a need to process improvement, getting competitive advantage, etc. could all

be possible reasons to embark on projects. There are various stakeholders, entities who are affected by its outcome, of a project. Project owners has dire interest of getting the benefit as soon as possible. However, the lack of accurate and detailed requirements definition at the beginning makes getting the project done difficult. This is most often prevalent in software projects. So as to minimize the occurrence of mishaps in projects, it is important to follow and adhere to project management principles.

Project management is the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements. Project management is accomplished through the appropriate application and integration of the project management processes identified for the project. Project management enables organizations to execute projects effectively and efficiently (PMI, 2017). Regardless of the scope and objective of the project, project management possesses sequence of activities that need to be managed and controlled. As projects have definite time that begins with initiating it and end with closing, managing projects is essentially managing the life cycle of the project; this is achieved using processes. A project life cycle is the full project duration from beginning to end, including all project phases (Sanghera, 2019). The processes describe the activities to be completed throughout the life cycle of the project.

The description of these processes is too general and the actual processes and activities vary depending on the scope, the industry, the team and the nature of the project itself. While carrying out these processes, in order to remain relevant, and viable a project management need to adapt to the ever changing business environment. Modern project management methodologies are developed in a way embracing this change and allow the project to be effective. This is enabled by making project management processes to be adaptable to the context.

This adaptability requires agility while managing projects. The dictionary meaning of agile refers to the ability to move quickly and easily; it is also defined as ability to think quickly and clearly. This implies the ability to adapt to changes coming in the way. Agile is a descriptor of a mindset approach to project management that focuses on early delivery of business value, continuous improvement of the product being created and the processes used to create the product, scope flexibility, team input, and delivering well-tested products that reflect customer needs (Layton et al., 2020). Although, the methodology can be applicable to any kind of projects that requires flexible interactions to the environment, it is highly acceptable and popular in software projects.

During the late 1990s the increased number of software projects following the popularity of software intensive systems has had challenges to keep scope changes at bay. However, the frequent changes in requirement specifications and the pace of technological advancements was a problem in project management. Hence, the need to develop an adapting framework.

In 2001, a group of software and project experts got together to talk about what their successful projects had in common. This group created the Manifesto for Agile Software Development (commonly referred to as the Agile Manifesto), a statement of values for successful software development (Layton et al., 2020). This phenomenon has used the term 'agile' for project management and heralded the era of agile software methodologies. Agile is a broad term for a variety of techniques and methodologies that share the principles and values outlined in the manifesto. Its use of iterative and incremental approach makes it suitable for software projects.

Even though there are different types of software projects, Software project management is the integration of management techniques into software development (Villafiorita, 2014). Software project management deals with software projects and the challenges of human-based development (as opposed to the more deterministic processes in traditional projects). The higher flexibility in software development approaches puts new demands on the capabilities of software project management (Ruhe & Wohlin, 2014). Software development is an activity involving creation and/or customization so that a new software product will be produced.

Technology trends and the continuing need for product differentiation, globalization, and its effect on the market and processes for new technology introduction accelerate system change, which makes the trade-off between the speed of development and the costs of development a necessity. When this is coupled with the increasing complexity of software and software intensive systems, it makes difficult handling software projects using the traditional project management methodologies. Therefore, agile project management methodologies involving a small member of self-organizing teams with short cycles is adapted. This enables responding to changes and allow communication and collaboration with clients.

This advantage of forming a collaborative team with short life cycles enable to embrace change in product requirements. It allows handling the volatility of client requirements, which helps to control scope bloat or minimize its effect on the overall project performance. Due to this reason, currently most software development projects have utilized one or another kind of agile

methodologies. It is particular very useful for companies engaged in offshore delivery services, where client involvement is invaluable. One such company operating in Ethiopia is Excellent Technology Solutions. The company opened its Ethiopia delivery center and started its operation in Ethiopia around 2019.

Founded in 2012, Excellent is a US-headquartered provider of technology and human resource solutions with professionals located within the US and Ethiopia. The company leverages both local in international talent via its offshore service delivery centers to deliver digital transformational products and services to clients. Currently, it has three offices located in Pittsburgh, Addis Ababa, and Mumbai (Excellent Technology Solutions, 2020).

Excellent Technology Solutions is a technology company specializing in providing digital solutions to clients through innovative methods and processes at an uncompromised level of quality. It uses the state-of-the-art technology stack in software development and aspires to the frontier in the field in Africa. Excellent is putting effort into fitting its name and being the excellence. In doing so, the company recruits top software engineers from the market. These engineers will engage in software development processes that are expected to deliver quality products faster. In a competitive world, the company has to perform better so as to increase customer satisfaction.

Recently, Excellent merged with Relevance Lab, along with CIGNEX (a global consulting company offering solutions, services, and platforms on Open Source, Cloud, and Automation technologies), to become a global powerhouse in digital transformation and cloud services. With this merger, Relevance Lab, headquartered in Singapore, will have a delivery presence across North America, India, and Ethiopia.

## **1.2 Statement of the Problem**

In today's world of software-intensive systems, it is clearly visible that there are an increased number of software projects. Particularly focusing on software development, organizations embark on automating their processes using new product, improved or customized product, or commercial-of-the-shelf products. Current technological developments also allow companies to utilize software as a service provisions. All these endeavors require careful execution of project management processes. The nature of software dictates the frequent change in requirements,

attributed to consumer characteristics, technological advancements, ICT infrastructure constraints, etc. Managing such projects using the traditional project management where by upfront planning is promoted and almost perfect requirement is available before hand, is very challenging and leads to minimal success. Hence, organizations engaged in software development projects are heavily applied agile project management methodologies.

In order to increase customer satisfaction along with their efficiency and effectiveness, technology companies embrace agile methodologies that enable delivering quality products incrementally. Offshore companies such as Excellerent Technology Solutions are reliable in their performance for their viability in the industry. Therefore, the company has to evaluate how it is performing, how it practices the agile project management methodology, whether it can improve, etc. The level of growth of the software development practice in the country coupled with the personality characteristics of team members could affect its effectiveness.

Due to the limited number of offshore companies in the country, there is a limited knowledge base on software development project management that would help to evaluate their practices. Such business operations were expanding due to the relative cost-saving effect and advances in networking technology. Therefore, this study would help to evaluate how the company currently practices agile project management and tries to explore the challenges thereof. The company would be using the result from this study to improve the implementation of agile software development to improve product quality and enhance customer satisfaction. This would also help other companies of similar nature in the same context.

### **1.3 Research Objectives**

The main aim of this project work is to assess the agile project management practices for software development projects in Excellerent technology solutions, in the context of its Ethiopia delivery center.

Specifically, the work has the following objectives:

- To explore the current agile project management practice in the company
- To assess the challenges of implementing agile project management
- To suggest areas of improvement in applying agile software development methodologies

## **1.4 Research Questions**

This study tries to provide answers to the following questions:

1. How do project teams practice the agile project management methodologies in Excellent Technology Solutions?
2. What are the challenges in practicing agile project management methodologies for software development projects?

## **1.5 Significance of Study**

The case study on Excellent Technology Solutions presents a valuable opportunity for the company to critically evaluate its project management approaches, particularly the application of agile methodology. By assessing current practices, the company can identify strengths to build upon and areas needing improvement. Understanding the challenges faced during the implementation of agile practices is crucial for developing strategies to overcome obstacles, thereby enhancing team efficiency and project success rates. This study not only contributes to the refinement of project management processes within Excellent Technology Solutions but also adds to the broader discourse on best practices in agile project management. Ultimately, the insights gained from this study could lead to more robust systems that support the company's objectives and drive its growth in an increasingly competitive technological landscape.

This study is believed to put forward valuable insights for developers and management teams involved in software development projects. It will highlight the challenges that need to be overcome and improvement areas so that the organizations can devise mechanisms as to how they can utilize their resources efficiently and able to implement the agile methodology effectively.

The study of agile project management in software companies is crucial for identifying the existing gaps in its implementation. By examining a specific company's approach to agile methodologies, researchers and academicians can gain valuable insights into the practical challenges and adaptations in the field. This research not only contributes to the academic discourse but also serves as a guide for software companies looking to refine their agile practices.

## 1.6 Scope of the Study

It is understood that Agile methodology can be applied to not only software projects but also non-software projects. However, this study focuses on a software development project, so the focal area is on software projects. Nonetheless, the outcome of this study can be applicable to other project types too. Moreover, from the preliminary survey, it is identified that Excellent Technology Solutions is implementing the Scrum framework. So, the study primarily focuses on this framework.

This study is conducted based on the company's Ethiopia delivery center located in Addis Ababa. Hence the findings will be based on the participants from this office, not considering teams in other locations.

## 1.7 Limitations of the Study

Despite the popularity of agile project management in software development companies, the lack of specialized training available to professionals, and the study being done on a particular company; the study is limited in its generalizability. As clearly stated, due to the nature of the projects, and the way agile software development methodologies are implemented, the study mainly presents the company's perspective thus, the findings are presented as their own understanding and perception. Therefore, these limitations shall be taken into account when referring to this study report.

## 1.8 Operational Definition of Terms

**Software** – It refers to the programs and other operating information used by a computer. It covers a wide range of applications and systems that instruct the computer on how to perform specific tasks.

**Software Development** – is the process of designing, creating, testing, and maintaining software applications and systems.

**Software project management** – is a specialized discipline within project management that focuses on planning, executing, monitoring, controlling, and closing software development projects.

**Agile software project management** - is an iterative approach to managing software development projects that focuses on continuous releases and customer feedback.

**Framework vs Methodology** — Methodology is a system of ways of doing, teaching, or studying something. It is a documented approach for performing activities in a coherent, consistent, accountable, and repeatable manner. Whereas a framework is defined as a system of rules, ideas, or beliefs that are used to plan or decide something. A framework is a structured approach to a problem that is needed to implement a model or at least, part of a model.

**Requirements** — Desired things that a product or service is supposed to do to satisfy the needs of the sponsor or the client and deliver expected business value.

**Scrum framework** — A simple framework for clearly defining roles and organizing your actionable work so that you're more effective in prioritizing work and more efficient in completing the work selected.

**Self-organizing teams** – Teams that are responsible for assigning and overseeing their work and progress. This helps not to spend extra resources to manage tasks, deadlines, etc.

**Sprint** - It is a fixed period within which development is done to produce a working, potentially deliverable product. A typical sprint might be one or two weeks in duration but could be as little as a day and as much as a month.

## **1.9 Organization of the Study**

This study is structured into five chapters; the first chapter gives a brief introduction to the subject matter of the study. It starts by presenting the background of the study and continues by providing the statement of the problem, It also highlights the objective of the study, the basic research questions, and the scope of the study. At the end of the chapter, the organization of the study is described.

The second chapter of a study often delves into the theoretical framework and empirical evidence that underpins the research topic. It typically outlines the fundamental concepts of project management, exploring various agile methodologies that prioritize flexibility and efficiency. The third chapter, on the other hand, is usually dedicated to the research methodology, detailing the systematic approach taken to gather and analyze data. This includes defining the research design,

describing the population and sample size, explaining the sampling procedures and techniques, and specifying the types of data collected. Additionally, it justifies the chosen methods, addresses the reliability and validity of the research instruments, and discusses the ethical considerations and analytical methods employed. Proper citation and referencing are crucial to lend credibility and allow readers to trace the sources.

Chapter four delves into the analysis of data, which is collected using the methods and instruments outlined in the methodology section. This analysis is pivotal as it translates raw data into meaningful insights. Following this, chapter five serves as a capstone, synthesizing the entire study into a concise summary, drawing conclusions based on the analyzed data, and offering recommendations for future research, practice, or application. The final section of the document provides the references, ensuring academic integrity and allowing readers to trace the study's scholarly foundation. Additionally, including the questionnaire in the appendices enhances the transparency and replicability of the study.

## Chapter 2 Related Literature Review

In its life cycle, a typical project management process has five process groups: Initiation, Planning, Execution, Monitoring and Controlling, and Closure. A process group is a logical grouping of project management processes to achieve specific project objectives (PMI, 2017).

### 2.1 Overview

Various authors define projects more or less similarly, albeit using different expressions. PMI defined project as (PMI, 2017):

*A project is a temporary endeavor undertaken to create a unique product, service, or result.*

Another definition is also given (Gechman, 2019)

*A project is a temporary endeavor to produce a specific product, service, upgrade or result within a defined period of time and is usually constrained by funding.*

Wysocki (2019) has also defined a project as:

*A project is a sequence of unique, complex, and connected activities that have one goal or purpose and that must be completed by a specific time, within budget, and according to specification.*

A more verbose definition is also given by Kerzner (2017) as *a project can be considered to be any series of activities and tasks that:*

- *Have a specific objective, with a focus on the creation of business value, to be completed within certain specifications*
- *Have defined start and end dates*
- *Have funding limits (if applicable)*
- *Consume human and nonhuman resources (i.e., money, people, equipment)*
- *Are multifunctional (i.e., cut across several functional lines)*

Despite all these different definitions provided by various authors, all describe consistent attributes to define a project. Generally, a project could have the following characteristics:

- Comprised of Activities: a project has a number of activities that needs to be completed in a certain order. The sequence of the activities is based on technical specification not on mere management prerogatives. These activities are unique and complex. Something is changed when the activities of a project are repeated. Moreover, these activities are interrelated, implying that implementing them should keep their sequence.
- Has objective: projects must have a single goal that contributes towards the larger organizational goal.
- Temporary endeavor: projects are finite; they have a definite beginning and end. This does not mean they have short duration, though.
- Within budget: projects have resource limits, such as a limited amount of people, money, or machines that can be dedicated to the project. These resources can be adjusted up or down by management, but they are considered fixed resources by the project manager (Wysocki, 2019).
- According to Specification: The client, or the recipient of the project's deliverables, expects a certain level of functionality and quality from the project. Although the project manager is tempted to treat these specifications as fixed, there are factors that cause them to change. Hence, need to deal with the risk of working with a moving target. This problem gets worsen in certain project types, such as software projects.

The project activities are started to achieve some stated goals and have limited period. Not only time, projects are using limited resources and these resources and budget are consumed during the life cycle of the project. After the project is finished, the unconsumed resources and budget should be released. Since each project is started for a customer, a fourth dimension in the project is also added. It is customer satisfaction. The customer must be satisfied with the goal achieved by the project (Ahmed, 2012).

In today's dynamic business environment, organizations more often initiate projects. On top of the fact that changes drive their initiation, projects do drive change in an organization. Projects are also expected to enable business value creation, and they could be a response to some actions against the business interest. Generally, projects are initiated in an organization for different reasons including:

- Create, improve, or fix products, processes, or services

- Implement or change business or technological strategies
- Meet regulatory, legal, or social requirements
- Satisfy stakeholder requests or needs

Based on the outcome or the objective of the project, projects can be specialized as construction projects, software projects, etc. In this study, the emphasis is on software projects and even though there are different forms of software-related projects, we further focus on software development projects.

## **2.2 Software Projects**

Before discussing software and software projects, we need to understand the difference between IT and software, and likewise between IT projects and software projects. IT is a field where systems comprising hardware, software, data, communication, and other related components enable a running system. However, it is common to refer to a combination of computer hardware and software systems as IT.

Software is a set of instructions, data, or programs used to operate computers and execute specific tasks. Software is a direct product of the cognitive processes of individuals engaged in innovative teamwork (Ruhe & Wohlin, 2014). A software project is a software development activity that involves creating, upgrading, or customizing software.

Information Technology projects encompass a broad range of complex activities, often requiring meticulous planning and execution. Large-scale enterprise systems, for instance, demand a comprehensive approach that includes the development of software tailored to business needs, the acquisition of compatible hardware, the establishment of robust infrastructure, and the meticulous installation and configuration of software. In contrast, a focused software development project typically involves creating a detailed design that aligns with customer requirements, followed by the actual coding in a suitable programming language. Before deployment, the software undergoes rigorous testing to verify its functionality and performance against the expected standards, ensuring it meets the quality criteria demanded by the client. Both types of projects are critical in their own right, contributing significantly to the operational efficiency and strategic goals of an organization.

On the other hand, in software maintenance projects, an existing software product is modified to remove software defects, add new functionality, port the software product to some other operating system, etc. Software development and software maintenance projects together are referred to as software projects (Ahmed, 2012). These types of projects, most commonly, include different tasks (see Figure 2.1) that have various activities to complete each task.

According to PMI (2013b), software projects may also be undertaken to satisfy service requests, and maintenance needs, or to provide operations support. These activities may occur as level-of-effort (LOE) activities; they are considered projects when they are specified as temporary endeavors to provide deliverables and outcomes.

A software project has two main activity dimensions: engineering and project management. The engineering dimension deals with building the system and focuses on issues such as how to design, test, code, and so on. The project management dimension deals with properly planning and controlling the engineering activities to meet project goals for cost, schedule, and quality (Jalote, 2016). Hence, a software project requires not only management skills, but also good software engineering skills.

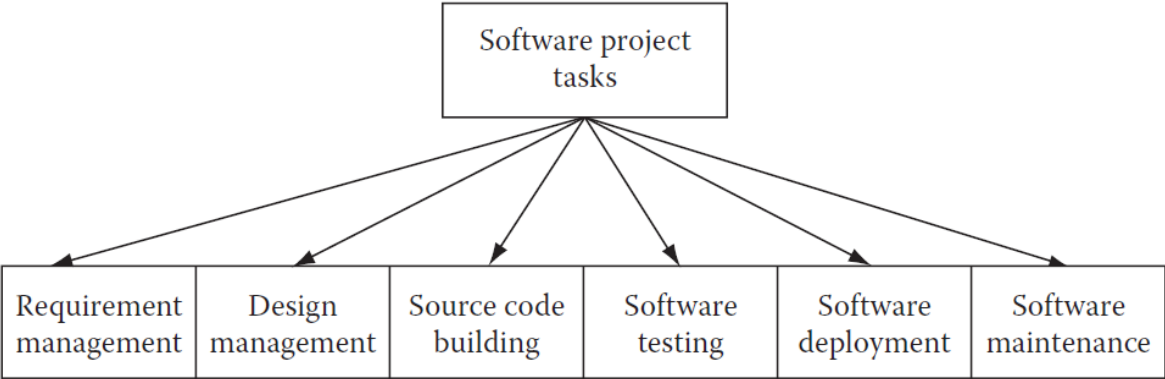


Figure 2.1 Tasks in Software Projects (Ahmed, 2012)

Software development is both human-intensive and knowledge-intensive, which makes people the most important asset in any software development endeavor. Software projects are different from other projects in a number of ways. Consequently, management of software projects cannot be done in the same way as in traditional project management and needs to be adjusted correspondingly (Ruhe & Wohlin, 2014).

## 2.3 Challenges of Software Projects

Mainly from the nature of software projects and the ever-changing nature of technology, software projects face various challenges. Today's best technology may be of no use tomorrow. The user may not be able to visualize or foresee the system usage and experience would change. Here are a few of the challenges that a project manager, who is working on a software project, would look after.

**Unclear or Undefined Requirements** - It would be difficult, if not impossible, to gather requirements from users, which further complicates the progress. The expectations and objectives become ambiguous, and the output deviates from the actual results. This will definitely affect the schedule, and it will also increase cost, and affect the client's expectations.

**Time Constraint** - Because of high competition and changing technology, it has become customary for the development team and project managers to work on tight deadline. On top of this, unrealistic deadlines from clients make it worse and affect the quality of the software.

**Changing project requirements and priorities** - One of the substantial challenges in software project management is changing requirements and priorities. The needs of the clients may change as the project progresses. And when that happens, extra time, cost, and resources need to be allocated, which further impacts the project.

**Steep Learning Curve** - It is not uncommon for the development team and project managers are using tools for the first time. Moreover, they have to use new technologies. And in the process, they have to learn to use them from scratch. As a result, it slows down the project's progress.

These and other challenges necessitate software projects to be managed carefully for their successful completion. Apart from the traditional one, there are different project management approaches that are proposed to handle software projects.

## 2.4 Project Management

Project management is the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements. Project management is accomplished through the appropriate application and integration of the project management processes identified for the project. Project management enables organizations to execute projects effectively and efficiently

(PMI, 2017). Like any other management profession, it involves coordinating and administering the various activities of people to achieve objectives.

Project management is a profession that is based on having a team to initiate, plan, execute, monitor, control, and close a set of tasks and activities in order to successfully achieve a set of goals in accordance with predefined criteria, within predefined schedules, and predetermined budgets (Radaideh, 2024). As each discipline has unique aspects that differentiate from other disciplines, we need to adapt the general principles of project management to account for those special aspects. This report focuses on software projects, hence the focus towards exploring software project management.

Software project management is a discipline of project management that is specific to software engineering and software projects. According to Villafiorita (2014), it is the integration of project management techniques into software development. Attributed to the unique nature of software projects, software project management resulted in distinct challenges. Managing projects involves balancing the competing project constraints, which include, but are not limited to cost, schedule, scope, quality, resources, and risk. On top of these, there are technological factors that can place constraints on software projects and software products. These constraints include the state of hardware and software technology, software development tools, hardware and software platforms, operating systems, software architecture, etc.

Other factors that can place constraints on software projects include but are not limited to requirements for system safety, security compliance, reliability, availability, scalability, performance, testability, information assurance, localization, maintainability, supportability, regulations, customers' policies, infrastructure support, team member availability and skills, software development environment and methods, and organizational maturity and capability (PMI, 2013b).

To this end, software project management can become challenging to carryout using the ordinary project management processes, tools and techniques. Therefore, it requires adoption to suit its unique nature. The distinct nature of software project management is hugely attributed to the challenging factors of software projects, which are strongly related to the nature of the discipline of software engineering.

As clearly stated in PMBOK Guide, the knowledge described should be used carefully and wisely. The organization and/or project team is responsible for determining what is appropriate for any given project or situation. A software project shall be handled differently based on the surrounding environment. Organization capability, team members, skill, experience and knowledge of project manager, etc. will affect how to run the project. The complexity and uncertainty of requirements as well as the need to deliver business value play crucial role to determine defining the project management approaches to be followed. The less complex and most certain kind of projects are those with very limited (if any) scope change that can be comfortably managed using traditional approaches. However, this is also less likely to occur in today's fast-paced, continuously changing business world. Change is inevitable, so, project managers must open their minds to the basic principles on which project management is based so as to accommodate change. 'Agile' practices are being aggressively used to handle these difficulties. These different project management approaches recognize that major differences exist among projects and that those differences require different management approaches if the project is to be managed and successfully completed.

### **2.4.1 Traditional Project Management**

Traditional project management denotes a highly structured project management framework where the project's requirements are defined upfront. This requires a clear understanding of exactly what is to be done. What needs to be done is specified in terms of the scope of the project, and the scope of a software project is most often described as the requirements of the project. In traditional project management, the project's scope is expected to be complete and clear. This is possible through the analysis of the requirements breakdown structure (RBS), an artifact that characterizes a project. This in turn helps to correctly choose which model to use for the project.

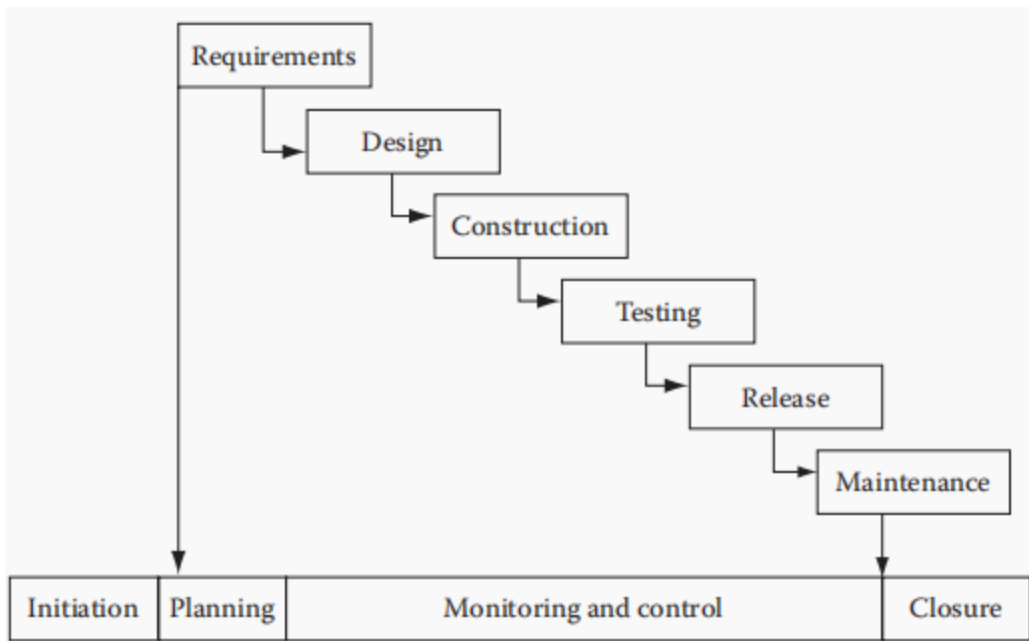
According to Villafiorita (2014), the underlying assumption is that efficiency can be achieved with a top-down and planned organization of work. Only with this approach, according to their advocates, it is possible to eliminate misunderstandings, errors, and rework, while at the same time promoting an efficient use of resources.

The early practice of software development projects involved writing code first without putting much effort into planning and designing (Cho, 2010). The approach of fixing bugs later can indeed be viable for small-scale projects where changes are less costly and easier to manage. However,

as projects grow in size and complexity, this method becomes less effective. The decrease in efficiency and effectiveness is often due to the reliance on traditional project management methodologies that are not well-suited for the dynamic nature of software development. These methodologies, borrowed from engineering disciplines, are plan-driven and require comprehensive documentation and understanding of all requirements before design and development begin. This works well for stable, well-understood environments but can lead to challenges when requirements change or are not fully known at the outset.

One well-known method of traditional project management, which is hugely implemented in the software industry, is the Waterfall model. The waterfall development activities proceed sequentially, from conceptualization of the problem to delivery of the final product. Each activity of the process takes as input the outputs produced by the previous activity of the chain, uses them to produce artifacts closer to the final product, and passes the outputs to the next activity in the chain (Villaforita, 2014).

In this model, the software engineering processes are executed linearly (as depicted in Figure 2.2), and thus, in a software project, each of the software engineering processes occurs just once. The phases in this model are highly sequential and once one phase is complete the development moves to the next phase.



*Figure 2.2 Project management in the waterfall model (Ahmed, 2012)*

The waterfall model is best suited for large software development projects for government, military, and other industries that have relatively stable requirements and require disciplined planning upfront. Again, the waterfall model is best suited for projects where well-developed software requirement specifications (SRSs) exist (Ahmed, 2012).

This rigid nature of the waterfall model is both an advantage and a weakness at the same time. In projects where the requirements are very clear or in which a controlled development environment is very beneficial, the waterfall process shows its advantages. However, in many other cases, the rigidity of the waterfall can hinder, rather than speed up, development (Villafiorita, 2014).

Over the past decades, traditional waterfall-style software development methods have been widely used for large-scale projects in the software industry and in the government sector due to their straightforward, methodical, and structured nature as well as their capability to provide predictability, stability, and high assurance. However, traditional software development models have a number of key shortcomings, including slow adaptation to constantly changing business requirements, and a tendency to be over budget and behind schedule with fewer features and functions than specified (Cho, 2010). After evaluating more than 10,000 software projects, the CHAOS report (The Standish Group International, 2015) has shown for all projects that agile

projects have almost four times the success rate as waterfall projects, and waterfall projects have three times the failure rate as agile projects.

It is understood that change is inevitable and software projects are highly uncertain. Any model has to embrace this imminent characteristics and deal it without affecting the project time, cost, and quality. The traditional methods, despite all their advantages, do not scale well and fail to respond to changes. In today's highly changing world, with frequent changes in requirements or projects having unclear or incomplete requirements other adaptive methodologies are introduced.

## **2.4.2 Agile Project Management**

As evident in different discussions about software projects, there are peculiar differences from other kinds of projects. On top of the ever-changing technology, and their nature of being knowledge-intensive, requirements are best gathered in many iterations. It has been observed that requirements change during different phases of the software development life cycle (SDLC) and this change plays a vital role in success or failure of any project (Bano et al., 2012).

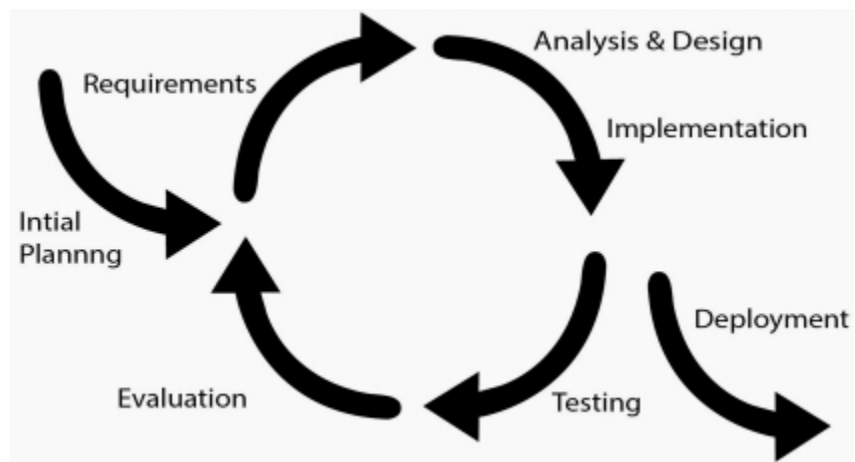
Agility is defined as the ability to act proactively in a dynamic, arbitrary, and constantly changing environment, and organizational agility is an organization's ability to be adaptable to changing conditions without being forced to change (Salameh, 2014).

Also, according to Layton et al (2020), agile is a descriptor of a mindset approach to project management that focuses on early delivery of business value, continuous improvement of the product being created and the processes used to create the product, scope flexibility, team input, and delivering well-tested products that reflect customer needs. Though, business agility is popular in software product development, agile values, principles, and techniques apply in a multitude of industries and applications – not just software.

Agile project management is about managing the impact of complexity and uncertainty on a project. Therefore, unlike the linear sequence of well-defined activities of traditional project management, agile project management is characterized by short cycles of iterative and incremental delivery of product features and continuous integration of code changes (Ruhe & Wohlin, 2014).

Starting from the mid-1990s to early 2000's Internet and related technologies were changing rapidly and the people working in the booming dot-com industry were under constant pressure to

be the first to market with fast-changing technologies. Development teams worked day and night, struggling to deliver new software releases before competitors made their companies obsolete (Layton et al., 2020). Agile software development represents a new approach to planning and managing software projects, arguably handling requirement changes. It puts less emphasis on up-front plans and strict control and relies more on informal collaboration, coordination, and learning.



*Figure 2.3 Incremental and Iterative nature of Agile methods (Stanley & Gross, 2020)*

Agile software development promotes adaptive planning, rapid response to changes through a project life cycle, and continuous improvement and early delivery (Alhroub & Jaaron, 2019).

In 2001, a group of 17 independent software and project experts got together to talk about what their successful projects had in common. This group created the Manifesto for Agile Software Development (commonly referred to as the Agile Manifesto), a statement of values for successful software development (Layton et al., 2020; The Agile Manifesto Authors, 2001):

## **Manifesto for Agile Software Development**

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

**Individuals and interactions** over processes and tools

**Working software** over comprehensive documentation

**Customer collaboration** over contract negotiation

**Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

*Figure 2.4 The Agile Manifesto (The Agile Manifesto Authors, 2001)*

Agile methodologies, which focus on iterative development, customer collaboration, and flexibility to change, are often more effective in managing the complexities of modern software projects. They allow teams to adapt to changes quickly and efficiently, ensuring a more responsive and effective development process. Agile software development consists of methods and practices based on values and principles stated in the Agile Manifesto.

## **2.5 Agile Project Management Frameworks**

Agile is an umbrella term for a vast variety of frameworks and techniques that share the principles and values described above. Each of them has its own areas of use and distinctive features. A few of these methodologies are described below.

### **2.5.1 Scrum**

The term “scrum” comes from a rugby term that consists of teammates interlocking their arms and pushing forward into opponents. According to Schwaber & Sutherland (2020), Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems. Scrum is a collaborative agile development framework, describes a group of meetings, set of tools, and roles that supports teams work together to manage product development (Hema et al., 2020).

While Scrum was introduced far before the Agile Manifesto, it relies on Agile principles and is consistent with the values stated in the document.

Scrum is a simple framework for clearly defining roles and organizing actionable items so that a team is more effective in prioritizing work and more efficient in completing the work selected. Though it is a framework and expected to be less prescriptive as it doesn't describe what to do in every situation. Scrum simply offers a framework (see Figure 2.5) and set of practices that keep everything visible. This allows Scrum practitioners to know exactly what's going on and to make on-the-spot adjustments to keep the project moving toward the desired goals (Schwaber, 2004).

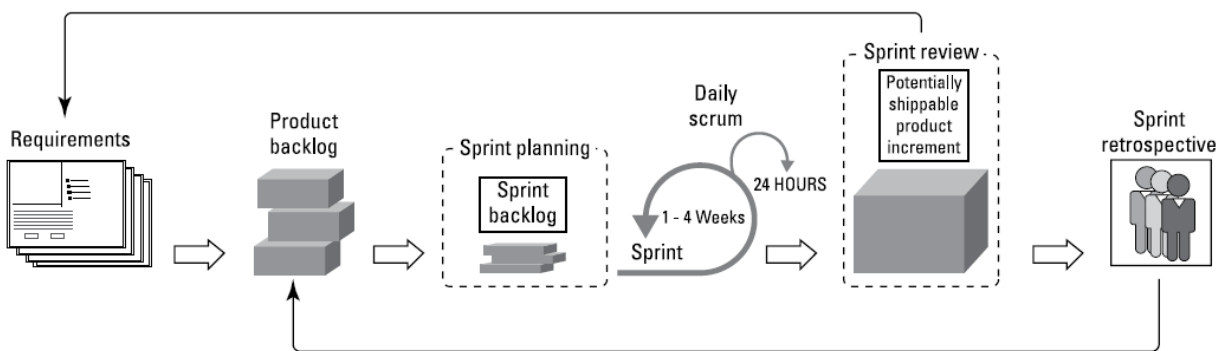


Figure 2.5 The Scrum Approach (Layton et al., 2020)

Scrum consists of three main processes: Roles, Artifacts, and Events. Scrum is often used to handle complex problems in the software development process and has been proven to increase productivity and reduce software development costs. Scrum delivers the software in iterations, which means the software is delivered partially so any changes or problems can be resolved before all software parts are ready to get delivered (Wonohardjo et al., 2019).

### Scrum Roles, Artifacts, and Events

The self-organizing and cross-functional scrum team consists of three roles. The team knows how best to accomplish the work and has all the required competencies. The three roles of a scrum team are product owner, scrum master, and development team.

The product owner is responsible for managing the product backlog. The Development Team consists of professionals who deliver a potentially releasable increment of usable product at the end of each Sprint. As a servant-leader of the scrum team, the scrum master is responsible for ensuring the Scrum is understood and well run.

Artifacts defined by Scrum are specifically designed to maximize transparency of key information so that everybody has the same understanding of the artifact. Scrum has three artifacts.

1. Product Backlog – This is an ordered list of everything that might be needed in the product and is the single source of requirements for any changes to be made to the product.
2. Sprint Backlog – The set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and realizing the Sprint Goal.
3. Product Increment - The Increment is the sum of all the Product Backlog items completed during a Sprint and the value of the increments of all previous Sprints.

There are about five time-boxed events with a maximum duration. These events create regularity and avoid meetings that are not defined in Scrum (Schwaber & Sutherland, 2020).

1. The Sprint – The heart of Scrum is a Sprint, a time-box of one month or less during which a “Done”, useable, and potentially releasable product increment is created. Sprints best have consistent durations throughout a development effort.
2. Sprint Planning – This is used to plan the work to be performed in the Sprint. This plan is developed by the collaborative effort of the scrum team.
3. Daily Scrum – This is a 15-minute time-boxed event to update complete activities and plan for the next 24 hours. This event is also known as daily standup.
4. Sprint Review – A Sprint Review is held at the end of the Sprint to inspect the Increment and adapt the Product Backlog if needed.
5. Sprint Retrospective – The Sprint Retrospective is an opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint.

### **2.5.2 Extreme Programming (XP)**

Extreme programming (XP) is an agile project management methodology that targets speed and simplicity with short development cycles and less documentation. In XP, the driver of the software project is the customer and acknowledges the fact that everything in software changes. It tries to resolve the problem of the inability to cope with this change by reducing the cost of changing software (Beck, 2000). In XP, simplicity is the driving factor that applies to the software development practices including communications with the customer. Beck (2000) has described

four values – Communication, Simplicity, Feedback, and Courage – of XP that would be used as criteria for a successful solution.

Communication is seen as the most important ingredient for effective cooperation within a team, which is enhanced through respect: an attitude to accept different viewpoints, constructively evaluate every possibility, and care about the contributions of each member. Simplicity emphasizes that a team should try to be efficient: achieve the maximum business value with the minimum effort. Feedback is gathered on everything the team does including discussions on alternative ideas, evaluating the performance of a system through a test, confronting customers with intermediate versions of a solution, and so on. Finally, courage is necessary to be ready “to speak truths, pleasant or unpleasant,” to “discard failing solutions,” and to “seek real, concrete answers” within the team (Janes & Succi, 2014). These values are interrelated. Simplicity minimizes the need for communication, reducing the feedback required. Also, the things that necessitate courage will be minimal if communicate frequently and effectively.

These are the 12 basic practices used by XP throughout the process (Al-Zewairi et al., 2017; Beck, 2000):

- **Planning Game:** the planning starts at the beginning of each iteration where the stakeholders of the project meet to define, estimate, and prioritize the “User Stories” (i.e. requirements) for the next release.
- **Small Releases:** there are two types of version releases: initial version and working version. The initial version is produced after a few iterations and it does not implement all features, but only essential ones. While, a working version is produced after a few weeks and contains most of the features.
- **Metaphor:** it is used in the modelling of the software system and is constructed by all the stakeholders.
- **Simple Design:** It is the base of the XP methodology and applies to the requirements gathering, system design, coding, and communications with the customer.
- **Testing:** In XP methodology, testing is considered one of the major activities to ensure high quality product in addition to high customer satisfaction. Testing begins before the coding phase, where developers are required to prepare the test functions prior to writing the code itself. While, customers are required to prepare the functional test scenarios for each iteration.

- **Refactoring:** Programmers restructure the system without changing its behavior to remove duplication, improve communication, simplify, or add flexibility. It means that any changes made on the system must uphold the simplicity feature.
- **Pair Programming:** Coding of the software system is carried out in a group of two developers.
- **Continuous Integration:** New parts of the software system are integrated as soon as they pass both the unit and functional test cases.
- **Collective Ownership:** The ownership of the code produced belongs to all the developers.
- **On-Site Customer:** Someone from the customer side must work with the development team at all times.
- **40-Hour Weeks:** The maximum working hours per week for developers must not exceed 40 hours, which implies that the requirements must be revised to adhere to this rule.
- **Open Workspace:** All work, including coding and development must take place in a common environment.

XP is a software development methodology broken down into work sprints like other agile methods. Agile frameworks follow an iterative process – you complete and review the framework after every sprint, refine it for maximum efficiency, and adjust to changing requirements.

Extreme programming is so extreme that involves a rigorous and highly structured process. XP is highly disciplined, using frequent code reviews and unit testing to make changes quickly. It is also highly creative and collaborative, prioritizing teamwork during all development stages.

### 2.5.3 Feature-Driven Development

Feature-Driven Development (FDD) is indeed a highly structured yet flexible approach to software development. It emphasizes frequent, timely delivery of products, and is particularly well-suited for larger teams working on a project using object-oriented programming. FDD blends a number of industry-recognized best practices into a cohesive whole. These practices include domain object modeling, developing by feature, feature teams, inspections, configuration management, and regular builds. By focusing on features, which are actually client-valued pieces of work, FDD facilitates client-centric software development and helps teams deliver practical, usable software consistently and efficiently.

FDD starts with the creation of a domain object model in collaboration with Domain Experts using information from the modeling activity and from any other requirements activities that have taken place, the developers go on to create a features list. A feature is a small, client-valued function expressed in the form of “Action-result-object”. As shown in Figure 2.6, FDD consists of five processes (Palmer & Felsing, 2002).

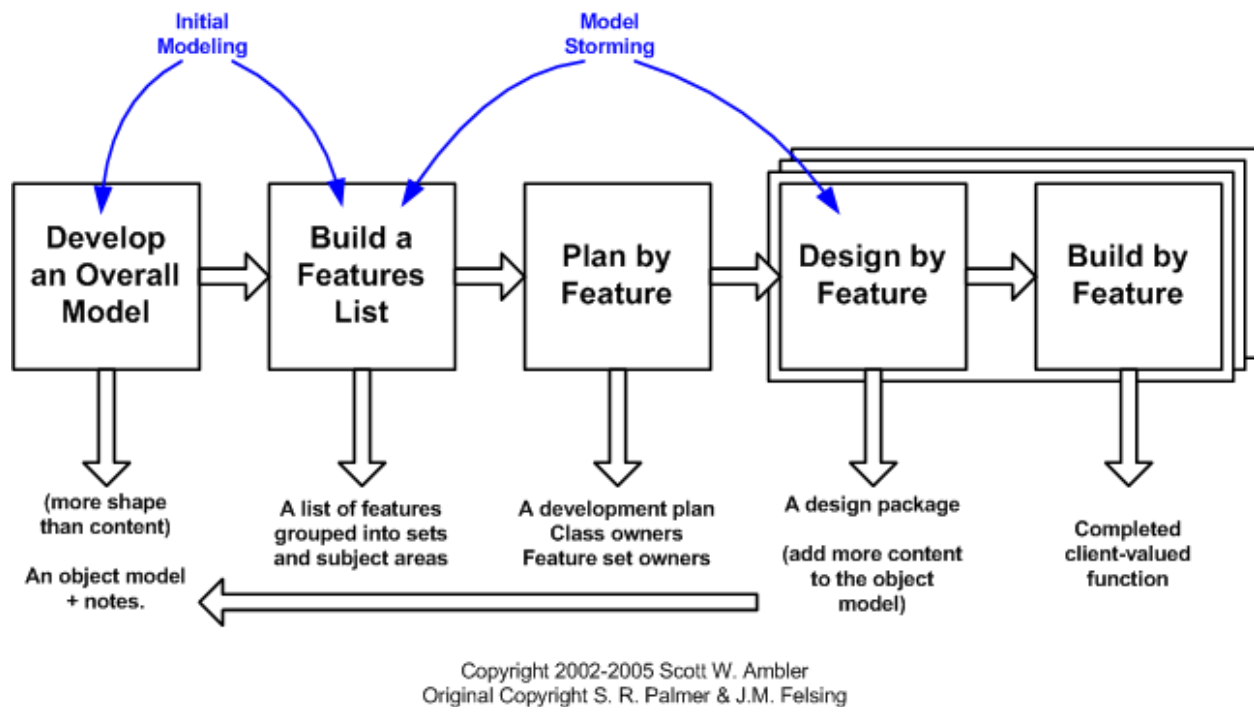


Figure 2.6 The FDD Lifecycle (AgileModeling.com, 2022)

During the first process, domain and development team members under the guidance of an experienced modeler develop overall model. Based on the knowledge gathered during the first process, the team then develop comprehensive list of features. The third process, plan by feature, is to sequence the feature sets or major feature sets (depending on the size of the system) into a high-level plan and assign them to programmers. The fourth and fifth processes are iterative set of development tasks where the lead programmer will select set of features to be completed within a few days.

## 2.5.4 Lean Software Development

Lean software development is the application of lean principles and thinking tools to software development. The pioneers of lean software development, Mary and Tom Poppendieck (2010) propose seven principles to characterize lean software development.

1. **Eliminate waste.** Waste is anything that does not add value to a product, value as perceived by the customer. Lean development aims to avoid whatever gets in the way of rapidly satisfying a customer need.
2. **Amplify learning (Create Knowledge).** Development is an exercise in discovery, while production is an exercise in reducing variation, and for this reason, a lean approach to development results in practices that are quite different from lean production practices.
3. **Decide as late as possible (Defer Commitment).** Development practices that provide for late decision-making are effective in domains that involve uncertainty because they provide an options-based approach. Delay the irreversible decisions that need to be made in the course of software development to the last responsible moment where you get the most information.
4. **Deliver as fast as possible.** Speed assures that customers get what they need now, not what they needed yesterday. It also allows them to delay making up their minds about what they really want until they know more. The discovery cycle of design, implement, feedback, and improve need to be shorter and this is important to eliminate waste.
5. **Empower the team (Respect People).** Top-notch execution lies in getting the details right, and no one understands the details better than the people who actually do the work. Involving developers in the details of technical decisions is fundamental to achieving excellence.
6. **Build integrity in (Build Quality in).** The goal of software development shall be to build quality into the code from the start, not to test later. In addition to meeting the customer perception over time, software needs to maintain its usefulness overtime and adapt to the future gracefully.
7. **See the whole (Optimize the whole).** Integrity in complex systems requires a deep expertise in many diverse areas. One of the most intractable problems with product development is that experts in any area (e.g., database or GUI) have a tendency to maximize the performance of the part of the product representing their own specialty rather than focusing on overall system performance.

### **2.5.5 Kanban**

Kanban methodology goes back to the late 1940s when Taiichi Ohno developed the system for Toyota in Japan based on the same model that supermarkets were using in stocking their shelves (Zayat & Senvar, 2020). Kanban is a visual signaling system that transformed manufacturing management at Toyota. This approach facilitated a more intuitive and efficient workflow by providing real-time communication and immediate responses to production needs. Its simplicity and visual nature allowed workers to easily follow the work process, understand their tasks, and quickly spot any bottlenecks or issues. This highlights the effectiveness of visual tools in improving communication and efficiency within complex systems. Today, Kanban's adaptability extends beyond automotive manufacturing into various industries, demonstrating its versatility and effectiveness in project management and team collaboration. It manages the entire value chain from supplier to consumer.

Kanban is a highly effective agile framework used in various industries to improve workflow and efficiency. The Kanban board, with its swim lanes (see Figure 2.7), offers a visual representation of the work process, allowing teams to see the status of each task at a glance. The 'To Do' column lists tasks that need attention, the 'In Progress' column shows what's currently being tackled, and the 'Done' column provides a satisfying visual cue of completed work. This system not only helps in managing work but also in identifying bottlenecks in the process, enabling teams to adjust and streamline their workflow for better productivity. Additionally, Kanban cards play a crucial role in this system; they represent individual tasks and carry important details such as deadlines, requirements, and the person responsible for the task. Together, these elements form a dynamic tool that promotes transparency, accountability, and continuous improvement within teams.

A Kanban card represents the item or task in progress. It helps communicate the workflow and progress to the team. It also determines vital information like the timeframe of the project, any upcoming deadlines, and the status of work (McCarthy, 2020).

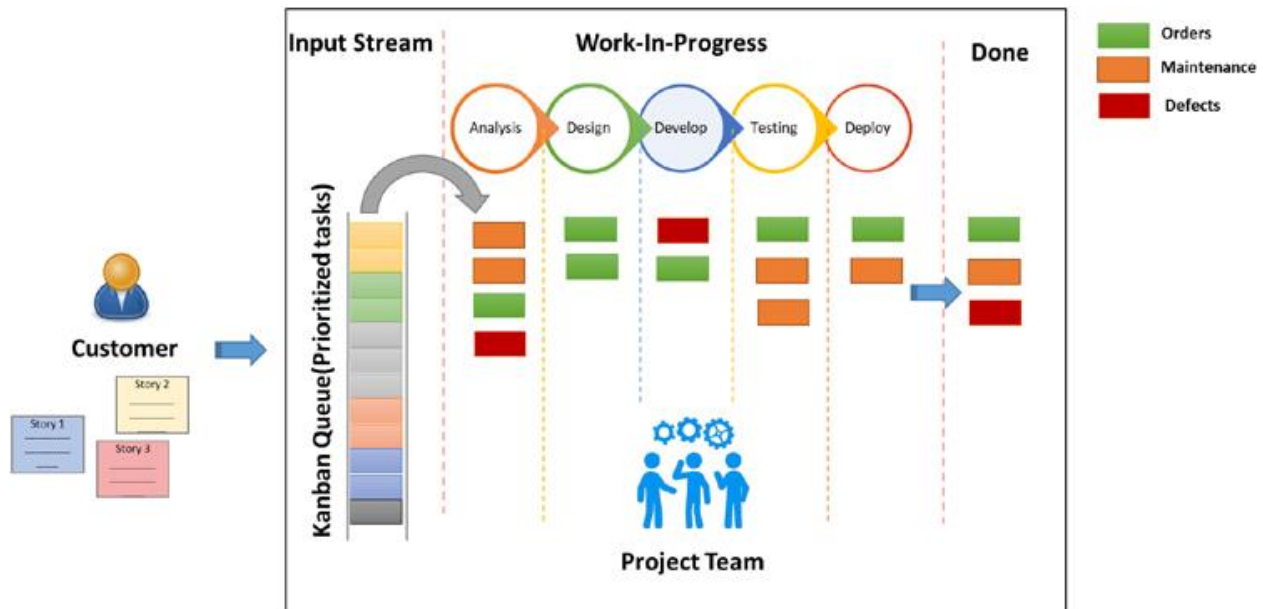


Figure 2.7 Kanban System (Zayat & Senvar, 2020)

Kanban offers a visual representation of the production process to deliver high-quality results by showing the workflow to determine any bottlenecks. The framework has different practices that forms its basis (Zayat & Senvar, 2020):

- **Visualize Work:** helps to provide teams shared views of the process along with all the workflow issues. A typical practice is to apply a board that visualizes specific work phases.
- **Limit Work in Progress (WIP):** This step is done through minimizing the size of work to be done before the task is delivered; this minimizes the time needed for an item to travel across the board of Kanban.
- **Focus on the Flow:** Kanban system can be optimized by analysing the flow of work through the processes. Some basic principles can be followed to limit the work in process and to avoid any bottlenecks in the system.
- **Continuous Improvement:** Great systems always have a mechanism of reaching a better state of performance through time. This mechanism is processed in Kanban through eliminating problems within the system with an objective to reach a shorter lead time, a better quality, and a more coherent flow.

## **2.6 Agile Project Management Issues and Challenges**

Different literatures discussed the merits and challenges of adopting agile project management. Though it enhances transparency of project tasks and increase accountability, it poses various challenges that range from ambiguous requirements to developer's lack of motivation.

Among others, the main challenges discussed in Hajjdiab & Taleb (2011) include:

1. Requirements: The product owners may not be well versed with business knowledge and the product backlog may contain user stories that are in different levels of abstraction and inconsistent with different versions.
2. Scrum implementation: Companies faced many challenges during the implementation phase. This is mainly due to fact that the project managers and many members of the agile team had limited knowledge of agile method.
3. Organization learning: Originally, the sprint review meeting was designed to support organization learning. Due to the fact that each Scrum master had his own view of what is Scrum, the learning has become nonexistent.
4. Other challenges related to key members of the agile team
  - The need for developers to be a “master of all trades”
  - Increased reliance on social skills
  - A lack of business knowledge among developers
  - Lack of developer motivation to use agile methods

## **2.7 Empirical Review**

According to the study conducted by Regassa, Bass, and Midekso (2017), most of the software companies in Ethiopia have used agile methods on projects. While working on their projects (primarily government projects), they have identified bureaucracy in government organizations and the tender and contract nature as major barriers to companies use of agile methods and success of projects. They also mention the lack of user involvement, because of lack of interest in the project, being unreachable, or users may believe the project is not their responsibility, is a common problem. It is also mentioned that these software companies working on government projects faced lack of knowledge or difficulty describing software requirements and the need for documentation.

Considering the agile team size and related factors, a study by Lalsing has identified that the larger the deviation the larger the team size. This deviation is attributed to team communication and collaboration, trust, team interaction, code integration, defect rate, etc. Management views such projects as difficult to manage the team using the agile approach and there were many instances where team members focused mainly on their deliverables rather than team output and forgot to pass information to everyone (Lalsing, 2012). Meanwhile, according to the Standish report complex projects tend to be failed or challenged. However, from all size projects 39% run with the agile method are successful, compared to 11% of the waterfall method (The Standish Group International, 2015).

Another study conducted in the Sri Lankan context by Safwan et al has stated that about 68% of their participants have adopted and been involved with Agile methodologies in their developing process. Among these practitioners, the study has found that the major reason for adopting this approach is to keep up with the changing environment (Safwan et al., 2013). Yet another study on the Sri Lankan context by Padumadasa and Peiris determined that 89% of software houses use agile methodologies (Padumadasa & Peiris, 2011).

A survey conducted by Papatheocharous and Andreou (2013) collected data from 377 respondents, and about 72% of them claimed their organization is following agile methodologies. Those using the agile paradigm have described it as the methodology that has helped the organization accelerate the time to market, enhancing the ability to manage changing requirements/priorities and contributing to software maintainability and extensibility.

A study by Iqbal, Omar, and Yasin (2019) analyzed the effect of agile teams on productivity. In this study, participants stated that the most widely used frameworks are Extreme Programming (XP), Scrum, and Kanban. Moreover, the study result shows that agile teams increase their productivity through the use of use of agile methodologies. A related survey of 15 software development companies has shown that Scrum framework is the most popular agile implementation (Matharu et al., 2015).

## **Chapter 3 Research Methodology**

This chapter outlines the research design and describes how research data is collected. It also mentions the way collected data is analyzed to get an insight. The section about participant samples and ethical considerations in data collection and analysis is also included here.

### **3.1 Research Design**

This undertaking is a descriptive type of study that attempts to assess the current practices of software project management methodologies within the company. As a software development company, there are various software projects currently being worked on. This assessment is then made based on data collected from employees of the company, who will be participants in the study.

By engaging employees across various roles, we will gain a comprehensive view of the existing practices and how they can be improved. This approach not only ensures a diverse range of perspectives but also fosters a sense of inclusion and value among the staff, as they contribute to the company's growth and efficiency. The data collected will serve as a valuable asset in identifying strengths, pinpointing areas for improvement, and ultimately, driving the company towards more successful project outcomes.

### **3.2 Research Approach**

Utilizing close-ended questions in a questionnaire is a strategic approach to gather quantifiable data, allowing for a more structured and statistical analysis of the responses. The subsequent processing of the response data through a spreadsheet application helps efficiently organize the information into tables and charts, making it easier to identify patterns, trends, and correlations. By converting raw data into visual formats, it was easier to communicate findings more effectively, facilitating a better understanding of the underlying results.

### **3.3 Target Population and Sampling**

The initial list of technical employees highlighted the need for a more comprehensive understanding of the workforce, leading to the deployment of a census survey. To encourage participation, a series of reminder emails were dispatched, successfully prompting approximately 50% of the employees to complete the questionnaire. This response rate, while indicative of a

significant engagement, also underscores the necessity for further strategies to increase participation and ensure a more representative data collection in future surveys.

### **3.4 Data Collection Techniques**

In the realm of research, utilizing a self-administered questionnaire distributed via Google Forms is a strategic choice for gathering data from employees across various projects. This method not only streamlines the process by allowing participants to complete the questionnaire at their convenience but also enhances the efficiency of data collection and analysis. By targeting technical team members, the study ensures that the insights gained are grounded in the practical experience and technical expertise of those directly involved in the company's projects. Such an approach can yield valuable data that reflects the nuanced realities of the company's operations and employee experiences.

Agile project management methodologies, such as Scrum, Kanban, and Lean, have become increasingly popular in various industries due to their flexibility, adaptability, and focus on continuous improvement. A questionnaire assessing the implementation of these methodologies within a company can provide valuable insights into the organization's processes and the team's proficiency. It can reveal areas where the company excels in agility and pinpoint opportunities for further training or process refinement. Understanding the level of experience team members have with agile practices is crucial for fostering a culture of collaboration and efficiency, ultimately leading to successful project outcomes. Such assessments are instrumental in guiding companies on their journey towards achieving a more responsive and dynamic project management environment. The questionnaire is designed to address the background of the participants including their specialization, which indicates whether they are well equipped to adapt to the flexibility required, and quickly understand the complex nature of software development. Queries related to the experiences of participants in agile project management and their practice of agile project management in Excellerent is also included.

### **3.5 Data Analysis**

The process of data analysis is a critical step in research that involves transforming raw data into meaningful insights. Utilizing appropriate tools allows applying various statistical techniques to understand trends, patterns, and relationships within the data. Most of the responses are qualitative

data, and it was needed to categorize responses to analyze them effectively. Coding responses from questionnaires allows for efficient categorization and analysis, which can then be visually represented through tables and graphs. These visual representations make it easier to communicate complex data in a more understandable and accessible way, facilitating informed decision-making based on empirical evidence.

### **3.6 Reliability and Validity**

The questions were designed based on the literature so as to enable answering the research questions and achieving the research objectives. These questions were included with achieving the study objectives in mind, implying the reliability of the instrument.

The development of the survey instrument passes through a process that involves several stages to ensure its validity. Initially, subject matter experts evaluate the survey to confirm that it accurately measures the intended variables. Their feedback is crucial and is integrated into the survey design to refine the questions. Subsequently, a pilot test is conducted with a select group of participants. This step was essential to evaluate the clarity and comprehension of the questions among the intended audience. It has served as a final check before the survey is distributed on a larger scale, ensuring that the data collected will be meaningful and reflective of the participants' true perspectives and experiences.

### **3.7 Ethical Considerations**

Ethical considerations are paramount in any research involving human participants. The success of a study often hinges on the willingness of participants to share honest and accurate information, which is why the principles of respect, dignity, and confidentiality are so crucial. By ensuring anonymity and obtaining informed consent, researchers can foster a trusting environment where participants feel safe and valued. This not only upholds the integrity of the research but also contributes to the richness and reliability of the data collected. Maintaining these ethical standards is not just a matter of protocol, but a commitment to the respectful and humane treatment of all individuals involved in the research process.

# Chapter 4 Data Analysis and Presentation

## 4.1 Overview

The process of data collection and analysis is a critical step in any research study. After distributing the questionnaire via Google Forms, the collected responses offer a rich dataset that requires careful examination. Utilizing a specialized software package for data analysis helps to ensure that the findings align with the study's objectives, allowing for a robust interpretation of the data. Furthermore, incorporating infographic elements into the presentation of results not only enhances the visual appeal but also aids in the comprehension of complex data, making the insights accessible to a broader audience. This chapter, therefore, serves as a pivotal point where raw data is transformed into meaningful patterns and trends, contributing to the overall narrative of the study.

## 4.2 Participants' Demographic Profile

The participants of the study are technical teams of different projects in the company. The general information about the participants in terms of sex, age, educational background, and role, is given below.

*Table 4.1 Demographic Profile of Participants*

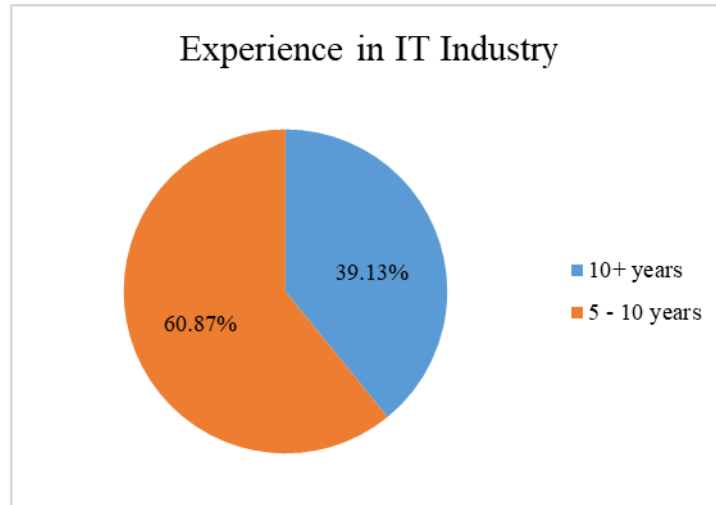
Description	Value	Count	Percentage
<b>Sex</b>	Female	6	13.04%
	Male	40	86.96%
<b>Age Group</b>	21 – 30	10	21.74%
	31 – 40	30	65.22%
	41 – 50	6	13.04%
<b>Educational Level</b>	Bachelor's Degree	24	52.17%
	Master's Degree	22	47.83%

<b>Role</b>	Software Engineer	32	69.57%
	Quality Assurance	8	17.39%
	Scrum Master	4	8.70%
	Other	2	4.35%

Among the participants, it is clear that male employees dominate it, which aligns with the reported percentage of women in tech jobs (Richter, 2021). Looking at tech jobs women take up fewer than 1 in 4 technical roles at each of the companies reporting. Therefore, the data in the survey implies the lack of female employees in the company; nonetheless, it is a global problem.

According to the collected data, most of the employees are aged between 31 and 40 years old. A balanced mix of graduates and experienced engineers comprise the teams in the company. The participants of the study have at least five years of experience in the information technology (IT) industry (see Figure 4.1). This indicates that the majority of the employees (about 86.96%) are experienced and belong to the productive age group of 21 – 40. This greatly helps to implement a project successfully.

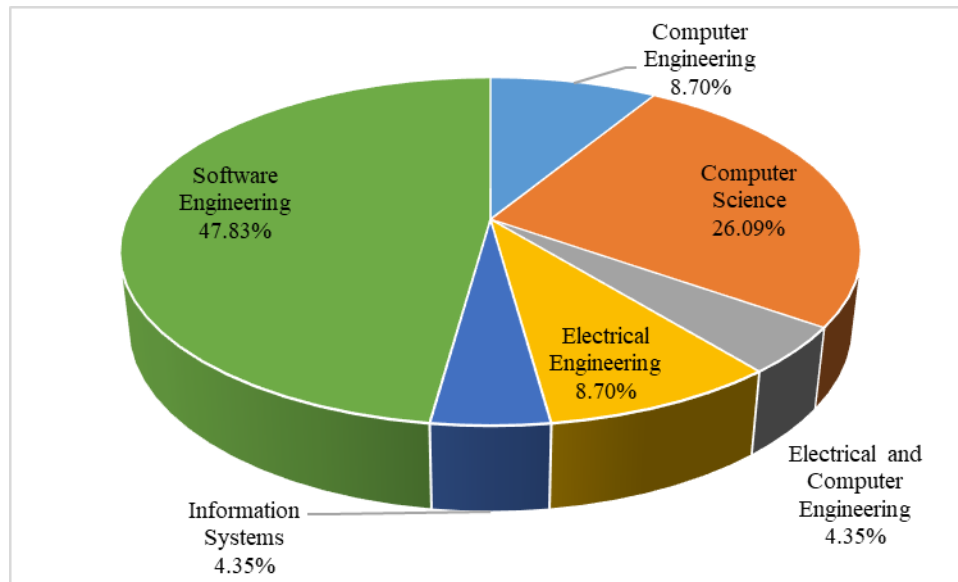
The data presents a promising scenario for the company, showcasing a workforce that is not only experienced but also falls within a highly productive age range. With most employees having a substantial tenure of at least five years in the IT industry (see Figure 4.1), the company benefits from a depth of knowledge and expertise that can drive innovation and efficiency. This blend of seasoned professionals and fresh graduates creates a dynamic environment conducive to learning and professional growth, essential for successfully executing complex projects. The high percentage of experienced employees also suggests a stable workforce, which can be advantageous for long-term planning and continuity in the company's strategic endeavors. Overall, the demographic and professional composition of the teams positions the company favorably for achieving its objectives and sustaining competitive advantage in the industry.



*Figure 4.1 Participants' experience in the IT industry*

The educational background of the staff, as indicated by the data (see Table 4.1), shows a significant level of academic achievement. With over half (52.17%) holding a bachelor's degree and nearly the other half (47.83%) possessing graduate degrees, it suggests a workforce that is well-equipped with the necessary knowledge and skills. This diverse educational composition is beneficial in understanding and navigating the intricate aspects of the sector, which often requires a multidisciplinary approach to problem-solving and innovation. Such a high level of education among the staff is likely to contribute positively to the organization's ability to meet its objectives and adapt to the evolving demands of the industry.

The composition of the staff within a company can significantly influence its operational efficiency, especially in the technology sector. With a majority of the workforce being software engineers and computer science professionals (see Figure 4.2), which comprises 73.91% of the employees, there is a robust foundation for innovation and development. This level of education along with the technical fields of specialization helps the workforce to understand the complex nature of software development projects. It also helps to quickly understand business requirements, when clients bring them, with limited or no prior business domain knowledge.



*Figure 4.2 Participants' Field of Specialization*

This specialized skill set is crucial for staying competitive in the fast-paced world of software development. Moreover, the inclusion of electrical and computer engineers complements the team, ensuring a comprehensive approach to product development that encompasses both software and hardware solutions. Such a diverse yet specialized team is well-equipped to tackle complex projects, streamline processes, and drive technological advancements.

### **4.3 Experience of Team Members**

The data presented (see Table 4.2) indicates a robust level of experience within the company's workforce. With a significant portion of employees possessing 5 to 10 years of experience, the company benefits from a mature and knowledgeable team. The fact that over a third of these individuals have remained with the company for three years or more suggests a stable and committed workforce. Additionally, the presence of employees with over a decade of experience in the IT industry is a valuable asset, contributing to the company's ability to stay ahead in the competitive and rapidly changing software development sector. This blend of experience levels not only fosters a dynamic work environment but also ensures a deep reservoir of industry knowledge and expertise.

Table 4.2 Participants' Experience

<b>Experience in the IT Industry</b>	<b>Experience in Excellent</b>	<b>Percentage of Participants</b>
<b>10+ years</b>	1 - 3 years	26.09%
	3+ years	13.04%
<b>10+ years Total</b>		<b>39.13%</b>
<b>5 - 10 years</b>	1 - 3 years	26.09%
	3+ years	34.78%
<b>5 - 10 years Total</b>		<b>60.87%</b>
<b>Grand Total</b>		<b>100.00%</b>

Apart from being experienced in the IT industry, employees with prior experience in software development using various methodologies are helpful to complete a project, in a way to enhance customer satisfaction.

The data presented in Table 4.3 is quite encouraging for the implementation of agile methodologies in projects. With half of the workforce already possessing prior experience in agile software development, and a significant portion of them having a substantial 4 to 7 years of experience, the foundation for adopting agile practices is well-established. This depth of experience not only facilitates a smoother transition to agile methods but also ensures that the team has a wealth of practical knowledge to draw upon, which is crucial for navigating the dynamic and often unpredictable nature of project development. Such a skilled workforce can adapt more readily to the iterative processes and rapid changes that characterize agile projects, ultimately leading to more efficient and successful outcomes.

Table 4.3 Prior Agile Software Development (ASD) Experience of Participants

<b>Have Prior ASD Experience?</b>	<b>Prior ASD Experience</b>	<b>Percentage of Employees</b>
<b>Yes</b>	1 - 3 years	17.39%
	4 - 7 years	26.09%
	8+ years	8.70%
<b>Yes Total</b>		<b>52.17%</b>
<b>No</b>	<i>(blank)</i>	47.83%
<b>No Total</b>		<b>47.83%</b>
<b>Grand Total</b>		<b>100.00%</b>

As shown in Table 4.3, the presence of a majority of employees (52.17%) with prior experience in agile software development is indeed a significant asset when transitioning to an agile methodology. This experience can facilitate smoother adoption as these employees are likely to be familiar with the principles and practices of agile project management.

The figures presented in Figure 4.3 show that 52.17% of employees are experienced with both Scrum and Kanban, while 43.48% were exclusively using Scrum, and 4.35% were using Kanban. This suggests a fertile ground for a flexible and iterative approach to project management, which is characteristic of agile practices.

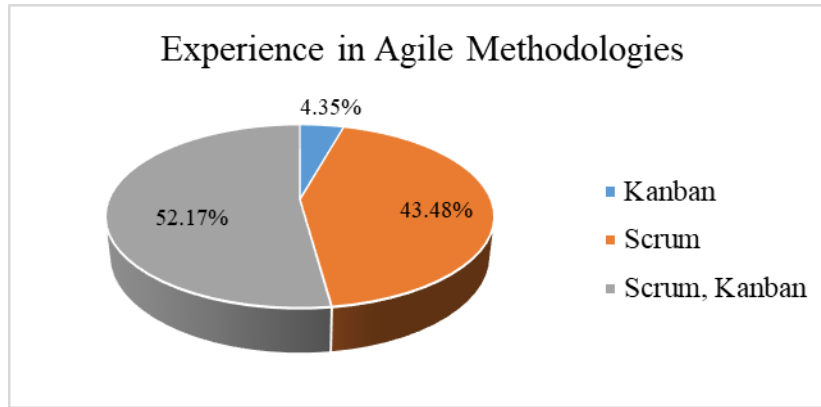


Figure 4.3 Agile Methodology Experience

## 4.4 Agile Methodology Practices

Due to the complex nature of software development, many technical professionals tend to have training in agile software development methodology. Based on the responses of participants (shown in Table 4.4), nearly half of the Exellere employees have received training on agile methodology.

Table 4.4 Project Management Training

Project Management Training?	Agile Methodology Training?	Number of Workforce	Percentage of Workforce
No	No	20	43.48%
	Yes	4	8.70%
<b>No Total</b>		<b>24</b>	<b>52.17%</b>
Yes	No	2	4.35%
	Yes	20	43.48%
<b>Yes Total</b>		<b>22</b>	<b>47.83%</b>
<b>Grand Total</b>		<b>46</b>	<b>100.00%</b>

After joining the company, a huge majority of participants responded that they are using agile methodology in their projects. With 91.3% of employees utilizing Agile project management (see Figure 4.4), it indicates a strong organizational shift towards more dynamic and responsive project management practices. This high percentage suggests that the company values the core aspects of agile methodologies. Such widespread adoption can lead to enhanced productivity, higher product quality, and increased employee satisfaction due to the empowerment and involvement in the project's development process.

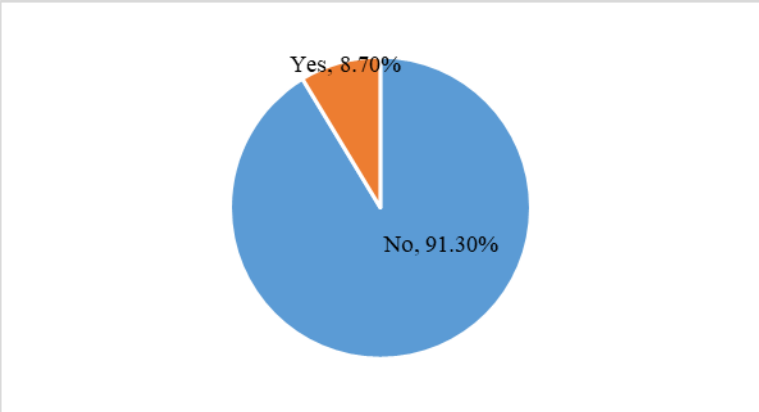


Figure 4.4 Have you worked on projects with non-agile software development methodology?

The high level of familiarity with agile methodologies among employees, as indicated by the data in Figure 4.5, is a strong asset for any company aiming to enhance its project execution capabilities. With over 87% of employees reporting a high degree of familiarity, the organization is well-positioned to leverage agile principles and practices to drive project success.

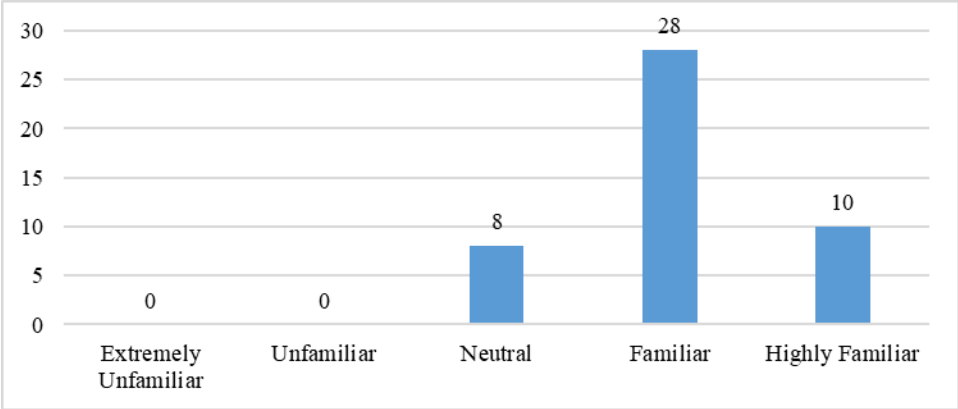


Figure 4.5 Familiarization with Agile Methodologies

All the participants have responded that they are using Scrum as an agile software development framework. However, a few participants have said to have worked on projects that do not use an agile methodology. This happens most of the time when employees are assigned to multiple projects.

In agile software development methodologies, communication with teammates as well as other teams is critical. The participants have replied to have used a host of communication media. There are clients who are abroad, and teams need to communicate as much as needed to achieve customer satisfaction and enhance product quality. Based on the response shown in Table 4.5, MS Teams and Slack chat communication is used by all employees, and more than half of participants said they are also using Email. Even though face-to-face or video call is preferred communication, employees replied the challenge to use them frequently is the time zone difference with the client.

*Table 4.5 Preferred Communication Means*

<b>Communication Means</b>	<b>No. of Participants</b>
Email, Microsoft Teams/Slack Chat	6
Email, Microsoft Teams/Slack Chat, Phone Calls, Face-to-face communication	2
Email, Microsoft Teams/Slack Chat, Video Calls	16
Email, Microsoft Teams/Slack Chat, Video Calls, Face-to-face communication, Slack, Webex	2
Microsoft Teams/Slack Chat	10
Microsoft Teams/Slack Chat, Face-to-face communication	2
Microsoft Teams/Slack Chat, Phone Calls, Video Calls	2
Microsoft Teams/Slack Chat, Video Calls	4
Microsoft Teams/Slack Chat, Video Calls, Face-to-face communication	2

The data indicates that Scrum is the agile software development framework of choice, emphasizing the importance of the product owner’s role in managing the product backlog. Regular communication between the team members and the product owner is crucial, as it allows for continuous feedback and discussion on project-related items. At Excellerent, the norm is to engage with the product owner frequently, either daily or a few times per iteration, as shown in Table 4.6. This consistent interaction is key to enhancing the quality of the product, ensuring that the team remains aligned with the product vision and that any issues are promptly addressed.

*Table 4.6 Frequency of Discussion with Product Owner*

<b>How often discuss with the product owner?</b>	<b>Percentage of Participants</b>
Almost on a daily basis	39.13%
As per the situation	4.35%
During Grooming	4.35%
Every other iteration	4.35%
On Iteration and on story bases	4.35%
Once or twice per iteration	39.13%
Only during the Iteration Planning	4.35%

On top of discussing with the product owner, scrum provides different events and artifacts. According to the responses from participants as shown in Figure 4.6, in Excellerent all of them are doing sprint planning and almost all are doing standup meetings. Most of the participants are also doing sprint retrospective and iteration review.

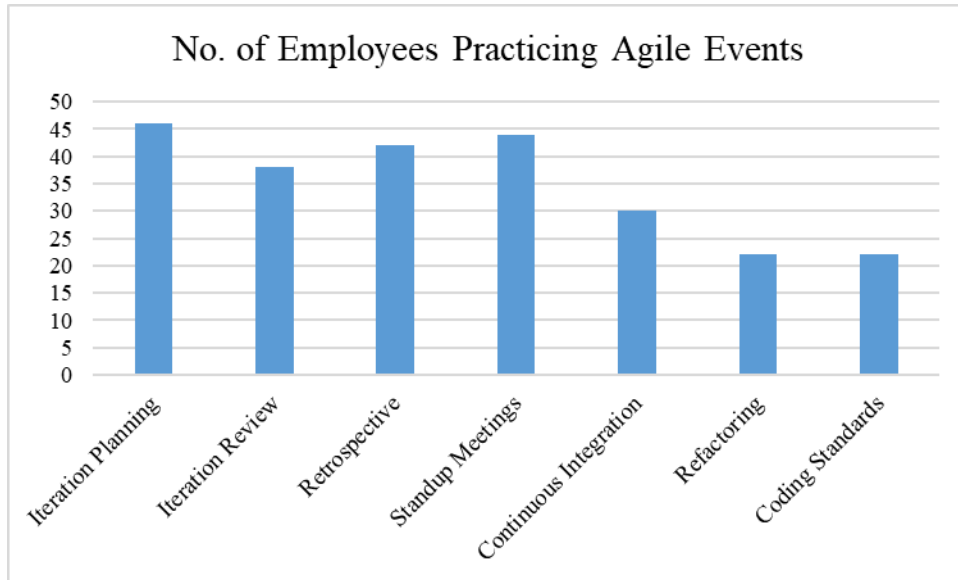


Figure 4.6 Participants practicing Agile Events

During sprint planning, the team needs to plan according to the capacity of the team. The amount of tasks a team can handle in a sprint is determined by team velocity. The data provided that most teams plan with 80% or more of their velocity (see Figure 4.7).

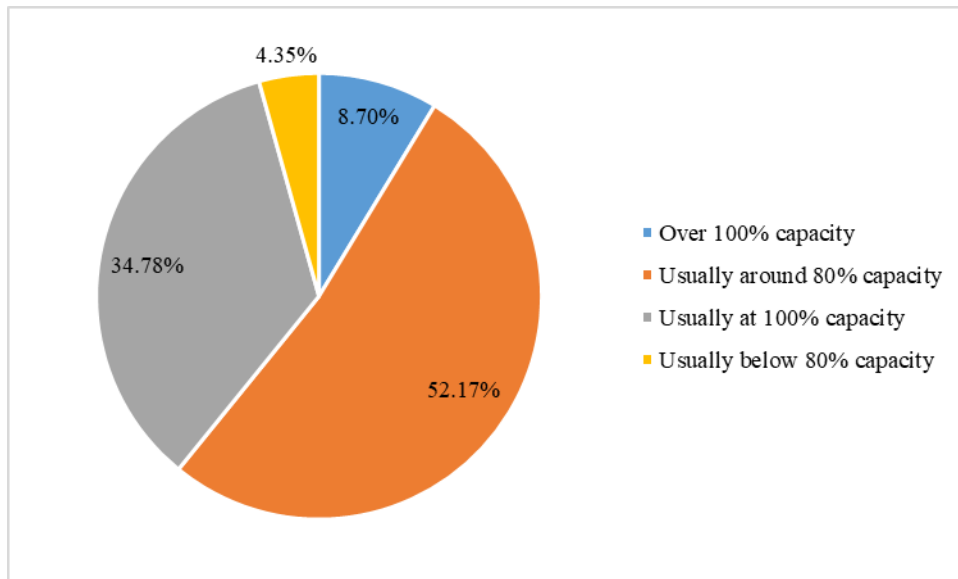


Figure 4.7 Sprint Planning with Velocity

More than half (52.17%) has responded to have planned with around 80% of capacity. However, a plan with over 100%, will likely fail to finish and contribute to the increment, and those planning below 80% are working slowly.

In scrum, standup meetings are used to discuss the progress of the sprint items and provide an opportunity to solve any blocker. In Excellerent, almost 87% of the teams do standup meetings daily; a few others do it irregularly and on a weekly basis (see Figure 4.8).

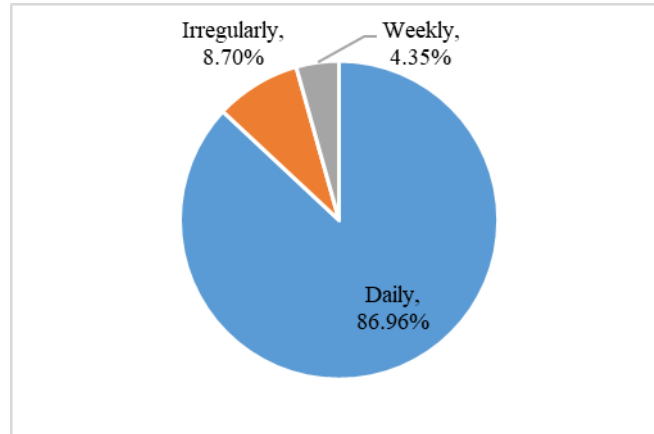


Figure 4.8 Standup meetings frequency

After a sprint is concluded, a review of what went well and identifying areas where the team can improve is important. According to the data, Excellerent has sprint retrospective meetings. Most of the employees have responded to having retrospective once per iteration, while others said to have rarely or once a month (see Table 4.7).

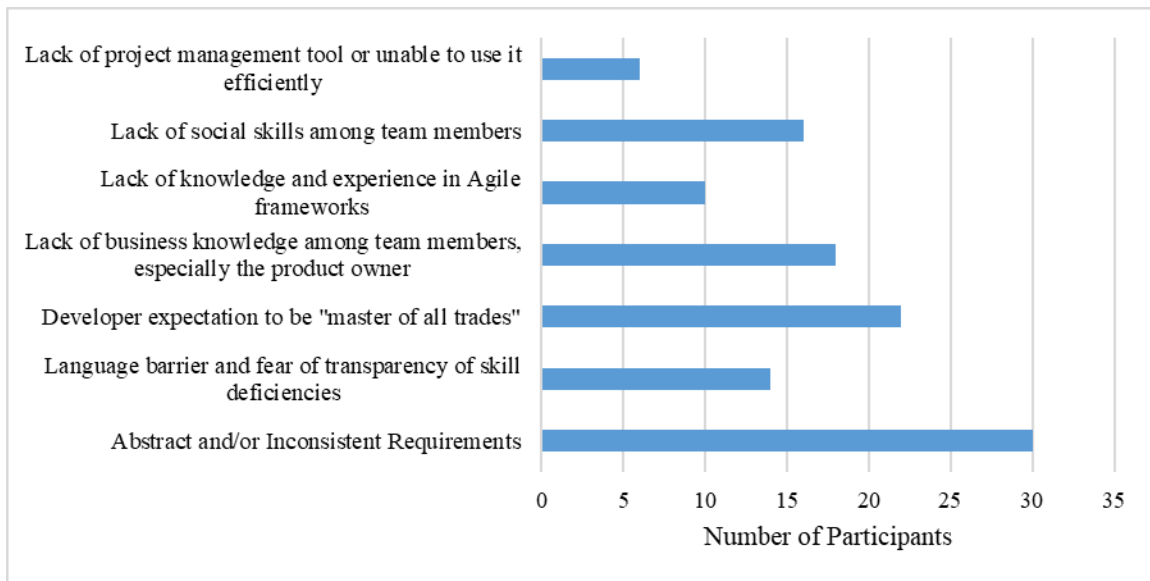
Table 4.7 Sprint retrospective meetings frequency

Retrospective meetings frequency	Percentage
Never	4.35%
Once a month	4.35%
Once per iteration	82.61%
Rarely	8.70%

## 4.5 Agile Software Development Methodology Challenges

Regarding the challenges they faced when working on a project using agile methodology, participants have overwhelmingly responded abstract or inconsistent requirements as the main challenge, which was mentioned by 65% of the participants. As shown in Figure 4.9, the next

major challenge faced by the company is the expectation from developers to be “masters of all trades”. In agile software development, developers are expected to be professionals technically as well as knowledgeable about the business domain. However, if the team were expected to be experts in every aspect ranging from planning to implementation, it would be a challenge affecting the productivity of developers.



*Figure 4.9 Challenges in Agile Software Development*

As an offshore company working with abroad clients, the data has shown that the team is getting challenged with a lack of social skills – about 35% and the other challenge is a lack of business domain knowledge about user stories, especially the product owner. When a team member needs clarification about a user story, the product owner is the first team member to contact; however, about 39% of the participants have said that lack of business knowledge is a challenge.

Another challenge that is mentioned by about 30% of the participants is the language barrier and fear of transparency of skill deficiencies. With the paramount importance of communication, the language barrier becomes a huge blow to the successful running of a project.

Despite all these challenges, while running agile software development, in Excellerent only one participant has responded that a project has failed. The reason for the failure is a lack of necessary skills among team members.

# **Chapter 5 Conclusions and Recommendations**

## **5.1 Overview**

As mentioned in the first chapter, the main objective of this study is to assess agile software development practices, and the challenges faced. In this section of the report, the conclusions of the findings are presented along with the recommendations of the study. The conclusions are made after carefully summarizing the findings in line with the supporting works of literature. The recommendations that would propose suggestions as to how to improve the agile software development practices in the company are also provided.

## **5.2 Summary of Major Findings**

The composition of the workforce, predominantly male and within the age range of 21 to 40 years, reflects a specific demographic trend in the industry. The educational background of the team members, holding bachelor's or master's degrees in software engineering or computer science, is instrumental in the adoption and effective implementation of agile methodologies. This specialized knowledge base is crucial for the dynamic and iterative processes that characterize agile practices, allowing for rapid adaptation to change, continuous improvement, and a focus on customer satisfaction through early and continuous delivery of valuable software.

The findings also indicated that the company's workforce has more than 5 years of experience in the IT industry. Moreover, it was found that 52.17% of the employees have prior agile software development experience, and they were practicing either Scrum or Kanban methodologies. Despite their experiences, nearly half (47.83%) of the employees have claimed to have training on agile methodology.

Almost all (91.3%) employees were using agile methodology for their project execution. This is also complemented by the high familiarization of employees towards agile methodologies. The result of the findings also indicated that scrum is the chosen framework and different communication tools are used. It is also shown that planning and standup meetings are the predominantly held events. In line with this, almost 87% of the teams do standup meetings daily; while others do it irregularly and on a weekly basis.

The result also showed that agile methodologies implementation in the company is challenged. The findings include identifying unclear requirements, which is mentioned by 65% of participants, is the main challenge, followed challenges such as lack of social skills and developer expectation to be master of all trades.

### **5.3 Conclusions**

The workforce composition within a company is a critical factor in maintaining a competitive edge in the market. With a majority of employees aged between 21 to 40 years, the company benefits from a blend of youthful energy and the potential for a long-term commitment to the organization's growth. Specialization in software engineering and computer science, along with related fields, indicates a strong foundation in technical expertise, which is essential to operate in today's software-intensive world. Furthermore, the experience gained from previous roles in the IT industry suggests that the employees are not only technically proficient but also well-versed in the nuanced business processes specific to the company. This combination of demographic vitality, specialized knowledge, and industry experience positions the company favorably in a competitive landscape where innovation and agility are paramount.

The establishment of an offshore software company office in Ethiopia marks a significant expansion for the company, allowing for a more diverse and global reach. The incremental delivery of requirements by clients from abroad aligns well with the principles of agile software development, which the local workforce is already proficient in. This methodology emphasizes adaptability and customer collaboration, making it ideal for handling piecemeal requirements and ensuring that the final product aligns closely with client needs. The team's experience in agile practices will be a cornerstone for success, enabling them to effectively communicate, analyze, and implement solutions in a dynamic and collaborative international business environment.

The adoption of the Scrum framework by over 91% of the workforce signifies a robust commitment to agile principles within the company. This widespread familiarity with agile methodologies not only fosters a dynamic and responsive development environment but also enhances collaboration and efficiency. As a result, the company is well-positioned to leverage the collective expertise of its employees in implementing agile software development practices effectively. This alignment with agile values is likely to drive innovation and maintain a competitive edge in the rapidly evolving tech industry. The company's dedication to these

methodologies demonstrates a forward-thinking culture that values adaptability and continuous improvement.

Scrum's framework fosters a collaborative environment where continuous feedback and iterative progress are key. The sprint planning and daily standup meetings that all teams at Excellerent are engaging in are crucial for setting goals and ensuring everyone is aligned on the day's tasks. The sprint retrospective and iteration review, while not as universally adopted, are equally important as they provide opportunities for teams to reflect on their processes, discuss what worked well, and identify areas for improvement. These practices are essential for maintaining a productive and adaptive team that is dynamic, and capable of delivering high-quality products in a fast-paced industry. It is encouraging to see that Excellerent is embracing these agile methodologies, which are vital for driving project success and fostering a culture of continuous improvement.

Excellerent's adoption of daily standups in 87% of its teams exemplifies a commitment to the agile methodology, ensuring a consistent rhythm and fostering a culture of transparency and collaboration. The remaining teams, opting for less frequent standups, might benefit from considering the potential advantages of increasing the regularity of these meetings to fully leverage the agile framework's strengths in promoting team alignment and efficiency.

During sprint planning, the development team has to participate actively to ensure the plan is according to the capacity of the team. In cases where necessary, additional parties shall be invited to provide business domain or technology domain information. Another scrum event that is attended by the scrum master and the team is the sprint retrospective. In Excellerent, about 83% of the time retrospective meetings are held once per iteration. This helps to understand and learn from what went well during the last sprint, and what to improve in the next sprint.

Agile software development offers a flexible and dynamic approach to project management, which can lead to faster delivery times and better responsiveness to changing requirements. However, the methodology has well well-recognized set of challenges. Inconsistent and abstract requirements can lead to confusion and misdirection, as user stories may not always provide the clarity needed for development. The expectation for developers to be versatile experts can be overwhelming, leading to burnout and reduced productivity. Additionally, when team members lack business knowledge, it can result in a disconnect between the developed solution and the actual business needs. Addressing these issues often requires a concerted effort to improve communication,

provide continuous learning opportunities for team members, and involve business stakeholders more closely throughout the development process.

## **5.4 Recommendations**

The scrum framework's flexibility allows for rapid adaptation to changes, which is crucial in today's fast-paced software development environment. With the workforce being well-versed in this methodology, the company is positioned to effectively manage its projects, regardless of size and nature. This proficiency in agile practices not only streamlines workflow but also fosters a collaborative culture that is conducive to innovation and continuous improvement. However, we need to have daily standup meetings to harvest the advantage of agility. Irregular scrum meetings will not provide the opportunity to adapt to changes as quickly as necessary.

At the end of each sprint, it is customary to hold sprint reviews that help to show what is done to the product owner. Additionally, the company has retrospective meetings that allow the team to understand what went well, and what to improve in the last sprint. The retrospective meeting is an opportunity for the team to reflect internally on their performance, processes, and relationships. To ensure the outcomes of these meetings are actionable, it is crucial to document insights and decisions and to translate them into tangible action items for upcoming sprints. This could involve adjusting workflows, adopting new tools, or revising goals to enhance productivity and team dynamics. By doing so, the continuous improvement cycle that is central to Agile can be effectively maintained, ensuring that each sprint is more efficient and successful than the last.

Despite all the positivity of agile software development, it has challenges to tackle. In Excellent employees has mentioned different challenges including unclear and inconsistent requirements, lack of social skills among team members, developer expectation, etc. The company shall address the major ones if not all these challenges to continue being the preferred software company that delivers quality products on time.

In the agile framework, the product owner plays a crucial role in bridging the gap between the technical team and the business requirements. Their deep understanding of the business domain is essential for accurately conveying the project's needs and priorities. Including a business domain expert in sprint planning can be invaluable for clarifying ambiguous user stories, and ensuring that the development team has a clear and shared understanding of what is required. Failing to do so,

shall pose another challenge specified by the employees – lack of business knowledge among team members. Recognizing that developers have specialized skills but are not universally knowledgeable is key to maintaining focus on coding and development tasks. This approach helps prevent the dilution of their efforts and keeps the team aligned with the core objectives of the project.

Excellent could also consider providing short-term training on social skills to boost developers' ability to communicate. This could be a must as most clients are abroad and effective delivery of a quality product hugely relies on effective communication. Another challenge mentioned is a language barrier and fear of transparency of skill deficiencies. Considering an optimal composition of the development team or pair programming could solve the skill deficiency issue; however, the language barrier is only removed by developing the language skill.

## References

- AgileModeling.com. (2022). *Feature Driven Development (FDD) and Agile Modeling*. Feature Driven Development (FDD) and Agile Modeling. <https://agilemodeling.com/essays/fdd.htm>
- Ahmed, A. (2012). *Software Project Management: A Process-Driven Approach*. Taylor & Francis Group, LLC.
- Alhroub, A., & Jaaron, A. A. M. (2019). Assessing agile project management practices: The case of Palestinian software development companies. *Middle East J. Management*, 6(1), 95–120.
- Al-Zewairi, M., Biltawi, M., Etaiwi, W., & Shaout, A. (2017). Agile Software Development Methodologies: Survey of Surveys. *Journal of Computer and Communications*, 05(05), 74–97. <https://doi.org/10.4236/jcc.2017.55007>
- Bano, M., Imtiaz, S., Ikram, N., Niazi, M., & Usman, M. (2012). *Causes of Requirement Change – A Systematic Literature Review*. 22–31.
- Beck, K. (2000). *Extreme Programming Explained: Embrace change*. Addison-Wesley Professional.
- Cho, J. J. (2010). *An Exploratory Study on Issues and Challenges of Agile Software An Exploratory Study on Issues and Challenges of Agile Software Development with Scrum Development with Scrum*. UTAH STATE UNIVERSITY.
- Excellerent Technology Solutions. (2020). About Excellerent. *Excellerent Technology Solutions*. <https://www.excellerentsolutions.com/about-us/>
- Gechman, M. (2019). *Project Management of Large Software-Intensive Systems: Controlling the Software Development Process*. CRC Press.
- Hajjdiab, H., & Taleb, A. S. (2011). Adopting Agile Software Development: Issues and Challenges. *International Journal of Managing Value and Supply Chains*, 2(3), 1–10. <https://doi.org/10.5121/ijmvsc.2011.2301>
- Hema, V., Thota, S., Naresh Kumar, S., Padmaja, C., Rama Krishna, C. B., & Mahender, K. (2020). Scrum: An Effective Software Development Agile Tool. *IOP Conference Series*:

*Materials Science and Engineering*, 981(2), 022060. <https://doi.org/10.1088/1757-899X/981/2/022060>

Iqbal, J., Omar, M., & Yasin, A. (2019). An Empirical Analysis of the Effect of Agile Teams on Software Productivity. *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 1–8. <https://doi.org/10.1109/ICOMET.2019.8673413>

Jalote, P. (2016). *Software Project Management in Practice*. Pearson India Education Services Pvt. Ltd.

Janes, A., & Succi, G. (2014). *Lean Software Development in Action*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-00503-9>

Kerzner, H. (2017). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling* (12th ed.). John Wiley & Sons, Inc.

Lalsing, V. (2012). People Factors in Agile Software Development and Project Management. *International Journal of Software Engineering & Applications*, 3(1), 117–137. <https://doi.org/10.5121/ijsea.2012.3109>

Layton, M. C., Ostermiller, S. J., & Kynaston, D. J. (2020). *Agile Project Management For Dummies* (3rd ed.). John Wiley & Sons, Inc.

Matharu, G. S., Mishra, A., Singh, H., & Upadhyay, P. (2015). Empirical Study of Agile Software Development Methodologies: A Comparative Analysis. *ACM SIGSOFT Software Engineering Notes*, 40(1), 1–6. <https://doi.org/10.1145/2693208.2693233>

McCarthy, R. (2020). *Lean Methodology: A Guide to Lean Six Sigma, Agile Project Management, Scrum and Kanban for Beginners*.

Padumadasa, E., & Peiris, M. (2011). *Investigation into Intention of Using Agile Methodology for Systems Development in Sri Lanka: Developers Perspective*.

Palmer, S. R., & Felsing, J. M. (2002). *A Practical Guide to Feature-Driven Development*. Prentice Hall Inc.

Papatheocharous, E., & Andreou, A. S. (2013). Evidence of Agile Adoption in Software Organizations: An Empirical Survey. In F. McCaffery, R. V. O'Connor, & R. Messnarz (Eds.),

- Systems, Software and Services Process Improvement* (Vol. 364, pp. 237–246). Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-642-39179-8\\_21](https://doi.org/10.1007/978-3-642-39179-8_21)
- PMI. (2013a). *A Guide to the Project Management Body of Knowledge (PMBOK® guide)* (5th ed.). Project Management Institute, Inc.
- PMI. (2013b). *Software Extension to the PMBOK® Guide* (5th ed.). Project Management Institute, Inc.
- PMI. (2017). *A guide to the project management body of knowledge (PMBOK® guide)* (6th ed.). Project Management Institute, Inc.
- Poppendieck, M., & Poppendieck, T. (2010). *Lean software development: An agile toolkit* (Nachdr.). Addison-Wesley.
- Radaideh, M. A. (2024). *Software Project Management*. De Gruyter.
- Regassa, Z., Bass, J. M., & Midekso, D. (2017). Agile Methods in Ethiopia: An Empirical Study. In *Information and Communication Technologies for Development* (Vol. 504, pp. 367–378). Springer International Publishing. [https://doi.org/10.1007/978-3-319-59111-7\\_31](https://doi.org/10.1007/978-3-319-59111-7_31)
- Richter, F. (2021, July 1). *Women's Representation in Big Tech*. <https://www.statista.com/chart/4467/female-employees-at-tech-companies/>
- Ruhe, G., & Wohlin, C. (Eds.). (2014). *Software Project Management in a Changing World*. Springer.
- Safwan, M. M. M., Thavarajah, G., Vijayarajah, N., Senduran, K., & Manawadu, C. D. (2013). An Empirical Study of Agile Software Development Methodologies: A Sri Lankan Perspective. *International Journal of Computer Applications*, 84(8), 1–7.
- Salameh, H. (2014). What, When, Why, and How? A Comparison between Agile Project Management and Traditional Project Management Methods. *International Journal of Business and Management Review*, 2(5), 52–74.
- Sanghera, P. (2019). *PMP® in Depth: Project Management Professional Certification Study Guide for the PMP® Exam* (3rd ed.). Apress.
- Schwaber, K. (2004). *Agile project management with Scrum*. Microsoft Press.

- Schwaber, K., & Sutherland, J. (2020, November). *The Scrum Guide*.
- Stanley, J. C., & Gross, E. D. (2020). *Project Management Handbook Simplified Agile, Scrum and DevOps for Beginners*. The Tech Academy.
- The Agile Manifesto Authors. (2001). *Manifesto for Agile Software Development*. Manifesto for Agile Software Development. <https://agilemanifesto.org/>
- The Standish Group International. (2015). *CHAOS Report 2015*. The Standish Group International, Inc.
- Villafiorita, A. (2014). *Introduction to Software Project Management*. CRC Press.
- Wonohardjo, E. P., Sunaryo, R. F., & Sudiyono, Y. (2019). A Systematic Review of SCRUM in Software Development. *JOIV : International Journal on Informatics Visualization*, 3(2), 108–112. <https://doi.org/10.30630/joiv.3.2.167>
- Wysocki, R. K. (2019). *Effective Project Management: Traditional, Agile, Extreme, Hybrid* (8th ed.). John Wiley & Sons, Inc.
- Zayat, W., & Senvar, O. (2020). Framework Study for Agile Software Development Via Scrum and Kanban. *International Journal of Innovation and Technology Management*, 17(04), 2030002. <https://doi.org/10.1142/S0219877020300025>

# Appendix

## I. Questionnaire

The goal of this questionnaire is to assess the agile project management practices in Excellent technology solutions and identify the challenges faced and will recommend possible solutions to these challenges based on the existing body of knowledge.

1. Sex
  - Female
  - Male
2. Age
  - 21 – 30
  - 31 – 40
  - 41 – 50
  - 50+
3. What is your level of education?
  - Diploma
  - Bachelor's Degree
  - Master's Degree
  - Ph. D.
  - Other \_\_\_\_\_
4. What is your field of specialization?
  - Computer Science
  - Software Engineering
  - Information Systems
  - Management Information Systems
  - Other \_\_\_\_\_
5. Years of experience in the IT industry
  - <1 year
  - 1 – 5 years
  - 5 – 10 years

- 10+ years
6. How many years of experience do you have in Excellerent Technology solutions?
- < 1 year
  - 1 – 3 years
  - 3+ years
7. Which best describes your current role in the organization?
- Software Engineer
  - Quality Engineer (Quality Assurance)
  - Scrum Master
  - Project Manager
  - Product Owner
  - Technical Delivery Manager
  - DBA
  - Other \_\_\_\_\_
8. Have you ever had formal training on project management?
- Yes
  - No
9. After you joined Excellerent, have you worked on or know a project that is not managed by an Agile method?
- Yes
  - No
10. Did you have experience of Agile Software Development before joining Excellerent?
- Yes
  - No
11. If you respond Yes to the above question, how many years of experience with Agile Software Development did you have?
- < 1 year
  - 1 – 3 years
  - 4 – 7 years
  - 8+ years
12. Which agile methodologies do you have experience with? (Select all that apply)

- Scrum
- Extreme programming (XP)
- Feature Driven Development (FDD)
- Lean Software Development
- Kanban
- Other \_\_\_\_\_

13. Have you ever attended any training on any Agile Methodology?

- Yes
- No

14. How familiarized do you feel with agile methodologies?

1   2   3   4   5

Not at all Familiarized                  Extremely Familiarized

15. Which Agile Methodology is implemented in your projects?

- Scrum
- Extreme Programming
- Feature Driven Development
- Kanban
- Other \_\_\_\_\_

16. Is there a project you are working on that is not using any Agile methodology?

- Yes
- No

17. Which Agile practices do you undertake in your projects? (Select all that apply)

- Iteration Planning
- Iteration Review
- Retrospective
- Standup Meetings
- Coding Standards
- Continuous Integration
- Refactoring
- Other \_\_\_\_\_

18. If you do standup meetings, how often do you meet?
- Daily
  - Weekly
  - Semi-Monthly
  - Monthly
  - Irregularly
  - Never
  - Other \_\_\_\_\_
19. Which means of communication do you prioritize to communicate with team mates and other scrum teams involved in the project delivery? (Select all that apply)
- Email
  - Microsoft Teams/Slack Chat
  - Phone Calls
  - Video Calls
  - Face-to-face communication
  - Other \_\_\_\_\_
20. If the answer to the above question is not calls or face-to-face communication, what is the biggest challenge not to make calls more frequently?
- Language barrier
  - Time zone differences
  - Agenda conflicts (unable to get available time slot)
  - Other \_\_\_\_\_
21. How often do you discuss project related items or request feedback on features/stories developed with your Product Owner?
- Almost on a daily basis
  - Once or twice per iteration
  - Every other iteration
  - Only during the Iteration Planning
  - Other \_\_\_\_\_
22. How often do you have retrospective meetings with your scrum team?
- Once per iteration

- Once a month
- Rarely
- Never
- Other \_\_\_\_\_

23. Are your retrospective meetings resulting in actionable items to bring improvements? If your team never has retrospective meetings, please select 'Not Applicable'

- Yes
- No
- Not Applicable

24. How is your team planning each iteration's capacity according to the team's velocity?

- Over 100% capacity
- Usually at 100% capacity
- Usually around 80% capacity
- Usually below 80% capacity
- Not planning capacity according to team's velocity

25. In your opinion, what is/are the main challenge(s) of running an Agile development project with remote teams? (Select all that apply)

- Abstract and/or Inconsistent Requirements
- Lack of knowledge and experience in Agile frameworks
- Developer expectation to be "master of all trades"
- Lack of social skills among team members
- Lack of business knowledge among team members, especially the product owner
- Language barrier and fear of transparency of skill deficiencies
- Lack of project management tool or unable to use it efficiently

26. Is there any failed project you have worked on?

- Yes
- No

27. If you answer Yes to the above question, what was the reason?

---