



Addis Ababa University  
College of Business and Economics  
School of Commerce

# **Adoption Level and Challenges of Implementing Agile Project Management Methodology at the Private Banking Industry in Ethiopia**

**Asegid Debebe Feyissa (GSE/6494/09)**

A Research Project Work Submitted to the School of Graduate Studies of Addis Ababa University, College of Business and Economics, School of Commerce In Partial Fulfillment of the Requirements for the Degree of Master of Arts in Project Management.

**Advisor: Adane Atara (Ph.D.)**

**August 2019  
Addis Ababa, Ethiopia**

**ADDIS ABABA UNIVERSITY**  
**SCHOOL OF GRADUATE STUDIES**  
**DEPARTMENT OF PROJECT MANAGEMENT**

**ADOPTION LEVEL AND CHALLENGES OF IMPLEMENTING AGILE  
PROJECT MANAGEMENT METHODOLOGY AT THE PRIVATE  
BANKING INDUSTRY IN ETHIOPIA**

**By: Asegid Debebe Feyissa**

**Approval Board Committee:**

Internal Examiner: \_\_\_\_\_ Signature: \_\_\_\_\_

External Examiner: \_\_\_\_\_ Signature: \_\_\_\_\_

Research Advisor: Dr. Adane Atara Signature: \_\_\_\_\_

## DECLARATION

I, the undersigned, declare that the study entitled, *“Adoption Level and Challenges of Implementing Agile Project Management Methodology at the Private Banking Industry in Ethiopia”* is the result of my own effort and study that all sources of materials used for the study have been acknowledged. I have conducted the study independently with the guidance and comments of the research advisor.

This study has not been submitted for any degree in any university. It is conducted for the partial fulfillment of the Master of Arts Degree in Project Management.

**Asegid Debebe**

---

**Signature and date**

## LETTER OF CERTIFICATION

This is to certify that Asegid Debebe has conducted this project work entitled *“Adoption Level and Challenges of Implementing Agile Project Management Methodology at the Private Banking Industry in Ethiopia”* under my supervision.

This project work is original and suitable for the submission in partial fulfillment of the requirement for the award of Master of Arts Degree in Project Management.

**Adane Atara (Ph.D.)**

---

**Signature and date**

## **Acknowledgment**

I would like to express my gratitude to my family who have encouraged and supported me throughout the course of this program since its beginning in 2009 E.C. I owe you a lot for patiently accommodating my demands and making it possible for me to complete this program. This wouldn't have been possible without your understanding and continuous support, especially while working on a full-time job.

I would also like to express my gratitude to our instructors at School of Commerce, who have taught us a great deal, supported, and encouraged us all the way from the beginning to the end of this program.

A big thank you also goes to the software developers and development team managers at the 7 private banks in Ethiopia who participated in this survey. They have kindly accepted my request and cooperated with the study and completed the questionnaire by allocating more than 20 minutes each to fill the data. The managers from the participating banks have also kindly allocated additional time and allowed me to hold a discussion with them for the open-ended part of the questionnaire which I appreciate very much. Thank you!

Asegid Debebe

## Abstract

This qualitative study has investigated the current level of adoption the Agile software project management methods at the private banking industry in Ethiopia and the challenges faced by the software development teams at the banks. The current 16 private banks in Ethiopia have been grouped into three categories using the decades of their year of establishments, and from each category, 3 banks were selected at random using the lottery method giving a total sample size of 9. Two sets of questionnaires have been used with one questionnaire distributed to the software development team members and the other distributed to the development teams' managers. Among the 9 banks, complete data was received from 7 banks with total individual responses of 30. The gathered data was analyzed using descriptive statistics. The analysis of the data shows that there is a low prevalence of application of agile methods at the private banks in Ethiopia with only 2 out of the 7 participating banks (28%) in the process of adopting the Scrum Agile method and with the remaining 5 (72%) still using the traditional waterfall method. There are several challenges that face the banks from adopting the agile method and these are presented in three categories as customer-related, team-related and management-related. Customer-related challenges include inconsistent requirements, lack of timely feedback, resistance to change, and lack of full commitment to the process. Team-related challenges include reluctance to work in a team, lack of experience, handling of helpdesk requests while working in software development and departure of experienced team members. Management-related challenges include setting unreasonably short deadlines, de-prioritization of ongoing projects out of work, not specifying standard software development tools and frameworks, absence of general long-term ICT Development strategic plan specific to the improvement of the software development process. In conclusion, there is a very low level of adoption of the agile methodology in the private banking sector in Ethiopia with only 28% of the banks attempting the move to the Scrum method while the remaining 72% is planning to adopt an agile method in the future and still using the traditional waterfall software development methodology.

**Keywords:** Agile Project Management, Agile adoption, Agile adoption challenges, Ethiopia, Private Banking, Software Development

## Acronyms

APM	Agile Project Management
ASD	Agile Software Development
ASDM	Agile Software Development Methodology
PMI	Project Management Institute
PMLC	Project Management Life Cycle
PMBok	Project Management Body of Knowledge
SDLC	Software Development Life Cycle
SDLCM	Software Development LifeCycle Model
TSD	Traditional Software Development
TSDM	Traditional Software Development Methodology
CI/CD	Continuous Integration / Continuous Delivery
XP	eXtreme Programming

## List of Tables and Figures

### List of tables

Table 1.2.1: Private Banks branches and total capital (source: NBE 2018) . . . . .	4
Table 2.3.1: Summary of factors and attributes for the success of Agile methods . .	23
Table 2.3.2: Failure factors for ASDMs . . . . .	24
Table 3.3.1: Sample clusters and sample size . . . . .	31
Table 4.1.1 Number of complete responses from banks . . . . .	34
Table 4.2.1 Number of participants by team roles . . . . .	35
Table 4.2.2 Number of participants by educational level . . . . .	35
Table 4.2.3 Number of participants by years of experience . . . . .	35
Table 4.2.4 Team communication frequency of participants . . . . .	35
Table 4.2.5 Team communication mode . . . . .	35
Table 4.3.1 SD responses by managers and team members . . . . .	37
Table 4.3.2.1 SD methodology at each private bank . . . . .	39
Table 4.3.4.1: Client- related challenges at each private bank . . . . .	40
Table 4.3.4.2: Development team-related challenges at each private bank . . . . .	41
Table 4.3.4.3: Management-related challenges at each private bank . . . . .	41
Table 4.3.4.1 summary of developer responses to questions in section II . . . . .	46

### List of figures

Fig. 1.2.1 Operational private banks in Ethiopia . . . . .	3
Fig. 2.1.1 Waterfall method (source: (Sutherland, 2014)) . . . . .	10
Fig. 2.2.1.1: The Scrum framework . . . . .	18
Fig. 2.2.1.2: The Scrum Process . . . . .	19
Fig. 2.5.1 Conceptual Framework . . . . .	28
Fig. 4.2.1 Number of participants by their team size . . . . .	36
Fig. 4.2.2 Number of participants and project durations . . . . .	36
Fig. 4.3.1 Participants overall perception of their current SDM . . . . .	38

## Contents

<i>DECLARATION</i> .....	<i>iii</i>
<i>LETTER OF CERTIFICATION</i> .....	<i>iv</i>
<i>Acknowledgment</i> .....	<i>v</i>
<i>Abstract</i> .....	<i>vi</i>
<i>Acronyms</i> .....	<i>vii</i>
<i>List of Tables and Figures</i> .....	<i>viii</i>
<b>CHAPTER 1: INTRODUCTION</b> .....	<b>1</b>
<b>1.1 Background of the study</b> .....	<b>1</b>
<b>1.2 Background of the private banking industry in Ethiopia</b> .....	<b>3</b>
<b>1.3 Statement of the problem</b> .....	<b>6</b>
<b>1.4 Research questions</b> .....	<b>7</b>
<b>1.5 Objectives of the study</b> .....	<b>7</b>
1.5.1 General objective .....	7
1.5.2 Specific objectives .....	7
<b>1.6 Significance of the study</b> .....	<b>8</b>
<b>1.7 Scope of the study</b> .....	<b>8</b>
<b>1.8 Organization of the study</b> .....	<b>9</b>
<b>1.9 Definitions of key terms</b> .....	<b>9</b>
<b>CHAPTER 2: LITERATURE REVIEW</b> .....	<b>10</b>
<b>2.0 Introduction</b> .....	<b>10</b>
<b>2.1 The waterfall software development method</b> .....	<b>10</b>
<b>2.2 Agile project management methodologies</b> .....	<b>12</b>
2.2.1 The Scrum framework .....	16
2.2.2 Extreme Programming (XP) .....	19
<b>2.3 Success and failure factors in Agile software projects</b> .....	<b>23</b>
<b>2.4 Empirical literature review of the study</b> .....	<b>26</b>
<b>2.5 Conceptual framework</b> .....	<b>28</b>

<b>CHAPTER 3: RESEARCH METHODOLOGY .....</b>	<b>29</b>
<b>3.0 Introduction.....</b>	<b>29</b>
<b>3.1 Research design.....</b>	<b>29</b>
<b>3.2 Data sources and data collection techniques .....</b>	<b>29</b>
<b>3.3 Study population and sampling method .....</b>	<b>30</b>
<b>3.4 Data collection procedures and instruments .....</b>	<b>32</b>
<b>3.5 Research approach.....</b>	<b>32</b>
<b>3.6 Data analysis method .....</b>	<b>32</b>
<b>3.7 Validity and reliability of data .....</b>	<b>32</b>
<b>3.8 Ethical considerations.....</b>	<b>33</b>
<b>3.9 Study location .....</b>	<b>33</b>
<b>CHAPTER 4: DATA ANALYSIS AND PRESENTATION.....</b>	<b>34</b>
<b>4.1 Introduction.....</b>	<b>34</b>
<b>4.2 Characteristics of respondents.....</b>	<b>35</b>
<b>4.3 Data analysis.....</b>	<b>37</b>
4.3.1 The current software development methodology.....	37
4.3.2 The current level of adoption of APM in the industry .....	39
4.3.3 Current challenges against adopting an agile methodology.....	39
4.3.4 Summary of developer perspectives on Agile and experience with agile tool .....	46
<b>CHAPTER 5: SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS.....</b>	<b>48</b>
<b>5.1 Summary of findings.....</b>	<b>48</b>
<b>5.2 Conclusion .....</b>	<b>50</b>
<b>5.3 Recommendations .....</b>	<b>51</b>
<b>5.4 Limitations and suggestions for further study .....</b>	<b>53</b>
<b>References .....</b>	<b>54</b>
<b>Appendices.....</b>	<b>57</b>
<b>Appendix A – Research Instruments .....</b>	<b>58</b>
Data Collection Questionnaire for Developers .....	58
Data Collection Questionnaire for Managers .....	64

## CHAPTER 1: INTRODUCTION

### 1.1 Background of the study

Software development is the process of creating valuable software system for a client based on systems requirements provided by the client to solve problems and create value for the client. Creating a software system follows a series of steps that define the software development life cycle (SDLC). Software development projects are among the riskiest project undertakings due to the inherent uncertainty involved in the initial stages of requirements definition to the complex coding work that is needed for implementation and the detailed communications requirement to clearly define the expectations of the software product. As a result of this, software development projects are among the riskiest endeavors with a higher failure rate as compared to other industries/sectors with almost 25% IT projects experiencing outright failure (Arcidiacono, 2017). To remedy this industry-level problem, several project management methodologies were devised and implemented to improve the success rate for projects in general and software development projects in particular.

The most recently introduced methodologies are given an umbrella term of *Agile Project Management Methodologies (APMs)*.

It has been about 18 years since the agile manifesto was written and signed by a group of 17 experienced software practitioners (Beck *et al.*, 2001). The manifesto was drafted and published in 2001 and has been popularized since.

Agile software development represents a major departure from traditional, plan-based approaches to software engineering (Dybå and Dingsøy, 2008).

The agile methodology has increasingly become popular since its release and has been widely adopted by the software development industry worldwide to deliver valuable and quality software in a timely manner with a higher rate of customer satisfaction.

In the current world with the ubiquity of computers & smartphones, it will be very difficult to find anyone that does not rely directly or indirectly, on the smooth operation of one or more software systems. Organizations in developing nations like

Ethiopia are increasingly creating and adopting software systems to facilitate and improve their day to day operations. Financial institutions, being no exceptions, need to use a reliable software system to run their core business and stay competitive in the market.

This study looks into the current state of the adoption of agile project management methodologies at the private banks in Ethiopia with the aim of establishing their adoption level and to identify any issues or challenges the development teams are facing while adopting an APM method.

## 1.2 Background of the private banking industry in Ethiopia

The government's reform of the financial system in 1994 allowed citizens to form financial institutions such as private banks and insurances and to engage in financial services such as loans and provision of insurance coverage. As a result of this, there were six private banks that were established between 1994 and 1999 G.C. in Ethiopia and currently, there are totally 16 (sixteen) private banks that operate within the country.



*Fig 1.2.1 Operational private banks in Ethiopia (as at 2019 GC)*

A previous study conducted between 2004 and 2010 indicates that the banking sector in Ethiopia has been doing well financially and the sector, in general, is exhibiting increasing volume of deposits, loan provision and asset possession (Abera, 2015).

According to National Bank of Ethiopia annual report for fiscal year 2017/2018, the total capital of the private banking sector in Ethiopia accounts to 39.9% of the overall national banking sector budget. As compared to the 2016/2017 percentage of the overall capital (35.3%), the 2017/2018 private banking capital has exhibited 5.6% increase showing that private banking sector is becoming more competitive in terms of capital gain.

Table 1.2.1: Private banks branches and total capital (source: NBE 2018)

#	Bank Name	Number of branches in 2018 GC	Total capital in 2018 GC (Billion birr)
1	Awash Bank	382	4.210
2	Dashen Bank	381	3.725
3	Abyssinia Bank	284	3.265
4	Wegagen Bank	292	3.195
5	United Bank	233	2.579
6	Nib International Bank	228	2.991
7	Cooperative Bank of Oromia	332	1.924
8	Lion International Bank	210	1.479
9	Oromia International Bank	260	1.890
10	Zemen Bank	25	1.391
11	Buna International Bank	176	1.667
12	Berhan International Bank	168	1.936
13	Abay Bank	162	1.514
14	Addis International Bank	59	0.789
15	Debub Global Bank	43	0.614
16	Enat Bank	40	1.045
	<b>TOTAL</b>	<b>3,275</b>	<b>34.214</b>

Delivery of quality financial services drives the capabilities of each bank and effective IT services including optimal software systems are among the primary enablers for the banks to provide quality services to their clients.

Software applications are the main enabling tools used by banks worldwide including high performance and efficient database systems with high-speed and reliable network infrastructure. Banks need to stay up to date on the implementation of software products that can cater for the needs of their clients satisfactorily and this demands

continuous learning of client needs and implementation cycles that are best addressed through agile management and delivery process.

As such, the banks are continually improving their products and are releasing new and improved products to satisfy the needs of their clients.

Recently implemented systems by the banks include:

- **Core banking** - centralized client account management and distributed data access allowing clients to deposit and withdraw money from their accounts at any branch.
- **Internet banking** – a service where clients can access and manipulate their accounts via the internet using computers or mobile devices.
- **Mobile banking** – allows access to the client account information and transactions using SMS messaging.

In addition to external client-directed software products that need to be maintained by the banks, there are also internal software systems that need continual improvement and maintenance to facilitate the internal processes that need to be managed by the banks. These include:

- Human Resource Management Systems
- Shareholders and Share Management Systems
- Inventory Management Systems
- Fixed Asset Management Systems
- Document/Contracts Management Systems
- Internal Financial Management Systems including staff payroll and credit processing systems

Hence, it is imperative that the banks implement an optimal software project management and delivery process so that quality software products can be made available to their internal and external clients in a cost-effective and timely manner.

### 1.3 Statement of the problem

The traditional waterfall software development methodology suffers from various shortcomings that risk the overall success of the project in terms of fulfilling its goals. Oyong and Ekong (2019) have identified some shortcomings which include the early complete planning that is mandated at the initial stage of the waterfall process leads to failures for large and complex projects. In addition, the delayed customer feedback to the last stage of the project leads to additional corrective work due to the bugs that are reported close to the completion of the project.

According to Sidky and Arthur (2015), Agile Project Management methodologies are being adopted worldwide by leading software providers mainly due to the flexibility the methodology provides, the cost-effective way of producing high-quality software in a reasonable period of time and a high degree of customer satisfaction. The benefits of agile include but not limited to, a quicker return on investment, better software quality, and higher customer involvement and overall satisfaction (Sidky and Arthur, 2015).

Hence, an organization that heavily relies on the optimal and reliable operation of software systems and works in a competitive environment will highly benefit from an implementation of a client-oriented, incremental, and adaptive project management framework which at the moment is the Agile Project Management methodology.

A study conducted by Regassa, Bass, and Midekso (2017) has shown that Agile practices are being used at private software companies in Ethiopia mainly to encourage user participation and clarify user requirements. However, the same study has shown that these companies work mainly on government projects that mandate extensive requirements and design documentation which require approval prior to delivery of working software that complies with a predefined delivery schedule. This will limit the application of agile methods which favor less documentation and incremental delivery of working software.

This research focuses on investigating the current level of the adoption of agile methodologies among software development teams at private banks currently operating in Ethiopia. During literature review, the author was not able find a research

addressing the current level of adoption of APM at private banks in Ethiopia and this research aims to contribute towards reducing the gap.

## **1.4 Research questions**

This study has tried to answer the following three research questions:

1. What is the current adoption level of the agile software project management methodology at private banks in Ethiopia?
2. What is the most commonly used software development method at private banks in Ethiopia?
3. What are the primary challenges that hinder software project managers at the banks from adopting agile methods?

## **1.5 Objectives of the study**

### **1.5.1 General objective**

The general objective of the study is to find out the current adoption level of the agile software project management methodology at private banks in Ethiopia?

### **1.5.2 Specific objectives**

1. To determine the most commonly used software development methodology at private banks in Ethiopia.
2. To identify challenges and issues facing the teams at the banks from adopting agile software development methods.

## **1.6 Significance of the study**

This research will be beneficial to private banks in Ethiopia to understand the overall industry-level adoption level of the agile methodology and helps set their future direction towards implementing a more agile software delivery process so as to reap the benefits the agile methodology.

Identification of primary hurdles in implementing agile at the banks and suggestion of possible solutions to these issues will positively help managers and the organizations in the adoption process.

It will also raise the awareness of the software team members about this methodology and encourage them to explore further and adopt the methodology for better software development and delivery process.

## **1.7 Scope of the study**

The scope of the research is limited to identification of the current level of adoption of APM by the software development teams and description of challenges related to the software development processes at the private banks in Ethiopia.

The private banks also work with consulting firms (usually of foreign origin) for the development and deployment of their core and strategic software systems. This study does not take into account the software development practices employed by the consulting firms which are external to the banks.

## 1.8 Organization of the study

This research paper is organized into five chapters. The first chapter describes the background of the study, statement of the problem, research questions, objectives and the significance of the study as well as the scope of the study. The second chapter is dedicated to the review of the related theoretical and empirical literature on the subject and the conceptual framework. The third chapter is about the research methodology which includes the research type and design, sampling method and participants, data collection techniques and instruments. Chapter four is on the data analysis and presentation and description of the results. Chapter five summarizes the findings of the study with conclusion and recommendations from the study. It also includes limitations of the study and suggestions for further study for future research.

## 1.9 Definitions of key terms

**Project Management (PM):** an organized common-sense approach that utilizes the appropriate client involvement in order to meet sponsor needs and deliver the expected incremental business value (Wysocki, 2019).

**Agile Project Management (APM):** is an iterative and incremental based development, where requirements are changeable according to customer needs (Sharma, Sarkar and Gupta, 2012).

**Software Development Lifecycle (SDLC):** a series of steps that software developers go through to develop software (Schaffer, 2013).

## CHAPTER 2: LITERATURE REVIEW

### 2.0 Introduction

Software development is the process of creating a software application to solve one or more problems for a client based on the specification provided by the client. Software development lifecycle tracks and manages the steps necessary to create and maintain software from initial ‘conception’ stage until the software is ‘retired’ or removed from production use (Schaffer, 2013). There are many software development lifecycle management models in use today. The oldest structured method was the Waterfall software development methodology and it was invented by Winston W. Royce in 1970.

### 2.1 The waterfall software development method

The waterfall method proposed the development of software in phases shown below with the output from the earlier phase being used as an input into the next phase. This method was borrowed from engineering discipline which puts a lot of weight on precise early planning making the model more or less plan-driven with documentation of a complete set of requirements preceding architectural and high-level design, development and implementation (Oyong and Ekong, 2019). The waterfall and its variant software development methods are generally known as Traditional Software Development Methodologies (TSDMs)

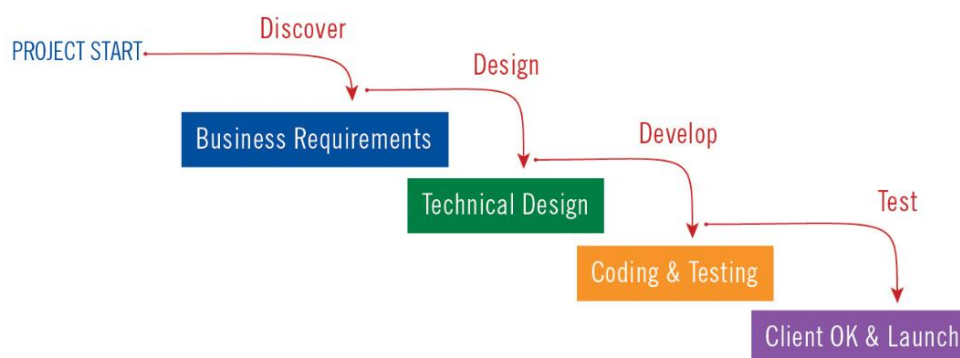


Fig 2.1.1 Waterfall method (source: (Sutherland, 2014))

The waterfall methodology works best when developers know all the requirements in advance and when requirements are relatively stable. Hence it will be difficult to

apply the waterfall method when it is either not possible for the development team to understand all the requirements necessary to build the software at an earlier stage or when the requirements are susceptible to changes due to one of the many reasons that result in the change of the requirements.

While the traditional software development methods are more applicable to small projects or larger projects that have their requirements not changing for a longer period of time, they are not very effective for complex projects that have changing requirements or where the requirements cannot be clearly be defined at the earlier stage of the development. Traditional development methods are more focused on following well-defined plans and are inflexible in a rapidly changing business environment (N.Vithana, G. S. Fernando and Kapurubandara, 2015). Oyong and Ekong (2019) have identified the following shortcomings the waterfall software development methods:

- Planning upfront for large projects leads to failures
- Delayed customer feedback and modules testing to the last phase of the project
- Not flexible to requirements changes
- Not all documented requirements are implemented
- Prone to litigations due to breach of contract items, not by meeting schedule dates, among others

The researchers quoted above also stated the following strengths of the traditional software development methods:

- Stability
- Planning upfront
- Detailed documentation
- Management control of decision making in the project
- Fixed cost estimates
- Fixed schedules
- Codified process
- Encourage reuse of pre-process modules
- Requirements known in advance, and fixed.

## 2.2 Agile project management methodologies

Agile systems development methodologies are a collection of software development methodologies that are design to accommodate changes in the environment during the implementation of the project activities. The traditional software development methodologies lock requirements in place before moving to the development of the system, agile SDM methods can adjust project requirements based on changes in the environment or in the requirements (Lee and Xia, 2010).

Research shows that one of the main reasons for failures for software projects is changing requirements (Bano *et al.*, 2012).

Bano *et al* (2012) have identified three primary sources of change for software requirements which are given below:

- Business reasons
  - Changes caused by the market demands resulting in a change in system requirements
- Organizational reasons
  - Changing organizational policy
  - Vague product vision and strategy
  - Business case not thoroughly evaluated
- Project-related reasons
  - Developers increased understanding of the product
  - Users understanding/learning of the system

The software requirements can change during the different phases of the SDLC and this calls for a better software development management methodology that better adapts to the changing nature of the requirements throughout the SDLC. Agile processes accept the reality of change and manage requirements by “embracing” rather than “controlling” changes.

The agile software development methodology was officially introduced in 2001 by a group of 17 senior software practitioners. Agile is a conceptual framework of practices and principles to develop software faster, incrementally and to produce a satisfied customer (Patel and Ramachandran, 2009).

Agile software development methodologies are characterized by iterative development, continuous integration (CI) and the ability to handle changing business requirements (Boehm and Turner, 2005).

The Agile Manifesto states that:

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

- ***Individuals and interactions over process and tools***
- ***Working software over comprehensive documentation***
- ***Customer collaboration over contract negotiation***
- ***Responding to change over following a plan”***

Principles behind the Agile Manifesto (Beck *et al.*, 2001) are:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is a face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Masood and Farooq (2017) have noted that agile project management presents a shift from the traditional project management approaches for its recognition that change is an inevitable part of projects and hence placing less emphasis on the development of rigorous project requirements and plans while on the other hand, emphasizing:

- Implementation of projects in iteration
- Revision of projects requirements after each cycle
- Collaboration between the project team and other stakeholders throughout the project implementation process

They have also noted that the goal of the agile approach is to make the project implementation process responsive to changes in the project environment.

*“A truly Agile method must be iterative (take several cycles to complete), incremental (not deliver the entire product at once), self-organizing (teams determine the best way to handle work), and emergent (processes, principles and work structure are recognized during the project rather than predetermined)”* (Boehm and Turner, 2005).

Since its popularization, the agile software development methodologies have introduced best practices into software development. The agile methodology is based on 4 core values and 12 principles (Beck *et al.*, 2001).

Annual State of Agile report (CollabNet VersionOne, 2019) has shown that the adoption of agile software project management methods in the last decade has been increasing steadily. The 13<sup>th</sup> Annual Stage of Agile report survey undertaken in 2019 on agile teams worldwide indicated that the most recent reasons for adoption of agile methodologies are reduced project cost and improvement of customer satisfaction.

According to (Masood and Farooq, 2017), benefits of agile methodologies include a reduction in the cost of re-work, fast completion of projects and greater customer satisfaction. They have also noted some challenges that arise as a result of using agile methods, these are; difficulty in scheduling tasks, difficulties in managing knowledge

and difficulties in managing large and multi-site projects can be surmounted by blending elements of traditional project management and agile methods.

According to Oyong and Ekong (2019), Agile software development methods have the following strengths over the traditional or Waterfall development methodology:

- Simplicity
- Hitting the market on target time
- Customer satisfaction guaranteed
- People-oriented
- Iterative delivery of small and complete pieces of software on time
- Face-to-face communication
- Close cooperation between developers and the clients
- Freedom to developers
- Regular adaptation to changes in requirements

On the other hand, these researchers have also noted the following limitations or shortcomings of agile software development methods:

- Agile software development methodologies are difficult to implement
- Limited documentation
- Customer must not only be knowledgeable but also interested in the ASDM to work.
- Only skilled developers can implement ASDM
- Unrealistic expectations
- Constant need for attention
- Testing is not cheap
- Not much empirical evidence to support as ASDM's claims
- No fixed project cost and schedule in advance

Combining the advantages of both ASDM and TSDM methods and attempting to reduce the shortcomings of both methods, Oyong and Ekong (2018) have suggested a new framework known as the Ambidextrous framework.

There are several specific methods included in the agile family while the most notable ones are listed below (Abrahamsson and Salo, 2002; Chow and Cao, 2008).

- Scrum
- Extreme Programming (XP)
- Dynamic Systems Development Model (DSDM)
- Crystal
- Feature Driven Development (FDD)
- Lean Software Development (LSD)

Among these methodologies, Scrum and XP are the most popular methodologies adopted by the software development community worldwide (VersionOne, 2018). The overview of these methodologies has been given below.

### **2.2.1 The Scrum framework**

Scrum was created by Jeff Sutherland and Ken Schwaber in 1995 well before the agile project management values and principles were officially introduced to the public in 2001 in the form of the Agile Manifesto.

*“At its root, Scrum is based on a simple idea: whenever you start a project, why not regularly check in, see if what you are doing is heading in the right direction and if it’s actually what people want? And question whether there are any ways to improve how you’re doing what you’re doing and ways of doing it better and faster, and what might be keeping you from doing that.”* (Sutherland, 2014)

Scrum is an adaptive and iterative project management and delivery framework that does not necessarily require an early and full provision of a detailed specification of the product requirements by the client to the development team. It is geared towards a discovery of more fine-grained and detailed product requirements and an optimal solution through small, manageable, and incremental steps called sprints.

*“Scrum focuses on project management in institutions where it is difficult to plan ahead, with mechanisms for ‘empirical process control’ where feedback loops constitute the core elements. Software is developed by a self-organizing team in increments (called ‘sprints’), starting with planning and ending with a review.*

*Features to be implemented in the system are registered in a backlog. Then the product owner decides which backlog items should be developed in the following sprint. Team members coordinate their work in a daily stand-up meeting. One team member, the Scrum Master, is in charge of solving problems that stop the team from working effectively.” (Dybå and Dingsøy, 2008)*

*“Scrum is an agile project management method which uses an empirical process based on flexibility, adaptability, and productivity and sets high-frequency management activities in order to track real-time problems during a project development process” (Cardozo et al., 2010)*

Scrum has three roles that participate in the framework and manage the whole product development process (Schwaber, 2004). These are:

- **The Product Owner** – responsible for representing the interests of everyone with a stake in the project and its resulting system. The Product Owner achieves initial and ongoing funding for the project by creating the project’s initial overall requirements, return on Investments (ROI) and objectives and release plans. The list of requirements is called the Product Backlog. The Product Owner is responsible for using the Product Backlog to ensure that the most valuable functionality is produced first and built upon; this is achieved by frequently prioritizing the Product Backlog to queue up the most valuable requirements for the next iteration (sprint).
- **The Team** – is responsible for developing the functionality. Scrum Teams are self-managing, self-organizing, and cross-functional, and they are responsible for figuring out how to turn Product Backlog into an increment of functionality within the iteration and managing their own work to do so. Team members are collectively responsible for the success of each iteration, and of the project as a whole.
- **The Scrum Master** – is responsible for the Scrum process, for teaching Scrum to everyone involved in the project, for implementing Scrum so that it fits within an organization’s culture and still delivers the expected benefits, for ensuring that everyone follows Scrum rules and practices.

The people who fill these roles are those who have committed to (as opposed to those involved in) the project. Scrum makes it clear and ensures that those responsible for the project have the authority to do what is necessary for its success and those who aren't responsible can't interfere unnecessarily.

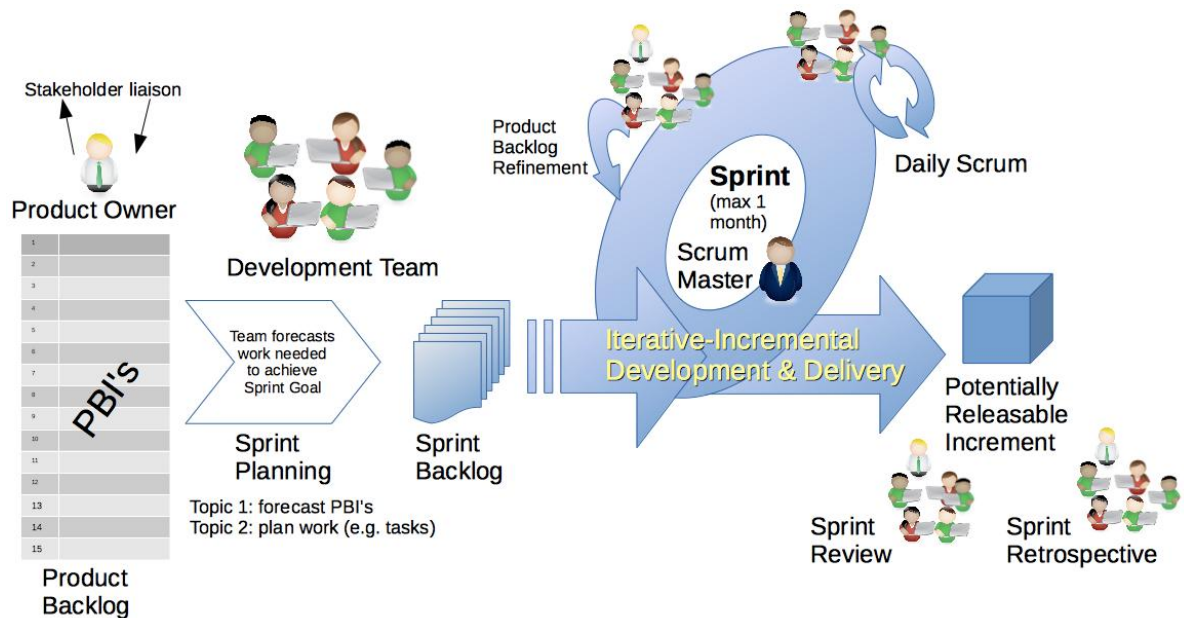


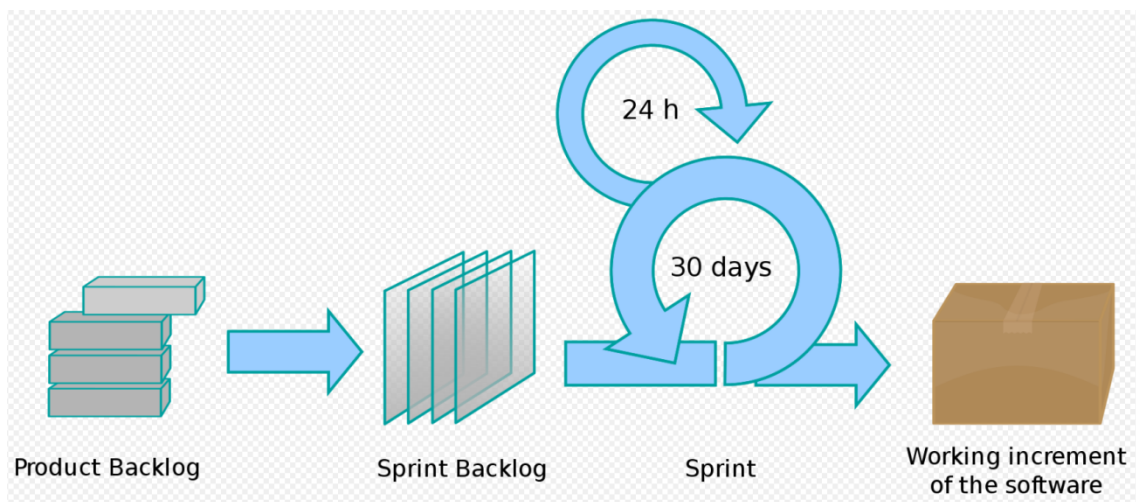
Fig. 2.2.1.1: The Scrum framework (Source: Wikipedia)

*“Scrum embraces uncertainty and creativity. It places a structure around the learning process, enabling teams to assess both what they have created and, just as important, how they created it. The Scrum framework harnesses how teams actually work and give them the tools to self-organize and rapidly improve both speed and quality of work” (Sutherland, 2014).*

Scrum introduces few new artifacts that are used throughout the Scrum process and are described below (Schwaber, 2004):

- **Product Backlog** – the total list of requirements for the system or product being developed by the project(s). The Product Owner is responsible for the contents, prioritization, and availability of the Product Backlog. It evolves as the product and the environment in which it will be used evolves and is never complete. It is dynamic as management constantly changes it to identify what the product needs to be appropriate, competitive and useful.

- **Sprint Backlog** – this defines the work or tasks that the Team defines for turning the Product Backlog it selected for that Sprint into an increment of potentially shippable product functionality. Tasks in the Sprint Backlog need to roughly take between 4-16 hours of work and only the Team can change the Sprint Backlog.
- **Increment of Potentially Shippable Product Functionality** – this is a new product functionality that has been completed at the end of the Sprint period and must be potentially shippable into the production environment and become available to users of the system if the Product Owner decides to deploy it.



*Fig. 2.2.1.2: The Scrum Process (Source: Wikipedia)*

A research conducted by Cardozo *et al.* (2010) using a systematic literature review of published articles on the Scrum methodology has identified that Scrum is significantly related to productivity in software projects. In the same research it has also been shown that in addition to productivity, the usage of the Scrum method is also positively related with better customer satisfaction, good quality software, team motivation, and cost reduction.

### 2.2.2 Extreme Programming (XP)

Extreme programming (XP) is an Agile Project Management Methodology that was invented by Kent Beck in March 1996. It has evolved since its release in 1996 and now it embraces five values that are reflected through its principles and practices to guide software development.

The five values of XP are (Beck and Andres, 2005):

- **Communication** – considered as that which matters most in the team software development. It is important for creating a sense of team and effective cooperation.
- **Simplicity** - refers to a team mindset of providing the simplest solution that solves the problem at hand. This does not mean the solution needs to be simplistic but rather, as much as possible, provide the simplest solution that works.
- **Feedback** – refers to the attitude of being satisfied with improvement rather than expecting instant perfection, using feedback to get closer and close to the team goals.
- **Courage** – is effective action in the face of fear. It can take many forms such as bias to action (doing something about a known problem) or as patience (as in waiting until the problem fully emerges before providing a solution for it)
- **Respect** – team members need to care about each other and about what they do (their project). Respecting others' worthiness and their contribution as human beings and as members of the software team is very important.

XP has several principles that more concrete than the values to guide human behavior. The principles of XP are Humanity, Economics, Mutual Benefit, Self-similarity, Improvement, Diversity, Reflection, Flow, Opportunity, Redundancy, Failure, Quality, Baby Steps, and Accepted Responsibility (Beck and Andres, 2005).

The goal of XP is to improve software quality and responsiveness to changing customer requirements. XP improves a software project in five essential ways: Planning, Managing, Coding, Designing and Testing (Kumar and Bhatia, 2012).

XP also has several practices that are subdivided into two categories. Practices are what the members of an XP team do day-to-day.

## Primary Practices of XP

The primary practices of XP can give the team immediate improvement and teams can start safely with any one of them.

- **Sit Together** – an open space big enough for the whole team.
- **Whole Team** – the team needs to have people with all the skills and perspectives necessary for the project to succeed, i.e. cross-functional team rather than just software development professionals.
- **Informative Workspace** – making the workspace about the team's work and an interested observer should be able to walk into the team space and get a general idea of how the project is in fifteen seconds.
- **Energized Work** – team members work as many hours as they can be productive and only as many hours as they can sustain.
- **Pair Programming** – write all production programs with two people sitting at one machine. Set up the machine so that the partners can sit comfortably side by side.
- **Pairing and Personal Space** – respect personal space for both partners with what they are comfortable about.
- **Stories** – plan using units of customer-visible functionality i.e. user stories.
- **Weekly Cycle** – plan works a week at a time and have weekly progress updates with the client and allow the client to select what will be done in the coming week.
- **Quarterly Cycle** – plan quarterly works and evaluate project progress on a quarterly basis in terms of alignment with larger goals.
- **Slack** – include some minor tasks that can be dropped if the team runs out of time.
- **Ten-Minute Build** – be able to automatically build and run all of the tests in 10 minutes.
- **Continuous Integration** – integrate and test changes after no more than a couple of hours.
- **Test-First Programming** – write a failing automated test before changing any code.

- **Incremental Design** – invest in the design of the system every day.

### Corollary Practices

The corollary practices can selectively be implemented by the team after first mastering the primary practices above.

- **Real Customer Involvement** – make people whose lives and business is affected by the system a part of the system.
- **Incremental Deployment** – when replacing a legacy system, gradually take over its workload beginning very early in the project.
- **Team Continuity** – keep effective teams together.
- **Shrinking Teams** – as a team grows in capability, keep its workload constant but gradually reduce its size.
- **Root-Cause Analysis** – every time a defect is found after deployment, eliminate the defect and its cause. This will ensure not just that this one defect won't ever recur, but that the team will never make the same kind of mistake again.
- **Shared Code** – share the code so that anyone on the team can improve any part of the system at any time.
- **Code and Tests** – maintain only the code and the tests as permanent artifacts and generate other documents from the code and tests.
- **Single Code Base** – let only one code stream to be used by the team. Developers can create temporary branches but never let it live longer than a few hours.
- **Daily Deployment** – put new software into production every night.
- **Negotiated Scope Contract** – write contracts for software development that fix time, costs and quality but call for an ongoing negotiation of the precise scope of the system.

Extreme programming (XP) methodology was found to be the most popular agile methodology about a decade ago (Livermore, 2007), however, the Scrum methodology has been found to be more popular in the most recent surveys (CollabNet VersionOne, 2019).

### 2.3 Success and failure factors in Agile software projects

Based on a survey conducted among Agile professionals from 109 projects in 25 countries around the world, Chow & Cao (2008) have identified 6 factors that positively contribute to the success of an Agile project. These factors along with the attributes that define each factor are given below according to their level of importance (Rank) where the earlier factor is more important than those following it in the table below.

Table 2.3.1: Summary of factors and attributes for the success of Agile methods (source: Chow & Cao (2008))

Rank	Factor	Attributes
1	Delivery strategy	<ul style="list-style-type: none"> <li>• Regular delivery of software</li> <li>• Delivering the most important features first</li> </ul>
2	Agile software engineering techniques	<ul style="list-style-type: none"> <li>• Well-defined coding standards upfront</li> <li>• Pursuing simple design</li> <li>• Rigorous refactoring activities</li> <li>• The right amount of documentation</li> <li>• Correct integration testing</li> </ul>
3	Team capability	<ul style="list-style-type: none"> <li>• Team members with high competence and expertise</li> <li>• Team members with great motivation</li> <li>• Managers knowledgeable in agile</li> <li>• Managers who have adaptive management style</li> <li>• Appropriate technical training to team</li> </ul>
4	Project management process	<ul style="list-style-type: none"> <li>• Following agile-oriented requirement management process</li> <li>• Following agile-oriented project management process</li> <li>• Following agile-oriented configuration management process</li> <li>• Good progress tracking mechanism</li> <li>• Strong communication focus with daily face-to-face</li> </ul>

		meetings <ul style="list-style-type: none"> <li>• Honoring a regular working schedule</li> </ul>
5	Team environment	<ul style="list-style-type: none"> <li>• Collaboration of the whole team</li> <li>• Coherent self-organizing teamwork</li> <li>• Projects with small team</li> <li>• Projects with no multiple independent teams</li> </ul>
6	Customer involvement	<ul style="list-style-type: none"> <li>• Good customer relationship</li> <li>• Strong customer commitment and presence</li> <li>• Customer having full authority</li> </ul>

While the above factors are important for a successful implementation of an ASDM, the authors have also noted that the factors listed below are associated with failures of ASDM based on studies that have looked into the lessons learned from past failed projects.

*Table 2.3.2: Failure factors for ASDMs (source: Chow & Cao (2008))*

Dimension	Factors
Organizational	<ol style="list-style-type: none"> <li>1. Lack of executive sponsorship</li> <li>2. Lack of management commitment</li> <li>3. Organizational culture too traditional</li> <li>4. Organizational culture too political</li> <li>5. Organizational size too large</li> <li>6. Lack of agile logistical arrangements</li> </ol>
People	<ol style="list-style-type: none"> <li>7. Lack of necessary skill-set</li> <li>8. Lack of project management competence</li> <li>9. Lack of teamwork</li> <li>10. Resistance from groups or individuals</li> <li>11. Bad customer relationship</li> </ol>
Process	<ol style="list-style-type: none"> <li>12. Ill-defined project scope</li> <li>13. Ill-defined project requirements</li> </ol>

	<ul style="list-style-type: none"><li>14. Ill-defined project planning</li><li>15. Lack of agile progress tracking mechanism</li><li>16. Lack of customer presence</li><li>17. Ill-defined customer role</li></ul>
Technical	<ul style="list-style-type: none"><li>18. Lack of a complete set of correct agile practices</li><li>19. Inappropriateness of technology and tools</li></ul>

## 2.4 Empirical literature review of the study

Regassa, Bass and Midekso (2017) conducted a study on the state of adoption of agile methods in small software firms in Ethiopia. The researchers conducted face-to-face interviews with 17 software professionals which were selected using a snow-ball sampling technique from 7 software companies. The analysis of the data has showed that Agile practices are used to encourage user participation and clarify requirements. It was also found that many projects undertaken by the professionals are for government clients that mandate extensive requirements and design documentation which must be approved prior to delivery working software that complies with predefined delivery schedules.

Another study on adoption of the Agile SCRUM method by an NGO project in Ethiopian was conducted in 2018 by Bizuayehu and Mamo which has found a moderate level of agile process adoption by the organization under study and has indicated there is a link between the level and maturity of the SCRUM method used and the project's overall success (Bizuayehu and Mammo, 2018).

There is a shortage of literature about the state of adoption of agile methods within the Ethiopian context. Besides the two research papers quoted above , the author was not able to find research articles on implementation or on attempts to adopt an Agile Project Management method in Ethiopia.

From the international literature of empirical studies on agile adoption, an empirical study conducted in 2009 in Sri Lanka to determine the factors that lead to the success of software development projects that want to adopt ASD practices have found that among 14 hypothesized factors, 9 have been found to be significantly related to the success of the project; these factors are (Misra, Kumar and Kumar, 2009) :

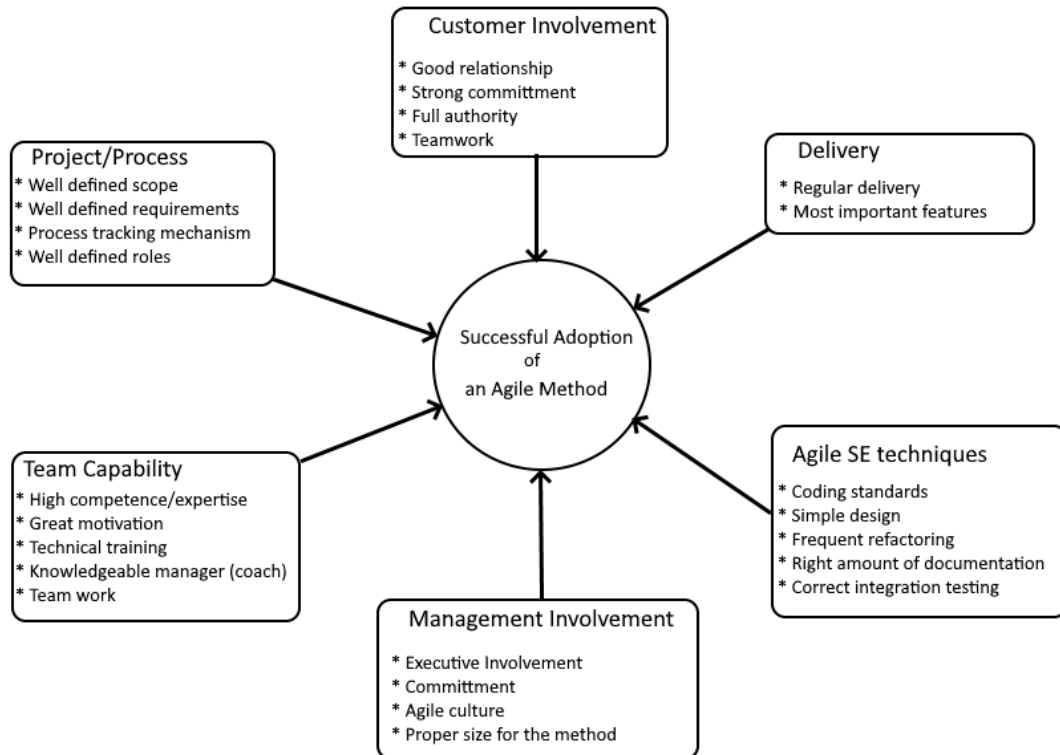
1. **Customer satisfaction** – highly motivated customer
2. **Customer collaboration** - continuous support and willingness to actively cooperate on the project.
3. **Customer commitment** – the most important customer-related factor for success
4. **Decision time** – fast decision making by the team is an enabler to agility

5. **Corporate culture** – organizational cultures that are adaptive to change should be supportive of working in a collaborative environment.
6. **Control** – plans, and qualitative control prepared and monitored by the team rather than external managers.
7. **Personal characteristics** – honesty, collaborative attitude, sense of responsibility, readiness to learn and work with others is as important as experience and competency.
8. **Societal culture**
9. **Training and learning** – emphasis on continuous learning and members of the team should be eager to share information with one another. Mentoring and professionally guided discussions were more useful in ASD than formal training, so that tacit knowledge could be transferred between individuals.

A study conducted in the South African provide banking industries on the exploration of usage and acceptance of agile methodologies by individuals working in the organizations as found out that factors known as Facilitating Conditions and Behavioral Intentions to be predictors of agile software development method in the banking sector (Mudarikwa and Grace, 2018). Facilitating conditions are the degree to which an individual believes that organizational and technical infrastructure exists to support of the of agile methodologies while Behavioral intention refers to how hard an individual tries to perform a behavior or how much effort an individual exerts in order to perform a behavior.

## 2.5 Conceptual framework

Based on the above theoretical and empirical literature review, the following conceptual framework has been developed for this study.



*Fig. 2.5.1 Conceptual Framework*

## **CHAPTER 3: RESEARCH METHODOLOGY**

### **3.0 Introduction**

This section consists of research design, data sources and data collection techniques, study population and sampling method, data collection procedures and instruments, research approach, methods of data analysis and interpretation.

### **3.1 Research design**

This undertaking is cross-sectional research of mainly of descriptive type. It attempts to understand the current level of the adoption of the agile methodology at private banks in Ethiopia through questionnaire and personal interview of key informants on the topic at randomly selected banks and through the distribution of questionnaires to team managers and the members of software development teams.

### **3.2 Data sources and data collection techniques**

The sources of data for this research are the private banks in Ethiopia and data has been obtained from software development team managers and the members of software development teams (developers) at the banks. The data from the developers was collected using a self-administered questionnaire (see part 1 of appendix A) the data from the software team managers were collected using a questionnaire that is partly self-administered and included open-ended questions (see part 2 of appendix A) which were then discussed with the data collector to clarify responses to these questions.

The questions used in the questionnaire were adapted from the following two sources given below. The first source uses multiple choice questions to identify the APM framework and practices used by the organization and the second source uses open ended questions to identify the issues that the team has come across while implementing agile.

- A survey that was conducted by Scott Ambler and Associates in 2011 to find out the agile adoption strategies (Ambler, 2011) and
- An empirical study of Agile methods in Ethiopia (Regassa, Bass and Midekso, 2017)

### 3.3 Study population and sampling method

There are 16 private banks that are fully operating in Ethiopia at the time of the survey. The study general population is all the 16 private banks in the country. As it is not practically possible within the time available to survey all the banks in the population for this project's scope of the study, it is imperative that representative samples be drawn from the population using a reasonable sampling technique.

These banks have been established within the last three decades with the oldest private bank been established in 1994 GC and the latest being in 2012 (NBE, 2019).

To obtain a sample that represents the whole population with equal representation of older and relatively newer banks alike, a stratified random sampling technique has been used in this study. Stratified random sampling (proportional or quota random sampling) involves dividing the population into homogeneous subgroups and taking a simple random sample in each group (Trochim, Donnelly and Arora, 2016). By the same authors' definition, a homogeneous subgroup is a group in which all the members are relatively similar. In this particular study, the 16 private banks were categorized into 3 non-overlapping subgroups (strata) by using their establishment years. This is with the assumption that, due to their long years of experience, the banks that were established earlier will have a more mature software development process as compared to those that were established later and hence will be more similar or homogeneous to each other regarding their adoption level of the agile software development methodology.

Hence, grouping the banks into 3 strata based on the number of years after establishment and drawing samples from each subgroup provides a more representative sample of the total population by removing the possible bias that could have resulted from selecting more banks with the same number years of experience while excluding others.

Below is the list and number of banks that were established in each decade:

*Table 3.3.1: Sample clusters and sample size*

Established	Banks	Count	Samples
1990-1999 G.C.	Awash Bank Dashen Bank Abyssinia Bank Wegagen Bank United Bank Nib International Bank (NIB)	6	3
2000-2009 G.C.	Cooperative Bank of Oromia (COOP) Lion Bank Oromia International Bank Zemen Bank Bunna International Bank	5	3
2010 G.C. and after	Berhan Bank Abay Bank Addis International Bank Debub Global Bank Enat Bank	5	3

The study selected 3 samples from each category using a random sampling technique where each item in the group had an equal chance of being included in the study. The total sample size is 9 banks.

### **3.4 Data collection procedures and instruments**

The data collection was undertaken using a paper-based questionnaire and it resulted in obtaining qualitative data from participants. As this is a case study of specific participants that includes several open-ended questions, the most important part of the information is of qualitative type.

### **3.5 Research approach**

The design used in this research is a qualitative method of data analysis. This research employs two survey instruments in the form of questionnaires where one was administered to members of the software development teams and includes multiple-choice questions as well as statements regarding their development practice and their teams, against which developers state their general agreement or disagreements.

The other questionnaire was filled by the heads of the software development teams at the banks and includes multiple-choice questions along with some open-ended questions. The open-ended questions were also discussed with the data collector in an interview session with the software team managers in the organizations.

### **3.6 Data analysis method**

The qualitative data were obtained through the questionnaires and open ended interviews with the development team managers were categorized, aggregated and were analyzed by descriptive statistics using tables, frequency charts, and percentages.

### **3.7 Validity and reliability of data**

The data collection instruments used in the study were tested on server IT professionals and the responses have been found to produce consistent responses between professionals of similar experience.

### **3.8 Ethical considerations**

The data collection for this study was obtained from the participants after receiving their informed consent. Organization names have been made anonymous in this report and no data that can possibly be used identify organizations or individuals have been collected.

### **3.9 Study location**

This study was conducted in Addis Ababa as all the private banks have their headquarters in Addis Ababa.

## CHAPTER 4: DATA ANALYSIS AND PRESENTATION

### 4.1 Introduction

The total number of private banks that were reached out for the survey data collection was 9 which account for 56% of the population. With 2 banks not being able to provide the information to the study due to various reasons, it was possible to collect responses from 7 private banks (about 44% of the total population).

The participating banks have been identified using letters and designated as Bank A, Bank B, Bank C, etc... to ensure the anonymity of the information provided by the banks as the main objective of the study is to identify their agile adoption level and challenges faced in doing so by the banks generally and not specific to any particular bank.

At the end of the data collection work, it was possible to collect totally 31 responses from the survey participants. Out of these, one questionnaire response has been discarded due to the lack of responses to the last section of the questionnaire distributed to the development team members. Hence the total number of complete responses received and included in this analysis is 30. Among these are 7 are managers (CIO, Deputy CIO, and Development Team heads) while 23 are members of the software development teams at the banks.

The table below shows the number of complete questionnaire responses from survey participants at each bank.

*Table 4.1.1 Total number of complete responses from each bank.*

<b>Bank</b>	<b>CIO / Managers</b>	<b>Developers</b>	<b>Total number of complete responses</b>
Bank A	1	2	3
Bank B	1	3	4
Bank C	1	3	4
Bank D	1	3	4
Bank E	1	5	6
Bank F	1	2	3
Bank G	1	5	6
<b>TOTAL</b>	<b>7</b>	<b>23</b>	<b>30</b>

## 4.2 Characteristics of respondents

The general characteristics of the participants in terms of their various attributes such as team roles, educational level and years of experiences are shown in the tables below:

*Table 4.2.1 Number of participants by team roles*

Team role	Number of responses
Business Analyst	1
Developer	22
Head of Software Development	5
Head of ICT / CIO	2
<b>TOTAL</b>	<b>30</b>

*Table 4.2.2 Number of participants by education level*

Education level	Number of responses
Degree	20
Master's degree	10
<b>TOTAL</b>	<b>30</b>

*Table 4.2.3 Number of participants by years of experience*

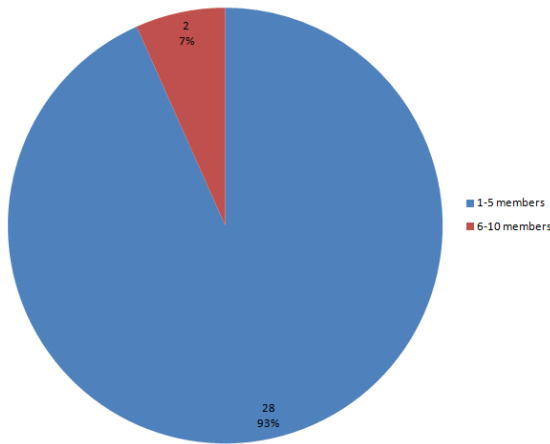
Team role	Number of responses
0 – 2 years	2
2 – 5 years	4
5 – 10 years	15
10 – 15 years	5
15+ years	4
<b>TOTAL</b>	<b>30</b>

*Table 4.2.4 Team communication frequency*

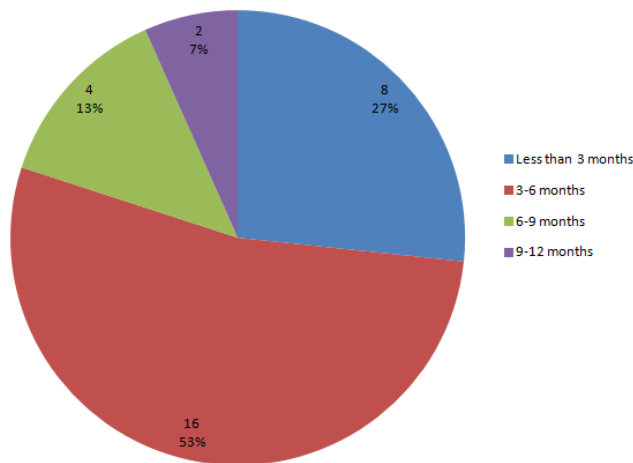
Meeting frequency	Number of responses
Weekly	7
Every two weeks	3
Irregular & ad-hoc meeting	20
<b>TOTAL</b>	<b>30</b>

*Table 4.2.5 Team communication modes*

Meeting frequency	Number of responses
Face-to-face communication	28
Email based communication	2
<b>TOTAL</b>	<b>30</b>



*Fig. 4.2.1 Number of participants by their team size*



*Fig. 4.2.2 Number of participants and project durations*

Summary of profiles of the survey participants:

- Significant majority of the survey participants (80%) have 5 or more years of experience in the IT field.
- About 66% of the respondents are degree graduates and the remaining 33% are master's degree holders.
- The majority of the developers (93%) work in a team of 5 members or less.
- The majority of the developers (53%) work in projects that last between 3-6 months.
- Face to face communication is by far the most dominant mode of communication between team members and with clients.

### 4.3 Data analysis

By its nature, it is difficult to analyze qualitative data using statistical methods. The approach followed in this study is to code, categorize and aggregate the individual data obtained from respondents in the written or verbal responses and express them in percentages so to see the overall picture and to come to relevant conclusions.

#### 4.3.1 The current software development methodology

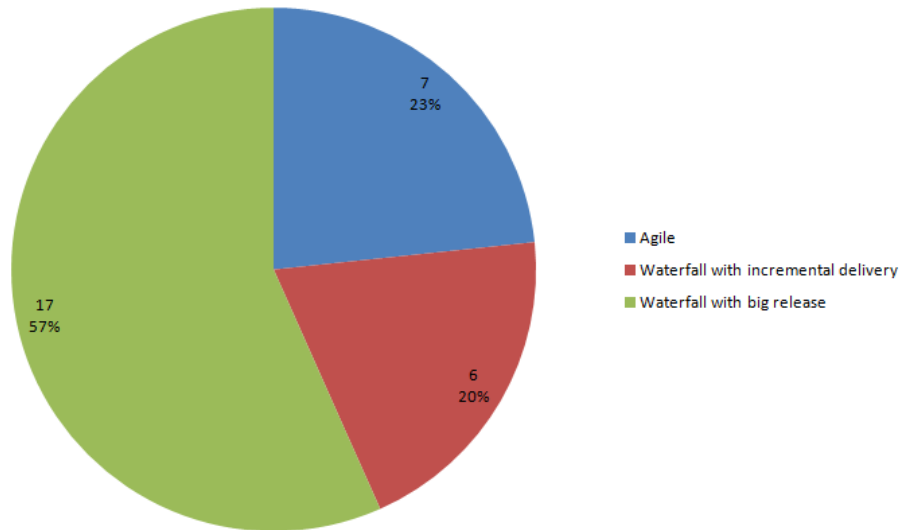
*Table 4.3.1 Software development methodology responses by managers and individuals*

<b>SDM</b>	<b>Managers</b>	<b>Individual responses</b>
Agile (Scrum)	2	7
Traditional	5	23
<b>TOTAL</b>	<b>7</b>	<b>30</b>

At organizational level, 2 banks have indicated to have started using the Scrum agile project management out of the total 7 banks.

This shows that only 28% of the total surveyed has started the process of adopting the Scrum agile method which is a lower proportion of the total considering the agile methods have been introduced to the world more than 18 years ago as if today.

Considering the responses from individuals, from the total 30 respondents, 7 respondents (23%) identify their current SDLC model to be the Scrum Agile Project Management Methodology. The remaining 23 respondents (77%) identify their development method as the TSDM. Among these, 6 respondents (20%) identify their methodology as Waterfall with incremental delivery approach while 17 respondents (57%) identify their methodology as Waterfall with big release at the end of the project.



***Fig. 4.3.1 Participants overall perception of their current software development methodology***

The data analysis shows the following points with regards to the participants.

- 77% of the respondents characterize their software development method as the traditional Waterfall method while the 23% characterize this SDM as Agile.
- Scrum is the most commonly used agile method at the software development teams at private banks in Ethiopia.
- 33% of the respondents have indicated that in their previous software development experiences, they have come across using an agile methodology mainly Scrum but also XP, FDD as well as DSDM.
- For 28 respondents (93%), face-to-face communication is by far the dominant form of communication between the developers, client and other stakeholders during the software development process. With the remaining 7% using Email as the primary form of communication.
- Based on the responses to the open-ended question to the managers concerning the bank's experience in using the Agile method, the responses show that from the 7 banks, only two banks (28%) have attempted to use the agile method while the remaining 5 (72%) are using the waterfall method with incremental delivery or with the big release at the end of the development process.

### 4.3.2 The current level of adoption of APM in the industry

The aggregated responses with regards to the current software development methodology, as given in the previous section shows that the extent of the adoption of an APM method in the industry is 2 out of the 7 participant organizations in the survey making the total percentage of 28%.

*Table 4.3.2.1: Software development methodology at each private bank*

Methodology	Private bank							TOTAL	%
	A	B	C	D	E	F	G		
Traditional method (waterfall)	X	X	X	X		X		5	72
Agile Scrum method					X		X	2	28
<b>TOTAL</b>								<b>7</b>	<b>100</b>

### 4.3.3 Current challenges against adopting an agile methodology

After classification and aggregation of the individual responses to the multiple-choice and open-ended questions presented to the development team managers, the challenges that were faced by the managers and the teams are given below. Based on the apparent source of the challenges from the point view of the development team, the challenges are divided into three broad categories as: Client-related, Management-related, and Software development team-related. The identified issues were then prioritized in accordance with the degree to which they were affecting the software development and delivery process at each bank. The challenges in each category are given below and listed in their order of importance from high to lower.

*Table 4.3.2.3: Client- related challenges at each private bank*

Challenge	Private bank							TOTAL
	A	B	C	D	E	F	G	
Lack of timely feedback from the client	X	X	X	X	X	X	X	7
Incomplete requirements provided to the development team.	X	X	X	X	X	X	X	7
Inconsistent understanding of requirements and software features among members of the client unit.	X	X	X		X	X	X	6
Resistance to change and tendency to go back to old ways of doing things.		X	X	X	X	X	X	6
Lack of motivation from clients for collective ownership of systems throughout the SDLC.	X	X	X	X	X	X		6
Client representative not having the expected knowledge on product requirements and not sufficiently cooperative and communicative to the software development team.		X		X		X	X	4
Fear of being replaced with software systems or reduction workforce due to an automated work process.	X			X	X			3
Client focal person’s concern on being left out of opportunities in their “home” unit.			X		X	X		3

*Table 4.3.2.3: Development team-related challenges at each private bank*

Challenges	Private bank							TOTAL
	A	B	C	D	E	F	G	
Developers handle helpdesk support request from users utilizing the systems.	X	X	X	X	X	X	X	7
Lack of experience in working in teams.	X	X	X	X	X		X	6
Development team leader working as a full-time developer.	X	X	X	X	X	X		6
The departure of experienced staff members amidst an ongoing work	X	X	X	X	X	X	X	7
A reduced interest of individual developers to work in a team environment and preference to solely work on a system.		X	X	X	X			4

*Table 4.3.2.4: Organizational/Management related challenges at each private bank*

Challenges	Private bank							TOTAL
	A	B	C	D	E	F	G	
Setting short project deadlines.	X	X	X	X	X	X	X	7
Allocation of insufficient funds for training and other resources and tools such as software and hardware.	X		X	X	X	X	X	6
Lack of organization-wide definition of standard tools and frameworks.		X	X	X	X	X	X	6
De-prioritizing of ongoing projects out of work.	X		X	X	X	X	X	5
ICT Development strategic plan for improvement of software process management.	X	X	X	X	X	X		5

## Client-related challenges

1. **Lack of timely feedback** - delayed provision of testing results and acceptance of systems and features that have been completed for a client unit in the organization.
2. **Incomplete requirements provided to the development team** - client requirements not being provided to the development team at the expected time and creating unexpected gaps in the development workflow. This can result in delayed delivery of software systems, increased costs and can possibly affect project quality.
3. **Inconsistent requirements and software features among members of the client unit** - lack of a consistent understanding of the system requirements from the client unit members - lack of a single and coherent set of requirements and individuals having different expectations on how the system should work.
4. **Resistance to change and tendency to go back to old ways of doing things** - reluctance to cooperate with the software development team in the middle or at the end of the project and tendency to drag the system development or deployment process so as to continue the current way of doing things.
5. **Lack of motivation from clients for collective ownership of systems throughout the SDLC** - lack of motivation to finalize and employ the system and tendency to see the system as the sole responsibility of the development team and shifting into blame whenever difficulties arise in the whole process.
6. **Client focal person's concern on being left out of potential opportunities in their "home" unit** - client representative to the software development team sometimes being concerned or having doubts of being left out of future promotions and opportunities at their "home" department as they spend more and more time with the software development team.

7. **Fear of being replaced with software systems or reduction of unit/department workforce due to an automated work process** - underlying fear by client unit team members and managers that the employment of a software system will eventually result in the reduction of their workforce resulting in resistance to optimally cooperate with the development team. In some organizations, a reduced workforce in a department can possibly lead to the reduction of the status of the department into a section or some other lesser level in the organization.
  
8. **Client representative not having the expected knowledge on product requirements and not sufficiently cooperative and communicative to the software development team.** The client unit not able or willing to provide their most knowledgeable and skilled team member to work with the software team. A tendency to allocate an individual with lower communication skills or allocating someone who is known for not being very productive within the client team.

## **Development team related challenges**

1. **A reduced interest of individual developers to work in a team environment and preference to solely work on a system** - team members tend to prefer to work alone on a single system from the beginning to the end of the project and less interest to cooperate with other team members.
2. **Developers handle helpdesk calls from users utilizing the systems they developed** - developers also handle helpdesk calls from clients from various branch offices and perform troubleshooting on the systems they developed previously while at the same time working on new systems.
3. **Lack of experience in working in teams** - lack of previous experience of working in teams.
4. **Development team leader working as a full-time developer** - team leaders spend full time on development and maintaining systems with less time left to facilitate the works of their team members.
5. **Departure of experienced staff members amidst an ongoing work** - the departure of experienced programmers with a deeper knowledge of existing systems leave the team in a difficult situation and finding a suitable replacement takes a long time until existing staff takes over the systems and be able to provide support on the systems maintained by the outgoing programmer.

## **Descriptions of organizational/management related challenges**

1. **Setting short project completion deadlines** - higher management setting project completion deadlines based on organizational priorities resulting in exerting more pressure on development team members.
2. **Allocation of insufficient funds for training and other resources such as software and hardware tools** - funds allocated to the development team not sufficiently allowing for staff-development training that may be needed to upgrade the skills of the programmers or resources needed in the procurement of software and hardware such as server computers for the purposes of installing team collaboration applications such as Git or TFS.
3. **De-prioritizing of ongoing projects out of work** - management putting ongoing projects on-hold while in the middle of the development process.
4. **ICT Development strategic plan for improvement of software process management** - lack of clearly stated overall ICT Development strategic plan to improve the organizations' software development processes and a tendency to keep using existing traditional methods.
5. **Lack of organization-wide definition of standard tools and frameworks** - lack of organization-wide definition of standard software development tools, frameworks including coding conventions, programming languages, database platforms, coding standards/conventions, and unit test requirements.

#### 4.3.4 Summary of developer perspectives on Agile and experience with agile tool

Table 4.3.4.1 summary of developer responses to questions in section II

Statement	Yes	No
1. I am familiar with Agile Project Management methods and practices	65%	35%
2. In general, the organization has supported and encouraged the use of agile methods in the development team	26%	74%
3. Adopting an agile method will make the team more effective.	82%	18%
4. I have participated in a training program (formal or informal) in an Agile Project Management method such as Scrum, XP, FDD, ...	39%	61%
5. There is a member of your team who has experience in agile project management and acts as a coach to the team.	30%	70%
6. There is someone in our team that represents the client's interest and explains issues from the client point of view and prioritizes issues on behalf of the client.	43%	57%
7. The team holds regularly scheduled meetings between team members and customer representative.	43%	57%
8. At the end of our delivery cycle (game/sprint), the team reflects on the challenges faced by the team during the delivery cycle and suggests remedies to ensure the next cycle will be improved.	48%	52%
9. The team welcomes modifications of requirements during and after the implementation of a feature or software functionality.	96%	4%
10. User stories are always estimated before a developer can start working on them.	57%	43%
11. Our team leader/scrum master is empowered to remove impediments that are faced by our team members during the cycle.	52%	48%
12. My team employs an issue tracking system (JIRA/Atlassian, Redmine, GitHub, GitLab, ...) to facilitate team communication and document requirements, and track progress	39%	61%
13. My team employs a source code management (Git, TFS, SVN,) to facilitate code collaboration between developers.	39%	61%
14. The team conducts Code Reviews on a developer's work before it can be deployed to the production server.	34%	66%
15. My team employs automated UI tests to ensure requirements are met at handover and prevent unintentional changes in the future.	34%	66%
16. My team employs a CI/CD server to prevent regression and ensure new code commits to comply with the standards (e.g. Unit Tests, Coding Standards)	34%	66%

- A significant proportion of the developers (65%) believe they are familiar with an agile method with 39% of developers being involved in some kind of training or past experience using an agile method.

- Low overall support and encouragement provided to software developers to improve the software development process (26%)
- Provision of trainings to developers on APM is minimal (39%)
- Availability of a team member with APM experience is at 30%.
- About 82% of the developers agree that adopting an agile method will help their team and organization more effective in the software development process. On the other hand, only 26% of the developers agree that the organization or the management has provided or encouraged them to use an agile method in the software development process.
- There is a limited usage of team collaboration tools among the software developers:
  - Only 39% have indicated to have used a version control system in the projects.
  - Only 39% have indicated to have used a project management system to document requirements and report their work progress with the clients or the management.
  - Only 34% of developers employ code reviews before the product is deployed to a production environment.
  - Only 34% of developers employ automated user interface tests to ensure requirements are met by the system.

## **CHAPTER 5: SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS**

This chapter presents the summary of findings, conclusion and recommendations of the study. Hence, it reviews the problem of the research and concludes the findings with regards to the objectives of the study. The recommendations that focus on how the challenges identified by the research could be addressed are also included in this chapter. The limitations of the study and suggestion for further research are also included at the end of this chapter.

### **5.1 Summary of findings**

The findings show that among the 7 participating banks in the survey, only 2 of the development team managers (28%) indicated to have started using the Scrum agile project management methodology. The remaining 72% identify their methods to be the traditional waterfall method with either full release at the end of the SDLC or using the incremental delivery approach. This closely agrees with the development team members' response that shows 77% of developers identify their development methodology to be TSDM and 23% identifying their methodology as APM. Hence the adoption level is can be said to be low as the majority of the banks have not started to adopt the agile methodology in the software development process.

The two organizations that have started to adopt the Scrum method are also struggling to make the method work for them smoothly due to a number of challenges they have faced while in the adoption process. On the other hand, the banks that currently use the traditional methodology have indicated they plan to adopt an agile method in the future but they do not have a specific timeline at to move to implementing an agile method.

The major challenges faced by the private banks in their software development process are presented in three categories as customer-related, team-related and management-related.

Customer-related challenges include inconsistent requirements, lack of timely feedback, resistance to change and lack of full commitment to the process.

Team-related challenges include reluctance to work in a team, lack of experience, handling of helpdesk requests while being engaged in software development and departure of experienced team members.

Management-related challenges include setting unreasonably short deadlines, de-prioritization of ongoing projects, not specifying standard software development tools and frameworks, absence of general long-term ICT Development strategic plan specific to the improvement of the software development process.

Developers and team managers are familiar with an agile method from either theoretical or practical encounter to the methodology. There is a high degree of awareness by developers about the fact that agile is a beneficial methodology for the development team to incorporate in their software development processes.

Also there is minimal involvement and championing of higher management in the overall process of the adoption of an agile method in the banking industry with only 26% of the survey participants feeling the encouragement and supporting from the management. The predominant software development process is the traditional waterfall method with minimal team based effort; with a single developer being assigned to work on projects throughout the life of a system.

## 5.2 Conclusion

The objective of the research was to find out the most commonly used software development methodology and the current adoption level of APM practices at the private banks in Ethiopia. This research also aimed at identifying the major challenges that confront software development team managers from adopting an APM.

The study has collected and analyzed data from 7 of the 16 private banks in Ethiopia that were selected using a stratified random sampling method from 3 categories of banks that were assigned to each category based on the decades of their respective establishment years. A total response of 30 completed questionnaires was obtained and analyzed using descriptive statistics.

The result shows that the most commonly used software development methodology is the traditional waterfall method and the current adoption level is at its early stage with only 2 participant organizations having actually taking the first steps and piloting the process of implementing the Agile Scrum method. There are several challenges that hinder teams from moving into an agile method which were categorized into three categories: client-related, team-related and management related.

### 5.3 Recommendations

Due to delayed project delivery, more rework as a result of modifications to account for unexpected changes to project requirements, lower customer satisfaction associated with the traditional waterfall method of software development and delivery, development teams at the banks will improve their output and customer satisfaction by adopting one of the Agile software development methods such as the Scrum methodology

To successfully adopt an agile method by their software development teams, a previous study has found that the following factors have an impact on the process of adopting an agile software development methodology in an organization (Livermore, 2007) :

- **Training** - provide training to the development team and the client members on agile values, principles, tools and techniques.
- **Management involvement** – involve management in the planning and implementation stages to ensure continue support while working with the clients and other stakeholders involved in the process.
- **Access to external resources** – enable team members to easily access resources resource that will facilitate the agile implementation process.

Familiarize developers to various software project management and collaboration tools:

- Provide training to familiarize developers to use collaboration tools that are crucial for successful collaborative teamwork between the members of the software development team.
  - **Version Control Systems** – allows the development team to share and centrally access a project codebase to all team members easily. This facilitates instant sharing of updated code among team members and code changes will also be traceable to the individual how makes the changes. Inclusion of automated tests and coding standards as team requirements can easily be enforced through the employment of continuous integration (CI) server that compiles

and runs the code to verify compliance to the expected test coverage and coding conventions. There are several version control system which can be used freely such as Git, Mercurial, and SVN.

- **Project Management Systems** – these allow the team to have a central location that is accessible by the whole team and provide detailed information on the project requirements and progress of tasks assigned to each developer. It also facilitates team communication and clarification of requirements through comments on user stories or tasks. Helps managers to track progress and identify progress gaps and to identify and resolve issues that may hinder progress at an early stage.
- Define and adhere to organizational standards on system development tools including:
  - A standard programming language or languages for various types of system development work to be undertaken in the organization.
  - Database management systems
  - Development and automated tests frameworks and IDEs for the developers in the organization.
- Encourage a team of developers to be involved in the development process of all their software systems that the organization intends to use for a considerable length of time. This avoids knowledge silos where a single person knows the ins and outs of systems making system upgrades or further developments dependent on the availability of a single person.

#### **5.4 Limitations and suggestions for further study**

This study was conducted on local software development teams and their managers at seven private banks in Ethiopia. Hence the information obtained is from the perspective of the developers and the managers at these banks. It does not take into account the perspectives of other stakeholders such as the software clients and the higher management in the organization.

This study also was able to collect data from 7 banks out of the total 16 provide banks operating in the country at the time of the study. This provides data for less than 50% of the population and given the small size of the overall population, a census study is required to obtain a reliable measure of the adoption level of banks in the industry. Due to time and resource constraints, the census study was not possible to conduct in this study. A more accurate conclusion can be made by performing this study by looking in detail at all the 16 private banks' software development team's experiences of the process of adopting an agile method.

In addition, since the sources of information for this study are developers and development-team managers, a future research that includes the views of the higher management and the internal software clients on the issues raised in this study can provide helpful information for the improvement of the industry's overall software development processes and can be a good topic for further investigation to challenge or validate the results of this study.

## References

- Abera, H. B. (2015) 'Financial Performance of the Ethiopian Banking Sector', *International Journal of Science and Research (IJSR)*, 4(5), pp. 2743–2747.
- Abrahamsson, P. and Salo, O. (2002) 'Agile software development methods', *Relatorio Technico*, pp. 1 - 122
- Ambler, S. (2011) *Agile Adoption Strategies: November 2011 Survey Results*. Available at: <http://www.ambysoft.com/surveys/agileStateOfArt201111.html> (Accessed: 16 April 2019).
- Ambler, S. W. and Lines, M. (2012) *Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise*. Boston: Pearson Education, Inc.
- Arcidiacono, G. (2017) 'Comparative research about high failure rate of IT projects and opportunities to improve', *PM World Journal*, 6(2), pp. 1–10.
- Bano, M. *et al.* (2012) 'Causes of requirement change - A systematic literature review', *IET Seminar Digest*, 2012(1), pp. 22–31.
- Beck, K. *et al.* (2001) *Agile Manifesto*. Available at: <https://agilemanifesto.org/> (Accessed: 15 April 2019).
- Beck, K. and Andres, C. (2005) *Extreme Programming Explained: Embrace Change (XP series)*. 2nd Editio. Boston: Addison-Wesley.
- Bizuayehu, G. and Mammo, S. (2018) *Agile Software Project Management Practices and Success Rates of Monitoring and Evaluation Software Projects in the NGO Sector – the case of PRIME: a scrum software project*. Addis Ababa University.
- Boehm, B. and Turner, R. (2005) 'Management Challenges to Implementing Agile Processes in Traditional Development Organizations', *IEEE Software*, 22(5), pp. 30–39.
- Cardozo, E. S. F. *et al.* (2010) 'SCRUM and Productivity in Software Projects : A Systematic Literature Review', in *International Conference on Evaluation & Assessment in Software Engineering (EASE)*.

Chow, T. and Cao, D.-B. (2008) 'A survey study of critical success factors in agile software projects', *Journal of Systems and Software*, 81(6), pp. 961–971.

CollabNet VersionOne (2019) *13th Annual STATE of AGILE Report*. Available at: <https://www.stateofagile.com/13th-annual-state-of-agile-report/>. (Accessed: 20 April 2019)

Dybå, T. and Dingsøy, T. (2008) 'Empirical studies of agile software development: A systematic review', *Information and Software Technology*, 50(9–10), pp. 833–859.

Kumar, G. and Bhatia, P. K. (2012) 'Impact of Agile Methodology on Software Development Process', *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*, 2(4).

Lee, G. and Xia, W. (2010) 'Toward agile: An integrated analysis of quantitative and qualitative field data on software development agility', *MIS Quarterly: Management Information Systems*, 34(1), pp. 87–114.

Livermore, J. (2007) 'Factors that impact implementing an agile software development methodology', in *Proceedings 2007 IEEE SoutheastCon*. IEEE, pp. 82–86.

Masood, Z. A. and Farooq, S. (2017) 'The Benefits and Key Challenges of Agile Project Management under Recent Research Opportunities', *International Research Journal of Management Sciences*, 5(1), pp. 20–28.

Misra, S. C., Kumar, V. and Kumar, U. (2009) 'Identifying some important success factors in adopting agile software development practices', *Journal of Systems and Software*, 82(11), pp. 1869–1890.

Mudarikwa, G. and Grace, T. D. (2018) 'Agile System Development Methodologies Usage and Acceptance in South African Banking Firms: An Exploratory Analysis', in *Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists on - SAICSIT '18*, pp. 248–257.

N.Vithana, V., G. S. Fernando, S. and Kapurubandara, M. (2015) 'Success Factors for Agile Software Development A Case Study from Sri Lanka', *International Journal of Computer Applications*, 113(17), pp. 10–18.

- NBE (2019) *Banks Information page*. Available at: <https://nbe.gov.et/financial/banks.html> (Accessed: 5 April 2019).
- Oyong, S. B. and Ekong, V. E. (2019) 'An explorative survey of formal and agile software development methods', *Global Journal of Pure and Applied Sciences*, 25(1), pp. 71–79.
- Patel, C. and Ramachandran, M. (2009) 'Agile Maturity Model (AMM): A Software Process Improvement framework for Agile Software Development Practices', *Journal of Software*, 2(1), pp. 3–28.
- Regassa, Z., Bass, J. M. and Midekso, D. (2017) 'Agile Methods in Ethiopia: An Empirical Study', in *IFIP Advances in Information and Communication Technology*, pp. 367–378.
- Schaffer, S. C. (2013) *A brief introduction of Software Development and Quality Assurance Management*.
- Schwaber, K. (2004) *Agile Project Management with Scrum*. Microsoft Press.
- Sharma, S., Sarkar, D. and Gupta, D. (2012) 'Agile Processes and Methodologies: A Conceptual Study', *International Journal on Computer Science and Engineering (IJCSE)*, 4(05), pp. 892–898.
- Sidky, A. and Arthur, J. (2015) 'A Structured Approach to Adopting Agile Practices: The Agile Adoption Framework', *The Journal of Systems and Software*, (1), p. 338.
- Sutherland, J. (2014) *Scrum: The Art of Doing Twice the Work in Half the Time*. Crown Publishing Group.
- Trochim, W. M., Donnelly, J. P. and Arora, K. (2016) *Research Methods: The Essential Knowledge Base*. 2nd Editio. Delhi.
- Wysocki, R. K. (2019) *Effective Project Management: Traditional, Agile, Extreme, Hybrid*. Eighth Edit. Indianapolis: WILEY.

## **Appendices**

## Appendix A – Research Instruments

### Data Collection Questionnaire for Developers

Dear Participant,

My name is Asegid Debebe and currently, I am a final year student studying Master of Arts in Project Management at AAU, School of Commerce. I really appreciate that you are giving me your precious time to complete this questionnaire.

The main purpose of this questionnaire is to collect data for research that attempts to identify the current level of adoption of the Agile Project Management (APM) methods in Ethiopia in general and at private banks in particular. In addition to describing the current level of adoption of APM methods, it will also identify the major issues and challenges the private banks are coming across while going Agile and will recommend possible solutions to these challenges based on the existing body of knowledge and experiences and researches conducted in other countries. The research findings will be shared with the participating organizations so that it can be used as an input to improve their adoption processes in the future.

To keep participant responses anonymous, please do not write your name on this questionnaire. In addition, **I kindly request your cooperation and sincere response to the questions included in this questionnaire** so that the results of the undertaking will be realistic and useful.

**Note: The estimated time to complete this questionnaire is between 20 – 25 minutes.**

**Thank you for your time and kind cooperation!**

1. Which one of the following best describes your current role in your organization?
  - a. Software Developer
  - b. Software Tester
  - c. Software team leader
  - d. Scrum master
  - e. Product owner
  - f. Other: \_\_\_\_\_
  
2. What is your current level of education?
  - a. Diploma (2 years)
  - b. Degree
  - c. Master's degree
  - d. Ph.D
  - e. Other: \_\_\_\_\_
  
3. How many years of experience do you have in the IT profession?
  - a. 0 – 2 years
  - b. 2 – 5 years
  - c. 5 – 10 years
  - d. 10 – 15 years
  - e. 15+ years
  
4. Can you please indicate the size of your software development team?
  - a. 1 – 5
  - b. 6 – 10
  - c. 11 – 15
  - d. 16 – 20
  - e. 20+
  
5. How is the geographic distribution of your development team?
  - a. Co-located in the same office
  - b. Near (but not in the same office)
  - c. Far (located with no possibility of face to face communication)
  - d. Other: \_\_\_\_\_

6. How do you characterize your team's current software development methodology?
- Ad-Hoc (unstructured)
  - Waterfall method with big release at the end.
  - Waterfall method with incremental delivery
  - Agile method (Scrum, XP, FDD, ...)
  - Other: \_\_\_\_\_
7. If you responded Agile in the above question, which practice do you use?
- Scrum
  - Extreme Programming (XP)
  - Feature Driven Development (FDD)
  - Dynamic Systems Development Methodology (DSDM)
  - Crystal
  - Lean Software Development (LSD)
  - Other: \_\_\_\_\_
8. Have you been a part of a team that has employed Agile PM methods in your previous work experience (other than the current organization) in Ethiopia?
- Yes
  - No
9. If YES, to the above question, which method have you employed?
- Scrum
  - Extreme Programming (XP)
  - Feature Driven Development (FDD)
  - Dynamic Systems Development Methodology (DSDM)
  - Crystal
  - Lean Software Development (LSD)
  - Other: \_\_\_\_\_
10. In your current software development team, how do you collaborate/communicate with your team members, managers, customers, and other project stakeholders?
- Mainly face-to-face communication on an ad-hoc basis
  - Primarily through email
  - Formal, structured and regularly scheduled meetings
  - No regular communication as I work alone on a project with occasional telephone calls meetings with managers for progress update.
  - Other: \_\_\_\_\_

11. In your current software development team, how often do you meet with your team members, managers, customers, and other project stakeholders?
  - a. Ad-hoc or irregular meeting sessions upon request by the team lead or manager
  - b. Daily
  - c. Multiple times a week, but not daily.
  - d. Once a week
  - e. Once every two weeks
  - f. Once a month
  - g. Less than once a month
  
12. Typically, how long is the duration of your typical software development projects?
  - a. Less than 3 months
  - b. 3 – 6 months
  - c. 6 – 9 months
  - d. 9 – 12 months
  - e. 1 – 2 years
  - f. More than 2 years
  
13. How often do you and your team members need to report on the work progress of their respective user stories?
  - a. Ad-hoc or irregular progress updates upon request by the team lead or manager
  - b. Daily
  - c. Multiple times a week, but not daily.
  - d. Once a week
  - e. Once every two weeks
  - f. Once a month
  - g. Less than once a month
  
14. As a developer, how often do you Demo your work to the team and clients on user stories you completed?
  - a. Once a week
  - b. Once a month
  - c. Less than once a month
  - d. At the end the project
  - e. I don't do the demo, a designated person does it at the end of the project

15. As software development team, which of the following software code compliance standards and guidelines do you use (please mark all that apply?).
- a. No, we don't have these
  - b. Coding standards (StyleCop or similar for Visual Studio, PEP8 for Python)
  - c. Unit Tests
  - d. Test Scripts
  - e. Other: \_\_\_\_\_
16. What metric do you use to estimate the level of effort needed to complete user stories or features during software development?
- a. Hours
  - b. Days
  - c. Weeks
  - d. Story points
  - e. Other: \_\_\_\_\_
17. As a software developer, do you test the implementation of a feature yourself or is there a dedicated tester in your team for this purpose?
- a. I test my own implementation and confirm that it works and it is ready for release.
  - b. The tester tests and confirms it works before it can be released.
  - c. Other: \_\_\_\_\_
18. How often does your team's output be delivered / installed into the production server after the completion of your work?
- a. Weekly
  - b. Every two weeks or less
  - c. At least once a month
  - d. Every 3 months
  - e. Once every 6 months or so

**Please see the last section on the next page.**

**Part II: Please indicate your agreement or disagreement on the statements below with regards to your current software development team and practice**

Statement	Yes	No
1. I am familiar with Agile Project Management methods and practices	<input type="radio"/>	<input type="radio"/>
2. In general, the organization has supported and encouraged the use of agile methods in the development team	<input type="radio"/>	<input type="radio"/>
3. Adopting an agile method will make the team more effective.	<input type="radio"/>	<input type="radio"/>
4. I have participated in a training program (formal or informal) in an Agile Project Management method such as Scrum, XP, FDD, ...	<input type="radio"/>	<input type="radio"/>
5. There is a member of our team who has experience in agile project management and acts as a coach to the team.	<input type="radio"/>	<input type="radio"/>
6. There is someone in our team that represents the client's interest and explains issues from the client point of view and prioritizes issues on behalf of the client.	<input type="radio"/>	<input type="radio"/>
7. The team holds regularly scheduled meetings between team members and customer representative.	<input type="radio"/>	<input type="radio"/>
8. At the end of our delivery cycle (game/sprint), the team reflects on the challenges faced by the team during the delivery cycle and suggests remedies to ensure the next cycle will be improved.	<input type="radio"/>	<input type="radio"/>
9. The team welcomes modifications of requirements during and after the implementation of a feature or software functionality.	<input type="radio"/>	<input type="radio"/>
10. User stories are always estimated before a developer can start working on them.	<input type="radio"/>	<input type="radio"/>
11. Our team leader/scrum master is empowered to remove impediments that are faced by our team members during the cycle.	<input type="radio"/>	<input type="radio"/>
12. My team employs an issue tracking system (JIRA/Atlassian, Redmine, GitHub, GitLab, ...) to facilitate team communication and document requirements and other stakeholders	<input type="radio"/>	<input type="radio"/>
13. My team employs a source code management (Git, TFS, SVN, ...) to facilitate code collaboration between developers.	<input type="radio"/>	<input type="radio"/>
14. The team conducts Code Reviews on a developer's work before it can be deployed to the production server.	<input type="radio"/>	<input type="radio"/>
15. My team employs automated UI tests to ensure requirements are met at handover and prevent unintentional changes in the future.	<input type="radio"/>	<input type="radio"/>
16. My team employs a CI/CD server to prevent regression and ensure new code commits to comply with the standards (eg. Unit Tests, Coding Standards)	<input type="radio"/>	<input type="radio"/>

## Data Collection Questionnaire for Managers

**Dear Sir / Madam,**

My name is Asegid Debebe and currently, I am a final year student studying Master of Arts in Project Management at AAU, School of Commerce. I really appreciate that you are giving me your precious time to complete this questionnaire.

The main purpose of this questionnaire is to collect data for research that attempts to identify the current level of adoption of the Agile Project Management (APM) methods in Ethiopia in general and at private banks in particular. In addition to describing the current level of adoption of APM methods, it will also identify the major issues and challenges the private banks are coming across while going Agile and will recommend possible solutions to these challenges based on the existing body of knowledge and experiences and researches conducted in other countries. The research findings will be shared with the participating organizations so that it can be used as an input to improve their adoption processes in the future.

To keep participant responses anonymous, please do not write your name on this questionnaire. In addition, **I kindly request your cooperation and sincere response to the questions included in this questionnaire** so that the results of the research will be realistic and useful.

**The estimated time to complete this questionnaire is between 25 – 30 minutes.**

**Thank you for your time and kind cooperation!**

1. Which one of the following best describes your current role in your organization?
  - a. CIO / Head of ICT
  - b. Head of Software Development Unit
  - c. IT Project Manager
  - d. Other: \_\_\_\_\_
  
2. What is your current level of education?
  - a. Diploma (2 years)
  - b. Degree
  - c. Master's degree
  - d. Ph.D
  - e. Other: \_\_\_\_\_
  
3. How many years of experience do you have in the IT profession?
  - a. 0 – 2 years
  - b. 2 – 5 years
  - c. 5 – 10 years
  - d. 10 – 15 years
  - e. 15+ years
  
4. Can you please indicate the size of your software development team?
  - a. 1 – 5
  - b. 6 – 10
  - c. 11 – 15
  - d. 16 – 20
  - e. 20+
  
5. How is the geographic distribution of your development team?
  - a. Co-located in the same office
  - b. Near (but not in the same office)
  - c. Far (located with no possibility of face to face communication)
  - d. Other: \_\_\_\_\_
  
6. How do you characterize your team's current software development methodology?
  - a. Ad-Hoc (unstructured)
  - b. Waterfall method with big release at the end.
  - c. Waterfall method with incremental delivery
  - d. Agile method (Scrum, XP, FDD, ...)
  - e. Other: \_\_\_\_\_

7. If you responded Agile in the above question, which practice do you use?
  - a. Scrum
  - b. Extreme Programming (XP)
  - c. Feature Driven Development (FDD)
  - d. Dynamic Systems Development Methodology (DSDM)
  - e. Crystal
  - f. Lean Software Development (LSD)
  - g. Other: \_\_\_\_\_
  
8. If you responded to Q6 as other than Agile, have you previously attempted to adopt APM methods into your organization?
  - a. Yes
  - b. No
  
9. If Yes, which method have you tried to adopt?
  - a. Scrum
  - b. Extreme Programming (XP)
  - c. Feature Driven Development (FDD)
  - d. Dynamic Systems Development Methodology (DSDM)
  - e. Crystal
  - f. Lean Software Development (LSD)
  - g. Other: \_\_\_\_\_
  
10. If No, to question #8 above, what is your primary reason for not trying to adopt APM methods?
  - a. The management is not fully aware of APM and its advantages.
  - b. The management does not believe there is a need to change the way we develop software.
  - c. Management thinks it will be too expensive and time taking to move to APM methods.
  - d. Lack of trained personnel to start APM in the organization.
  - e. APM is not well suited to our organization.
  - f. Other: \_\_\_\_\_
  
11. Do you plan to adopt an APM method in your software development team in the future?
  - a. Yes, in less than 2 years.
  - b. Yes, after 2 years or so.
  - c. No, no plan right now.

12. Have you been a part of a team that has employed an Agile PM method in your previous work experience (other than the current organization) in Ethiopia?
- Yes
  - No
13. If YES, to the above question, which method have you employed?
- Scrum
  - Extreme Programming (XP)
  - Feature Driven Development (FDD)
  - Dynamic Systems Development Methodology (DSDM)
  - Crystal
  - Lean Software Development (LSD)
  - Other: \_\_\_\_\_
14. In your current software development team, how do you collaborate/communicate with your team members, managers, customers, and other project stakeholders?
- Mainly face-to-face communication on an ad-hoc basis
  - Primarily through email
  - Formal, structured and regularly scheduled meetings
  - No regular communication as I work alone on a project with occasional telephone calls meetings with managers for progress update.
  - Other: \_\_\_\_\_
15. How often do you meet with your team members, customers, and other project stakeholders?
- Ad-hoc or irregular meeting sessions upon request by the team lead or manager
  - Daily
  - Multiple times a week, but not daily.
  - Once a week
  - Once every two weeks
  - Once a month
  - Less than once a month
16. Typically, how long are the durations of your software development projects?
- Less than 3 months
  - 3 – 6 months
  - 6 – 9 months
  - 9 – 12 months
  - 1 – 2 years
  - More than 2 years

17. How often do your team members need to report on the work progress of their respective user stories?
- Ad-hoc or irregular progress updates upon request by the team lead or manager
  - Daily
  - Multiple times a week, but not daily.
  - Once a week
  - Once every two weeks
  - Once a month
  - Less than once a month
18. As software development team, which of the following software code compliance standards and guidelines do you use (please mark all that apply?).
- No, we don't have these
  - Coding standards (StyleCop or similar for Visual Studio, PEP8 for Python)
  - Unit Tests
  - Test Scripts
  - Other: \_\_\_\_\_
19. What metric do you use to estimate the level of effort needed to complete user stories or features during software development?
- Hours
  - Days
  - Weeks
  - Story points
  - Other: \_\_\_\_\_
20. As a software developer, do you test the implementation of a feature yourself or is there a dedicated tester in your team for this purpose?
- I test my own implementation and confirm that it works and it is ready for release.
  - The tester tests and confirms it works before it can be released.
  - Other: \_\_\_\_\_
21. How often does your team's output be delivered into the production server after the completion of your work?
- Weekly
  - Every two weeks or less
  - At least once a month
  - Every 3 months
  - Once every 6 months or so

22. What difficulties do you face in carrying out your current SD projects in terms of:

a. Teamwork

---

---

---

---

---

b. The software development process used at your organization?

---

---

---

---

c. Organizational management

---

---

---

---

d. Customers

---

---

---

---

23. What improvements do you recommend to deal with difficulties or how did you try to resolve them? Are the solutions working?

---

---

---

---

---

---