FACTORS IN AGILE METHODS ADOPTION IN SMALL AND MEDIUM ENTERPRISES

A DOCTOR OF PHILOSOPHY THESIS

in

THE DEPARTMENT OF MODELING AND DESIGN OF ENGINEERING SYSTEMS

(Main Fields of Study:Software Engineering)

Atilim University

by

ABDALHAMID SAMIA

DECEMBER 2017

FACTORS IN AGILE METHODS ADOPTION IN SMALL AND MEDIUM ENTERPRISES

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

ATILIM UNIVERSITY

BY

SAMIA ABDALHAMID

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

THE DEPARTMENT OF MODELING AND DESIGN OF ENGINEERING SYSTEMS

(Main Fields of Study: Software Engineering)

DECEMBER 2017

Approval of the Graduate School of Natural and Applied Sciences, Atılım University.

_____

Prof.Dr. Ali Kara

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy.

_____

Assoc.Dr.Ender Keskinkılıç

Head of Department

This is to certify that we have read the thesis "Factors in Agile MethodsAdoption in Small and Medium Enterprises" submitted by "SamiaAbdalhamid" and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

_____ _____

Assoc. Prof .Dr. Deepti Mishra Prof.Dr. Alok Mishra

Co-Supervisor Supervisor

Examining Committee Members

Prof. Dr. Alok Mishra _____

Asst. Prof. Dr. Murat Ozbayoglu _____

Assoc. Prof. Dr.Nergiz Ercil Çağıltay _____

Asst.Prof.Dr.Abdül Kadir GÖRÜR _____

Asst. Prof. Dr. Serhat Peker _____

Date: 26/12/2017

I declare and guarantee that all data, knowledge and information in this document has been obtained, processed and presented in accordance with academic rules and ethical conduct. Based on these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: SAMIA ABDALHMID

Signature

ABSTRACT

FACTORS IN AGILE METHODS ADOPTION IN SMALL AND MEDIUM ENTERPRISES

Abdalhamid, Samia

Ph.D., Modeling and Design of Engineering Systems (Software Engineering)

Supervisor: Prof.Dr. Alok Mishra

Co-Supervisor: Assoc.Prof.Dr. Deepti Mishra

December 2017, 144 pages

Recently, agile methods have become more popular in the software development industry, but adopting Agile methods by software development organizations can be an easy process or a hard one depending on certain factors. So to make the process of adopting the Agile method successfully, there are some factors that can help organizations to adopt agile without fear of failure. There is not enough research in terms of adopting Agile in SMEs in particular. For this reason, we studied the factors of adopting Agile methods in small and medium software development organization to provide guidelines for success and failure factors. In this research, the use of agile methods is explored in small- and medium-scale software. Based on rigorous literature review number of models and hypotheses were developed and examined by data collected from 52 software organizations from 7 countries based on comprehensive questionnaire. As results some significant success factors were identified such as : Assigning essential features first. Frequent delivery of software, and the use of tools. In terms of failure factors, the most significant factor that can cause failure is too-large size of an organization.

Keywords: Agile methods, Success factors, Failure factors, Adoption agile in organizations.

ÖZ

KÜÇÜK VE ORTA İŞLETMELERDE ÇEVİK YÖNTEMLERİN UYARLANMASINDAKİ FAKTÖRLER

Abdalhamid, Samia

Doktora Tezi, Mühendislik Sistemlerinin Modellenmesi ve Tasarımı

(Yazilim Mühendısliği)

Danışman: Prof.Dr. Alok Mishra

Eş-Danışman: Doç.Dr. Deepti Mishra

Aralık 2017, 144 sayfa

Son zamanlarda çevik yöntemlerin yazılım mühendisliği endüstrisinde giderek daha popüler olmasına rağmen yazılım geliştiren kurumların çevik yöntemleri benimsemelerinin kolaylığı veya zorluğu bazı faktörlere bağlıdır. Çevik yöntemlerin benimsenme sürecini başarılı hale getirmek ve başarısızlık korkusu olmadan benimsenmelerini sağlamak için kurumlara yardımcı olmak üzere, bazı faktörler bulunmaktadır. Küçük ve orta ölçekli firmaların çevik yöntemleri benimsemesini araştıran yeterli sayıda çalışma bulunmamaktadır. Bu sebeple, biz küçük ve orta ölçekli firmaların çevik yöntemleri benimsemesini etkileyen faktörleri başarı ve başarısızlık faktörleri için prensipleri sağlamak üzere araştırdık.Bu araştırmada, küçük ve orta ölçekli yazılım firmalarında çevik yöntemlerin kullanımı araştırıldı. Titiz bir literatür araştırması sonucunda birçok model ve hipotez geliştirildi ve kapsamlı bir anket ile 7 ülkedeki 52 yazılım kurumundan toplanan verilerle değerlendirildi. Sonuç olarak önemli özelliklere öncelik vermek, sık yazılım teslimatı yapmak ve araç kullanımı gibi bazı önemli başarı faktörleri belirlendi. Başarısızlığa yol açacak en önemli başarısızlık faktörü olarak kurumun çok büyük olması belirlendi.

Anahtar kelimeler: Çevik yöntemler, başarı faktörleri, başarısızlık faktörleri, çevik yöntemlerin kurumlarda benimsenmesi.

To My Dear Family,

ACKNOWLEDGEMENTS

I express sincere appreciation to my supervisor Prof. Dr. Alok Mishra for his guidance, supportive and constructing approach throughout the research. Thanks also go to my co-supervisor Assoc. Prof .Dr. Deepti Mishra

I offer sincere thanks to my thesis jury members Assoc. Prof. Dr.Nergiz Ercil Çağıltay and Asst. Prof.Dr. Murat Ozbayoglu for their guidance in evaluation and review of my study. Also I would like to thank Prof.Dr. Sofia Ostrovska.

Grateful thanks to my dear husband Husam, my children Mohammed, Adam and Youssef, and my sister Suhila for their love, encouragement, patience, and understanding throughout my study journey. I appreciate their support.

Table of Contents

LIST OF TABLES

TABLE **PAGE**

LIST OF FIGURES

LIST OF ABBREVIATIONS

ASD - Agile Software Development

CMMI - Capability Maturity Model Integration

OO - Object Oriented

XP - Extreme Programming

RUP - Rational Unified Process

DSDM - Dynamic Software Development Method

FW - Frame Work

ASDM - Adaptive Software Development

FDD - Feature Driven Development

SF - Success Factors

FF - Failure Factors

A - Attribute

F - Factor

H - Hypthosis

CHAPTER 1

INTRODUCTION

This research is an attempt to identify and explore the factors essential to the process of Agile adoption. The purpose behind the investigation is reveal the main aspects which can make Agile adoption in software development companies and enterprises succeed or fail. In order to do so, guidelines will be provided for the introduction of Agile alongside a list of factors important for realizing the positive impacts and benefits involved therein.

1.1. Agile Methodology Revolution

In the early years of programming in the 1950s, structured languages such as Fortran were first used. Then in the 1960s, this was improved and developed into object-oriented languages ("Object-oriented programming", 2009). Through time and in a similar way, software improvement approaches have been continuously developed. In terms of making software, the improvement approaches are usually selected by the developers who divide the process into stages. Also, in every phase or stage, they determine the guidelines required for that particular phase. With the software crisis in the 1970s, the first major development approaches started to appear (Klimeš et al., 2006).

Programming engineers attempted to lessen the impact of the crisis and prevent it by coming up with structured strategies to develop a product. As stated earlier, these approaches break down the software development procedure into phases so that the developers can concentrate more on one stage at a time. Utilizing these methods led to reducing the number of unsuccessful or unfinished projects, eliminating the expense

and extended advancement period of software ventures, and restricting the impact of the crisis in the industry (Klimeš et al., 2006).

Nowadays and as a general rule, all information technology applications depend on software at all levels (Mens et al., 2005). The need to develop new software methods came as a result of changing requirements and a dynamic business environment, forcing the developers to try and find new methods that can be adopted in these scenarios (Lan and Ramesh, 2007). In addition, it is vital at any rate for the developers to react to the changeable environment, and to understand that the process of developing software is now a changeable subject itself (Abbas et al., 2008).

During the development process, it is hard to define the requirements at the very beginning and the reason behind this is that there are many changes that cannot be prevented alongside the improvement (Williams and Cockburn, 2003). This is distinctive in construction and manufacturing enterprises, in which nature is appropriate for all requirements in advance (Rajlich, 2006).

The issues in the software methods have not only been brought on by the environment of software, they have likewise originated from the past methodologies including problems that surface during programming itself. The previous methodologies were observed to be heavyweight, authenticated and concentrated on plan- driven approaches. In the Waterfall approach, the most serious issue is found in characterizing all the necessities at the beginning of the venture. The specifics of requirements are difficult to clarify without seeing the functionality of the developing system (Beck, 2000). Once the requirements are gathered, the improvement will continue and nothing can be changed until the project is completed. This situation will bring about high expenses as changes can be made only after the thorough execution of the system.

Nevertheless, clients are frequently unable to specify without errors their requirements until the point when they can make a comparison in between the working programme and their expectations within the context of the task (Brooks, 1987). Traditional methods - for example, the waterfall - were created to allow delivering the software at the end of the venture. As a result, clients frequently do not come in contact with the product until the point it is completed at the end of the venture (Tripp, 2012).In

view of these reasons, Agile approaches were proposed by experts to deliver substantially positive effects in the performance of projects. Demands with respect to the effects of the utilization of Agile-development approaches fall into some common trends as follows.

First, Agile approaches increase output by making a tenable speed in the development process (Beck, 1999), thereby building an organizational environment in which individuals wish to work (Highsmith, 2002). Second, they tend to provide confidence between the product development group and their clients as a result of the involvement of those very clients in the venture and an orderly and refined demonstration of working programming (Fowler and Highsmith, 2001). Third, these approaches handle turbulent surroundings by means of concentrating on interaction and adjustments between design and delivery. As a result of this iterative delivery paradigm, Agile approach experts focus on delivering better and more valuable programming. This leads to increase in delivery achievements and reduction of any potential hazards while giving a satisfactory return on the investment (Moran, 2010) (Highsmith, 2002).Apart from these benefits, Agile development has a tendency to concentrate on right- on- time and quick production of working codes, repeated, little, or incremental changes, pair programming, short redundancy, fast and continuous client comments processing and collaboration (Beck and Fowler, 2001) (Cockburn, 2001).

On the other hand, traditional development has a tendency to confirm inclusive analysis before coding, creation, and up keeping of models, comparatively a longer period of time between deliverables and little value given to client cooperation (Boehm and Turner, 2004)(Turk at el., 2005). Numerous specialists have announced that Agile methods can possibly bring about far more customer satisfaction, bring down bug rates, and involve a shorter development cycle and faster adjustment to quickly changing business requirements (Sidky at el., 2007).

In essence, Agile principles were more commonly addressed towards the late 1990s, but the Agile Manifesto was pronounced in 2001(“Agile Manifesto”, 2001). This was when some IT experts began to work exclusively on new ways to deal with

developing a software. As a result of their researches, new methodologies were developed, each with its own popular features. The name "Agile Manifesto" came into spotlight when at a conference in Utah in 2001("Agile Alliance. Manifesto for Agile software Development", 2001).These methodologies were produced in view of a similar rule that an ideal way to check a system is to produce working renditions for the client and, then, refresh it as indicated by their comments. The Agile methodologies are built on three principles by their authors ("Agile Alliance. Manifesto for Agile software Development", 2001). These are:

- to create a software that ensures customer satisfaction through constant delivery of working programming and receiving feedback from the client;
- to deal with the change of requirements at any phase during the developing process in such a way that the client would be satisfied with the development process;
- to promote interaction between the developers and the clients every day and all through the process of developing a project; and
- finally, to develop on a test-driven premise, which requires writing a test before writing the code. Here, the test suite is run on the application.

Recently, Agile methods have become more popular in the industry of software development, and these methods are applied with considerable controversy around their applications (Highsmith, 2002) .as a result, adopting Agile methods has helped to deal with some problems that can be faced during the development of software since they allow for a faster delivery of the software and guarantee that the product meets the clients' changing needs (Paetsch at el., 2003).

There are different Agile methods such as: Agile Scrum Methodology, Lean and Kanban Software Development, Extreme Programming (XP), Crystal, Dynamic Systems Development (DSDM), Feature-Driven Development (FDD), Rational Unified Process (RUP), and Adaptive software development (ASD). Every one of these methods shares some fundamental standards, for example, promote consumer satisfaction, deal with changing requirements, oftentimes deliver working programming, and create close cooperation between developers and business people (Paetsch at el., 2003).

In order to find out about the number of establishments that adopted Agile since the beginning of its use, there are two different studies administered in 2005 that provide such data. Initially, an online survey directed by MethodsAndTools.com ("Adoption of Agile Methods", 2007) signifies that around 40% of the 232 member enterprises have used Agile approaches, and that 20% were assessing them in pilot ventures. The second review, directed by Schwaber and Fichera for Forrester Research, states that around 14% of all North American and European organizations were utilizing Agile methodologies and another 19% intended to do so in the prompt future (Schwaber and Fichera, 2007). This review additionally infers that while the early adopters were normally firms of smaller sizes creating high-tech products, the current adopters had a tendency to be information technology groups inside bigger organizations.

Moreover, a positive feedback was received about the adoption of Agile methods by organizations which have implemented them. According to them, there are a number of benefits gained by adopting Agile, such as raising productivity, extended test scope, enhanced quality, decreased time and expense, better understanding, maintainable and extensible code, better cooperation and increased client satisfaction (Vijayasarathy and Turk, 2008).

1.2. Agile Methods

In this section, eight Agile methods used in our research will be introduced followed by a brief explanation for each.

Agile processes are considered as another host attempting to challenge the constraints of traditional software development methodologies by utilising a certain strategy (Chan and Thong, 2009) .There are many types of Agile approaches, the most common eight of which can be listed as following:

- Agile Scrum Methodology
- Lean and Kanban Software Development
- Extreme Programming (XP)
- Crystal
- Dynamic Systems Development (DSDM)
- Feature-Driven Development (FDD)

- Rational Unified Process (RUP)
- Adaptive software development (ASD)

While traditional methodologies, for example the life-cycle approach and object oriented (OO), still have control over the area of systems development, many articles and several opinion polls clearly show the growing popularity of Agile methodologies (Nerur, 2016).The appearance of these methodologies led to a division in the sector and the development of software to oppose the traditional views, with each group advocating their own methodology's benefits. In addition, a more balanced perspective is offered of the two competing methodologies by the few who propose that each method has its strengths as well as limitations, and that they may not be suitable for specific types of projects (Nerur,2016).

1.2.1. Scrum

The Scrum method was particularly intended to deal with quickly changing business prerequisites. The name is taken from a methodology utilized as a component within the English Rugby game, and the approach propels ventures by developing communication between colleagues and breaking the work into a progression of "sprints" lasting anywhere between one and four weeks (Schatz et al., 2005).

Scrum has earned ever-expanding popularity in the Agile programming improvement circles due to its clarity and confirmed output and the potential to wrap up what is needed for different designing practices as proposed other Agile methods (Mann et al., 2005). Despite these, its limitations are as follows: the venture is exceptionally reliant on cohesiveness of the group and the individual responsibilities of each member and minor lack of coordination may bring about counter-productive results in the sprint. Also, it does not clearly handle the problem of criticality, the client is offsite, closer client cooperation is impossible and improved group dynamics empowered by Scrum are not accessible in one-developer ventures (Flora and Chande, 2014).

Scrum is a framework for software development to be used for projects and manage such products or application developments by IT establishments (Permana and Bali, 2015). This approach can be utilized as with different designing practices advanced by other Agile strategies (Mann et al., 2005).

1.2.2. Lean And Kanban Software Development

Lean is an interpretation of Lean industrialization and IT practices within the product development area. The expression "Lean Software Development" originates from the book "Lean Software Development: An Agile Toolkit" written by Tom and Mary Poppendieck in 2003, and depends on 7 Principles and 22 Tools (“Lean Software Development”, 2003).

Lean Software Development (LSD) is an iterative approach that concentrates on decreasing waste and optimizing the whole procedure to accomplish the utmost conceivable gain. It merges well with the idea of six sigma (Poppendieck and Poppendieck, 2003), and is suitable for any product advancement venture where there is a requirement for radical change. (Flora and Chande, 2014). Its limitations include the following: the venture is exceedingly reliant on cohesiveness and the individual duties of the colleagues; in this way, group building becomes a basic factor and absence or improper participation could lead to a scope (Flora and Chande, 2014).

The Kanban methodology in programming development was constructed in 2004, when David J. Anderson was helping a small IT group at Microsoft. Kanban Software Development depends on a number of principles, mainly: imagining the work process, constraining the work in progress, measuring and managing flow, improving process policy and collaboration, and utilizing models and logical methods (David and Anderson, 2010). The Kanban methodology in software development encourages venture groups to imagine the work process, confine work in advance (WIP) at every work process stage, and measure cycle durations (Kniberg, 2012). The Kanban board provides a larger vision of the procedure by demonstrating the contribution of every developer, in this way establishing priorities and highlighting bottlenecks. Additionally, the strategy means rapid adjustment of the procedure by utilizing shorter feedback loops (Ahmad et al., 2013).

There are, of course, limitations, such as insignificant breakdowns in the framework's procedure resulting in the whole line to close down and recovery requirements. Also, the throughput of the Kanban framework is not administered as an alternative and comes as a solution of controlled WIP and already-known cycle durations (Flora and Chande, 2014).

1.2.3. Extreme Programming

Extreme Programming was created at Chrysler by Kent Beck while participating at a payroll venture as an individual from a 15-member group. (Highsmith, 2002). According to Kent, Extreme Programming (XP) is also "a well-known and a light weight discipline of software development that focuses on engineering practices"(Kent,2000). The XP engineering practices can be used with the Scrum to increase the productivity of the venture team (Mar and Schwaber, 2002). The XP is a useful method that can be used for Web-based software ventures (Maurer and Martel, 2002).

The XP method has the ability to develop programming quality while reducing functionality delivery plans. It depends on a set of ideas and practices that involve having the client meet with the development group as well as the element of simplicity.[46]. There are some limitations when using the XP method, such as the need for larger coordination between the software engineers while doing pair programming. Other issues are client cooperation not being very powerful and the stages of testing and development of codes being done by the same person. Pair programming is essential to practice in the XP method but it cannot be used in one-developer projects (Flora and Chande, 2014).

1.2.4. Crystal Method

This is a toolkit or combination of approach components that companies integrate into proper methods to suit singular ventures. In this respect, large ventures and projects that affect open public safety demand more approach components than small non-critical projects (Highsmith, 2002). There are some benefits in using the Crystal methodology, namely reflective improvement, close contact, individual safety, better and concentrated work, more frequent delivery and simple access to clients

(Cockburn, 2004). The main limitations of this method are that lack of success leads to loss comfort, unrestricted money, basic money, and life (Flora and Chande, 2014). It is hard to define the scope of Crystal in view of the fact that the method offers a foundation for choosing and tuning other methodologies (Flora and Chande, 2014).

1.2.5. Dynamic Systems Development Method (DSDM)

The DSDM method was produced in the United Kingdom in the mid-1990s and is an outgrowth of, as well as an extension to, fast application advancement practices. The DSDM offers preparing and documentation of any ASDE (Stapleton, 1997) and is especially useful for systems that need to be developed in a short period of time and where the requirements cannot freeze at the beginning of the application ("Dynamic System Development Method", 2015). In terms of time and budget, it is highly effective (Flora and Chande, 2014).

However, there are some limitations in DSDM method. It depends on client contribution which cannot be possible in each venture, on account of its strictness and eight standards, the primary issue with DSDM is that it can be restrictive and hard to work in contrast with other Agile methodologies, and it is not appropriate for all ventures, specifically frameworks that are real-time and safety-critical (Flora and Chande,2014).It is independent of tools, so it can be utilized with both structured analysis and design approach or object-oriented approach ("Dynamic System Development Method",2015).

1.2.6. Feature Driven Development

Feature Driven Development or FDD was created for a bank venture in Singapore by Coad and DeLuca, comprising a five-stage procedure that does not demand extensive preparation for a development group to be utilized (palmer and Felsing, 2002). FDD benefits are that it concentrates on the plan and building stages, gives priority to quality matters all through the procedure and involves recurrent and tangible deliveries, alongside precise checking of the progress of the venture. its limitations are that does not particularly address venture criticality (Flora and

Chande,2014). FDD has the ability to combine Agile development mechanisms from different strategies, and works exceptionally well with the XP practices of pair programming and day-by-day stand up gatherings (Livermore, 2008) .It is also used to develop secure Websites like the XP and Scrum (Firdaus et al., 2013).

1.2.7. Rational Unified Process (RUP)

The RUP is a well-characterized and very much organized programming engineering process which characterizes who is in charge of what, how things are done, and when to do them. The RUP, likewise, organizes the lifecycle of projects (Kroll and Kruchten, 2003), and offers a customizable procedure format for programming engineering. Its configurations can be made to bolster small or large groups and trained or less-formal ways to deal with improvements. The RUP is utilized by a wide variety of organizations in the industry sector (Kroll and Kruchten, 2003).

1.2.8. Adaptive Software Development

Adaptive Software Development (ASD) was created by James A. Highsmith and provides an Agile and adjustable approach to deal with fast and highly-changeable programming ventures (Highsmith, 2000). Adaptive Software Development life cycle contains three stages: speculation, collaboration, and learning (Agile Methodologies") and offers a framework as to the best way to encourage cooperation and learning inside the project (Stapleton, 1997). It helps project economy by raising the returns (Arthur, 1996). High speed and change factors, obviously, distinguish this economy across different projects (Livermore, 2008).

1.3. Research Overview

There are about twelve different methods of Agile focused on in this research with respect to their use in small and medium enterprises. In addition, there are many factors that can affect the decision of choosing one method over others for a particular project, for which purpose during this research we will investigate such factors as well.

1.4. Research Motivation

In the last few years, Agile approaches have become more popular in the software development industry, and the methods applied have brought about considerable controversy (Highsmith, 2002). In this way, adopting Agile methods has helped to deal with problems faced during the development of software. These methods make room for delivery of software in a faster pace and to guarantee that the product meets clients' changing needs (Paetsch et el., 2003). Numerous organizations have sought to adopt Agile methods so as to make use of their various advantages, which comprise, to mention a few, quicker return on investment, better programming quality, and greater client satisfaction (Highsmith, 2002).

However, adopting Agile methods by software development organizations can be an easy process or a hard one depending on certain factors, such as the individuals involved (Cockburn and Highsmith, 2001; Lee, 2008) as well as the organizational factor (Strode et al., 2008b). These are considered vital before the adoption of Agile methods and while software development is in progress (Iivari and Huisman, 2007). As a result, to make the process of adopting the Agile method successfully, there are some factors that can help organizations without fear of failure provided that they also know the failure factors that can make the adoption of Agile methods turn into an unsuccessful venture.

Such knowledge helps organizations to be aware of the facts that can affect their adoption of Agile and to do so in a proper way. There are many researches related to these factors in adopting Agile methods in organizations, such as: Chow and Cao (2008) and Kumar and Goe (2012). Yet, there is not enough research in terms of adopting Agile in SMEs in particular. For this reason, we studied the factors of adopting Agile methods in small and medium software development organization to provide guidelines for success and failure factors.

Exploring the combination of Agile approaches with traditional methods In SMEs and what type of projects require using specific methods of Agile will be addressed during this research, thereby contributing to the present literature in the

following ways: First, we will explore if adopting Agile in large projects is the reason for the project to succeed or fail. Next, new aspects will be added to the scope by comparing Agile adoption in SMEs and large enterprises, such as providing better control over the work is viewed as the primary advantage of the Agile methods within large and small companies, while for the medium-size companies, the priority is switched to coping with changing user requirements.

As for the scope in detail, the goal of this study is to explore the factors of Agile adoption methods in software development organization that have already such methods in place. It does not focus only on the success factors but also explores the failure factors. Obtaining information from such companies would provide us with their experience and how they developed their work in the process. Such data will encourage other companies to adopt Agile methods without fear of failure because the information can be used to help them to adopt Agile methods successfully and avoid failure by learning the lessons from those who used Agile in their works before.

1.5. Objectives of the Study

The main aim of this research was to examine the adoption of Agile approaches in SMEs, with other objectives achieved during the process as mainly:

- Explore the use of Agile methods in small- and medium-scale software development organizations;
- Investigate the factors that influence the decision to choose one method over others for a particular project;
- Explore the success factors of adopting Agile methods;
- Find out about the failure factors in adopting Agile methods;and
- Determine if the methods used are alone in projects or in combination with other methods.

1.6. Report Structure

Chapter 2 provides information and introduction to the Agile approaches and also will discuss how these methodologies can be adopted for new environments. there will be the limitations and benefits of using Agile methodology, and what factors can affect the adoption of Agile methods by software development organizations according to the related work.

Chapter 3 includes an explanation of the objectives of the study which were detailed above, and presents the research questions used to determine the survey questions. It also includes an explanation of the software used to design the survey and states how the process is carried out.

Chapter 4 illustrates the results of this research and the statistical analysis used to examine the findings of the survey.

Chapter 5 provides the discussion of the results as explained in Chapter Four and compares our results with previous studies in attempting to find out the similarities and differences with these studies.

Chapter 6 as the conclusion summarizes the entire thesis, including the findings of this research and which of the aims have been achieved.

CHAPTER 2

RELATED WORK

This chapter will discuss the development of Agile approaches, the different types, and what benefits and limitations there are in adopting Agile methodologies. Then, it will address how these methodologies can be adopted for new environments, as well as the factors essential in the adoption of Agile methodology.

2.1. Agile Methodology

This section provides an introduction to Agile methodology in general and the Agile Manifesto in addition to the related practices.Agile approaches were developed by a group of software developers and methodology authors, who met in 2001.The objective was to offer approaches to software development of less bureaucratic a nature and to concentrate less on documentation to pay more attention, instead, to user interaction and early delivery of a working software (Bannet et al., 2010).

These days, programming development is faced with never-ending and fast changes. To help the improvement and immediate progress of complex systems and meet the particular needs of such an approach to improvement, diverse methodologies have been proposed over the years. Among these methodologies, the present work will concentrate on "Agile methodologies"(Strode et al, 2009), which are an extension of the "Agile Manifesto" and make room for more noteworthy reactivity as per the changing needs of clients (Chan and Thong, 2009). According to Williams and Cockburn (2003), Agile advancement is related to the ideas of “feedback and change ". This is the reason Agile methodologies have been produced keeping in mind the

end goal to adopt rather than dismissing new demands and requirements that can later make way for progress.

The Agile Manifesto offers a group of principles; yet, it does not uncover the correct meaning of the concept (Chan and Thong, 2009). In reality, a number of researchers have attempted to characterize Agile methodology through the idea of "agility". In terms of Agile practices, it is believed that they resemble whatever other practices have to offer. However they are learned through examples, applications, and preparations. On the off chance that practices are separated from logic, the outcome is likely to fail. For instance, if an organization that needs to be more "Agile" would, in any case, need to know precisely when projects would begin and stop before the requirements were outlined (Shankarmani et al, 2012).

2.2. Agile Benefits and Limitations

In spite of the fact that the Agile procedure provides diverse methods to deal with the way programming ought to be created, the techniques ought not to be thought out as the main "right" route for software development. In spite of the many points of interest that these methodologies have, a few limitations exist. In this section, the concept and notions will be clarified as to the advantages and disadvantages of Agile methods.

One of the studies in this respect has been directed towards recognizing the benefits of Agile strategies for software development. The achievement of the strategy was classified in four various sets, which are: quality, scope, time and cost (Chow and Cao, 2008). In terms of the benefits, this study reveals that (i) the delivery strategy, (ii) Agile engineering techniques and (iii) group capacity were the three variables that proved to be advantageous, being also regarded as the basic success components for Agile development.

Another study has proposed that, in order to understand the benefits of Agile, the strategies ought to be practiced within the scope of favourable conditions (Strode et al., 2008a). The enterprise, the administration, and the group must cooperate while attaching value to feedback, learning, social connections, joint efforts and competency

(Strode et al., 2008a). According to Strode (2008a), environmental factors assume a part in delivering the advantages of Agile.

The environmental factors characterized by Strode incorporate enterprises, application area, individuals, venture, and technology. It is, however, acknowledged that these aspects are different from one organization to the next; thus, more learning is required as to these issues as the information can be used to serve as a recommendation and reference to other people from various environments concerned with adopting Agile. Another systematic study was conducted in which the authors expressed that the advantages of Agile are directed towards client cooperation, work procedures for dealing with defects, learning in combine-programming, thinking ahead by the administration, concentrating on the existing work by the engineers, and estimation (Dyba and Dingsoyr, 2008). These angles demand the involvement of whoever they may concern.

There are many other works with regard to about the proof for the advantages of Agile strategies. Based on those, it is important to know how Agile works and what obstacles can prevent it from succeeding. To do so, the limitations should likewise be taken into account. This is a substantial step in order know the fitness of these methods in the development process of software. One of the studies debate the limitations of Agile practices as retrieved from 11 presumptions discovered in the standards and practices of Agile (Turk et al., 2002). In more detail, the authors introduce the restrictions formed on anecdotal proof. It was expressed that Agile methods are restricted in their support of a distributed development environment, subcontracting, building reusable artefacts, advancement including large-size groups, developing safety-critical software and developing large, complex software products.

There is another issue to be considered, that is face-to-face communication, which has been regarded challenging in cases when the development team is not co-located. In other words, when they are separated and should have a similar form of progress for product improvement. The Agile procedure can turn out to be more unpredictable when one organization outsources their venture and includes subcontracting organizations. It has been proposed that for this situation, the contract ought to incorporate both settled and variable parts (Turk et al., 2002).

In this respect, the Waterfall approach is clearly appropriate in projects that include subcontracting. That is because it gives detailed requirements and specifications before the execution begins. The other points mentioned are that Agile is not appropriate for large groups and complex systems as it is likely to lessen the agility. Should the environment fail to bolster Agile elements, the efficiency of Agile methods will not be complete during the development process.

In spite of the fact that the techniques - for instance, feedback, the little extent of requirements and clients cooperation – are regarded as the positive features of the Agile procedure, one review asserts that the iteration in XP needs to be treated with under Agile circumstances (Bahli and Abou Zeid, 2005). The challenges that appear in controlling the iteration demand the venture administrator to have three plans. In line with this, Bahli and Abou Zeid (2005) recommended that the making of such arrangements for iteration ought to incorporate "one for the current iteration, one for the upcoming iteration and one for the overall iteration". Likewise, the Lean improvement and pair programming methods were observed to be useless (Dyba and Dingsoyr, 2008).

One research has asserted that product proprietorship, co-found groups, knowledge and committed customers are conceivable weaknesses in the Agile adoption (Ilieva et al., 2004). In the review by Ilieva and associates (2004), there were three groups practicing XP and Scrum, utilizing a college as their subject setting. One of the groups did not go with the client requirements and, to create a useable system, programming was produced openly as per their own thoughts and requirements.

In another case, there was a developer group which had erroneously comprehended the requirements of the item proprietor; however, they did not ask for further explanation. For this situation, it was proposed that the item proprietor (substituting for the customer) ought to have had an unmistakable recollection of what was really required. He ought to have been instructed about the system asked. Having said this, experienced development groups are preferred for Agile to work (Dyba and Dingsoyr, 2008). In a similar way, the developers ought to cooperate and voice their observations to the item proprietor at whatever point they find deceptive data about

the system. According to the review, the greater part of the groups was in agreement that a common and collaborative working environment ought to exist in the development process.

2.3. Adoption of Agile Methods

This study aims to investigate the suitability for adoption of Agile Methods for various organizations and environments. This is on the grounds that, in consideration of making it work, Agile methodology aims to concentrate on skilled people and how they cooperate. For this reason, the adoption of Agile specialists' observations will be discussed at this stage together with how these findings will improve efficient adoption. Moreover, this study intends to shed light on the factors related to the adoption and use of Agile methodologies.

2.3.1 Understanding the Perceptions Surrounding Agile Methods

One study has discussed the significance of understanding the perceptions of social (for example, organizational structure, people, environment, etc) and technical features for the adoption of software development strategies as a whole (Vavpotic and Bajec, 2009). This section presents the view of practitioners in connection to the adoption and utilization of Agile strategies. By first understanding these observations, one can also determine the reasons why practitioners choose to go for Agile techniques. It is important to understand these perceptions in light of the fact that the distinctions in the factors of adoption cannot be studied unless the level of such conceptual influence from their perspective is seen first.

A systematic review was carried out of practical investigations directed up to 2005 and related with Agile (Dyba and Dingsoyr, 2008). The purpose of these studies was to identify the perception of developers (Ilieva et al., 2004, Mann et al., 2005; Bahli and Abou Zeid, 2005), clients (Ilieva et al., 2004; Martin et al., 2004, Koskela and Abrahamsson, 2004) and students (Melnik et al., 2005; Melnik and Maurer, 2002) when utilizing Agile strategies. For instance, Ilieva et al. (2004) noticed that developers found XP which is one of Agile techniques to be extremely beneficial.

Notwithstanding all these, practicing the strategies - for example, 40 hours of working in seven days in XP - proved to make developers exhausted over time. Mann et. al (2005) recognized that developers saw Agile to possess some advantages for them. As for client cooperation and obligations, it was found from the review that developers are assured given that they claimed to be building the product as per the clients' desired characteristics. Another review found that workers in their examinations saw XP as simple and valuable to utilize (Bahli and Abou Zeid, 2005).

Having knowledge in utilizing Agile strategies and understanding the advantages Agile can convey, the developers from the review said that they would utilize Agile in their future undertakings (Ilieva et al., 2004; Mann et al., 2005; Bahli and Abou Zeid, 2005). Additionally, Agile received well-deserved attention from clients; Mann et al (2005) expressed that clients trust that every-day stand up gatherings help them to keep up-to-date. The review additionally found that arranging meetings was useful in lessening perplexity about the system to be created .

In any case, bearing in mind all advantages to be experienced by the clients, Mann et al. (2005) stress that one should prepare to utilize Agile (Scrum). This is critical for the clients to see how Agile functions and what developers anticipate that clients will do. According to Martin et al (2004), clients were experiencing anxiety and saw Agile as defying to them. Since the clients work intimately with developers, they need to adopt to various cultures and organizations of the developers. Melnik and Maurer led a study on students' perceptions as well (Melnik and Maurer, 2002; Melnik et al., 2005). Generally speaking, the reviews found that Agile assisted the students to develop professional abilities, for example, in correspondence, collaboration, obligation and adjustability.

Some of the studies were mostly carried out in western countries, for example, Canada, and others in the EU (Melnik et al., 2005; Bahli and Abou Zeid, 2005) as well as the United States (Mannaro et al., 2004). In view of these investigations, mixed outcomes emerge as to the advantages that Agile can convey, as stated by Tessem's qualitative examination (Tessem, 2003), which concentrated on XP practices.

In addition, there was an effort to examine the view of Agile methods inside the Microsoft Corporation around the world (Begel and Nagappan, 2007) conducted after 2005. In spite of the fact that the review by Begel and Nagappan (2007) was led on a worldwide scale, there were no debates or contrasting points in regard to the hierarchical culture or structure dominant in different nations. Likewise, the organizational viewpoints may have had very similar traits in light of the fact that the review was just centered around the Microsoft organization. One study researching the perception found that the most critical issues rose out of communication and feedback procedures, where they were depicted to be "mainly of psycho-social nature, with practically no dependence on technology" (Misic 2006). From the perception in this review, Misic (2006) was convinced that the advantages from the two strategies (communication and feedback) take up a key role in successful execution of programming utilizing Agile.

Regardless of the way that the point of software development is to create technology, then again, it was additionally found that the most difficult challenges were posed by social and human-related viewpoints or activities. This appeared to be a major finding for developers and organizations so as to prioritise the two viewpoints and, in the meantime, not to disregard the technical aspects. The fact is that technical aspects still remain a fundamental variable for software development. One investigation was led in Nokia, and the outcomes demonstrated that the more extended experience adopters practice Agile, the more positive their views become with respect to Agile value (Laanti et al., 2011).

In terms of the importance of social aspects and Agile techniques, the features or perspectives critical for the adoption and use of Agile methodologies have indicated significant contributions in social terms; for example people and the organization and the environment impacting one another in a given setting. What's more, the “people” aspect is expressed as the way to success (Lee, 2008), as it is acknowledged that people and society are much more integral to Agile methodologies (Cockburn and Highsmith, 2001). Earlier thoughts regarding the significance of these viewpoints will be addressed here.

Products of software include technologies and are driven by business and individuals. Subsequently, to create quality products, both organizations' and clients' needs should be considered. The advent of Agile methods has changed the thoughts regarding software development, thereby proposing an equivalent view of technical and social aspects. Currently, the focus has started to concentrate on the latter (Law and Charron, 2005; Robinson and Sharp, 2005a; Robinson and Sharp, 2005b, Moe et al., 2008; Strode et al., 2008a; Siger et al., 2008). This means social perspectives are being given more consideration and importance than the technical aspects. Furthermore, this is a result of the method for the Agile itself, for example in XP (Robinson and Sharp, 2005a).

Among other Agile techniques, it is stated that the first methodology practiced by the organizations was Extreme Programming (Tolfo and Wazlawick, 2008). Livermore (2006) clarified that while adopting it, companies were, for the most part, fitting the practices of Extreme Programming to meet their organizational culture and development environment (Livermore, 2006). In this situation, it can be seen that product methodologies are adopted in manners similar with the goal to be met as far as the social viewpoints of the organization is concerned.

Furthermore, it is essential to notice that a cooperative culture is important when it comes to XP methodology. Tolfo and Wazlawick (2008) demonstrated that their venture was developed after adopting the practice of short daily gatherings. The practice gives a medium to debate, problem-sharing, and problem-solving (Tolfo and Wazlawick, 2008).

In addition, Agile methods require intensive cooperation with clients and people in the group (Cockburn and Highsmith, 2001; Lycett et al., 2003). In Agile, each developer and client plays a role and not at all like the traditional methods, for example, Waterfall, V-model and Spiral Model, where clients are included fundamentally in the specification phase at the starting and have insignificant participation in other subsequent actions (Nerur et al., 2005). As far as the product building viewpoint is concerned, one review has been led in human and social factors (John et al., 2005). From here, the two variables (human and social elements) are addressed and viewed as vital during the time spent throughout the development of a programme.

As programming is done for individuals and by individuals, it can be said that these two elements “have an exceptionally solid effect on the success of developing software efforts and the resulting system” (John et al., 2005). Whether human and social elements are thought to be essential, while keeping in mind the end goal to deliver a successful item, it is paramount not to underestimate these two perspectives.

In connection with the above reviews, there is an idea regarding the social perspectives of Agile methods with focus on Extreme Programming (Robinson and Sharp, 2005a). Client cooperation makes it necessary to have clients available and to utilize the planning game while pairing practices during programming, refactoring, test-first advancement and simple design stages.

The social aspect of Agile, including the involvement of the entire group, has also been put in the spotlight by Whitworth (2008), who found that involvement and cooperation are the outcomes of group cohesiveness. Trust, esteem, and mass thinking are likewise fundamental for such group cohesiveness. A lack of these components will bring about failure (McAvoy and Butler, 2007). From the review, the most executed technique in XP is found to be nonstop code integration; small functional releases and refactoring are done with the on-site customers, and coding standards and regular releases were statistically observed to be similarly vital.

Building up effective interplay with the group is the most critical part of Agile. In spite of the fact that there is no specific preference as to which tools to use, Agile techniques tend to clarify what to go with. For instance, test-first improvement requires an automated admission test and, yet, the group can likewise utilize different tools which are appropriate for them. To supplant the use of the automated admission tools, one review utilized a less complex instrument, regarded as a cheap tool in the review, to help beginner users with coming to terms as to how Test Driven Development (TDD) really operates (Miller, 2004). In Agile, tools are picked relying upon their appropriateness for the user.

One essential characteristic in Agile is having feedback, for which purpose major communication efforts have to be made. Both the clients and the developers ought to have a fair amount of information in order for them to debate, improve and deliver

proper requirements for their venture. From an article formed by the inventors of Agile methodologies, it can be seen that people in joint efforts with good communication and interplay skills can work at higher and more productive scales than if they work separately (Cockburn and Highsmith, 2001).

For this reason, good communication is in essence demanded with the specific end-goal to deliver success in Agile methodologies. By the same token, it will help in creating requirements which can fulfill the clients' needs. In general, the prominence of Agile methods lies in highlighting group competencies as opposed to processes (Cockburn and Highsmith, 2001).

2.3.2 Understanding the Adoption and Usage of Agile Methods

Since the concentration of Agile initiatives is about cooperation with clients, it comes to no wonder that most reviews center around the efficiency and adoption of Agile techniques by helping those involved to embrace the strategies; for example the production of information for those attempting to utilize XP (Bahli and Abou Zeid, 2005). Furthermore, the people factor (such as team capability) was observed to be an important one as far as the use of Agile techniques is concerned (Chow and Cao, 2008).

In addition, from its initial presentation, the technique has been explained as a group of approaches that focus on people and social aspects (Cockburn and Highsmith, 2001). Obviously, the technical facets cannot be set aside, and as one study describes, these are important (Chow and Cao, 2008). Chow and Cao (2008) identified the critical factors pertaining to the technical aspects, for example, the Agile delivery methodology and techniques. Despite the fact that the technical angles have been labeled as important, they rely on upon the people who are practicing them. The second most essential perspective, after the technical angle, is the people aspects including 'group capacity' (Chow and Cao, 2008).

In terms of the people factor, this is a piece of the Agile ecosystem (Cockburn and Highsmith, 2001) and the motivation behind why their abilities, skills, experiences and communication have become the main reason for the adoption of Agile methods (Lindvall et al., 2002). Nevertheless, it was also found that

professionalism and experience went about as a deterrent to actualizing Agile (Krasteva and Ilieva, 2008). As one of the studies suggests, communication, commitment, cooperation and adaptability are seen as four professional skills required for the adoption of Agile strategies (Melnik et al., 2005). They are essential in light of the fact that most Agile activities demand and include these four aspects.

Consumer satisfaction and good communication were among other critical factors in the reviews and regarded as vital for Agile adoption. Client satisfaction is the intended outcome while performing Agile. The technical viewpoint still exists in practices, for example, in pair programming, yet it additionally needs considerable collaboration and expert individuals to materialize it.

One investigation (Robinson and Sharp, 2005b) states that the technical practices of XP themselves are in reality social up to a high level. In addition, Agile methodologies do not depend on dense documentation as much as the implicit understanding of the group. However, people continually inquire how the group can utilize the strategy in the event that it does not have any documentation to correspond with the work they are doing. This is one issue that might be raised by the early adopters of Agile strategies, especially under conditions where few reviews on the strategy can be found.

It is known that Agile methodologies utilize documents just when they are in real need of them , and that this need is a justifiable and realistic one. One approach to guarantee that the strategy actually works is by guaranteeing that the team members additionally obtain the skills and knowledge as expressed in one review, “without these kinds of persons, the chosen approach would probably have little possibility to success” (Hilkka et al., 2005). Such information creation has also been depicted as a factor in the adoption of Extreme Programming, one of the Agile techniques' practices (Bahli and Abou Zeid, 2005) as it is difficult for the group to depend just on the knowledge they share. Another review proposed that the group ought to obtain the skills in demand in programming development while being motivated at the same time (Madeyski and Biela, 2007). In addition to the knowledge perspectives, (i) preparation, (ii) administration, (iii) involvement and (iv) access to outer resources have been the

other factors regarded as affecting the implementation of the Agile programming strategy (Livermore, 2007).

Moreover, there are a few viewpoints to be considered if an organization is to embrace and present Agile techniques. The environment, settings, culture, and administration support are among the qualities that are most critical for the organization that is thinking about adopting Agile. One review researched the effect of organizational culture (Strode et al., 2008b) and demonstrated that the this environment can impact the successful utilization of Agile techniques. In the review, eight of the projects were conducted in New Zealand and one in the United Kingdom. Also, Strode et al. in 2008 debate the organizational culture, and the related factors can help in deciding about their appropriateness for practitioners to consider in Agile methodologies. Achievement in adopting Agile is reliant on the biological system of Agile and the people who participate in it.

From the reviews covered so far, one can see that not all societies and environments are appropriate for Agile use. Lindvall et al. (2002) expressed that “to be Agile is a cultural thing. If the culture is not right, the organization cannot be Agile". There was another review, which discovered that no significant outcomes can be yielded in terms of organizational culture with Agile techniques adoption (Chow and Cao, 2008).Aside from understanding the elements and environmental appropriateness when utilizing Agile, another essential aspect of the organizational factor is to consider the characteristics of ventures and groups in the organization. In this respect, two reviews state that these concepts can also be addressed in connection to organizational factors (Krasteva and Ilieva, 2008, Strode et al., 2008a). Nevertheless, Chow and Cao (2008) disagree on this matter And regard it as insignificant. Another work specified that Agile is appropriate just for small and co-located groups, in the meantime offering recommendations to large organizations to adopt Agile strategies (Elshamy and Elssamadisy, 2007).

Conversely, another review detected no considerable connection between successful utilization and group size (Livermore, 2007) and only observed a negative relationship between the organization size and the execution of Agile. In addition, this review demonstrates that large organizations may confront challenges in practicing

Agile as opposed to small organizations. Based on the experiences of the enterprises in a review, the authors maintain that Agile can likewise be utilized in large organization; however, it is best in small and co-located teams (Lindvall et al., 2004).There was one study which clarified that size turns into an issue since more people means harder correspondence (Lindvall et al., 2002). As far as co-location is concerned, (Misra et al., 2009) discovered critical negative outcomes, while Livermore did not locate any major outcomes for this factor either (Livermore, 2007).

In the same way, the organizational setting which shapes the Agile work environment is vital for the appropriateness of Agile techniques. The open-plan office is, for the most part, favored in Agile Methods (Law and Charron, 2005). Nevertheless, two reviews characterize conflicts about the work environment or the organizational setting aspect (Chow and Cao, 2008; Chan and Thong, 2007). As it is shown in the literature review above, the nature of Agile practice which stresses communication demands the Agile group to co-find.

In terms of distributed software development, the practice is viewed as a restriction in Agile programming development and hard to practice (Turk et al., 2002). Conversely, another research found that Agile strategies can assist in decreasing three sorts of 'distance' or issues: temporal, geographical, and socio-cultural, all of which are recognized in universal software advancement (Holstrom et al., 2006). The review recommended that to utilize Agile techniques in universal or distributed development environments, clients should truly comprehend the attributes of Agile strategies. As far as the socio-cultural issues go, the review found that the language viewpoint can be an impediment in many ventures (Holstrom et al., 2006).

Evidence has been provided by Ramesh et al. (2006) for how distributed software development can turn into Agile; whereas, another one (Maria and Casper, 2006) anticipates the advantages of having a distributed software development environment utilizing Agile. These reviews show that Agile can tackle issues in connection to distributed development. Moreover, it was likewise found that cultural variation can be handled by applying Agile techniques (Nisar and Hameed, 2004). From the reviews introduced in this section, it can be inferred that Agile is not for

everybody (Cockburn and Highsmith, 2001) and that the adoption must be adjusted after the appropriateness of the Agile ecosystem is determined.

From the literature, it can be seen that it is essential to adjust the Agile methods for adoption in connection to appropriation, comprehension organizational matters. Agile methodologies can have certain deficiencies that should be handled so as to harvest the benefits of the approach. The utilization of Agile methodologies needs to be tailored. A review illustrated the need to adjust such utilization of Agile strategies and plan-driven techniques (Boehm and Turner, 2003). Previously, Boehm and Turner (2003) found that practitioners have to examine their environment and organizational capabilities.

At the beginning of adopting the Agile methodology, clients may be provided with the related strategies and the plan-driven techniques. On account of fitting Agile and plan-driven properties, organizations can allude to a polar graph created by Boehm and Turner (2003). There are five axes which incorporate personnel, dynamism, size, criticality and culture. The culture axis mirrors that Agile methodologies will have a better possibility of accomplishment in "a culture of 'flourishes and chaos' than one that in 'blossoms with the request', while the inverse is valid for the plan-driven strategies (Boehm and Turner, 2003). As far as dynamism, Agile methods perform well in both high- and low-change rates; however, plan- driven works best with the latter.

The polar chart introduced by Boehm and Turner (2003) does not involve the technical angles in fitting the utilization of Agile techniques in enterprises. Nonetheless, it is expressed in the review that plan-driven techniques work effectively with clients that are having both high and low skills levels; yet, Agile requires people with more abilities (Boehm and Turner, 2003). There was a survey regarding Agile adoption directed by Scott Ambler from 2006 to 2008. The information from the survey was provided descriptively and, looking from the early adoption survey perspective (Ambler, 2006), he found that most organizations gained positive outcomes upon adopting Agile strategies or methodologies. In view of these outcomes, he states that obviously piloting Agile ventures can moderate any potential hazard when organizations adopt the techniques.

While the adoption of an Agile study in 2006 by Ambler demonstrates that over half of the respondents did not adopt Agile, 65% of the respondents concurred that they have embraced one or some of the Agile methods. From here, Ambler deduced that Agile techniques are adopted more, while further noticing that many groups are doing a subset of XP practices, for example, refactoring, test-first plan and few others. This means that organizations are fitting Agile strategies to their environment by adopting the techniques before embracing the entire methodology. The adoption rates end up plainly higher in 2007 and 2008. From the outcomes, Ambler expressed that it is low-risk to consider adopting Agile strategies (Ambler, 2008).

Fitting Agile methodologies is depicted in another research (Lindvall et al., 2004), which states that the environments and conditions for Agile methodologies are as yet misty and need more examination. The study demands that fitting Agile into a system turn into a full requirement, adding that presenting Agile in a large organization without broad fitting is, for the most part, infeasible. In the end, the review deduced that Agile can be utilized in large organizations and, in particular, for little and co-located teams. Organizations need to merge Agile ventures into their environment and new practices with existing practices. The review demonstrates that fitting the Agile strategies to meet the appropriateness of the organizational environment and ventures is actually important.

2.3.3. Factors In Adopting Agile

The achievement and failure factors in adopting Agile methods have been studied by many researchers (Abdalhamid and Mishra, 2017a; Chow and Cao, 2008). In this section, those related to achievement will be exhibited first, later to move on to the previous attempts to identify the failure components. In 1996, inquiry began when Belassi and Tukel came up with a new frame to determine the vital success and failure aspects of Agile by suggesting a new diagram for essential elements and the impact of these elements on the execution of projects. They divided the factors into four groups using an empirical study, namely project, managers and colleagues, organizations, and the environment. The outcomes demonstrated that venture directors, administrative skills, colleagues and their technical backgrounds, venture attributes and

environmental factors are as practical and, hence, important as any other organizational factors, and that the criticality of these components vary amongst industries (Belassi and Tukel, 1996).

There were many other studies between 1996 and 2006. A survey was made by Mahanti (2006) with the aim to examine real difficulties in Agile practices adoption by companies. Accordingly, efficient selection of Agile methods includes management, instruction, and support, in regard to outside processes, starting pilot exercises, reporting on them, corresponding changes, and keeping up agility. Put differently, the achievement of adopting Agile techniques is particularly related to how the new system is introduced within a company (Mahanti, 2006).

Also, there was a study by Tsun chow, Dac-Buu Cao in 2008 to examine the achievement factors in Agile programming projects based on a quantitative approach. This work was an investigation of 109 Agile ventures from 25 countries all over the world and utilized multiple regression methods, with the results stating that 10 out of 48 speculations were reinforced among these establishments and perceived as primary achievement factors for Agile programming projects. In detail, the factors include: delivery of strategy, Agile programming engineering methods and group ability. It was found that differing achievement components should be further elaborated and the success of Agile ventures with different strategies further demonstrated (Chow and Cao, 2008). Then, Stankovic et al. (2013) continued with the survey by Chow and Cao (2008) in an attempt to check the rating of vital achievement factors defined by them. They utilized regression analysis for the accumulated data which displayed three more factors that could possibly be considered as fundamental achievement factors (Stankovic et al, 2013).

There was another investigation done by Misra et al. (2009), who built up a speculative and hypothetical achievement components frame to identify the examination questions. To confirm the hypotheses, information analysis methods were used with a broad scale-based technique to make the review, including respondents who practice Agile programming improvement and who had been already been involved in applying plan-driven programming. According to the results, there were 9

out of 14 anticipated factors to be basically related to success, namely: client cooperation, client content, client obligation, conclusion time, company culture, people's characteristics, societal culture, and preparing and learning (Misra et al, 2009).

In addition, another examination led by Zulkefi, Saadiah and Noor in 2010 reviewed the literature to assemble data from past reviews, with the outcome that client participation, communication, least change requirement, companies' culture, active testing, clarity, allocation of time, client cooperation and code checking define the success in Agile programming development methods (Saadiah et al, 2010).

Wan and Wang in 2010 have clarified a "P organization achievement factors" model in the following way: "leading (recognition of summit heads, contribution of summit leaders), company (clarifying vision, building the Agile hierarchical culture, substituting the technique for organization), gadgets and technology (setting up the basic apparatuses and foundation, using arrangement plans and other created outline systems, using programming reuse technology), appropriate import (choosing pertinent import project, brilliance execution staff, selecting fitting Agile procedure practice), planning and teaching (appropriate understanding and capacity of Agile methodologies, enhancing the professional abilities of the individual), calculating success (flexible and inventive development methodology, fast response to the demand, successfully developing learning association) which they checked them by an inquiry in P organization" (Wan and Wang, 2010). They found that teaching and planning have a positive impact in Agile advancement. As a result, Agile strategies should be set up in the culture of Agile and with due interest in the designing and utilization of the related technology.

There were two case studies made by Melo et al. in the industry in 2011 analysing the information from two ventures. They also examined three reviews and showed the basic aspects of the most related factors, which included: item (reuse, programming attributes), venture(resource limitations, agenda, group structure, correspondence), crew (group experience and ambition) and procedure (client involvement, daily construction, documentation, early prototyping, incremental and

iterative improvement, present-day programming work on, programming language abstraction, programming strategies, tool use) (Melo et al, 2011). According to this work, there are a few variables influencing the capacity of Agile strategies; these elements are group break-down and distribution, outer dependencies, and staff turn-over.

Another review by Asnawi et al. (2011) 13 members including CEOs, ventures managers, originators, and engineers involved in Agile adoption. Their review showed that social and human viewpoints are critical when Agile strategies are to be utilized. They utilized qualitative semi-structure meetings, which pointed to the issues and difficulties faced during the process of adoption as attitudes, information, venture, people, learning exchange, communication, administration participation, technical angles, and organizational structure (Asnawi et al., 2011).

One study by Kumar and Goe in 2012 was directed to present and clarify the factors considered by programming professionals while adopting Agile strategies, as well as the impacts of Agile adoption techniques on clients and business while practicing in Agile. Their study proposed six premises: effect of group size, effect of requirements collection for Agile methods, efficient requirement capturing process, time needed to solve issues and effect of insufficient corresponding time with the client in software improvement. The outcome of this research showed that Agile adoption can increase the output of companies and also raise the clients' level of content (Kumar and Goel, 2012).

In terms of failure-related research in software development, works are commonly depended on lessons learned from specific sorts of ventures; however, they are for the most part similar and generalized (Coram and Bohner, 2005). For example, in 1999 Reel concentrated more on general software development ventures and collected 10 indications of programming development ventures failure, no less than seven of which are located even before an outline is produced or a line of code is written (Reel, 1999).

In addition, the issues in transitioning organizations to Agile methods was studied by Cohn and Ford in 2003, whereas in 2004 Larman debates in detail the errors and confusions occurring while developing Agile projects (Larman, 2004). There was a research done by Boehm and Turner in 2005, which confirms administration challenges during the implementation of Agile projects; whereas another study was conducted by Nerur et al. in 2005 that includes issues from the administrative angle as well as the people, process, and technology aspects of the transition to Agile ventures. In light of the previously mentioned literature, failure can be classified into four groups: organizational, individuals, procedural, and technical

Moreover, in 2008 Vijayasarathy and Turk show that a portion of the factors that cause failure in Agile projects involve the absence of preparing and associate support, inexperience regarding the methodologies, absence of offices for pair programming, people's resistance, and depending just on financial assessment criteria. Another concern is administrative disregard and organizational impedance to change. In this respect, Chow and Cao (2008) discuss the failure factors by classifying them into four group which are: organizational, people, process, and technical (Chow and Cao, 2008).

2.4. The Combination of Agile Methods and Traditional Methods

The product procedure comes in various shapes, such as a stiff plan-driven approach or as slim Agile strategy, and since the 2000's two fundamental streams have come to exist. Traditional procedures intend to address the entire software venture lifecycle, e.g. by giving inclusive guidelines, venture planning designs, standardized processes, and interfaces to additional company processes. This occurs while Agile strategies go for reducing the software procedure to its base to stay away from "bureaucracy", and to supply clients with just the required number of rules and guidelines needed to carry out a venture (Theochari et al., 2015).

In software development processes, there is a requirement for new methods to deal with the process. With this in context, and as per Raith et al. (2016) and Soundararajan and Arthur (2009), a delicate organized Frame Work (FW) can be made as a mix of Agile standards and traditional methods to address the primary issues in

regards to adjustment to change and related to large-scale organizations and systems. This FW is a mix of two sections of software organized requirements with an Agile attitude, prompting the use of Agile methods model and traditional improvement procedures which can be appropriate for both small- and large-scale systems.

For instance, Solinski and Petersen (2014) outlined Scrum and XP as the most well-known and adopted methods, additionally suggesting mixes of Waterfall/XP, and Scrum/XP as the most widely and ordinary combinations to emerge. Another late published research also signifies a circumstance in which traditional and Agile methodologies exist together and make the greater part of practically utilized hybrid approaches (Kuhrmann and Fernandez, 2015; Kuhrmann et al., 2014).

2.5. Adoption of Agile Methods By Software Development Organizations

It goes without saying that, nowadays, organizations are tested by fast-changing and constantly improving business surroundings and fully-experienced clients with continually rising anticipations so as to reduce the time and obtain the best possible services (Cooke, 2010, Gunasekaran, 2001).

During the development of any software, there are always the same challenges to be faced as regards uncertainty. This is because from the beginning of the developing venture, it is hard to tell if a project's requirements were in deed identified in the right way (Abdalhamid and Mishra, 2017b). There is also the possibility that the requirements change during the process of development. Such problems can be solved by using Agile development methods, and one of the reasons that forced many companies to announce an adoption of Agile methodologies is that the specialists related to Agile made strong and compelling claims with regards to the benifits of using Agile (Thakur and Kaur, 2013).

The general agreement is that software is a vital part of everyday life, and in proceeding with the development of better software, the industry has brought about new organizations (small- and medium-size) over the past decade. The quick pace with which such organizations are established makes them face a few downsides, for

example, casualness in the product improvement procedure and technological insufficiencies. Software development organizations have found in Agile methods a conceivable answer for developing their practices and procedures (Sarmiento and Vasquez, 2012).

Small- and medium-size programming improvement companies are shifting to Agile methods in light of the fact that these approaches were considered for small groups without complex organizational structures (Sarmiento and Vasquez, 2012). As indicated by Forrester's report (2008), design-driven methods procedures, for example Waterfall, lead to waste, add risk and contain other hindrances against increasing quality and predictability.

The Agile programming advancement strategies have been considered as an extreme-quality new model for product improvement, with small work groups without order or bureaucracy (Nerur et al., 2005; Forrester, 2007; Dybå Dingsøyr, 2009; Leffingwell, 2007; Nerur, 2005) Meanwhile, Agile approaches are also effective in other situations, where large and complicated software items frequently require orderly training with the required extra procedure to guarantee achievement. Agile designing is a comparatively unofficial process with numerous but small undertakings to guarantee ideal delivery outcomes (Cohn and Ford, 2003).

The pertinence of Agile approaches to vast organizations is frequently regarded as challenging (Simons, 2002; Allen,2001). In very large scale projects, this matter becomes even more so as the complication of the application space is frequently beyond the experience or expertise of certain clients', not to mention the developers'. For this reason, there is an evident requirement for a continued client engagement in large-scale complex projects, as also the main key for XP project achievement (Cao et al, 2004).

For quite some time, organizations have been progressively deploying Agile methods in their product development ventures. The fact is that Agile methods were initially intended for use in small, single-group projects (Boehm and Turner, 2005). Nonetheless, their appeared and potential advantages have made them alluring

likewise outside this situation, especially both for larger projects and in bigger organizations (Dikert and Paasivaara, 2016).

2.6. Adoption of Agile Methodologies In Large Projects

Large and complex programming goods frequently demand systematic discipline with the wanted extra procedure to guarantee achievement. Moreover, Agile outlining is a comparatively unofficial process with numerous little tasks to guarantee ideal delivery outcomes (Cohn and Ford, 2003). The appropriateness of Agile methodologies to large organizations is frequently regarded as defying (Simons,2002; Allen,2001). In large ventures, this issue is intensified as the complication of the application domain is frequently beyond the experience or expert of a small number of clients, not to mention the developers

The fact is that, in the Agile approach, achieving communication and cooperation among the members who appreciate and trust each other is an essential issue for the success. Some of the Agile methods have individual-centered activities, for example, programmable pair in XP. In this case, the selection of appropriate personnel managers is vital to provide them with the necessary training and guidance and create a set of business practices that promote excellence in the process. These types of activities, especially for senior amd traditional developers, may not come easily. In addition, customers should be involved in the process of development which can be an issue by itself in some cases (Gandomani et al., 2013).

Presently, organizations are progressively redistributing Agile methods in their software development ventures. Nevertheless, a methodical and large-scale adoption of Agile practices in companies is as yet evasive, and it is principally centered around venture-level exercises (Salo and Abrhamsson, 2005). Large rate and complex ventures confront active changes in necessities and time-to-market demand. To reduce such hazards, endeavors have been made in using the XP in large ventures, though with conflicitng results (Cao et al., 2004).

Even though some Agile methods work well with small and medium projects, it can be difficult to use them in large projects for different reasons, the main one being

that some Agile methods are hard to adopt for large projects such as team size. These methods work effectively with small-size ventures. At this point, some of the attempts are shown to improve the XP and Scrum to work with large projects.

2.6.1. Extreme Programming (XP)

According to Jacobi and Rumpe (2001), there is a way of improving the Extreme Programming (XP) approach by using elements of various-leveled strategy. They agreed to choose the XP approach to build the hierarchical structure, While maintaining the lightweight methodology, it becomes possible to restructure large projects to a group of small XP projects with still the same goal to achieve. However, there are five main principles in this various-leveled approach generated as the result of mixing and adjustment of each corresponding XP standard with those from the hierarchical reorganization approach, and these principles are: Client/worker involvement for aloft admission, using clear and tiny interfaces to divide the system, each sub-venture has a sub-objective of the primary target, 80% resolutions and after that incremental development, and well-orderly with venture groups and directing comity.

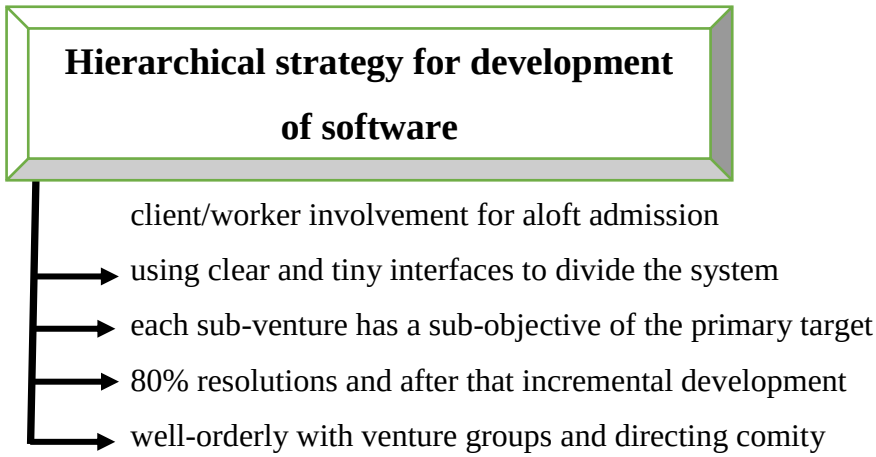

Figure 1. The five main standards of the progressive approach. Adopted from (Jacobi and Rumpe,2001)

2.6.2. Agile Scrum Methodology

In terms of adoption, the challenge will be in transitioning a medium-to-large-scale organization to Scrum since it may take more than one year for full implementation to make the transition from the traditional method based on the waterfall to Scrum framework as it requires major changes. To manage the process of such adoption effectively and maintain the change, it will be helpful to use a formal model change, as described in what follows:

According to Weidner (2014), the organization should develop a strategy for change and, then, measure the progress against the plan. While such an Agile process is being implemented, it should monitor the progress against the plan, and the plan must be adapted based on observations and/or in response to unintended consequences. At this stage, one may pose the question as to what can be done to make the adoption of scrum little bit easier.

To answer this, initial efforts should be focused towards the evaluation of the organization ready for agility, and provide initial training of the early participants. By looking at the accumulation of products of the initial projects, the first pilot project can then be specified. This would highlight the positive benefits to improve the lightness software movement within the organization, thus increasing the interest and excitement for others to try out the new process (Weidner, 2014).

CHAPTER 3

RESEARCH METHODOLOGY

This chapter explains the aims of the study as detailed in the introduction, with further elaborations on how they are achieved. In term of the programs used to carry out the research, we will use a questionnaire along with Snap10 Evaluation edition program and Google form.

3.1. Research Overview

3.1.1. Objectives of the Study

Regarding what section 1.2 has provided, the main aim of this research is to design a questionnaire that contains a variety of items to examine the use of Agile approaches in general in addition to the use of Agile methods in small- and medium-scale software development organizations, the factors that influence the decision to choose one method over the others for a particular project and, finally, the success and failure factors that can help to make the process of adopting Agile methods easier and more productive.

3.1.2. Research Questions

Q1.which of the Agile methods are used more than the others in large and complex projects?

Q2. Is the decision of choosing one of the Agile methods over the others affected by the type of the project - small, medium, or large in scale?

Q3. What are the factors that can make the process of adopting Agile methods successful?

Q4. What are the factors that can make the process of adopting Agile methods fail?

Q5. What are the benefits that can be gained by adopting Agile methods?

3.1.3. Achieving the Objectives

In order to achieve the objectives in section 3.1.1, a survey is carried out using Snap10 Evaluation edition software to create a questionnaire form which allowed us to explore the factors influencing the decision to choose one method over another for a particular project on top of some results for the remainder of the objectives set forth. A Website was designed to collect responses for the questionnaire using Google forms as appears in the address: https://drive.google.com/drive/my-drive .

The questionnaire was divided into six sections, each containing a number of questions as follows:

- Section I Background Information
- Section II Characteristics of Projects That Adopted Agile
- Section III The Success Factors in Adopting Agile Methods
- Section IV The Failure Factors in Adopting Agile Methods
- Section V The Acceptance of Agile
- Section VI Comments

Section I - Background Information

This section contains questions that allow us to know about the type of firms, such as small or medium in size, as well as information about the level of experience they have in using Agile through a series of questions related to the number of projects

they had developed with Agile and if the firms are CMMI-certified (The Capability Maturity Model Integration) or not.

Section II - Agile-Adopting Projects' Characteristics

It is important to know about the types as well as the nature of the projects that need to be developed using Agile methods, that is whether small, medium or large in size. In addition, there are many methods for Agile, making it necessary to know which method is which specific project. For this, section II was developed to gather such information.

Section III - The Success Factors of Adopting Agile Methods

The purpose of this section is to find out the success factors in adopting Agile methods in projects. A five-point Likert scale was utilized to reflect the impressions of the respondents. Achievement factors were classified into five classifications: organizational, people, process, technical, and project. To find out which factors can affect the process of Agile adoption by software development organizations, a number of hypotheses was developed in this research.

However, to define the research hypotheses of success factors, certain related attributes are needed to delineate the general view of 'success2 for a specific venture. In this respect, Cohn and Ford (2003) and Lindvall et al. (2004) recommend these criteria: quality (i.e., providing a working item), scope (meeting all prerequisites set by the client), timeliness, and Cost.

In addition, Misra et al. (2009) identified decreased delivery agenda and increased return on investment (ROI) as success attributes, adding that output, functionality and client satisfaction can also be seen as quality criteria. These features are listed in Table 1.

Table 1. Success Attributes

Dimension	Attribute
Overall acknowledged level of success	1. Quality (producing good products or venture outcomes, client satisfaction) 2. Scope (dealing with requirements in a better way) 3. Time and cost estimation 4. Reducing the delivery schedules 5. Increasing return on investment

In light of existing literature, a preparatory rundown of potential basic achievement factors of Agile ventures were identified and arranged, as results there are about 25 hypotheses related to five dimensions of the success factors as shown in Figures 2 to 6. For more detials these hypotheses are listed below:

1. Hypotheses associated with the organization aspect

H 1. The culture of organizations is a crucial success determinant that adds to effective Agile product development ventures in terms of: (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H 2. Team structure is a crucial success determinant that adds value to effective Agile product development ventures with respect to: (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H 3. Management support is a crucial success determinant that improves the effective Agile product development ventures as to: (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e)

customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H 4. Team organization is a crucial success determinant in effective Agile product development ventures in regarding: (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H 5. Maintaining agility is a crucial success determinant that adds value to effective Agile product development ventures in regarding: (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H 6. Acceptance of Agile methodology is universally is a crucial success determinant that adds to the effective Agile product development ventures in regarding to: (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

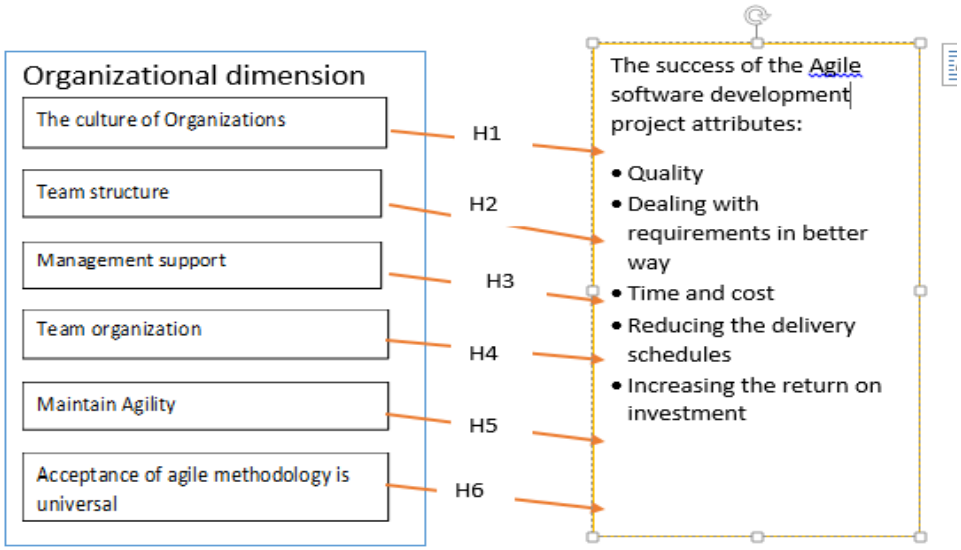

Figure 2. The research model for success factors in terms of organizational dimension

2. Hypotheses associated with the people aspect

H 7. Customers' dominant issues is a crucial success determinant that adds to effective Agile product development ventures in regard to: : (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H 8. Having team members with high capabilities is a crucial success determinant that adds to the value of effective Agile product development ventures in: : (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H 9. Practice and learning is a crucial success determinant that improves effective Agile product development ventures for: : (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H 10. Communication and arbitration is a important and adds to the effective Agile product development ventures for: : (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H 11. Encouragement is a crucial success determinant that adds to the value of effective Agile product development ventures in: : (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H 12. Project champion is a crucial success determinant that adds to the effective Agile product development ventures in: : (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer

satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

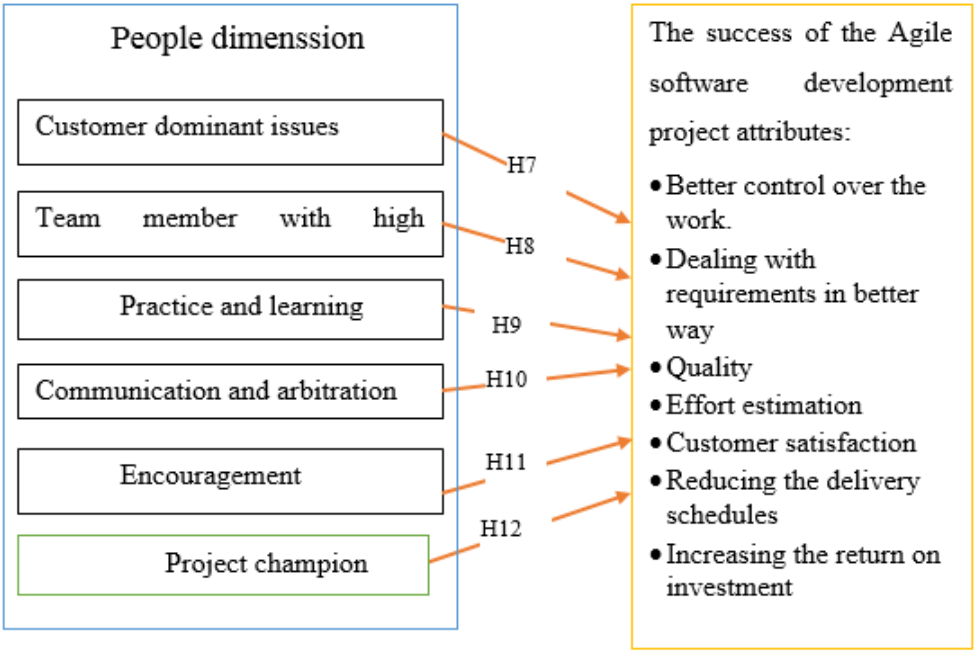

Figure 3. The research model for success factors in terms of people dimension

3. Hypotheses associated with the technical aspect

H 13. Assigning essential features first is a crucial success determinant that adds to the effective Agile product development ventures for: : (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H 14. Frequent delivery of software is a crucial success determinant that adds to the value of effective Agile product development ventures in: : (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H 15. High competence for team and organizational issues is a crucial success determinant that adds to the value of effective Agile product development ventures in:

: (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H 16. The use of tools is a crucial success determinant that adds to the value of effective Agile product development ventures in: : (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H 17. Correct integration testing is a crucial success determinant that adds to the value of effective Agile product development ventures in: : (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

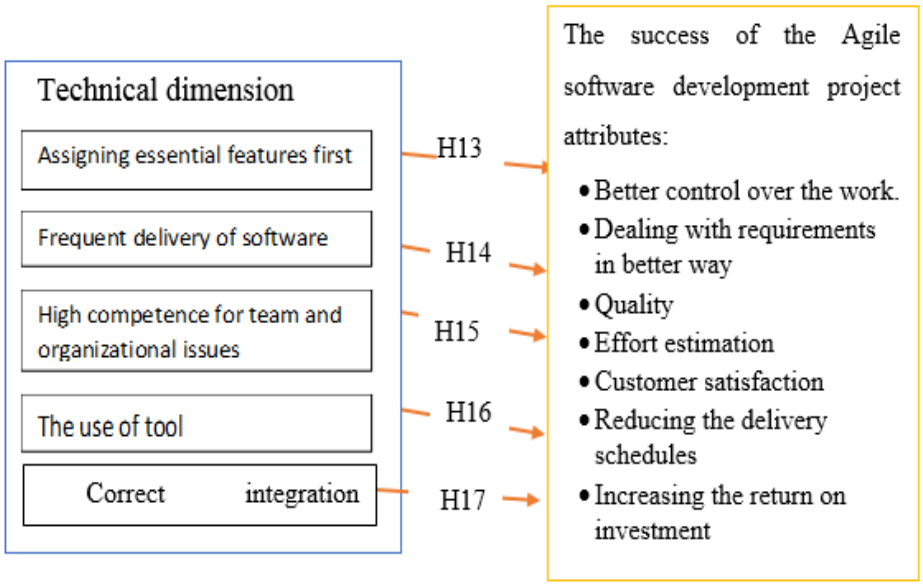

Figure 4. The research model for success factors in terms of technical dimension

4. Hypotheses associated with the project aspect

H 18. Project category is a crucial success determinant that adds to the value of effective Agile product development ventures in: : (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H 19. Smaller size team is a crucial success determinant that adds to the value of effective Agile product development ventures for: : (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H 20. Agenda is a crucial success determinant that adds to the value of effective Agile product development ventures regarding: (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H 21. Least changes of requirement is a crucial success determinant that adds to the value of effective Agile product development ventures regarding: : (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H 22. Projects within advance hazard analysis performed is a crucial success determinant that adds to the value of effective Agile product development ventures in: : (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

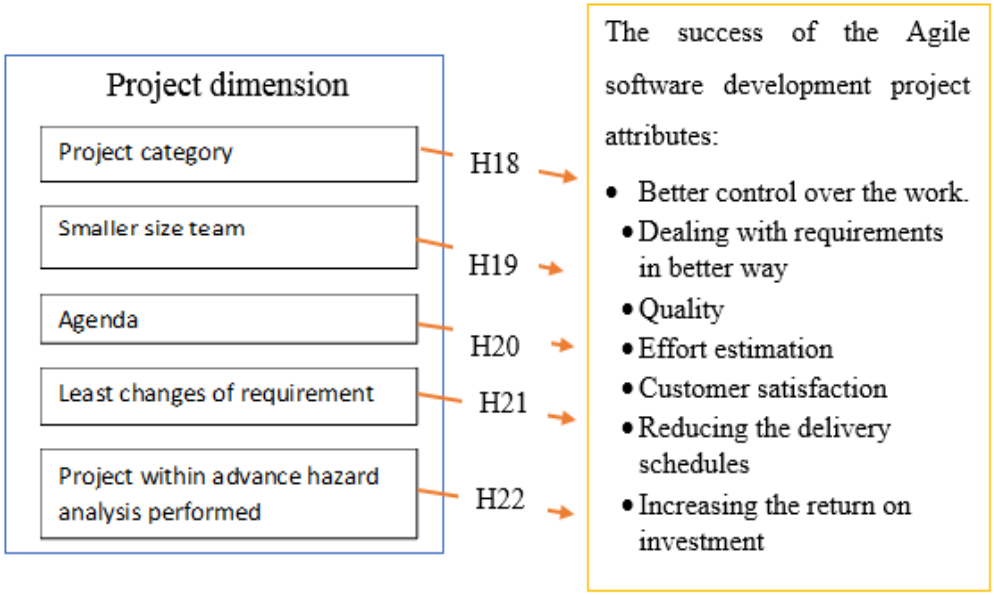

Figure 5. The research model for success factors in terms of project dimension

5. Hypotheses associated with the process aspect

H 23. Clarity performed is a crucial success determinant that adds to the effective Agile product development ventures in regarding: : (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H 24. Strong customer performed is a crucial success determinant that adds to the effective Agile product development ventures in regarding: : (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H 25. Efficient requirements gathering method performed is a crucial success determinant that adds to the effective Agile product development ventures in regarding: : (a) better control over the work, (b) dealing with requirements in a better

way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

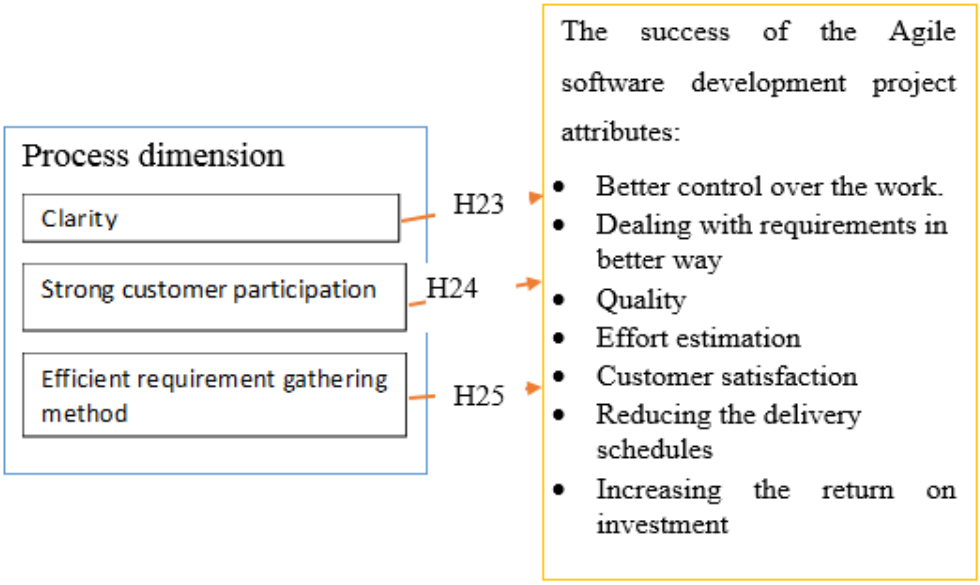

Figure 6. The research model for success factors in terms of process dimension

Section IV - The Failure Factors of Adopting Agile Methods

Finding out about the failure factors of adopting Agile methods in developing projects is one of our objectives, for which a five-point Likert scale was used to reflect the opinions of the respondents in this matter. Failure factors were classified into four categories: organizational, people, process, technical. A number of hypotheses was developed to find out which factor has the most impact on the process of adopting Agile methods. In light of existing literature, a preparatory rundown of potential basic failure factors of Agile ventures were identified and arranged, as results there are about 11 hypotheses related to four dimensions of failure factors as shown in Figures 7 to 10.

1. Hypotheses associated with the organization aspect

H1.Absence of management support in projects can be the main reason for failure in ASD projects in terms of: (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H2. When the size of the organization is large, the possibility of ASD projects' failure is increased in terms of : (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H3. When the culture of the organization is regarded as traditional and political, the possibility of ASD projects' failure is increased in terms of: (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

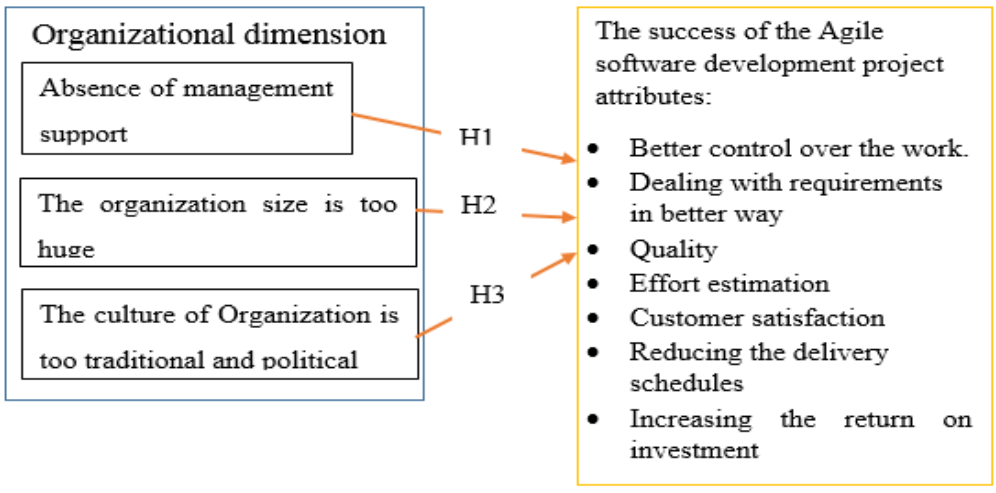

Figure 7. The research model for failure factors in terms of organizational dimension

2. Hypotheses associated with the people aspect

H4. Having a negative relationship with customers in projects can be a main reason for ASD projects to fail in terms of: (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H5.The absence of cooperation between the clients and while developing a project can be a main reason for ASD projects to fail in terms of: (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H6. When essential skill-sets are not provided, the possibility of failure of the ASD projects is increased in terms of: (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

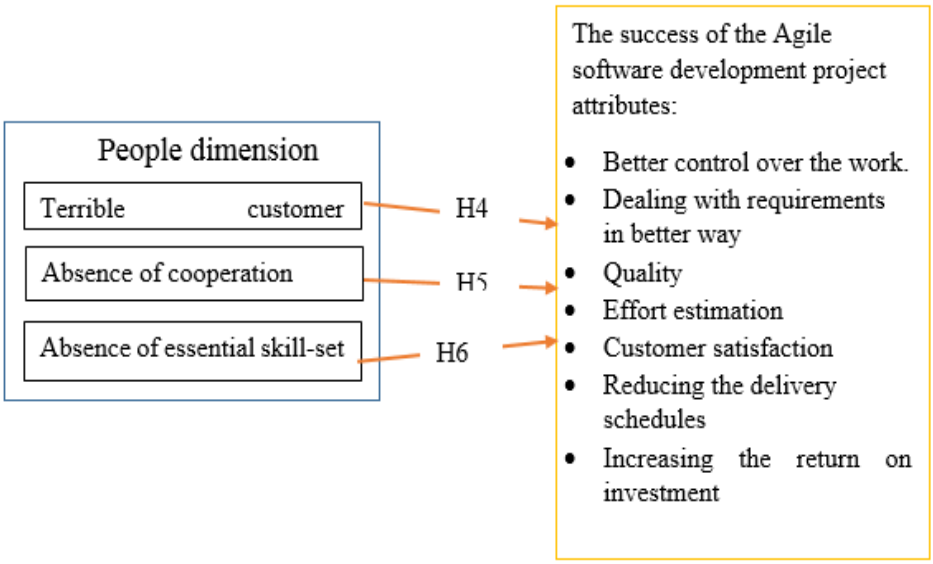

Figure 8. The research model for failure factors in terms of people dimension

3.Hypotheses associated with the process aspect

H7. Absence of customer's presence in projects can be a major reason for the failure of ASD projects in terms of: (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H8. Absence of tracking mechanisms during the Agile progress in projects can be a major reason for the failure of ASD projects in terms of: (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

.

H9. When the role of the customer is determined in projects, the possibility of the ASD projects' failure is increased in terms of: (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

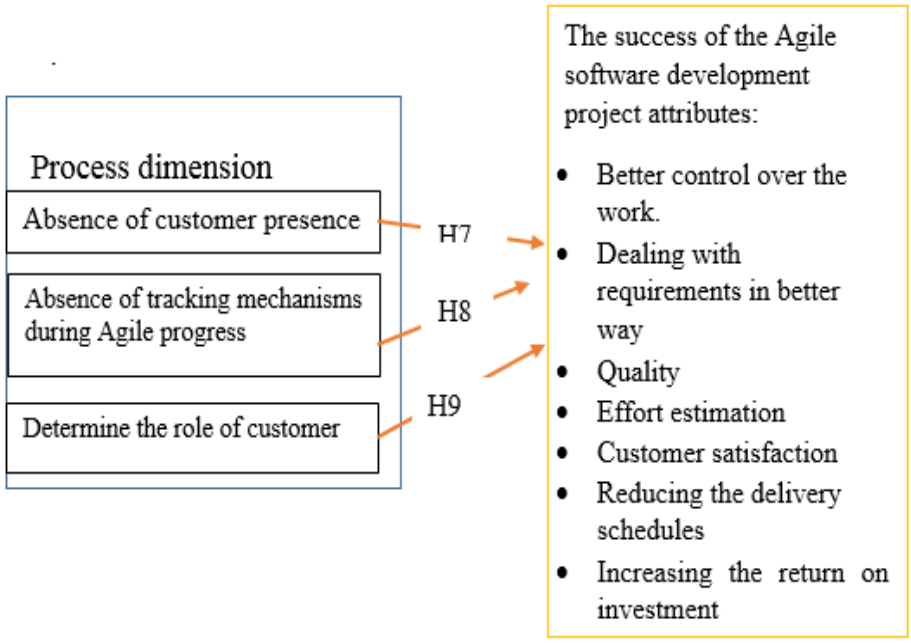

Figure 9. The research model for failure factors in terms of process dimension

4. Hypotheses associated with the technical aspect

H10. The absence of a full set of right Agile practices in projects can be a major reason for the failure of ASD projects in terms of: (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

H11. Inadequacy of technology and tools in projects can be a major reason for the failure of ASD projects in terms of: (a) better control over the work, (b) dealing with requirements in a better way (c) quality, (d) Effort estimation, (e) customer satisfaction, (f) reducing the delivery schedules, and (g) increasing return on investment.

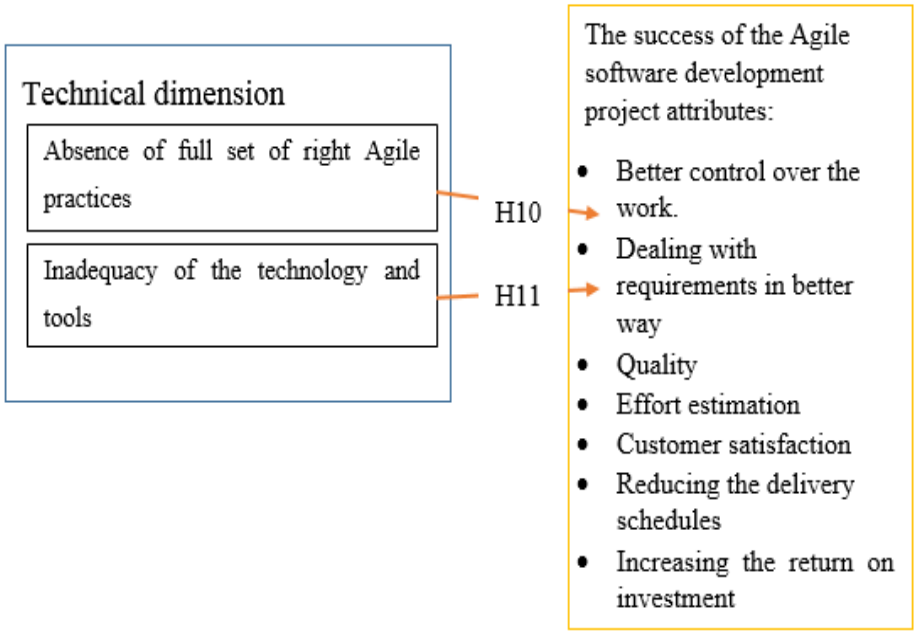

Figure 10. The research model for failure factors in terms of technical dimension

Section V - The Acceptance of Agile

Section V asks questions to gain a clear idea about the acceptance of Agile; in other words, why Agile methods are adopted in projects: is it because it can help companies to have more control over their projects, or because it copes and adapts better with changing user requirements. There may exist many other possible reasons that deserve further investigation. .

Section VI – Comments

This section provides the opportunity to better understand the responses from the participants by allowing for any additional comments they may have about the use of Agile methods in their work environment. The respondents were encouraged to enter any feedback which may be utilized for follow-up clarifications and elaborations, if important.

3.2. Software Introduction

3.2.1. The Snap10 Software

Snap program is a reliable and perceptive software for designing a questionnaire, and data publishing, collection, and analysis. It is strong and has the ability to analyze tables, charts, and descriptive statistics with the advantage of SQL database connectivity. The rating edition has 25 variables and 25 cases of restriction ("Snap 10 Professional").

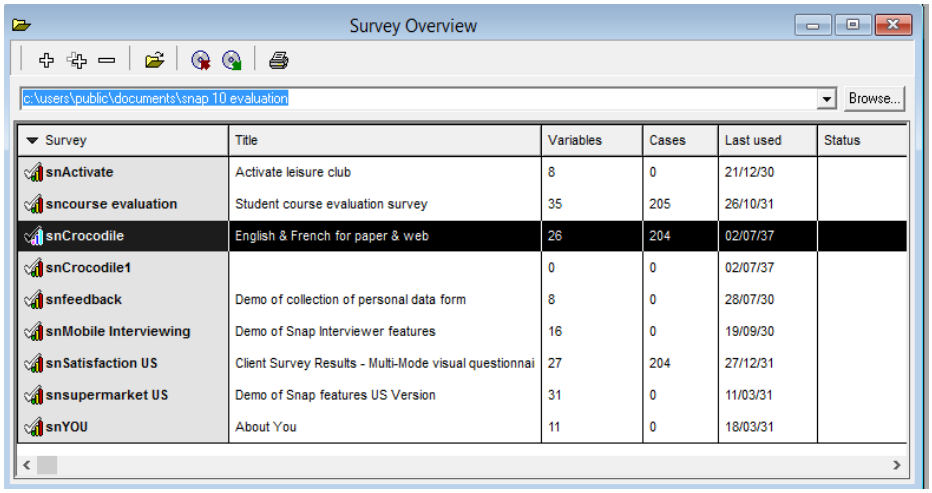

Figure 11. The main window in Snap 10 software

3.2.2. Google Form

Google Forms is a part of Google Drive, which is a tool designed to make surveys, tests, or Web input forms. Google frames is available to anyone and can be attached to a spreadsheet where one can track results and post them on the Web without any knowledge of programming (“Create a survey using Google Forms”).

3.2.2.1. Create a Survey Using Google Forms

Google Form can arrange occasions, make a review or survey, give understudies a test, or gather other data in a simple way. The frame can be made from Google drive or from into a current spreadsheet that allows recording the replies onto the form. To create a form, there are some steps that should be followed (“Create a survey using Google Forms”):

1. Go to docs.google.com/frames.
2. In the upper left, click Blank.
3. An automatically new frame will open.

In addition, all responses to the survey are automatically gathered in forms with information about real-time response and even charts.The data can be taken even further by viewing it all in sheets (“Collect and organize information big & small with Google form”).

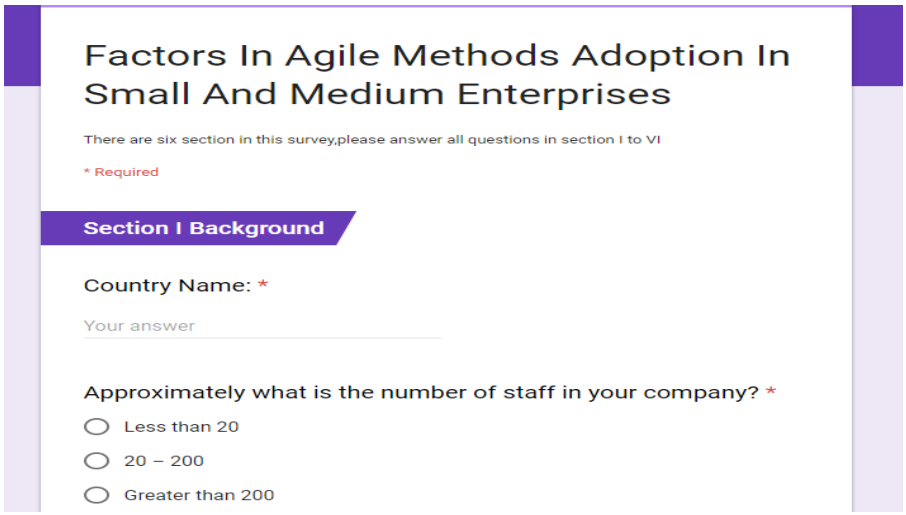

Figure 12. The first section in the survey web window

3.3. The Process of Designing the Questionnaire

The design of the poll can be divided into three components. As indicated by Burgess (2001), these components are:

a) Determine the inquiries to be made.

b) Select the question form of each inquiry and formulate the wording.

c) Design the inquiry sequence and layout for the questionnaire.

a. Determine the inquiries to be inquired

The research questions in section 1.3 were used to determine the type of inquiries that need to be made during the survey questions' drafting. For example, questions One and Two (which of the Agile methods are used more than the others in large and complex projects?, Is the decision of choosing one of the Agile methods over the others affected by the type of the project - small, medium, or large in scale? helped us to develop Section Two in the questionnaire, Characteristics of Projects That Adopted Agile) . While Questions Three and Four (what are the factors that can make the process of adoption Agile methods successful? what are the factors that can make

the process of adoption Agile methods fail?) were used to outline the items in Section Three and Four (Section III The Success Factors in Adopting Agile Methods, and Section IV The Failure Factors in Adopting Agile Methods).

b. Select the question form of each inquiry and formulate the wording

During the process of designing, various kinds of questions can be used, e.g. open vs. closed, single vs. multiple, ranking, and rating. However, most of the questions in the survey were multiple-choice type or rating. Some general rules were followed on the items' wording, mainly that the questions should be concise and clear, duplicate questions to be avoided, specific answers to be sought, and leading questions removed.

c. Design the inquiry sequence and layout for the questionnaire

The layout of the questions was made in the next step. Though, to decide on a layout and the sequence of questionnaire, step one should be completed successfully. Since the number of sections and titles were already decided by using the research questions in step one, the layout of our questionnaire is as follows:

Title of the questionnaire: Factors in Agile Methods Adoption In small And Medium Enterprises

Sections in the questionnaire:

- Section I Background Information
- Section II Agile-Adopting Projects' Characteristics
- Section III The Success Factors of Adopting Agile Methods
- Section IV The Failure Factors of Adopting Agile Methods
- Section V The Acceptance of Agile
- Section VI Comments

In addition, The final version of the questionnaire was obtained after many iterations of developing it. Firstly, it was distributed to prospective respondents and later on, it was improved 3-5 times based on their valuable suggestions. Then, one meeting committee members also reviewed it and finalized it.

3.4. Data Collection

To collect responses for the survey, the Web site (https://goo.gl /forms /yNAd6Aiqr ON2AKFF3) was employed to gather the data. All responses were stored immediately in Excel file. The target audience are individuals from companies that have adopted Agile in their ventures. The sample size is fifty two software development companies. The questionnaire was filled by software development companies from eight different countries, but most of the responses (57.7%) are from Turkey, followed by India Brazil, and Malta at 15.4%, 13.5% and 7.7%, respectively. The other countries present the lowest number of responses, standing at 1.9% as shown in Table 2.

Table 2. The countries and percentages as participants in the survey

Country Name	Frequency	Percent
Turkey	30	57.7
India	8	15.4
Brazil	7	13.5
Malta	4	7.7
Finland	1	1.9
Saudi Arabia	1	1.9
U.A.E	1	1.9
Total	52	100.0

In addition, The purpose of this research is to explore the adoption of Agile methods in small and medium enterprises; therefore, most of the responses were collected from companies of such size, namely 24 small companies and 11 medium

companies. This is while some responses were also collected from large companies, 17 firms to be exact, as part of the samples as shown in Table 3.

Table 3. The size of the companies

number of staff	Frequency	Percent	Company size
Less than 20	24	46.2	Small
Greater than 200	17	32.7	Large
20- 200	11	21.2	Medium
Total	52	100.0	

CHAPTER 4

DATA ANALYSIS AND RESULTS

To analyze the data, a statistical approach is adopted and, for this purpose, the IBM SPSS version 20 program has been used. This software, produced by SPSS Inc, and later obtained by the IBM Company in 2009 ("SPSS"), it is commonly utilized for logical batched and non-batched statistical analysis. Initially, descriptive statistics were performed to help in outlining the data obtained, based on the information collected as per the survey.

In terms of success and failure factors, since this research is an exploratory investigation to discover which factors can emphatically affect the achievement of Agile ventures, linear multiple regression analysis is regarded as suitable where the connection between multiple independent variables (success and failure factors) and the dependent variable (Agile project success) is concluded, and where the relative prescient significance of the independent factors can be established (Williams and Monge, 2001).

4.1. Reliability and validity test

Since this study is of exploratory nature, there is a need for a reliability analysis, for which purpose the Cronbach's alpha is used as it is the most well-known and efficient technique today to calculate inner consistency reliability (Rubin and Babbie, 1997). Higher estimations of Cronbach's alpha respectively demonstrate more noteworthy consistency in variance of the specimen test scores when the value exceeds 0.7 as the standard in a survey study.

Cronbach's alpha for a set of test scores in this research yield 0.8 for the failure factors and acceptance of Agiles, while for success factors this value stands at 0.9 as shown in Table 4. According to these results, there is an indication of clear accuracy of the statistical deductions from the information; that is, there are no issues with the inner consistency reliability tests.

Table 4. Reliability statistics summary

Items in	Number of items	Chronbach's alpha
Success factors	25	0.913
Failure factors	11	0.895
Acceptance of Agile	9	0.804

4.2. Data Analysis

There are six different sections regarding the survey data, and each section is analyzed and explained as in the following:

4.2.1. Background Information

This section provides information about the companies and individuals participating in the survey. The level of experience of the individuals in using Agile through a series of questions related to the number of projects they had developed using Agile and whether or not the organizations are CMMI-certified.

4.2.1.1. Respondents' Profile

This section presents the profile of respondents, based on which developers present the highest number at 18, followed by project managers and senior managers as second and third highest numbers at 13 and 7, respectively. Table 5 shows the break-down.

Table 5. Profile of responders

Job Title	Frequency	Percent
Developer	18	34.6
Project Manager	13	25.0
Senior Manager	7	13.5
Business Analyst	3	5.8
Designer	3	5.8
Tester	3	5.8
digital marketing EXP	1	1.9
Product Manager	1	1.9
Scrum Master	1	1.9
Subject Matter Expert	1	1.9
Total	52	100.0

4.2.1.2. Experience Levels

In terms of years of experience in Agile development, differences can be observed from organization to organization; nevertheless, the highest number stands at 17 years of experience which presents 1.9% of the sample. Most organizations have 3 to 10 years of experience in Agile development while the lowest is one year, representing 5.8% as shown in Table 6.

Table 6. Years of experiences in Agile development

Years of experience	Frequency	Percent
1	3	5.8
2	5	9.6
3	7	13.5
4	5	9.6
5	1	1.9
6	2	3.8
7	6	11.5
8	4	7.7
9	2	3.8
10	7	13.5
11	1	1.9
12	5	9.6
14	1	1.9
16	1	1.9
17	1	1.9
18	1	1.9
Total	52	100.0

4.2.1.3. Level of projects complexity

The purpose of this question is to find out the level of complexity in projects developed by these companies as represented by the undertaken projects' size. It is commonly acknowledged that, as this size increases, the complexity of a project is likely to increase as well. According to the results, most companies develop their projects with a medium level of complexity at 62%, followed by 38% at high levels of complexity as shown in Table 7.

Table 7. Level of project complexity

Level of projects_complexity	Frequency	Percent
Medium	32	62.0
High	20	38.0
Low	0	0.0
Total	52	100.0

4.2.1.4. Number of Agile projects

The range of the projects which used Agile methods is 1 to 30; that is at least 10 companies used Agile in four projects, and another nine companies developed five projects in this way. One company developed thirty projects, which is the highest number, and the rest developed two to twenty as shown in Table 8.

Table 8. Number of projects that used Agile

Number of projects_ have you used Agile methods	Frequency	Percent
4	10	19.2%
5	9	17.3%
1	5	9.6%
7	5	9.6%
10	4	7.7%
2	4	7.7%
3	4	7.7%
6	3	5.8%
8	3	5.8%
15	2	3.8%
20	2	3.8%
30	1	1.9%
Total	52	100

4.2.1.5. Companies CMMI Certified

Most firms that responded to the survey were not CMMI-certified by 67%, except 33% who have CMMI certification as shown in Table 9.

Table 9. Number of companies with CMMI certification

Organizations have_ CMMI certification	Frequency	Percent
No	35	67.0
Yes	17	33.0
Total	52	100.0

4.2.1.6. CMMI Level

According to the collected responses, 70% of the firms that have CMMI certification are in level three, which is the defined level, and only 17.6% in level five, which is the optimizing level. Also, 5.9% presents levels two and four as managed and quantitatively managed, respectively as shown in Table 10.

Table 10. Participant Companies' CMMI levels

Level of CMMI	Frequency	Percent
Level 3:Defined	12	70.6
Level 5:Optimizing	3	17.6
Level 2:Managed	1	5.9
Level 4:Quantitatively Managed	1	5.9
Total	52	100.0

4.2.1.7. Other Certifications

The related question on this topic was not answered by any of these companies, which means that, except for those with CMMI certification, the others do not have any type of software process improvement certifications.

4.2.1.8. Reasons for Not Adopting Agile Methods

Most companies used Agile methods in more than 3 projects as shown in Table 11, but there were four respondents who came up with the reason for not using Agile any more, stating that they do not have enough experience and skills in Agile.

Table 11. Reasons for not using Agile methods

Reasons for not_ adopting Agile methods	Frequency	Percent
In our organization, any kind of change meets with resistance	1	25%
some projects are so big and well defined requirements	1	25%
The projects we are making are not suitable for Agile methods	1	25%
We do not have enough experience and skills in Agile	1	25%
Total	4	

4.2.2. Agile-Adopting Projects' Characteristics

The purpose of this question was to ascertain which methods were used more in their projects, and which ones worked in effective ways with small, medium and large projects. Specifically, the question is whether project characteristics can determine which methods are to be used. To answer this question, a series of other inquiries were made, as follows:

4.2.2.1. Type of Systems Developed

There are many types of systems that can be developed using Agile methods. In this survey, seven systems were provided as options for the companies involved. Most develop more than two types of systems, but the Windows-based one presents the most developed systems, followed by business systems at 69.2% and 55.8%, respectively. Embedded systems and safety-critical systems present the lowest ones at 26.9% and 25%, respectively as shown in Figure 13.

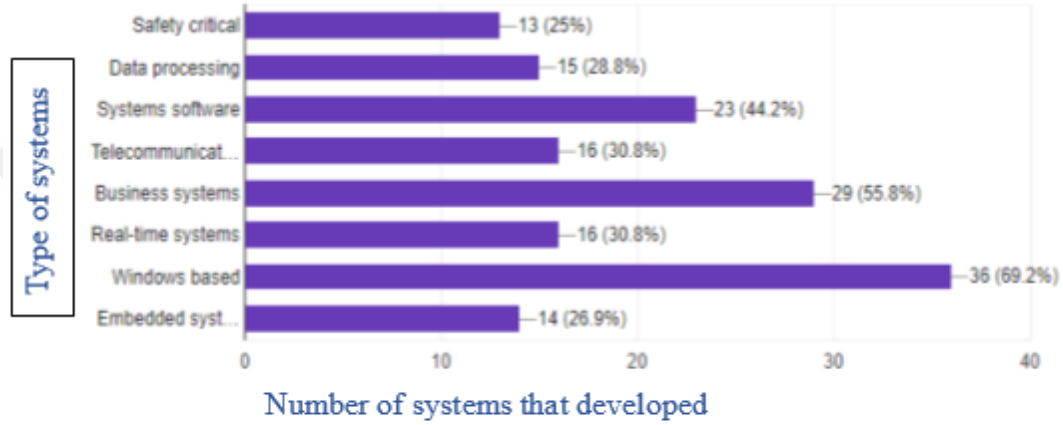

Figure 13. Type of systems developed

4.2.2.2. Agile Methods' Joint Application

Most of the responding firms used Agile methods along with other structure methods, as Table 12 shows, with 73% while some also used Agile methods solely (19.2%), and finally the lowest percentage that used Agile rarely (7.8%).

Table 12. Agile methods Joint Application

Agile methods were used in the project	Frequency	Percent
Used along with other structure methods	29	73.0
Completely	19	19.2
Rarely used	4	7.8
Total	52	100.0

4.2.2.3. Using Agile with other Structured Software Development Methods

Since most companies use Agile with other structured methods, it is significant to know what these methods are. There were three options provided in the survey as Prototyping, Spiral, and Waterfall lifecycle. The results show that the Waterfall lifecycle is the most common method used with Agile at 27.9%, the prototyping method is second at 23.6%, and the Spiral method at 9.3% and the lowest as shown in Table 13.

Table 13. Methods used with Agile

methods that used with Agile	Frequency	Percent
Waterfall lifecycle	12	27.9%
Prototyping	10	23.6%
Prototyping, Spiral, Waterfall lifecycle	6	13.9%
Prototyping, Waterfall lifecycle	6	13.9%
Prototyping, Spiral	5	9.6%
Spiral	4	9.3%
Total	43	100%

4.2.2.4. The Agile Methods Often Used

In general, most companies used more than one method of Agile, and the question would be which ones. There are many methods of Agile and some can be preferred over the rest for different reasons. By looking at the responses in Figure 14, one can see that the Scrum was the most common at 88.5%, followed by the XP methods come as second at 32.7%. The Dynamic Systems Development Method (DSDM) and Crystal present the lowest percentage at 5.8%.

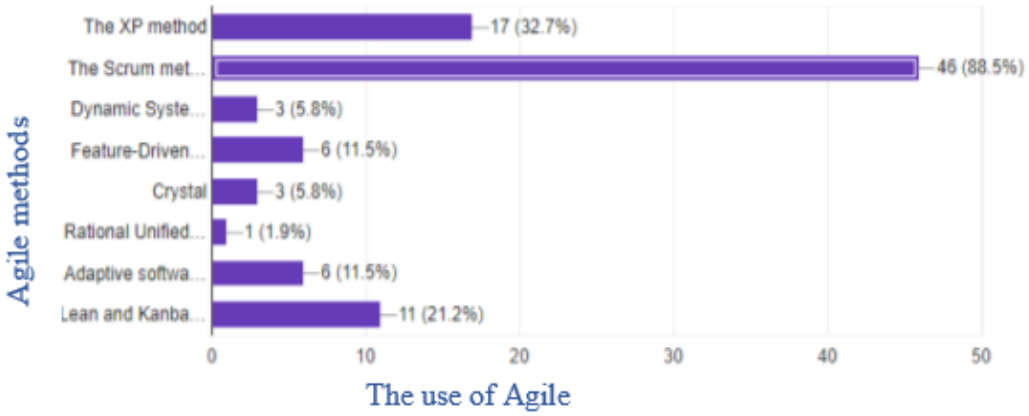

Figure 14. The Agile methods often used

4.2.2.5. Agile Methods Effective in Small Projects

In terms of small projects, many Agile methods can be used to develop small projects, and the question is which ones are effective for this purpose. According to the results, Scrum method and the XP method are the most commonly used in this respect because they are more effective than the rest. According to Figure 15, the figures stand at 67.3% and 40.4%, respectively. Meanwhile, the Crystal and the Dynamic Systems Development Method (DSDM) represent the lowest percentages at 5.8% and 3.8%, respectively.

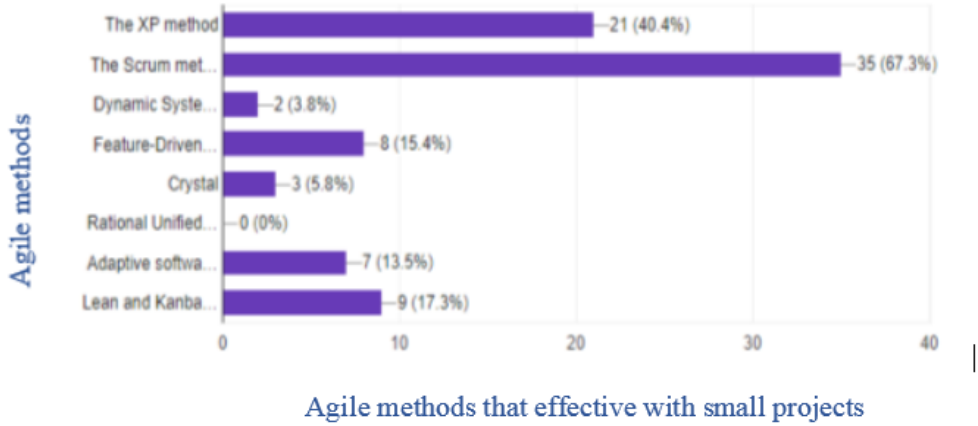

Figure 15. Agile methods effective in small projects

4.2.2.6. Agile Methods Effective in Medium Projects

Most agreed that the Scrum method is the most effective one used to develop medium projects at 82.7%, followed by the Lean and Kanban methods as the second one at 21.2%. Some expressed other insights, indicating the crystal method as effective in developing medium projects, but this presents only 3.4% and, hence, is the lowest one as shown in Figure 16.

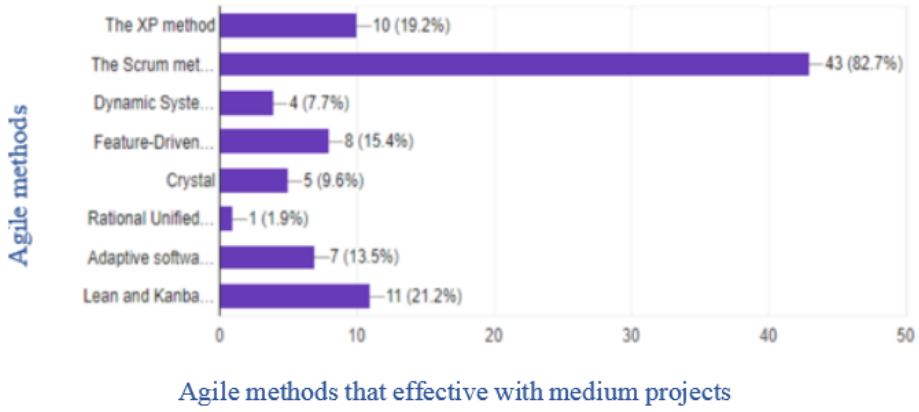

Figure 16. Agile methods effective in medium projects

4.2.2.7. Agile Methods Effective with Large and Complex Projects

In terms of large projects, the most effective method of Agile is the Scrum as by 65.4% of the respondents, followed by the Feature-Driven Development (FDD) method (32.7%) and the XP method, Lean and Kanban (28.8%) as shown in Figure 17.

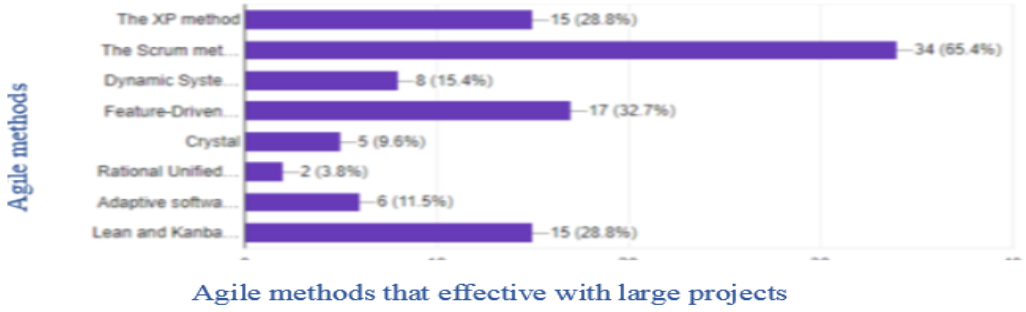

Figure 17. Agile methods effective with large and complex projects

4.2.2.8. Agile Methods as Reason_for Project Failure

Agile methods are being used to develop large projects, but adopting these methods may cause the failure of the project. To further investigate into this matter, questions were asked and most disagreed by 76.9% as shown in Table 14, indicating that adopting Agile methods was indeed successful and not the other way.

Table 14. Agile methods as the reason for project failure

Adopting Agile methods in complex project were the main reason_ behind the project failure	Frequency	Percent
No	40	76.9
Yes	12	23.1
Total	52	100.0

4.2.2.9. Agile Practices in Companies

There are many practices of Agile eight of which were provided in the survey with the option to indicate any other practice not included in the options. All of these Agile practices in the survey were used, but some of them were preferred more than the rest, namely Scrum daily meetings, small release cycles, continuous integration, code and design reviews, use of design patterns, and code standard (Figure 18) at 73.1%,69.2%,67.3%,55.8%,51.9%, and 46.2% respectively.

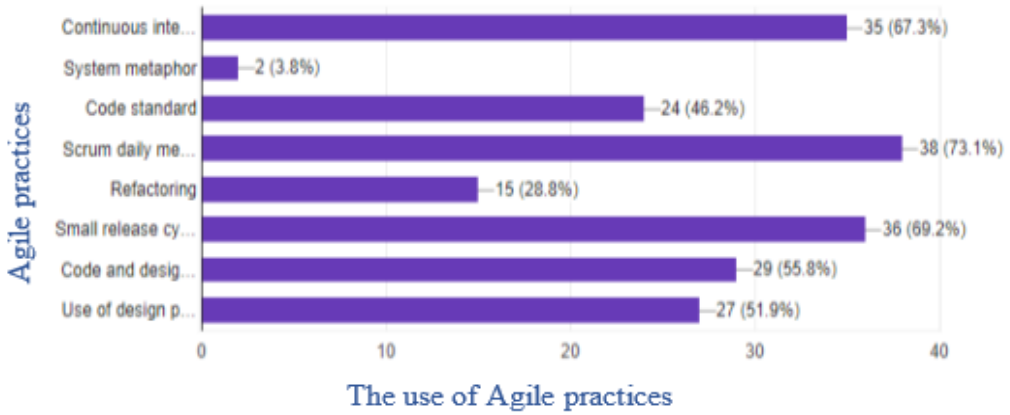

Figure 18. Agile Practices in Companies

4.2.3. The Success Factors of Adopting Agile Methods

The purpose of this research question is to analyze the importance of various factors possibly leading to success in the use of Agile software projects. These factors have been categorized into five dimensions, specifically the organizational dimension consisting of factors F1-F6, the people dimension which includes factors F17-F12, the technical dimension comprising factors F13-F17, the project dimension at F18-F22, and the process dimension incorporating factors F23-F25. The factors (independent variables) have been determined as stated below:

F1. The culture of organizations (Indep1)
F2. Team structure (Indep2)
F3. Management support (Indep3)
F 4. Team organization (Indep4)
F 5. Maintaining agility (Indep5)
F 6. Universal acceptance of Agile methodology (Indep6)
F 7. Customers' dominant issues (Indep7)
F 8. Having team members with high capabilities (Indep8)
F9. Practice and learning (Indep9)
F10. Communication and arbitration (Indep10)
F 11. Encouragement (Indep11)
F12. Project champion (Indep12)
F13. Assigning essential features first (Indep13)
F14. Frequent delivery of software (Indep14)
F15. High competence for team and organizational issues (Indep15)
F 16. The use of tools (Indep16)
F 17. Correct integration testing (Indep17)
F18. Project category (Indep18)
F19. Smaller-size team (Indep19)
F20. Agenda (Indep20)
F21. Least changes in requirement (Indep21)
F22. Projects within advance hazard analysis (Indep22)
F 23. Clarity (Indep23)
F24. Strong customer participation (Indep24)

F25. Efficient requirements gathering method (Indep25).

4.2.3.1. Correlation Analysis:

To understand the correlation between success factors and success attributes, we consider the independent variables $\mathbf{X_i}$, whose numerical values are assigned according to the response as follows:

Very important	5
Important	4
Neutral	3
Unimportant	2
Very unimportant	1

Correspondingly, dependent variables **Yi** represent the success attributes and attain numerical values according to the breakdown below:

Strongly agree	5
Agree	4
Neutral	3
Disagree	2
Strongly disagree	1

The success attributes defined as dependent variables are:

1. Better control over the work (Depen1).
2. Dealing with changing requirements (Depen2).
3. Increasing quality (Depen3).
4. Effort estimation (Depen4).
5. Customer satisfaction (Depen5).
6. Reducing the delivery schedules (Depen6).
7. Increasing return on investment (Depen7).

Afterward, for each factor X_i and each quality attribute Y_j, the Pearson correlation coefficient r_{ij}, is computed with the help of the SPSS software. It is well-

known that uncorrelatedness of random variables is a strictly stronger condition than their independence. Therefore, a non-zero correlation coefficient indicates relationship between variables.

In essence, this coefficient describes how close a relationship exists between the variables to a linear one, while the sign of r demonstrates whether the relationship is positive (r>0) or negative (r<0). After a correlation coefficient is calculated, its significance is tested. To be more specific, the following test procedure is applied for each success factor and each success attribute: after a numerical value is assigned to each response, the correlation coefficient r_{ij} is computed using the SPSS software.

After that, each r_{ij} is tested as to whether it provides a significant relationship at the level of significance $\alpha=0.05$ and if X_i is a significant explanatory variable for Y_j. This is done by using the hypotheses of the form:

H_0: $r_{ij} = 0$ (X_i is not a significant explanatory variable for Y_j).

H_1: $r_{ij} \neq 0$ (X_i is a significant explanatory variable for Y_j).

The test is a two-tailed t-test, with t (n-2) = t (50) distribution and the t-statistic

$$t=r\ \frac{\sqrt{n-2}}{1-r^2}\ .$$

From the observed value of the test statistic, the P- value is obtained and the null hypothesis is rejected if and only if P<0.05. The Table in appendix C shows which of the correlation coefficients appear to be significant. By looking at the Table, one can notice that 8 factors do not demonstrate any noticeable relation to the considered success attributes, namely F4, F5, F6, F9, F10, F21, F22, and F25. On the whole, we assume that 25 success factors are all essential to achieve each and every one of the success attributes A1-A7. After conducting 25*7 tests as described earlier, 8 of the factors were removed as unessential. In addition, it is observed that the remaining factors are important only for some, and not all, attributes. Further going into details, we can see that the most effective factor is F16 with a positive relationship with 6 attributes, followed by F11, F14 and F17, all of which have positive relationships with 5 attributes.

However, depending on the significance values and the values of correlation coefficients, we either accept or reject our 25 hypotheses introduced in chapter 3 as tested with 7 attributes (a-g). This means that there are 175 tests or hypotheses to test,

out of which 45 hypotheses are accepted and 129 rejected. The results of the acceptance/rejection of these hypotheses are shown in Table 15.

Table 15. Summary of hypothesis testing results of success factors using correlation analysis

SF/SA	control over the work	Dealing with changing requirements	quality	Effort estimation	Customer satisfaction	Reducing the delivery schedules	Increasing return on investment
The culture of organizations	H1a	H1b	H1c	H1d	H1e	H1f	**H1g** √
Team structure	H2a	**H2b** √	**H2c** √	H2d	**H2e** √	H2f	H2g
Management support	H3a	H3b	H3c	H3d	**H3e** √	H3f	H3g
Team organization	H4a	H4b	H4c	H4d	H4e	H4f	H4g
Maintaining agility	H5a	H5b	H5c	H5d	H5e	H5f	H5g
Universal acceptance of Agile methodology	H6a	H6b	H6c	H6d	H6e	H6f	H6g
Customers' dominant issues	**H7a** √	H7b	H7c	H7d	H7e	H7f	H7g
Having team members with high capabilities	H8a	H8b	**H8c** √	H8d	H8e	H8f	H8g

Practice and learning	H9a	H9b	H9c	H9d	H9e	H9f	H9g
Communication and arbitration	H10a	H10b	H10c	H10 d	H10e	H10f	H10g
Encouragement	**H11a** √	**H11b** √	**H11c** √	H11 d	**H11e** √	**H11f** √	H11g
Project champion	H12a	H12b	H12c	H12 d	**H12e** √	**H12f** √	H12g
Assigning essential features first	H13a	H13b	**H13c** √	H13 d	H13e	H13f	H13g
Frequent delivery of software	**H14a** √	H14b	**H14c** √	H14 d	**H14e** √	**H14f** √	**H14g** √
High competence for team and organizational issues	H15a	H15b	**H15c** √	H15 d	**H15e** √	H15f	**H15g** √
The use of tools	H16a	**H16b** √	**H16c** √	**H16 d** √	**H16e** √	**H16f** √	**H16g** √
Correct integration testing	H17a	**H17b** √	**H17c** √	H17 d	**H17e** √	**H17f** √	**H17g** √
Project category	**H18a**√	H18b	H18c	H18 d	H18e	H18f	H18g
Smaller-size team	H19a	**H19b** √	H19c	H19 d	**H19e** √	**H19f**√	**H19g** √
Agenda	H20a	H20b	H20c	H20 d	H20e	H20f	**H20g** √
Least changes in requirement	H21a	H21b	H21c	H21 d	H21e	H21f	H21g

Projects within advance hazard analysis	H22a	H22b	H22c	H22 d	H22e	H22f	H22g
Clarity	H23a	H23b	H23c	H23 d	H23e	H23f	**H23g** √
Strong customer participation	**H24a** √	**H24b** √	H24c	H24 d	**H24e** √	H24f	**H24g** √
Efficient requirements gathering method	H25a	H25b	H25c	H25 d	H25e	H25f	H25g

4.2.3.2. Linear Multiple Regression Analysis

The impact of the success factors on the success attributes can be observed in the Table in appendix B, based upon which seven dependent variables are affected by success factors which are: Better control over the work, dealing with changing requirements, increasing quality, cost and time, customer satisfaction, reducing the delivery schedules, and increasing return on investment. Consequently, the linear multiple regression analysis is conducted between each one of the dependent variables (success attributes) and seventeen independent variables (success factors accepted in the correlation test). The significant (sig) and the regression coefficient values can be used to measure the relationship between the dependent variables and independent variables.

4.2.3.2.1. Better Control Over the Work Attribute

There are five factors in the model: Strong customer participation, frequent delivery of software, project category, customer dominant issues, and encouragement. By looking at the value of sig in Table 16, it can be observed that not all the factors are significant (p> 0.05), implying that they are not related to better control over the work.

Table 16. Regression coefficients for control over work attribute

Coefficients[a]

Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
(Constant)	1.822	.636		2.865	.006
People dimension _F7_Customer dominant issues	.090	.096	.144	.945	.350
People dimension_F11_ Encouragement	.069	.122	.091	.566	.574
Technical dimension _F14_Frequent delivery of software	.236	.126	.255	1.870	.068
Project dimension _F18_Project category	.124	.120	.155	1.039	.304
Process dimension _F24_Strong customer participation	.079	.115	.100	.688	.495

a. Dependent Variable: The usage of Agile provides _ better control over the work

4.2.3.2.2. Dealing with Changing Requirements Attribute

There are six factors in the model: Team structure, encouragement, the use of tool, correct integration testing, smaller-size team, and strong customer participation. By looking at the value of sig in Table 17, it can be seen that, again, not all the factors are significant (p> 0.05), which means that they are not related to dealing with changing requirements in venture.

Table 17. Regression coefficients for dealing with chaning requirements attribute

Coefficients[a]

Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
(Constant)	1.364	0.791		1.724	0.092
Organizational dimension am structure	0.034	0.164	0.034	0.208	0.836
People dimension_F11_ Encouragement	-0.006-	0.150	-0.006-	-0.038-	0.970
Technical dimension _F16_The use of tool	0.118	0.175	0.127	0.671	0.506
Technical dimension _F17_Correct integration testing	0.128	0.215	0.107	0.595	0.555
Project dimension _F19_Smaller size team	0.161	0.135	0.183	1.188	0.241
Process dimension _F24_Strong customer participation	0.245	0.147	0.264	1.675	0.101

a. Dependent Variable: Agile methods were used because it copes with_ changing user requirements in better way

4.2.3.2.3. Increasing Quality Attribute

There are eight factors in the model as team structure, team members with high capability, encouragement, assigning essential features first, frequent delivery of software, high competence for team and organizational issues, the use of tool, and correct integration testing. By looking at the value of sig in Table 18, one can see that assigning essential features first is a significant factor by p=0.04 since b=0.33 is positive, which indicates that an increase in assigning essential features leads to increase in the quality in venture. On the other hand, the rest of the factors are not significant because (p> 0.05), meaning they are not related to quality. By looking at Figure 19 it can be seen that the shape of the histogram almost follows the shape of the normal curve.

Table 18. Regression coefficients for quality attribute

Coefficients[a]

Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
(Constant)	-.385-	.859		-.448-	.657
Organizational dimension _F2_Team structure	.197	.147	.208	1.342	.187
People dimension_F8_Team members with High capability	-.018-	.133	-.021-	-.137-	.892
People dimension_F11_ Encouragement	-.042-	.135	-.050-	-.316-	.754
Technical dimension _F13_Assigning essential features first	.333	.164	.273	2.031	.048
Technical dimension _F14_Frequent delivery of software	.322	.173	.310	1.857	.070
Technical dimension _F15_High competence for team and organizational issues	.112	.188	.096	.597	.553
Technical dimension _F16_The use of tool	.258	.155	.291	1.667	.103
Technical dimension _F17_Correct integration testing	-.166-	.211	-.145-	-.786-	.436

a. Dependent Variable: Agile adoption allows _ to achieve better quality

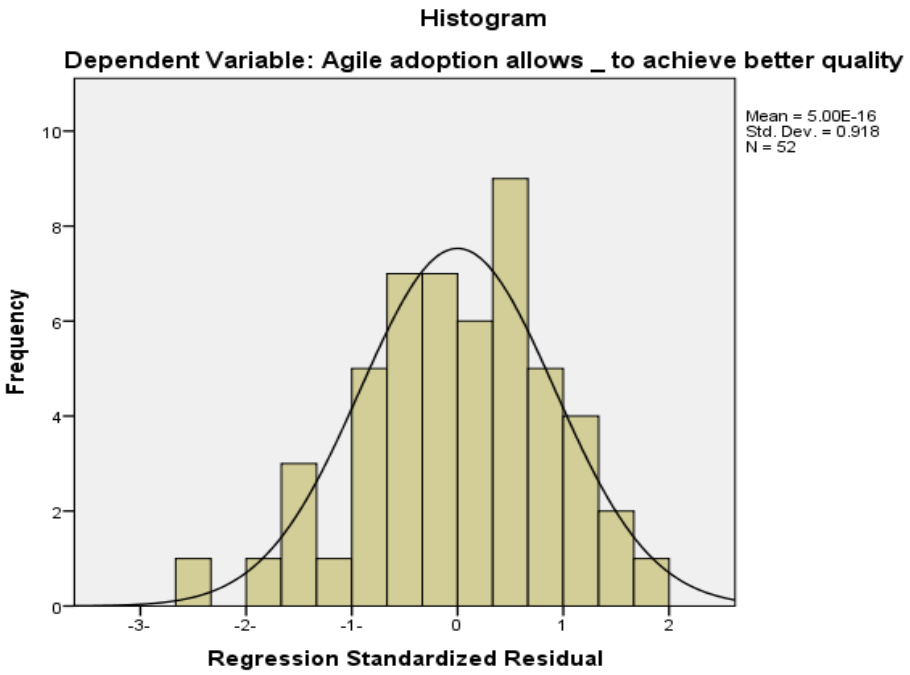

Figure 19. Frequency versus regression standardized residual for quality attribute

4.2.3.2.4. Effort Estimation (cost, schedule) Attribute

There is only one factor in the model - the use of tool. By looking at the significant value in Table 19, it can be noticed that p=0.006, which means that it is a significant factor since b=0.39 is positive, in turn indicating that the increase in using tool leads to the increase in effort estimation. By looking at Figure 20 it can be seen that the shape of the histogram follows the shape of the normal curve.

Table 19. Regression coefficients for effort estimation attribute

Coefficients[a]

Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
(Constant)	2.185	.553		3.954	.000
Technical dimension _F16_The use of tool	.393	.136	.378	2.890	.006

a. Dependent Variable: Agile is used because it helps in effort estimation_(cost,schedule)

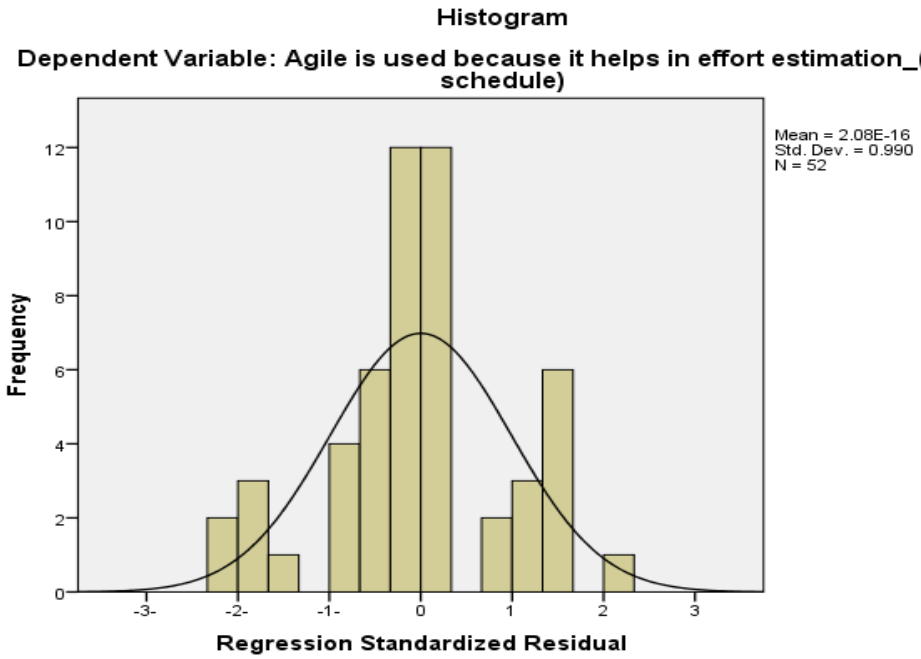

Figure 20. Frequency versus regression standardized residual for effort estimation attribute

4.2.3.2.5. Customer Satisfaction Attribute

There are ten factors in the model, namely team structure, team organization, head of project first, encouragement, frequent delivery of software, high competence for team and organizational issues, the use of tool, correct integration testing, smaller-size team, and strong customer participation. The value of sig in Table 20 shows the use of tool as a significant factor (P=0.003), and that b=0.55 is positive which indicates that a increase in using tool leads to an increase in customer satisfaction. The other factors are not significant (p> 0.05), implying that they are not related to reducing the delivery schedule. On other hand, encouragement, head of project first, and correct integration testing factors have a negative relationship (b =- 0.108, b=-0.018, and b=-0.316, respectively), which means when these factors increase, customer satisfaction decreases. By looking at Figure 21 it can be seen that the shape of the histogram almost follows the shape of the normal curve.

Table 20. Regression coefficients for customer satisfaction attribute

Coefficients[a]

Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
(Constant)	.207	.894		.231	.818
Organizational dimension _F2_Team structure	.026	.208	.024	.126	.900
Organizational dimension _F3_ Team organization	.025	.195	.024	.129	.898
People dimension_F11_ Encouragement	-.108-	.158	-.111-	-.681-	.500
People dimension_F12_Head of project first	-.018-	.165	-.017-	-.112-	.912
Technical dimension _F14_Frequent delivery of software	.356	.185	.300	1.923	.061
Technical dimension _F15_High competence for team and organizational issues	.050	.231	.037	.217	.830
Technical dimension _F16_The use of tool	.558	.173	.551	3.219	.003
Technical dimension _F17_Correct integration testing	-.316-	.243	-.242-	-1.301-	.200
Project dimension _F19_Smaller size team	.192	.161	.200	1.192	.240
Process dimension _F24_Strong customer participation	.186	.146	.183	1.276	.209

a. Dependent Variable: Agile methods were used because _ provide customer satisfaction

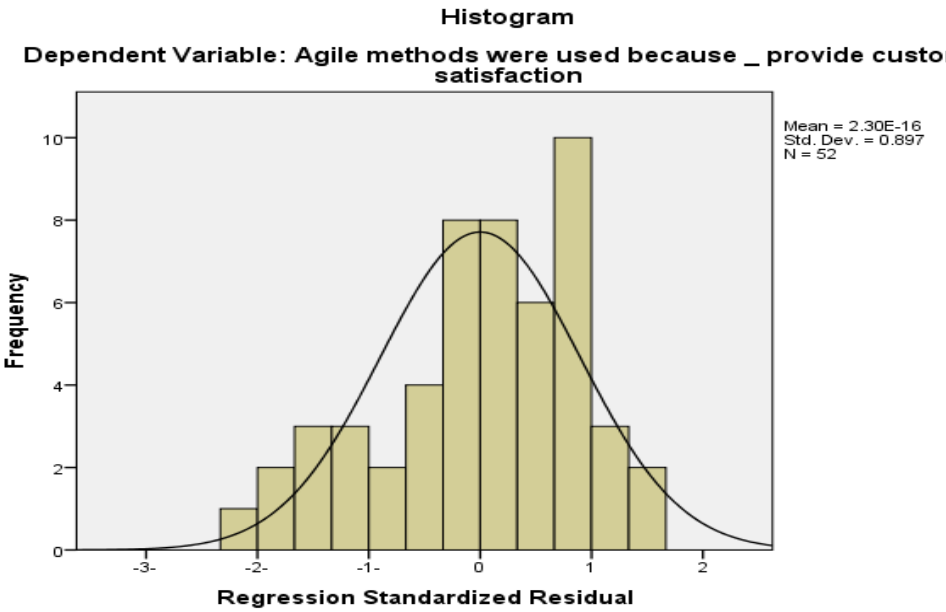

Figure 21. Frequency versus regression standardized residual for customer satisfaction attribute

4.2.3.2.6. Reducing the Delivery Schedule Attribute

There are six factors in the model, these are: encouragement, head of project first, frequent delivery of software, the use of tool, correct integration testing, and smaller-size team. Measuring the impact of each factor on the dependent variable gives us the sig as in Table 21. Accordingly, frequent delivery of software and the use of tool factors are significant (P=0.041 and P=0.042 respectively). In other words, the increase in frequent delivery of software and the use of tool factors lead to a reduction in the delivery schedule because their p values are positive and b=0.34 and b=0.36, respectively. The other factors are not significant (p> 0.05), which means that they are not related to reducing the delivery schedule. On other hand, the correct integration testing factor has a negative relationship (b= - 0.235) with reducing the delivery schedule. By looking at Figure 22 it can be seen that the shape of the histogram follows the shape of the normal curve.

Table 21. Regression coefficients for reducing the delivery schedule attribute

Coefficients[a]

Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
(Constant)	1.382	.775		1.783	.081
People dimension_F11_ Encouragement	.027	.140	.030	.190	.850
People dimension_F12_Head of project first	.042	.156	.043	.269	.789
Technical dimension _F14_Frequent delivery of software	.340	.161	.313	2.106	.041
Technical dimension _F16_The use of tool	.365	.174	.394	2.097	.042
Technical dimension _F17_Correct integration testing	-.235-	.230	-.197-	-1.023-	.312
Project dimension _F19_Smaller size team	.139	.127	.158	1.092	.281

a. Dependent Variable: Using the Agile methods helped _ to reduce the delivery schedules

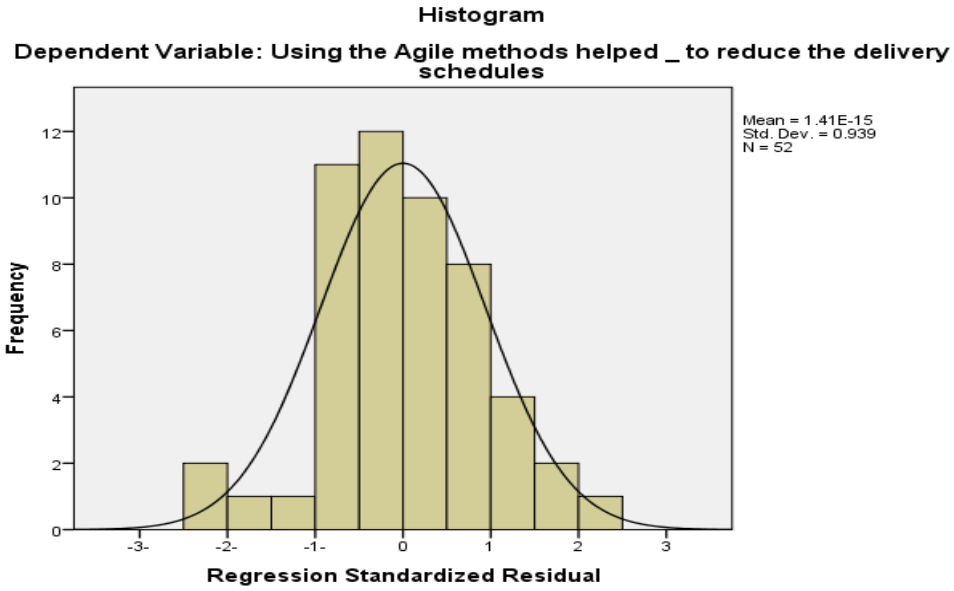

Figure 22. Frequency versus regression standardized residual for reducing the delivery schedule attribute

4.2.3.2.7 Increasing Return on Investment Attribute

There are ten factors in the model, namely the culture of organizations, assigning essential features first, frequent delivery of software, high competence for team and organizational issues, the use of tool, correct integration testing, smaller-size team, agenda, clarity and strong customer participation. To measure the impact of each factor on the dependent variable, the b and significant (sig) values in Table 22 can be used, showing that not all the factors are significant ($p > 0.05$). This implies that they are not related to the return on investment factor. In addition, some factors a have negative relationship with return on investment and, by looking at the b value, it can be noticed that agenda ($b = -0.134$) and correct integration testing ($b = -0.184$), when used more, decrease the return on investment.

Table 22. Regression coefficients for increasing return on investment attribute

Coefficients[a]

Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
(Constant)	-.978-	.991		-.987-	.330
Organizational dimension _F1_The culture of Organizations	.121	.119	.150	1.013	.317
Technical dimension_F13 Assigning essential features first	.309	.176	.243	1.757	.086
Technical dimension _F14_Frequent delivery of software	.224	.180	.207	1.249	.219
Technical dimension _F15_High competence for team and organizational issues	.153	.226	.125	.679	.501
Technical dimension _F16_The use of tool	.273	.175	.295	1.556	.127
Technical dimension _F17_Correct integration testing	-.184-	.233	-.155-	-.791-	.433
Project dimension _F19_Smaller size team	.114	.152	.130	.749	.458
Project dimension _F20_Agenda	-.134-	.153	-.152-	-.878-	.385
Process dimension _F23_ Clarity	.273	.183	.235	1.489	.144
Process dimension _F24_Strong customer participation	.032	.132	.035	.244	.808

a. Dependent Variable: The Agile methods were used to_ increase the return on investment

The results show that for time and cost, both model approaches arrive at the same conclusion, mainly that in each case, team capability and delivery strategy are selected as the most significant factors. To compare between the seven models, the adjusted R-square statistic is used because it compensates for the number of variables in the model and increases only if added variables contribute significantly to the model. The

customer satisfaction and quality models are the best because their adjusted R Square is 37.9% and 28.1%, respectively, as shown in Table 23.

Table 23. Regression model summary for success factors

1. Regression Model Summary				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.508[a]	.258	.178	.5921
2	.508[a]	.258	.159	.7041
3	.627[a]	.393	.281	.6222
4	.378[a]	.143	.126	.8043
5	.708[a]	.501	.379	.6615
6	.568[a]	.323	.233	.6719
7	.645[a]	.416	.274	.6527

a. Predictors: (Constant, Indep24, Indep14, Indep18, Indep7, Indep11.

b. Predictors: (Constant, Indep24, Indep16, Indep19, Indep2, Indep11, Indep17.

c. Predictors: (Constant, Indep17, Indep13, Indep11, Indep15, Indep8, Indep2, Indep14, Indep16.

d. Predictors: (Constant, Indep16.

e. Predictors: (Constant, Indep24, Indep12, Indep17, Indep15, Indep3, Indep14, Indep11, Indep19, Indep16, Indep2.

f. Predictors: (Constant, Indep19, Indep17, Indep12, Indep14, Indep11, Indep16.

g. Predictors: (Constant, Indep24, Indep1, Indep17, Indep13, Indep19, Indep23, Indep14, Indep20, Indep15, Indep16.

h. Dependent Variables: Depen1, Depen2, Depen3, Depen4, Depen5, Depen6, Depen7.

In both customer satisfaction and effort estimation models, the results achieved the same conclusion: the use of tool is selected as the most significant factors. This is while the assigning-essential-features-first factor is selected as a significant one in the quality model. The frequent delivery of software and the use of tool factors are selected as significant in reducing delivery schedule. Consequently, there are three main factors selected as significant factors over the four models, and these factors are within the technical dimension. The rest of the models show no significant factors. To finalize the hypothesis testing, we can use the observations above to reduce the number of hypotheses to five, rejecting 40 hypotheses, meaning that the presence of those factors did not make a significant difference to the value of the success dimensions. In brief, the accepted hypotheses are:

1. Assigning essential features first is a crucial success determinant that adds to the effective ASD ventures for: (c) quality

2. Frequent delivery of software is a crucial success determinant that adds to the value of effective ASD ventures in: (g) reducing the delivery schedules

3. The use of tools is a crucial success determinant that adds to the value of effective ASD ventures in: (f) reducing the delivery schedules.

4. The use of tools is a crucial success determinant that adds to the value of effective ASD ventures in: (e) customer satisfaction.

5. The use of tools is a crucial success determinant that adds to the value of effective ASD ventures in: (d) effort estimation.

In addition, the general model of multiple regression is shown in the equation below assuming that there are k independent variables (McClave and Benson, 1988):

$$y=\beta_0+\beta_1x_1+\beta_2x_2+\ldots.+\beta_k x_k+\varepsilon$$

Where y is the dependent variable, x_1, x_2... x_k are the independent variables, β is the regression coefficient, and Ɛ is the random error component. In the case of our study, the above translates to the following general equation:

$$Y\,(Q,E,C,R) = \beta_1 SF_{13} + \beta_2 SF_{14} + \beta_3 SF_{16} + \varepsilon .$$

Since the accepted factors have an impact on four success attributes, there are four equations, which are:

$$Y\,(Q) = 0.333 * SF_{13}. \quad (1)$$

$$Y\,(E) = 0.393 * SF_{16}. \quad (2)$$

$$Y\,(C) = 0.558 * SF_{16}. \quad (3)$$

$$Y\,(R) = 0.340 * SF_{14} + 0.365 * SF_{16}. \quad (4)$$

where Y is the failure dependent variable, *Q* is the quality dimension, effort estimation for the cost and time dimension, C is the customer satisfaction dimension, *R* is reducing delivery schedule dimension, and β is the partial regression coefficient for the ith success factor (*SF*). The multiple regression analysis was done on one level, the full model, where all 17 independent variables were entered into a regression model at the same time.

4.2.4. The Failure Factors of Adopting Agile Methods

The purpose of this research question is to analyze the importance of various factors possibly leading to a failure in the usage of Agile software projects. These factors have been categorized into four dimensions, specifically the Organizational dimension consisting of factors H1-H3, the People dimensions, which includes factors H4-H6, the Technical dimension comprising factors H7-H8, and the Process dimension incorporating factors H9-H11. The factors (independent variables) have been determined as stated below:

F1. Absence of management support (Indep1).
F2. The organization size is too huge (Indep2).
F3. The culture of Organization is too traditional and political (Indep3).
F4. Terrible customer relationship (Indep4).
F5. Absence of cooperation (Indep5)
F6. Absence of essential skill-set (Indep6).
F7. Absence of full set of right Agile practices (Indep7).
F8. Inadequacy of the technology and tools (Indep8).
F9. Absence of customer presence (Indep9).
F10. Absence of tracking mechanisms during Agile progress (Indep10).
F11. Determine the role of the client (Indep11).

4.2.4.1. Correlation Analysis

To understand the correlation between failure factors and success attributes, we consider the independent variables $\mathbf{X_i}$, whose numerical values are assigned according to the response as follows:

Very important	1
Important	2
Neutral	3
Unimportant	4
Very unimportant	5

Correspondingly, dependent variables $\mathbf{Y_i}$ represent success attributes attain numerical values according to the breakdown below:

Strongly agree	5
Agree	4
Neutral	3
Disagree	2
Strongly disagree	1

The success attributes which are defined as dependent variables are:

1. Better control over the work (Depen1).
2. Dealing with changing requirements (Depen2).
3. Increasing quality (Depen3).
4. Time and cost (Depen4).
5. Customer satisfaction (Depen5).
6. Reducing the delivery schedules (Depen6).
7. Increasing return on investment (Depen7).

Afterward, for each factor $\mathbf{X_i}$ and each quality attribute $\mathbf{Y_j}$, the Pearson correlation coefficient $\mathbf{r}_{ij}$, has been computed with the help of **SPSS** software. It is well-known that uncorrelatedness of random variables is a strictly stronger condition than their independence. Therefore, non-zero correlation coefficient indicates relationship between variables.

In essence, this coefficient describes how close is the relationship between variables to a linear one, while the sign of **r** demonstrates whether the relationship is positive($\mathbf{r}>0$) or negative ($\mathbf{r}<0$). After a correlation coefficient was calculated, its significance has been tested. To be more specific, the following test procedure has been applied for each success factor and each success attribute: after numerical value has been assigned to each response, the correlation coefficier $\mathbf{r}_{ij}$ has been computed using **SPSS** software.

After that, each $\mathbf{r}_{ij}$ has been tested whether it provides a significant relationship at the level of significance $\alpha=0.05$ or that is it has been checked if $\mathbf{X_i}$ is a significant explanatory variable for $\mathbf{Y_j}$. This has been done by using the hypotheses of the form:

H_0: $\mathbf{r}_{ij} = 0$ ($\mathbf{X_i}$ is not a significant explanatory variable for $\mathbf{Y_j}$).

H_1: $\mathbf{r}_{ij \neq 0}$ ($\mathbf{X_i}$ is a significant explanatory variable for $\mathbf{Y_j}$).

The test is a two-tailed t-test, with t (n-2) =t (50) distribution and the t-statistic

$$\mathbf{t=r}\ \frac{\sqrt{n-2}}{1-r^2}\ .$$

From the observed value of the test statistic, the P- value was obtained and the null hypothesis has been rejected if and only if P<0.05. Table B shows which of the correlations coefficients appear to be significant.

In terms of failure factors, the relationship between failure factors and success attributes is negative correlation, because variable X (failure factors) increases as the other variable Yi (success attributes) decreases, and vice versa. From the observed value of the test statistic, the P- value was obtained and the null hypothesis has been rejected if and only if P< 0.05. Table B shows which of the correlations coefficients appear to be significant. On the whole, we assumed that, 11 failure factors can cause failure for each and every quality attributes A1-A7.After conducting 11*7 tests as described above, 4 of the factors we removed as unessential which are F4,F5, F6, and F8. In addition, it has been found that the remains factors are important only for some rather than all attributes. Going in details the most effective factors are F2, and F11which have negative relationships with 3 attributes.

However, depending on the significance values and the values of correlation coefficients, we either accept or reject our 11 hypotheses that we have introduced in chapter 3 are test with 7 attributes(a-g) which means we have 77 test or hypotheses to test, there are 13 hypotheses accepted and 65 hypotheses are rejected . The results of acceptance/rejection of hypotheses are shown in Table 24

Table 24. Summary of hypothesis testing results of failure factors using correlation analysis

FF/SA	control over the work	Dealing with changing requirements	quality	Effort estimation	Customer satisfaction	Reducing the delivery schedules	Increasing return on investment
Absence of management support	H1a	H1b	H1c	**H1d** √	H1e	H1f	H1g
The organization size is too huge	**H2a** √	**H2b** √	H2c	H2d	**H2e** √	H2f	H2g
The culture of Organization is too traditional and political	H3a	**H3b** √	H3c	H3d	H3e	H3f	H3g
Terrible customer relationship	H4a	H4b	H4c	H4d	H4e	H4f	H4g
Absence of	H5a	H5b	H5c	H5d	H5e	H5f	H5g

cooperation							
Absence of essential skill-set	H6a	H6b	H6c	H6d	H6e	H6f	H6g
Absence of full set of right Agile practice	H7a	**H7b** √	H7c	H7d	**H7e** √	H7f	H7g
Inadequacy of the technology and tools	H8a	H8b	H8c	H8d	H8e	H8f	H8g
Absence of customer presence	H9a	**H9b** √	H9c	H9d	**H9e** √	H9f	H9g
Absence of tracking mechanisms during Agile progress	H10a	H10b	H10c	H10d	**H10e** √	H10f	H10g
Determine the role of the client	**H11a** √	**H11b** √	H11c	H11d	**H11e** √	H11f	H11g

4.2.4.2. Linear Multiple Regression Analysis

The impact of failure factors on success attributes can be observed in Table in appendix C, it can be noticed that four dependent variables are affected by failure factors which are: Better control over the work, dealing with changing requirements, cost and time, and customer satisfaction. So the multiple regression analysis are conducted between each one of the dependent variables (success attributes) and seven independent variables (failure factors that accepted in correlation test). The significant (sig) and the regression coefficient values can be used to measure relationship between dependent variable and independent variables.

4.2.4.2.1. Better Control Over the Work Attribute

There are two factors in this model which are determine the role of the client factor and organization size is too huge. By looking at the significant (sig) value in Table 25 it can be noticed that, organization size is too huge is a significant factors p=0.007, since b= -0.22 is negative which indicates that the increase organization's size lead to decrease control over the work in the venture. On the other hand, the rest of factors are not significant because (p> 0.05) which seem they are not related to control over work. By looking at Figure 23 it can be seen that the shape of the histogram almost follows the shape of the normal curve.

Table 25. Regression coefficients for control over the work attribute

Coefficients[a]

Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
(Constant)	5.070	.229		22.097	.000
Organizational dimension _F2_The Organization size is too huge	-.227-	.081	-.373-	-2.820-	.007
Process dimension _F11_Determine the role of client	-.133-	.086	-.206-	-1.552-	.127

a. Dependent Variable: The usage of Agile provides _ better control over the work

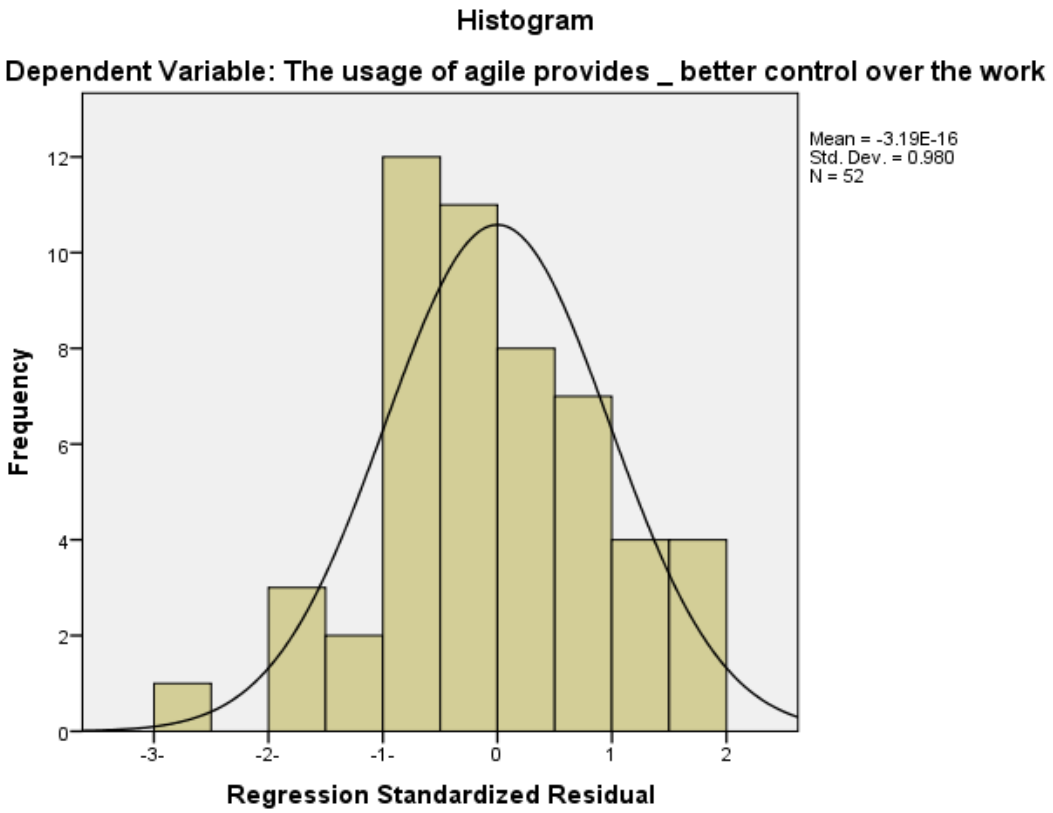

Figure 23. Frequency versus regression standardized residual for control over the work attribute

4.2.4.2.2.Dealing with Changing Requirements Attribute

There are five factors which have an impact on this attribute: determining the role of client, too-large an organization size, absence of full set of right Agile practices, absence of customer presence, and the culture of organization as traditional and political. The sig values in Table 26 show that not all the factors are significant (p> 0.05), meaning that they are not related to dealing with changing requirements.

Table 26. Regression coefficients for dealing with changing requirements

Coefficients[a]

Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
(Constant)	5.197	.341		15.222	.000
Organizational dimension _F2_ The Organization size is too huge	-.135-	.113	-.189-	-1.193-	.239
Organizational dimension _F3_ The culture of Organization is too traditional and political	-.017-	.132	-.023-	-.126-	.901
Technical dimension _F7_ Absence of full set of right Agile practices	-.134-	.143	-.151-	-.938-	.353
Process dimension _F9_Absence of customer presence	-.161-	.142	-.205-	-1.135-	.262
Process dimension _F11_ Determine the role of client	-.047-	.138	-.061-	-.339-	.736

a. Dependent Variable: Agile methods were used because it copes with_ changing user requirements in better way

4.2.4.2.3.Effort Estimation (cost and time) Attribute

There is only one factor which has an impact on this attribute: absence of management support. By looking at the sig value in Table 27, it can be noticed that the absence of management support is a significant factor at p=0.017, but since b= 0.29 is positive, it indicates that an increase in the absence of management support leads to an increase in effort estimation in venture. In other words, the relationship is not negative.

Table 27. Regression coefficients for effort estimation attribute

Coefficients[a]

Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
(Constant)	3.167	.263		12.050	.000
Organizational dimension _F1_Absence of management support	.292	.118	.329	2.463	.017

a. Dependent Variable: Agile is used because it helps in effort estimation_(cost,schedule)

4.2.4.2.4.Customer Satisfaction Attribute

There are five factors in this model which are: too large an organization size, absence of a full set of right Agile practices, absence of customer participation, absence of tracking mechanisms during Agile progress, and determine the role of client. The sig values in Table 28 show all the factors are not significant (p> 0.05), meaning that em they are not related to customer satisfaction.

Table 28. Regression coefficients for customer satisfaction attribute

Coefficients[a]

Model	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
(Constant)	5.195	.362		14.359	.000
Organizational dimension _F2_The Organization size is too huge	-.205-	.107	-.262-	-1.909-	.062
Technical dimension _F7_Absence of full set of right Agile practices	-.115-	.146	-.119-	-.791-	.433
Process dimension _F9_Absence of customer presence	-.206-	.169	-.241-	-1.224-	.227
Process dimension _F10_Absence of tracking mechanisms during Agile progress	-.031-	.168	-.037-	-.183-	.855
Process dimension _F11_Determine the role of client	-.008-	.151	-.009-	-.052-	.959

a. Dependent Variable: Agile methods were used because _ provide customer satisfaction

Overall, by studying the negative correlation between failure factors and success attributes, it can be observed that some factors have a negative relationship with the success attributes. A multiple regression analysis is later applied as a result of which in control over the work model the “organization size is too large” is selected as the most significant factor. The rest of the models show no significant factors. To compare between the seven models, the adjusted R-square statistic is used because it compensates for the number of variables in the model and will only increase if added variables contribute significantly to the model. The “control over the work” and “customer satisfaction” models are the best because the adjusted R Square are 20% and 16.4.1%, respectively, as shown in Tabl**e** 29.

Table 29. Regression model summary for failure factors

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.481[a]	.232	.200	.5840
2	.459[a]	.210	.125	.7183
3	.368[a]	.135	.100	.8162
4	.496[a]	.246	.164	.7674

a. Predictors: (Constant), Indep11,Indep2

b. Predictors: (Constant), Indep11, Indep2, Indep7, Indep9, Indep3.

c. Predictors: (Constant), Indep4, Indep11.

d. Predictors: (Constant), Indep11, Indep2, Indep7, Indep9, Indep10.

e. Dependent Variable: Depen1, Depen2, Depen4, Depen5.

To finalize the hypothesis testing, we can use the observations above to reduce the number of hypotheses to one, rejecting 11 hypotheses. This means that the presence of those factors did not make a significant difference in the value of the success dimensions. The accepted hypothesis is:

When the size of the organization is large, the possibility of ASD projects' failure is increased in terms of (a) control over the work.

The general model of multiple regression can be shown in the equation below, assuming that there are k independent variables (McClave and Benson, 1988):

$$y=\beta_0+\beta_1x_1+\beta_2x_2+\ldots..+\beta_k x_k+\varepsilon$$

where y is the dependent variable, x_1, x_2... x_k are the independent variables, β is the regression coefficient, and Ɛ is the random error component. In the case of our study, the above translates to the following general equation:

$$Y(L) = \beta_1 FF_1.$$

$$Y(L) = (-0.227)*FF_1.$$

Where Y is the failure dependent variable, *L* is the losing control over the work dimension, and β is the partial regression coefficient for the ith failure factor (*FF*). The multiple regression analysis was applied on one level which is a full-model where all 7 independent variables were entered into a regression model at the same time.

4.2.5. The Acceptance of Agile

The purpose of this research question is to indicate the main benefits gained by adopting Agile methods. Specifically, the following nine characteristics are selected for consideration:

1. The use of Agile provides better control over the work.
2. Using Agile methods allows to finish the tasks quickly.
3. Agile methods are used because they cope with changing user requirements in a better way.
4. Agile adoption allows to achieve better quality.
5. Agile methods are selected because of the type of the project.
6. Agile is used because it helps in effort estimation (cost, schedule).
7. Agile methods are used because they help to provide customer satisfaction.
8. Using Agile methods help to reduce the delivery schedules.
9. The Agile methods are used in order to increase the return on investment.

Furthermore, the size-related aspects (large, medium, and small) of companies are analyzed separately. The responses to the questionnaire are summarized in Table 30. To compare the results of the achievements of each statement after Agile methods were incorporated, the weighted Borda count is used where the score of a given benefit is calculated as follows:

$$\text{Score} = 3*S+2*A+N,$$

where S is the number of "Strongly Agree" responses, A is the number of "Agree" responses, and N stands for the number of "Neutral" responses. In the case when the score values are equal, the numbers of the "Strongly agree" replies are compared to determine the preference. Based on the aforementioned calculations, the following conclusions are reached:

The main benefit obtained by using Agile methods is 1: "Agile provides better control over the work" (score: 117). The next ones - rather close to each other in the opinion of the respondents – are 3 ("Agile methods cope with changing user requirements in a better way") (score: 111) and 2 ("Using Agile allows to finish the tasks quickly") (score: 110).

If we compare these outcomes with the data of the companies with different sizes, we observe that 1 is still viewed as the primary advantage of the Agile methods within large and small companies, while for the medium-size companies, the priority is switched to 3, and 1 comes at the 4-th place.

The importance of Agile in achieving the quality attribute 3 is indicated as the major one by the employees of medium companies, while mentioned as 3-rd and 4-th by large and small companies, respectively.

The role of Agile methods in achieving characteristic 2 was considered to be the third important one, and also it was determined as the 2-nd for large and medium companies, and the 3-rd for small ones.

As a conclusion, in the evaluation of the benefits coming from adopting Agile methods, the priorities are:

$$1 \rightarrow 3 \rightarrow 2$$

Along with finding the benefits gained mostly from the use of Agile methods, it is also determined which of them can be considered as receiving the weakest effect from the adoption of those methods. The data supplied in Table 30 shows that the quality which gains the least is 6 ("Agile is used because it helps in effort estimation - cost &schedule") (score:91), which remains also true for small and medium companies, whereas large companies mention as such attribute 7.("Agile methods are used because they help to provide customer satisfaction").

The impact on the Agile methods on attributes 5 ("Agile methods are selected because of the type of the project") (score: 92) and 4 ("Agile adoption allows to achieve better quality") (score: 95) are listed as the 2-nd and 3-rd least important ones. It has to be pointed out that, according to this questionnaire, the Agile is not used mainly to achieve better quality.To summarize, the least important reasons to adopt Agile can be listed as follows:

$6 \rightarrow 5 \rightarrow 4$.

Table 30. Reasons for accepting Agile

Factors\Success Attributes	Company size	Strongly agree	Agree	Neutral	Disagree	Str. disagree	*Total*	Borda count
Control over the work	Large	10	5	2	-	-	17	42
	Medium	2	8	1	-	-	11	23
	Small	7	14	3	-	-	24	52
	Total	19	27	6	-	-	52	117
Finish the task quickly	Large	8	4	5	-	-	17	37
	Medium	4	5	2	-	-	11	24
	Small	7	12	4	1	-	24	49
	Total	19	21	11	1	-	52	110
Coping with changing requirements	Large	7	5	4	1	-	17	35
	Medium	6	4	1	0	-	11	27
	Small	5	15	4	0	-	24	49
	Total	18	24	9	1	-	52	111
Achieving better quality	Large	5	8	1	3	-	17	32
	Medium	2	5	4	-	-	11	20
	Small	0	9	15	-	-	24	33
	Total	7	32	10	3	-	52	95
Type of project	Large	5	4	6	2	-	17	29
	Medium	2	7	-	2	-	11	20
	Small	4	12	7	-	1	24	43
	Total	11	23	13	4	1	52	92

Help in effort estimation	Large	6	4	6	1	-	17	32
	Medium	1	5	3	2	-	11	16
	Small	2	17	3	2	-	24	43
	Total	9	26	12	5	-	52	91
Cutomer satisfaction	Large	5	6	4	2	-	17	31
	Medium	4	6	-	1	-	11	24
	Small	5	13	6	-	-	24	47
	Total	14	25	10	3	-	52	102
Reducing delivery schedules	Large	6	4	6	1	-	17	32
	Medium	2	8	-	1	-	11	22
	Small	4	19	-	1	-	24	50
	Total	12	31	6	3	-	52	104
Incresing return on investment	Large	7	4	6	-	-	17	35
	Medium	3	6	1	1	-	11	22
	Small	3	15	6	-	-	24	45
	Total	13	25	13	1	-	52	102

4.2.6. Comments

This section gives an opportunity to the respondents to provide any additional comments they may have about the use of Agile methods in their work environment. Four responses were received in regards to the Comment section as shown in Table 31. The most interesting comment is that research and development projects need Agile.

Table 31. Additional comments provided by responders

Additional comment that may have about adopting Agile methods	Frequency
A good Agile system is very important in companies now. Also to make sure it is being implemented is also a requirement until it becomes a tradition.	1
I think the main goal of Agile is to help deliver artifacts soon and after, to avoid misunderstandings with the client.	1
Research and development projects need Agile	1
We are mainly research institute and develop SW occasionally (mainly research prototypes or PoC's) so my answer is not from the point of view commercial SW development. However, we have done strong Agile research and, therefore, have broader view to the capabilities of Agile and Agile practices in general.	1
Total	4

CHAPTER 5

DISCUSSION

These days, software development is faced with never-ending swift changes. To help the improvement and immediate progress of complex systems and meet the specific needs of such an approach to improvement, diverse methodologies have been proposed over the years. One of these methods is the Agile approach.

This study explores the factors involved in the adoption of agile methods in small and medium enterprises. In all, 52 respondents participated from around the world from which, 35 belonged to small and medium companies and the rest to large enterprises. The feedback provided is used to make comparisons between the small and medium companies with large ones.

Descriptive statistics are used to analyze the data, regarding small and medium enterprises in seven different countries based on their years of experiences and number of projects developed using agile - which, in comparison, is more than more than large companies. these were expected results since agile methods are initially intended for small, signal-group projects and are, as a result, popular in small and medium organizations (Boehm and Turner, 2005).

Most of the companies do not have CMMI certification, except 17 companies as mainly large and medium in size. Perhaps, the reason behind this is that large and medium companies tend to develop projects that require more time and involve larger scales of deliverables and more members in teams. Also, these companies can afford expenditure to attain CMMI certification. Most of the small companies developed projects with medium levels of complexity and the reason behind this is that, these

companies dealing with small projects that do not require much time and small number of people work on it. While medium and large companies have the same percentage of developing projects with medium and high levels of complexity because of size of the projects they are dealing with, which reqires more peolple and much time to work on it.

Most companies use agile methods along with other structured methods because both can co-exist, thus constituting the greater part of practically utilized hybrid approaches (Kuhrmann and Fern´andez, 2015; Kuhrmann et al., 2014). In terms of the methods used alongside agile, the Waterfall lifecycle tops the list. It is observed that a mix of the Waterfall/XP, and Scrum/XP are the most widely ordinary combinations (Solinski and Petersen, 2014), and this was also reiterated by the responses to the present research which shows that 27.9% of responses use Waterfall lifecycle with Agile . The waterfall is used with agile methods when requirements are identified in advance and the documentation is needed. The second method that uses with agile is prototyping by 23.6%, and the reason of using prototyping with agile is that this method has many features of Agile and is the best choice to make a combination with Agile when the customer cannot participate in developing the project.

Solinski and Petersen (2014) discovered that Scrum and XP are the most well-known and adopted methods, and this was also reiterated by the responses to the present research by 88.5% and 32.7% respectively. There are three types of projects (small/medium/large) provided in the survey and, in attempting to find out which methods are more effective with these projects, three questions were formulated. The XP and Scrum are more effective with small projects by 67.3% and 40.4% respectively , while Scrum and the Lean and Kanban methods are more effective with medium size projects by 82.7% and 21.2 %respectively, and in large projects, the Scrum and Feature-Driven Development are considered most effective by 65.4% and 32.7% respectively.

In addition, all small companies develop Windows-based systems along with others since they uses the XP method, which is for Web-based software ventures (Maurer and Martel, 2002). Basically, the most common systems developed by

medium companies are business systems and embedded systems, whereas the most used method is Scrum since it is a framework of software development which can be used for software projects, manage products or application developments such as Industrial Technology (IT) companies (Permana and Bali, 2015). Large organizations develop data processing and safety-critical items along with other systems.

In terms of the success factors, 25 conceivable basic achievement factors are listed in this research for each of the seven venture success classes; these are: control, dealing with changing requirements, quality, customer satisfaction scope, effort estimation, reducing the delivery schedules, and increasing return on investment. The correlation test was used to study the positive relationship between the success factors and success attributes, and the number of factors were reduced to 17 factors as the culture of organizations, team structure, management support, customers' dominant issues, having team members with high capabilities, encouragement, project champion, assigning essential features first, frequent delivery of software, high competence for team and organizational issues, the use of tools, correct integration testing, project category, smaller-size team, agenda, clarity, and Strong customer participation.

For further detailed investigation, the multiple regression techniques were utilized to increase the accuracy of our conclusions as result of which only 5 out of 45 theories were supported, distinguishing three basic achievement factors for Agile software development ventures which are: A) Assigning essential features first. B) Frequent delivery of software, and C) The use of tools. These factors have an impact on four success attributes which are: the quality, effort estimation (cost and time), customer satisfaction, and reducing delivery schedule.

Many studies were conducted between 1996 and 2012 related to the success factors and as explained in chapter 2. To compare our results with pervious investigations in this area, we used three studies to show the similarities and differences between other results and ours since they used different approaches. First, a study was done by Chow and Cao in 2008 to examine the achievement factors in agile programming projects based on a quantitative approach. This work was an investigation of 109 Agile ventures from 25 countries all over the world and utilized

multiple regression methods, with the results stating that 10 out of 48 speculations were reinforced among these establishments and perceived as primary achievement factors for Agile software development projects.

Another study was directed by Kumar and Goe in 2012 to present and clarify the factors considered by software professionals while adopting Agile strategies, as well as the impacts of agile adoption techniques on clients and business practicing with agile methods. Their study proposed six premises: effect of group size, effect of requirements collection for agile methods, efficient requirement collection process, time needed to solve issues and effect of insufficient corresponding time with the client in software improvement.

In detail, the factors include: delivery of strategy, agile programming engineering methods and group ability. According to their findings, three factors - one of which, we have already observed - is delivery of strategy, but the other factors are different from what we obtained. The same occurred with Misra et al. (2009), Zulkefi et al (2010), and Kumar and Goe (2012) studies; the first one identified 9 factors namely: client cooperation, client content, client obligation, conclusion time, company culture, people's characteristics, societal culture, and preparing and learning; while the second study found different factors as the outcome of client participation, communication, least change requirement, companies' culture, active testing, clarity, allocation of time.

Both studies Misra et al. (2009), Zulkefi et al (2010) share some factors such as client obligation, conclusion time, and company culture. The last study Kumar and Goe (2012) discovered six factors which are: Effect of group size, effect of requirements collection for agile methods, efficient of requirement collection process, time needed to solve issues and effect of insufficient corresponding time with the client in software improvement.

As a result, some factors are shred between the present work and the first study in previous paragraph, whereas the outcomes are quite different for the other studies shown above when multiple regression techniques results are taken into account. In the case of correlation test results, however, 9 factors appear to be similar to those

named in the previous four studies, except eight factors, which are: having team members with high capability, project champion, assigning essential features first, the use of tools, correct integration testing, project category, and team structure.

The reason behind such difference can be summarized in this way: first, the sample size is different from one study to next; large samples increase the variety of the responses. Second, each study uses different techniques, such as chow and Cao's study (2008) with multiple regression techniques similar to the present work; while Misra's et al (2009) used the broad scale-based technique. Third, the number of factors included in each study is different, with some investigating more factors than others. For example, Misra et al. (2009) included 14 anticipated factors, while Chow and Cao (2008) study involved 12 factors.

In addition, 11 failure factors were identified in this study based on the existing literature. Thus, a final set of 11 failure factors is obtained for each of the seven venture success classes – control, dealing with changing requirements, quality, customer satisfaction scope, effort estimation, reducing the delivery schedules, and increasing return on investment. After using a correlation test to study the negative relationship between the success factors and success attributes, the number of factors was reduced to seven, which are: absence of management support, too-large an organization size, too traditional and conservative an organization culture, absence of a full set of right agile practices, absence of customer presence, absence of tracking mechanisms during agile progress, and determining the role of the client.

Then, to increase the accuracy of our conclusions, multiple regression techniques were utilized. Consequently, on only 1 hypothesis was supported out of 13, identifying one significant factor that can cause failure which is related to organization size being too large and it may lead to lose control over the work, and does contributes to the failure factor, in addition to identifying a new failure attribute which is losing control over the work. Failure-related research in software development, works are commonly dependent on lessons learned from specific sorts of ventures; however, they are for the most part similar and generalized (Coram and Bohner, 2005) there has been no study to investigate the failure factors using a negative correlation relationship with

success attributes as accomplished in this research, thereby rendering our results new for the available literature.

Software development companies adopt agile approaches because of their benefits, which can be counted as better control over the work, reducing the delivery schedules, and finishing the tasks quickly. In addition, when comparing these outcomes with the data from companies of different size, one can observe that providing better control over the work is still viewed as the primary advantage of the Agile methods within large and small companies, while for the medium-size companies, the priority is switched to coping with changing user requirements. This is also a new observation and a first in the available literature.

Some of the results here are aligned with previous studies; finishing the tasks quickly is one of the benefits established here and also mentioned in other studies as Agile development has a tendency to concentrate on the right-on-time and quick property of methods applied(Beck and Fowler, 2001) (Cockburn, 2001).

Chow and Cao in 2008 classified the achievement of Agile strategy in four various sets, which are: quality, scope, time and cost (Chow and Cao, 2008). However, according to our results, quality is less important than others. According to the author, the reason behind this is that these companies do not have a separate unit for software processing and improvement. The results conclude that control over the work is the most important reason to adopt agile methods, meaning these companies adopted Agile because it provides better control over the work for both customers and developers alike.

Finally, most responses do not provide any comments regrading adopting agile, expect four. One of them believes that, nowadays, a good agile system is very important in companies and to make sure it is being implemented is also a requirement until it becomes a tradition in the organization. Another one thinks that the main goal of agile is to help deliver artifacts sooner and to avoid misunderstandings with the client. The most interested comment is about specific type of projects that need to use agile among which, one can refer to research and development (R&D)projects.

CHAPTER 6

CONCLUSION

Using quantitative methods, this study used survey data to examine adopting of agile methods in small- and medium-scale software development organizations, which agile methods are used in large projects, if the choice of one method over others is affected by the size of the project, which other methods are used along with agile and, finally, what critical success and failure factors exist in Agile software development projects. The data gathered from 52 Agile software development companies from different-size organizations and geographic areas provided sufficient data for statistical analysis to address core issues and arrive at the conclusions.

First, the Agile methods used more than the others in large and complex projects are Scrum, the Feature-Driven Development (FDD) , Lean and Kanban, and XP.

Second, the decision as to choose one of the Agile methods over the others is affected by the type of the projects - small, medium, or large in scale – and, for each size, there are specific methods preferred by different enterprises. For example, small-size projects commonly use the XP and Scrum methods, while in developing the medium projects the most used methods are Scrum, Lean, and Kanban methods.

Third, the results show that most companies prefer to adopt Agile methods in combination with other methods, and that these methods are mainly Waterfall Lifecycle, Prototyping, and Spirals.

Fourth, software development companies adopt the agile apporoach because of their benefits, mainly that the results present better control over the work, reduce the delivery schedules, and finish the tasks quickly. In addition, when we compare these

outcomes with the data from companies of different sizes, we can see that providing better control over the work is still viewed as the primary advantage of the Agile methods within large and small companies, while for the medium-size companies, the priority is switched to coping with changing user requirements.

Fifth, the outcomes show that only 5 out of 45 theories were supported, distinguishing three basic achievement factors for Agile software development ventures: a. Assigning essential features first; b. Frequent delivery of software, and c. The use of tools.

Sixth, in terms of failure factors, only 1 hypothesis was supported out of 13, identifying one significant factor that can cause failure which is too-large an organization size, besides identifying a new failure attribute, which is losing control over the work.

However, the main contribution of this study is to reduce a large number of recounted success factors variables to three basic ones in view of the survey information analysis, as well as to reduce failure factors to one main factor.

6.1. Research limitations

Despite the fact that this research achieved its objectives, there are still certain unavoidable restrictions that should be taken into account. These limitations are reflected in what follows.

To begin with, this study is constrained by the presumption limits that the information acquired across various work functions are equally critical. It would have been more interesting to explore if there are at all any differences in terms of the outcomes in light of the work elements of the respondents. Nevertheless, this ambition requires a change in the design of the original survey and instruments of research and, as such, shall be left to future initiatives.

Next, the sample size is rather small considering the large Agile society population. A larger sample size from more countries could, in turn, provide more

generic and precise statistical calculation and examination and, furthermore, give a chance to compare the other factors involved in companies of different sizes. Despite the fact that the responses were from different countries, we could not compare all the factors of adopting agile methods among these countries for the same reason as stated above.

Eventually, Samples collected from different countries, but the samples for each one of these countries was different and not even close to each other which make it hard to compare between different countries in terms of adopting agile methods.

6.2. Future Work

There are some issues that deserve more investigation in the future. Three important ones are:

1. As a result of this research, the failure attributes are identified. Hence, it would be beneficial to have further research and design new surveys to study these attributes in detail in software development companies.

2. The sample size is to be increased to have a chance to compare the adoption of agile methods between different companies based on the size of companies.

3. More respondents from different countries are to be involved in the survey to compare between these countries in terms of adopting agile methods in organizations.

REFERENCES

Abbas, N., Gravell, A. M. & Wills, G. B. (2008). Historical roots of Agile methods: Where did "Agile Thinking" come from?. IN ABRAHAMSSON, P.

Abdalhamid,S., & Mishra, A. (2017a). Factors in Agile Methods Adoption. TEM Journal,6(2), 416-421.

Abdalhamid,S., & Mishra, A. (2017b). Adopting of Agile methods in software Development organization: Systematic Mapping. *TEM Journal,6(4), 817-825*

Agile Alliance. Manifesto for Agile Software Development. (2016). Retrieved from http://www.Agilemanifesto.org .

Ahmad, M. O., Markkula, J., & Oivo, M. (2013). Kanban in software development: A systematic literature review. *In Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on* (pp. 9-16). IEEE.

Allen, P. (2001). Light methodologies and CBD. Component Development Strategies .*XI*:1–16.

Ambler, S. (2006). Results from Scott Ambler's 2006 Agile Adoption Rate Survey.

Ambler, S. (2008). Results from Scott Ambler's 2008 Agile Adoption Rate Survey.

Asnawi.A., Gravell.A., & Wills.G. (2011). Empirical investigation on Agile methods usage: issues identified from early adopters in Malaysia. *In Agile Processes in Software Engineering and Extreme Programming.Springer*, pp. 192–207.

Bahli, B. & Abou zeid, E. S. (2005). The role of knowledge creation in adopting extreme programming model: An empirical study. *IN SALEM, M. A. & ELHADIDI, M. T. (Eds.) ITI 3rd International Conference on Information and Communications Technology (ICICT 2005). Cairo, EGYPT, IEEE.*

Bannet, S., McRobb, S., and Farmer, R. (2010).object-oriented systems analysis and design using UML. *4th edition*.

Beck, K. (1999). Extreme Programming Explained: Embrace Change. *(First ed.).* Addison-Wesley Professional.

Beck, K. (2000). Extreme Programming Explained: Embrace Change. *Addison-Wesley, ISBN* 0-201-61641-6.

Beck, K., & Fowler, M. (2001). Planning Extreme Programming. Applied. *Addison-Wesley, Boston, MA.*

Begel, A., & Nagappan, N. (2007) .Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study. *First International Symposium on Empirical Software Engineering and Measurement*, 2007. ESEM 2007.

Belassi, W, & Tukel,O.I. (1996). A new framework for determining critical success/failure factors in projects. *Int. J. Proj. Manag*, vol. 14, no. 3, pp. 141–151.

Boehm, B., & Turner, R. (2003). Using risk to balance Agile and plan-driven methods. *Computer* 36 (6), 57–66.

Boehm, B., and Turner, R. (2005). Management challenges to implementing Agile processes in traditional development organizations. *IEEE* 22 (5).30-39.

Brooks, F. (1987). No Silver Bullet: Essence and Accidents of Software Engineering. *IEEE Computer* (20:4), pp. 10-19.

Burgess,T.F. (2001). A general introduction to the design of questionnaires for survey research.

Cao L, Mohan K, Xu P, Ramesh B. (2004). How Extreme does extreme programming have to be? Adapting XP practices to large-scale projects. *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) —Track 3—Vol. 3 (05–08 January 2004). HICSS. IEEE Computer Society: Washington DC, 2004; 30083.3.*

Chan, F. K., & Thong, J. Y. (2009). Acceptance of Agile methodologies: A critical review and conceptual framework. *Decision support systems*, 46(4), 803-814.

Chow, T., & Cao, D. (2008). A Survey Study of Critical Success Factors in Agile Software Projects. *The Journal of Systems and Software*, 81, 961-971.

Cockburn, A., & Highsmith, J. (2001). Agile software development: The people factor. Computer, 34, 131-133.

Cockburn, A. (2004). Crystal Clear: A Human-Powered Methodology for Small Teams. *Addison-Wesley.ISBN* 0-201-69947-8.

Cohn, M., & Ford, D.(2003). Introducing an Agile process to an organization. *Computer* 36 (6), 74–78.

Collect and organize information big & small with google. (2016). Retrieved from, "https://www.google.com/forms/about/.

Cooke, J. L. (2010). Agile Principles Unleashed: Proven Approaches for Achieving Real Productivity in Any Organisation. *Ely, U.K.: IT Governance Pub*.

Coram,M., & Bohner,S.(2005). The Impact of Agile Methods on Software Project Management A Brief Look at Agile Methods. *Engineering, pp*. 363–370.

Create a survey using Google Form. Retrieved from, https://support.google.com/docs/answer/87809?hl=en.

David. J, Anderson. (2010). Kanban: Successful Evolutionary Change for Your Technology Business. Sequim, *WA: Blue Hole Press*.

Dikert, M., Paasivaara, C., & Lassenius. (2016). Challenges and success factors for large-scale Agile transformations: a systematic literature review.

Dyba, T. & Dingsoyr, T. (2008) Empirical studies of Agile software development: A systematic review. *Information and Software Technology*, 50, 833-859.

Dybå, T., &s Dingsøyr,T. (2009). What do we know about Agile software development?. *IEEE Computer Society*, 2009.

Dynamic System Development Method. (2015). Retrieved from ,http://www.freetutes.com/systemanalysis/sa2-dynamic-system-development-method.html.

Elshamy, A., & Elssamadisy, A. (2007) Applying Agile to large projects: new Agile software development practices for large projects. *IN CONCAS, G., DAMIANI, E., SCOTTO, M. & SUCCI, G. (Eds.) Agile Processes in Software Engineering and Extreme Programming. 8th International Conference, XP 2007. Como, Italy, Springer-Verlag.*

Firdaus, A., Ghani, I., & Yasin, N.I.M. (2013). Developing Secure Websites Using Feature Driven Development (FDD): A Case Study. *Journal of Clean Energy Technologies*, Vol. 1, No. 4.

Flora,H.K., & Chande,S.V.(2014). A Systematic Study on Agile Software Development Methodologies and Practices. *International Journal of Computer Science and Information Technologies*, Vol. 5 (3).

Forrester. (2007). Lean software is Agile, “Metodologías Ágiles de desarrollo: Un panorama general.

Forrester. (2008). Lean software is Agile, fit-to-purpose, and efficient,” tech. rep., Forrester.

Fowler, M., & Highsmith, J. (2001). The Agile Manifesto," in: Software Development. pp. 28-35.

Gandomani, T. J., Zulzalil, H., Ghani, A. A. A., Sultan, A. B. M., & Nafchi, M. Z. (2013). Obstacles in moving to Agile software development methods; at a glance. *Journal of Computer Science, 9*(5), 620.

Gunasekaran, A. (2001). Agile manufacturing: The 21st century competitive strategy. *1st ed. Oxford; New York: Elsevier*.

Highsmith, J. (2002a). Agile Software Development Ecosystems. *Boston: Addison, MA.*

Highsmith,J.(2002b) .Agile Software Development .*The journal of defense software engineering*,Vol.15 No.10.

Highsmith,J.A.(2000). Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. *Addison Wesley*.

Hilkka, M. R., Tuure, T. & Matti, R. (2005). Is extreme programming just old wine in new bottles: A comparison of two cases. *Journal of Database Management*, 16, 41-61.

Holstrom, H., Fitzgerald, B., Agerfalk, P. J. & Conchuir, E. (2006) Agile practices reduce distance in global software development. *Information Systems Management*, 23, 7-18.

Iivari, J., & Huisman, M. (2007) .The relationship between organizational culture and the deployment of systems development methodologies. *Mis Quarterly*, 31, 3558.

Ilieva, S., Ivanov, P. & Stefanova, E. (2004) Analyses of an Agile methodology implementation. *IN STEINMETZ, R. & MAUTHE, A. (Eds.) 30th EUROMICRO Conference. Rennes, FRANCE, IEEE Computer Soc.J. Syst. Software, 119 (2016), pp. 87-108.*

Jacobi, C., & Rumpe, B. (2001). Hierarchical XP-Improving XP for large scale projects in analogy to reorganization processes.

John, M., Maurer, F. & Tessem, B. (2005) Human and social factors of software engineering: workshop summary.*ACM SIGSOFT Software Engineering Notes*, 301-6.

Klimeš, C & Procházka, J.(2006). New Approaches in Software Development. *In Acta Electrotechnica et Informatics*, 6(2).

Kniberg, H. (2012, July, 19). Kanban vs Scrum – how to make the most of both. Retrieved from, via: www.crisp.se/file-uploads/Kanbanvs-Scrum.pdf.

Krasteva, I. & Ilieva, S. (2008). Adopting an Agile methodology: why it did not work. *Proceedings of the 2008 international workshop on Scrutinizing Agile practices or shoot-out at the Agile corral. Leipzig, Germany, ACM*.

Kroll, P., & Kruchten, P. (2003). The rational unified process made easy: a practitioner's guide to the RUP. *Addison-Wesley Professional*.

Kuhrmann, M., & Fern´andez, D, M (2015). Systematic software development: A state of the practice report from germany. *In International Conference on Global Software Engineering*. IEEE.

Kuhrmann, M., Fern´andez, D.M., & Tiessler, M. (2014).A mapping study on the feasibility of method engineering. *Journal of Software: EvolutionandProcess*, 26 (12):10531073, 2014.

Kumar.A., & Goel.B.(2012). Factors Influencing Agile Practices: A Survey.

Laanti, M., Salo, O. & Abrahamsson, P.(2011). Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on Agile transformation. *Information and Software Technology, 53, 276-290.*

Lan, C. & Ramesh, B. (2007) Agile software development: ad hoc practices or sound principles? *IT Professional*, 9, 41-47.

Larman, C. (2004). Agile & Iterative Development. Addison-Wesley, Boston, Massachusetts.

Law, A. & Charron, R. (2005) Effects of Agile practices on social factors. *Proceedings of the 2005 workshop on Human and social factors of software engineering. St. Louis, Missouri, ACM*.

Lean Software Development. Retrieved from https://www.cs.colorado.edu/~kena/classes/5828/s12/presentation-materials/schweikertmarcbubernakchris.pdf.

Lee, E. C. (2008). Forming to performing: Transitioning large-scale project into Agile. *IN MELNIK. M. (Eds.) AGILE 2008 Conference. Toronto, CANADA, IEEE Computer Soc.*

Leffingwell, D. (2007). Scaling Software Agility - Best Practices for Large Enterprises. *Addison Wesley*.

Lindvall, M., Basili, V., Boehm, B., Costa, P., Dangle, K., Shull, F., TESORIERO, R., WILLIAMS, L. & ZELKOWITZ, M. (2002) Empirical findings in Agile methods. *Lecture Notes in Computer Science*, 197-207.

Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., et al., 2004. Agile software development in large organizations. *Computer* 37 (12), 26–34.

Livermore, J. A. (2006). What Elements of XP are being Adopted by Industry Practitioners. *SoutheastCon, 2006. Proceedings of the IEEE*.

Livermor, J. A. (2007) .Factors that impact implementing an Agile software development methodology. *SoutheastCon, 2007. Proceedings. IEEE*.

Livermore, J. A. (2008). Factors that Significantly Impact the Implementation of an Agile Software Development Methodology. *JSW,* 3(4), 31-36.

Lycett, M., Macredie, R., Patel, C. & Paul, R. (2003). Migrating Agile methods to standardized development practice. *IEEE, Computer*, 36, 79-85.

Madeyski, L. & Biela, W. (2007). Capable Leader and Skilled and Motivated Team practices to introduce eXtreme Programming. *IN MEYER, B., NAWROCKI, J. R. & WALTER, B. (Eds.) 2nd IFIP Central and East European Conference on Software Engineering Techniques (CEE-SET 2007). Poznan, POLAND, Springer-Verlag Berlin.*

Mahanti, A. (2006). Challenges in enterprise adoption of Agile methods-A survey. *CIT. J. Comput. Inf. Technol.*, vol. 14, no. 3, pp. 197–206.

Mann, C. & Maurer, F. A case study on the impact of scrum on overtime and customer satisfaction. *Proceedings of the Agile development Conference (ADC'05). Denver, CO*. 70-79. 2005.

Mannaro, K., Melis, M. & Marchesi, M. (2004). Empirical analysis on the satisfaction of IT employees comparing XP practices with other software development methodologies. *IN ECKSTEIN, J. & BAUMEISTER, H. (Eds.) 5th International*

Conference Extreme Programming and Agile Processes in Softwave Engineering. Garmisch-Partenkirchen, GERMANY, Springer-Verlag Berlin.

Mansor Saadiah Yahya,Z., & Habibah,N.A. (2010). Success_Determinants_in_Agile_Software_Development_Methodolog y_ICMLC_Updated.

Mar, K., and Schwaber, K.(2002). Scrum with XP. *Informit.com*.

Maria, P. & Casper, L. (2006). Could Global Software Development Benefit from Agile Methods?. *Global Software Engineering, 2006. ICGSE '06*.

Martin, A., Biddle, & R., Noble, J. (2004). The XP Customer role in practice: Three studies. *2nd Agile Development Conference (ADC 2004).* Salt Lake City, UT, IEEE Computer Soc.

Maurer, F., & Martel, S. (2002). Extreme programming: rapid development for web-based applications. *IEEE Internet Computing*. pp. 86-90.

McClave, J.T., & Benson, P.G. (1988). Statistics for Business and Economics. *Dellen Publishing, San Francisco, California.*

Mcavoy, J. & Butler, T. (2007). A Failure to Learn in a Software Development Team: The Unsuccessful Introduction of an Agile Method. *IN BARRY, C., CONBOY, K., LANG, M., WOJTKOWSKI, G. & WOJTKOWSKI, W. (Eds.) 16th International Conference on Information Systems Development. Galway, IRELAND, Springer.*

Melnik, G. & Maurer, F. (2002). Perceptions of Agile practices: a student survey. *IN WELLS, D. & WILLIAMS, L. (Eds.) Extreme Programming and Agile Methods - XP/Agile Universe 2002. Second XP Universe and Firs t Agile Universe Conference. Proceedings. Chicago, IL, USA, Springer-Verlag.*

Melnik, G., & Maurer, F. (2005) A cross-program investigation of students' perceptions of Agile methods. *27th International Conference on Software Engineering (ICSE 2005). St Louis, MO, Assoc Computing Machinery*.

Melo,C., Cruzes,D., Kon,F., & Conradi,R.(2011). Agile team perceptions of productivity factors. *in Agile Conference (AGILE) , pp. 57–66*.

Mens, T., Wermelinger, M., Ducasse, S., Demeyer, S., Hirschfeld, R. & Jazayeri, M. (2005). Challenges in software evolution. Principles of Software Evolution. *Eighth International Workshop on.*

Methods and Tools, "Adoption of Agile Methods ". (2007). Retrieved from, http://www.methodsandtools.com//dynpoll/oldpoll.ph p?Agile.

Miller, K. W. (2004). Test driven development on the cheap: text files and explicit scaffolding. *J. Comput. Small Coll*., 20, 181-189.

Misic , V. (2006) Perceptions of extreme programming: an exploratory study. *ACM SIGSOFT Software Engineering Notes*, 31, 1-8.

Misraa, S.G., Kumarb, V., & Kumarb, U. (2009). Identifying some important success factors in adopting Agile software development practices. *The Journal of Systems and Software* 1869–1890.

Moe, N. B., Dingsoyr, T., Dyba, T. & Society, I. C. (2008) Understanding selforganizing teams in Agile software development. *19th Australian Software Engineering Conference. Perth, CANADA, IEEE Computer Soc*.

Moran, D. (2010). Top 10 Reasons to Use Agile Development. Retrieved from, http://www.devx.com/enterprise/Article/44619.

Nasir, M., Ahmad, R. & Hassan, N. (2008). An Empirical Study of Barriers in the Implementation of Software Process Improvement Project in Malaysia. *Journal of Applied Sciences*, 8, 4362-4368.

Nerur,S.,Mahapatra,R.K., & Mangalaraj,G. (2005).Challengesofmigrating to Agile methodologies. *Communications of the ACM* 48 (5), 72–78.

Object-oriented programming. (2015). Retrieved from, http://en.wikipedia.org/wiki/Object_oriented.

Oluwole,D.(2013). Agile X Factors. Retrieved from, Https://www.scrumalliance.org.

Paetsch, F., Eberlein, A., & Maurer, F. (2003). Requirements Engineering and Agile Software Development. *In WETICE* (Vol. 3, p. 308).

Palmer ,S.R., & Felsing ,J. M. (2002). A Practical Guide to Feature-driven Development. *Prentice Hall. Upper Saddle River. NJ, ISBN* 0- 13- 067615-2.

Permana, P.A.G., & Bali, S.S .(2015). Scrum Method Implementation in a Software Development Project Management. *International Journal of Advanced Computer Science and Applications*, Vol. 6, No. 9.

Poppendieck, M., & Poppendieck, T. (2003). Lean Software Development: An Agile Toolkit for Software Development Managers. *AddisonWesley, Boston, ISBN* 0-321-15078-3.

R. Boehm., & Turner, R. (2004). Balancing agility and discipline: evaluating and integrating Agile and plan-driven methods. *In: Proceedings of the 26th International Conference on Software Engineering (ICSE'04), Edinburgh, Scotland*, pp. 718–719.

Raith, F., Richter, I., & Weber, M. (2016). Problems in Agile Global Software Engineering Projects especially within Traditionally Organised Corporations:[An exploratory semi-structured interview study]. In *Proceedings of the Ninth International C* Conference on Computer Science & Software Engineering* (pp. 33-43). ACM.

Rajlich, V. (2006). Changing the paradigm of software engineering. *Commun. ACM*, 49, 67-70.

Ramesh, B., Cao, L., Mohan, K. & Xu, P. (2006) Can distributed software development be Agile? . *Communications of the ACM*, 49, 41-46.

Reel, J.S. (1999). Critical success factors in software projects. *IEEE Software* 16 (3), 18–23.

Robinson, H. & SHARP, H. (2005a) .Organisational culture and XP: three case studies. Proceedings. Agile 2005. *Denver, CO, IEEE Computer. Soc*.

Robinson, H. & Sharp, H. (2005b) The social side of technical practices. *IN BAUMEISTER, H., MARCHESI, M. & HOLCOMBE, M. (Eds.) Extreme Programming and Agile Processes in Software Engineering. 6th International Conference, XP 2005. Proceedings. Sheffield, UK, Springer-Verlag.*

Rubin, A., & Babbie, E. (1997). Research Methods for Social Work. *Brooks/ Cole Publishing Company, Pacific Grove, California*.

Salo, O., & Abrahamsson, P. (2005). Integrating Agile software development and software process improvement: a longitudinal case study. *ISESE 2005, 4th International Symposium on Empirical Software Engineering, Noosa Heads, Australia, November*.

Sarmiento V.E., & Vasquez M.L. (2012). A model for measuring agility in small and medium software development enterprises. *Informatics (CLEI),Conference Latino Americana En (2012),* pp. 1-10, 10.

Schatz, B., & Abdelshafi, I. (2005). Primavera gets Agile: A successful transition to Agile development. IEEE Software. 22(3).

Schwaber, C., & Fichera, R. (2007). Corporate IT Leads the Second Wave of Agile Adoption.

Siger, T., Hazzan, O. & Bar-nahor, R. (2008). Agile Orientation and Psychological Needs, Self-Efficacy, and Perceived Support: A Two Job-Level Comparison. *Agile, 2008. AGILE '08. Conference.*

Sidky.A., Arthur, J., & Bohner, S. (2007). A disciplined approach to adopting Agile practices: the Agile adoption framework. Innovations in systems and software engineering, 3(3), 203-216.

Simons , M. (2002). Big and Agile?. Cutter IT Journal; 15:34–38.

Snap 10 Professional. http://download.cnet.com/Snap-10-Professional/3000-2650_4-75001541.html.

Solinski,A., & K.Petersen, K. (2014). Prioritizing Agile benefits and limitations in relation to practice usage. *Software Quality Journal,pages*1–36.

Soundararajan, S., & Arthur, J. D. (2009). A soft-structured Agile framework for larger scale systems development. In *Engineering of computer based systems. 16th annual ieee international conference and workshop on the* (pp. 187-195). IEEE.

SPSS. (2017, October, 08). Retrieved from https://en.wikipedia.org/wiki/SPSS.

Stankovic, D.,Nikolic.V., Djordjevic ,D., & Cao.D. (2013). A survey study of critical success factors in Agile software projects in former Yugoslavia IT companies. *Journal of Systems and Software*, 86(6): p. 1663-1678.

Stapleton,J. (1997). Dynamic systems development method – The method in practice. *Addison Wesley* .

Strode, D., Huff, S. L., & Tretiakov, A. (2008b). The impact of organizational culture on Agile method use. 2009 42nd Hawaii International Conference on System Sciences. HICSS-42. Big Island, HI,, IEEE.

Strode, D., Huff, S., Wellinton, N., Tretiakov, A., North, P. & Zealand, N. (2008a). Investigating the Target Environment for Agile Methods. *19th Australian Conference on Information System. Christchurch, Australia.*

Tessem, B. (2003). Experiences in learning xp practices: A qualitative study. *Extreme Programming and Agile Processes in Software Engineering*, 1012-1012.

Thakur, S. & A. Kaur,. (2013). Role of Agile Methodology in Software Development.

Theocharis,G., Kuhrmann, M., Munch,J., & Diebold,P. (2015). Is water- scrum-fall reality? on the use of Agile and traditional development practices. *In Proceedings of the International Conference on Product Focused Software Process Improvement, ser. Lecture Notes in Computer Science, vol. 9459. Berlin Heidelberg: Springer, pp. 149–166.*

Tolfo, C. & Wazlawick, R. S. (2008). The influence of organizational culture on the adoption of extreme programming. Journal of Systems and Software, 81, 1955-1967.

Tripp, J. (2012). The Impacts of Agile Development Methodology Use on Project Success: A Contingency View.

Turk, D., France, R., & Rumpe, B. (2005). Assumptions underlying Agile software development processes. *Journal of Database Management* 16 (4)62–87.

Turk, D., France, R. & Rumpe, B. (2002). Limitations of Agile software processes.

Vavpotic, D. & Bajec, M. (2009). An approach for concurrent evaluation of technical and social aspects of software development methodologies. *Information and Software Technology*, 51, 528-545.

Vijayasarathy,L. R., & Turk, D. (2008). Agile Software Development: A Survey of Early Adopters. *Journal of Information Technology Management*, 21(2). 701.

Vijayasarathy, L., & Turk, R.D. (2008). Agile Software Development: A Survey Of Early Adopters. *Journal of Information Technology Management ISSN* #1042-1319.

Wan,J., & Wang.R. (2010). Empirical Research on Critical Success Factors of Agile Software Process Improvement. *J. Softw. Eng. Appl*., vol. 3, no. 12, p. 1131.

Weidner, R. (2014). Agile adoption, How to Successfully Transition an Organization to the Scrum Framework.

Whitworth, E. (2008). Experience Report: The Social Nature of Agile Teams. Agile, 2008. *AGILE '08. Conference.*

Williams, F.,& Monge, P. (2001). Reasoning with Statistics. *Thomson Wadsworth, Belmont, California.*

Williams, L., & Cockburn, A. (2003). Agile software development: It's about feedback and change. *Computer*, 36, 39-43.

APPENDICES

Appendix A

Background Information Figures

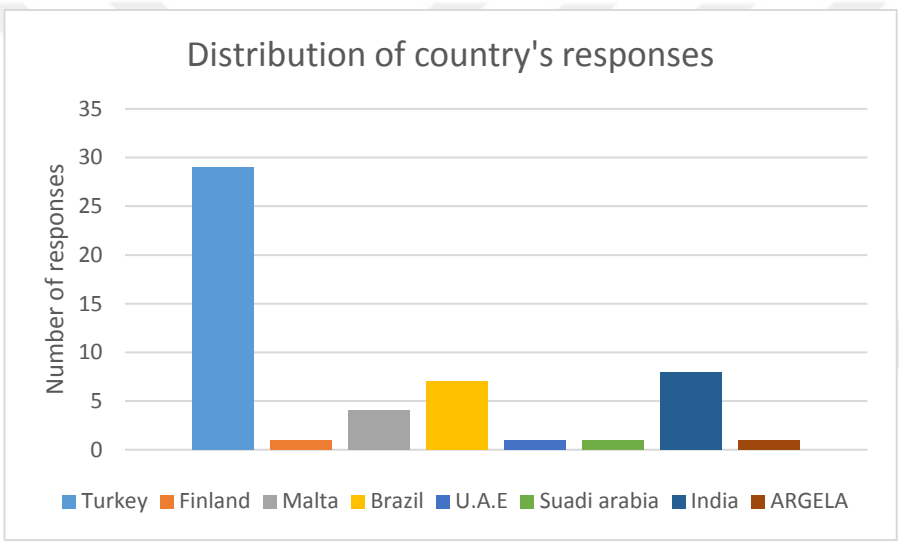

Figure 24. Country distribution

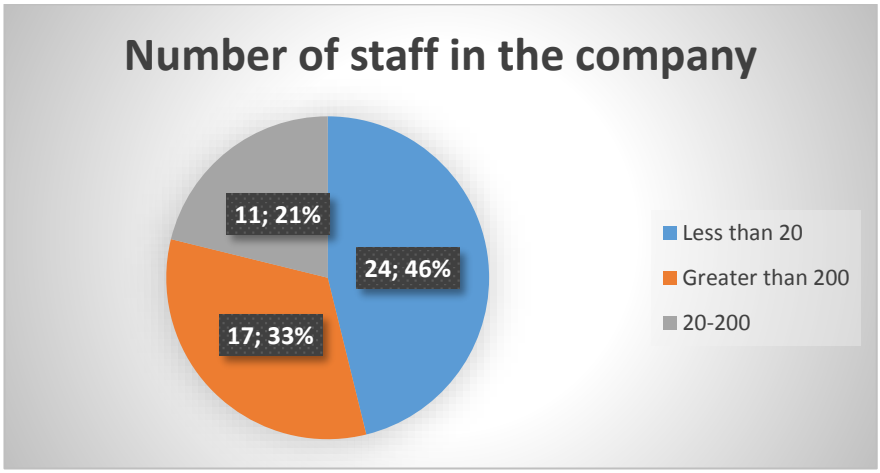

Figure 25. Companies size distribution

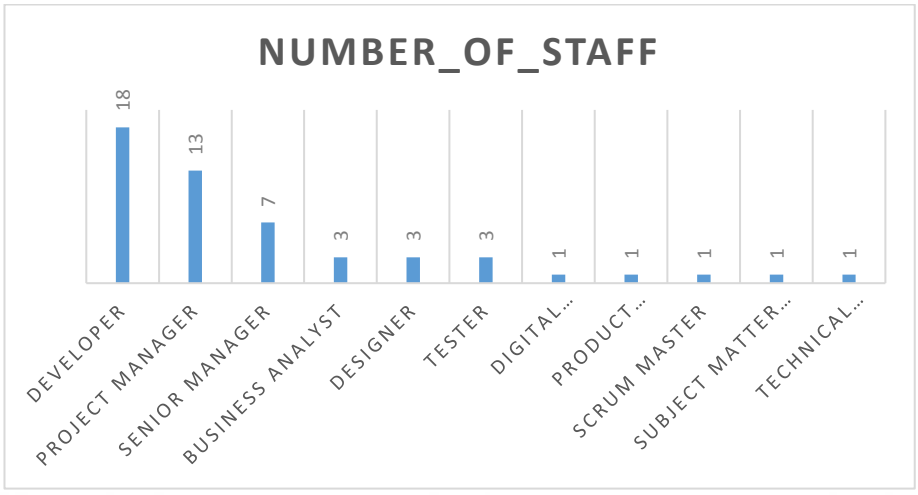

Figure 26. Profile of responders

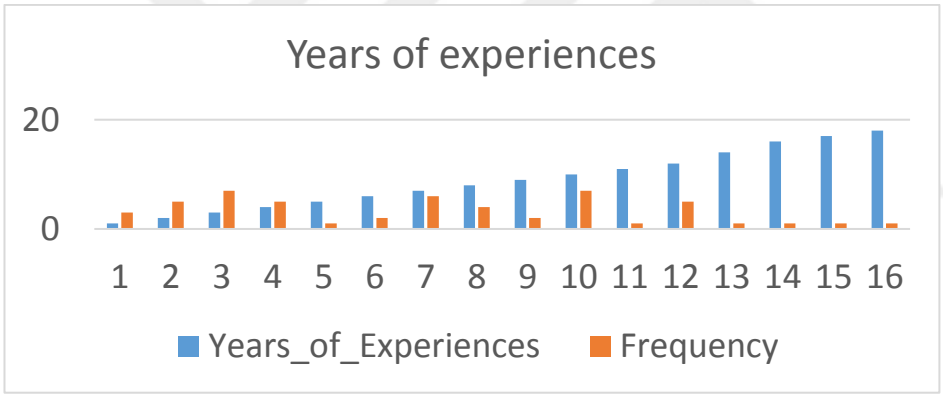

Figure 27. Years of experiences in Agile development

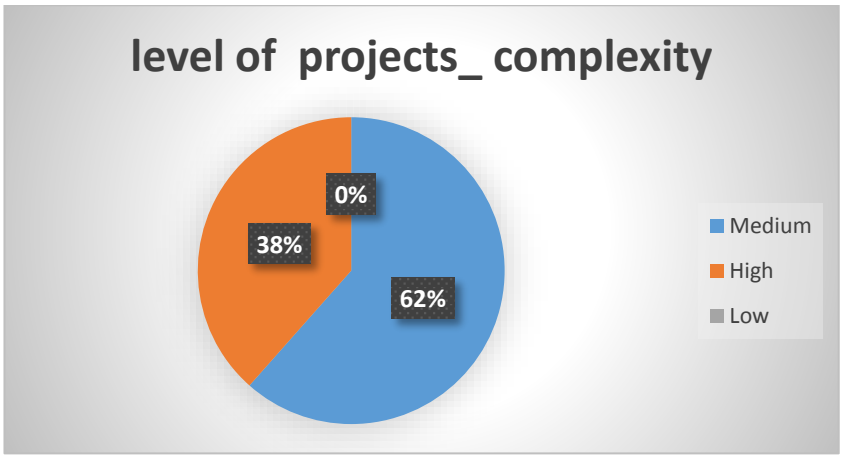

Figure 28. Level of projects complexity distribution

Figure 29. companies with CMMI certification distribution

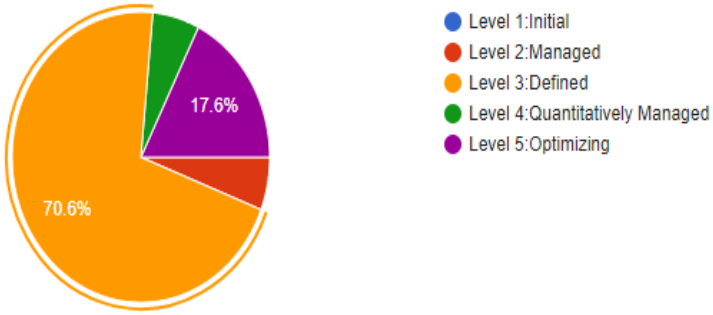

Figure 30. Participant Companies' CMMI levels distribution

Appendix B

Correlation Test Results for Success Factors

Table 32. Results of correlation test for success factors

SF/SA	Depen1	Depen2	Depen3	Depen4	Depen5	Depen6	Depen7
Indep 1	r=0.032 p=0.823	r=0.093 p=0.510	r=0.048 p=0.737	r=-0.241 p=-0.086	r=0.070 p=0.621	r=0.027 p=0.849	r=0.275 p=0.049
Indep 2	r=0.184 p=0.191	r=0.303 p=0.029	r=0.367 P=0.007	r=0.037 p=0.795	r=0.296 p=0.033	r=0.23 p=0.099	r=0.126 p=0.373
Indep 3	r=0.224 p=0.110	r=0.215 p=0.126	r=0.171 p=0.224	r=0.000 p=1.000	r=0.289 p=0.038	r=0.159 p=0.260	r=0.221 p=0.116
Indep 4	r= -0.112 p= 0.430	r=-0.100 p=-0.480	r=0.059 p=0.679	r=-0.133 p=-0.424	r= -0.156 p= -0.269	r=0.052 p=0.823	r=-0.032 p=-0.823
Indep 5	r=0.093 p=0.510	r=0.230 p=0.101	r=0.255 p=0.068	r=0.024 p=0.868	r=0.046 p=0.746	r=0.0177 p=0.210	r=0.227 p=0.105
Indep 6	r=0.268 p=0.054	r=0.160 p=0.256	r=0.212 p=0.131	r=-0.177 p=0.211	r=0.221 p=0.116	r=0.061 p=0.668	r=0.181 p=0.200
Indep 7	r=0.360 p=0.009	r=0.152 p=0.282	r= 0.078 p=0.583	r=-0.120 p=0.395	r=0.207 p=0.141	r= -0.025 p=0.863	r= 0.079 p=0.576
Indep 8	r=0.208 p=0.138	r=0.181 p=0.199	r=0.296 p=0.033	r=-0.131 p=0.356	r=0.170 p=0.277	r=0.309 p=0.062	r=0.218 p=0.121
Indep 9	r=0.043 p=0.760	r=0.043 p=0.764	r=0.102 p=0.473	r=0.073 p=0.604	r=0.074 p=0.603	r=0.030 p=0.835	r=0.022 p=0.879
Indep 10	R=0.141 p=0.318	r=0.173 p=0.220	r=0.036 p=0.799	r=-0.027 p=0.850	r=0.080 p=0.572	r=0.030 p=0.833	r=0.148 p=0.295
Indep 11	r=0.349 p=0.011	r=0.281 p=0.044	r=0.301 p=0.030	r=0.106 p=0.454	r=0.303 p=0.029	r=0.297 p=0.032	r=0.273 p=0.051
Indep 12	r=0.133 p=0.349	r=0.204 p=0.146	r=0.211 p=0.133	r=0.014 p=0.919	r=0.313 p=0.024	r=0.322 p=0.020	r=0.214 p=0.129
Indep 13	r=0.037 p=0793	r=0.219 p=0118	r=0.383 p=0.005	r=0.123 p=0.385	r=-0.007 p=0.005	r=0.255 p=0.069	r=0.374 p=0.006

Indep 14	r=0.372 p=0.007	r=0.173 p=0.220	r=0.437 p=0.001	r=0.040 p=0.776	r=0.441 p=0.001	r=0.398 p=0.003	r=0.447 p=0.001
Indep 15	r=0.167 p=0.235	r=0.271 p=0.052	r=0.415 p=0.002	r=0.018 p=0.898	r=0.424 p=0.002	r=0.244 p=0.081	r=0.424 p=0.002
Indep 16	r=0.082 p=0.565	r=0.313 p=0.024	r=0.382 p=0.005	r=0.378 p=0.006	r=0.535 p=0.000	r=0.432 p=0.001	r=0.308 p=0.026
Indep 17	r=0.187 p=0.185	r=0.312 p=0.024	r=0.365 p=0.008	r=0.106 p=0.453	r=0.349 p=0.011	r=0.278 p=0.046	r=0.343 p=0.013
Indep 18	r=0.321 p=0.020	r=0.174 P=0.218	r=0.154 p=0.275	r=0.063 p=0.659	r=0.151 p=0.286	r=0.156 p=0.286	r=0.165 p=0.242
Indep 19	r=0.129 p=0.362	r=0.383 p=0.005	r=0.246 p=0.079	r=0.033 p=0.818	r=0.512 p=0.000	r=0.381 p=0.005	r=0.385 p=0.005
Indep 20	r=0.250 p=0.073	r=0.167 p=0.237	r=0.158 p=0.263	r=0.125 p=0.379	r=0.210 p=0.135	r=0.177 p=0.211	r=0.318 p=0.021
Indep 21	r=0.083 p=0.559	r=0.125 p=0.379	r=0.036 p=0.800	r=0.084 p=0.555	r=0.134 p=0.344	r=0.094 p=0.507	r=0.100 p=0.482
Indep 22	r=0.109 p=0.440	r=0.110 p=0.439	r=0.071 p=0.617	r=0.019 p=0.893	r=0.110 p=0.438	r=0.115 p=0.418	r=0.264 p=0.059
Indep 23	r=0.272 p=0.051	r=0.098 p=0.530	r=0.099 p=0.483	r=0.069 p=0.627	r=0.098 p=0.490	r=0.232 p=0.098	r=0.378 p=0.006
Indep 24	r=0.300 p=0.030	r=0.402 p=0.003	r=0.117 p=0.410	r=0.021 p=0.844	r=0.356 p=0.010	r=0.217 p=0.122	r=0.297 p=0.032
Indep 25	r=0.174 p=0.217	r=0.233 p=0.096	r=0.134 p=0.344	r=0.000 p=1.000	r=0.263 p=0.060	r=0.259 p=0.064	r=0.140 p=0.323

Appendix C

Correlation Test Results for Failure Factors

Table 33. Results of correlation test for Failure factors

FF/ SA	Depen1	Depen2	Depen3	Depen4	Depen5	Depen6	Depen7
Indep1	r= -0.155 p= 0.273	r= -0.211 p= 0.134	r=0.000 p=1.000	r=0.329 p=0.017	r=0.000 p=1.000	r=-0.026 p= 0.853	r=- 0.106 p=0.456
Indep2	r= 0.440 p=0.001	r= -0.311 p=0.025	r= -0.100 p=0.480	r=-0.154 p= 0.276	r= -0.372 p=0.007	r=-0.143 p= 0.312	r= -0.122 p=0.389
Indep3	r= -0.163 p= 0.247	r= -0.311 p= 0.025	r=0.122 p=0.389	r=0.210 p=0.135	r= -0.111 p= 0.432	r=0.048 p=0.733	r= -0.146 p= 0.301
Indep4	r= -0.199 p= 0.157	r= -0.027 p= 0.850	r=0.094 p=0.509	r=0.252 p=0.072	r= -0.170 p= 0.228	r=0.158 p=0.263	r= 0.198 p=0.159
Indep5	r= -0.157 p= 0.265	r= -0.113 p= 0.424	r=0.083 p=0.557	r=0.228 p=0.104	r= -0.120 p= 0.397	r=0.122 p=0.389	r= 0.039 p= 0.781
Indep6	r= -0.057 p= 0.686	r=- 0.206 p= 0.142	r= -0.102 p= 0.474	r=0.193 p=0.171	r= -0.074 p= 0.604	r=0.028 p=0.844	r=0.031 p=0.826
Indep7	r= -0.052 p= 0.714	r= -0.306 p= 0.027	r=0.054 p=0.706	r=0.119 p=0.402	r= -0.287 p= 0.039	r= -0.059 p=-0.677	r=- 0.226 p= 0.108
Indep8	r= -0.104 p= 0.463	r= -0.168 p= 0.235	r= -0.171 p= 0.266	r=- 0.053 p=-0.711	r= -0.222 p= 0.144	r=- 0.118 p=-0.405	r= -0.095 p= 0.501
Indep9	r= -0.222 p= 0.113	r= -0.381 p= 0.005	r= -0.096 p= 0.498	r=0.215 p=0.125	r= -0.409 p= 0.003	r=0.052 p=0.713	r= 0.023 p= 0.871
Indep10	r= -0.173 p= 0.220	r= -0.113 p= 0.426	r=0.014 p=0.922	r=0.200 p=0.156	r= -0.348 p= 0.011	r=0.000 p=1.000	r=0.003 p=0.983
Indep11	r= -0.327 p= 0.018	r= -0.329 p= 0.017	r= -0.106 p= 0.455	r=0.068 p=0.633	r= -0.324 p= 0.019	r=- 0.127 p=-0.371	r= -0.203 p= 0.149

Appendix D

Questionnaire

Factors In Agile Methods Adoption In Small And Medium Enterprises

There are six sections in this survey. Please answer all questions. Thank you for your time and efforts!

* Required

Section I Background

1. Country : *

2. Approximately what is the number of staff at your company? *

- ◯ Less than 20
- ◯ 20 – 200
- ◯ More than 200

3. What is your job in the firm? *

- ◯ Project Manager
- ◯ Senior Manager
- ◯ Developer
- ◯ Tester
- ◯ Designer
- ◯ Scrum Master
- ◯ Business Analyst
- ◯ Other:

4.How many years of experiences do you have in Agile development? *

5.What is the level of your projects complexity? Note: As the project size increases, the complexity of the project often increase as well. *

- ◯ High
- ◯ Medium
- ◯ Low

6. So far, in how many projects have you used Agile methods? Please mention the number of projects?

7. Does your organization have CMMI certification? *

- ◯ Yes
- ◯ No

8. If your answer is 'yes' to Item 7, please specify the level of your company?

- ◯ Level 1: Initial
- ◯ Level 2: Managed
- ◯ Level 3: Defined
- ◯ Level 4: Quantitatively Managed
- ◯ Level 5: Optimizing

9. If your answer is 'no' to Item 7, what other type of software process improvement certification do you have?

- ◯ SPICE
- ◯ ISO/IEC JTC1 SC7 standard
- ◯ ISO/IEC 15504 standard
- ◯ Other:

10. If you have not adopted Agile methods, specify the reasons?

- In our organization, any form of change meets with resistance.
- We do not have enough experience and skills in Agile
- Other: _____

Section II Project Characteristic In Agile Methods Adoption .

Note: The project size depends on the following criteria: .Total financial resources available .Number of team members involved ·Number and size of deliverables to be produced ·Complexity of deliverables to be produced. Time frames involved in delivery

11. What type of systems does your company develop? You may choose more than one *

- [ ] Safety critical
- [ ] Data processing
- [ ] Systems software
- [ ] Telecommunications
- [ ] Business systems
- [ ] Real-time systems
- [ ] Windows based
- [ ] Embedded systems
- [ ] Other: _____

12. Agile methods were used in the project/projects *

- Completely
- Used along with other structured methods
- Rarely used

13. If you use Agile with other methods, please specify those other methods?

- [ ] Prototyping
- [ ] Spiral
- [ ] Waterfall lifecycle
- [ ] Other: _____

14. Which one of the Agile methods mentioned below do you often use? *

- [ ] The XP method
- [ ] The Scrum method
- [ ] Dynamic Systems Development Method (DSDM)
- [ ] Feature-Driven Development (FDD)
- [ ] Crystal
- [ ] Rational Unified Process (RUP)
- [ ] Adaptive software development (ASD)
- [ ] Lean and Kanban Software Development

15. In your opinion, which one of the Agile methods mentioned below is more effective with small projects? *

- [ ] The XP method
- [ ] The Scrum method
- [ ] Dynamic Systems Development Method (DSDM)
- [ ] Feature-Driven Development (FDD)
- [ ] Crystal
- [ ] Rational Unified Process (RUP)
- [ ] Adaptive software development (ASD)
- [ ] Lean and Kanban Software Development

16. In your opinion, which one of the Agile methods mentioned below is more effective with medium projects?*

- [ ] The XP method
- [ ] The Scrum method
- [ ] Dynamic Systems Development Method (DSDM)
- [ ] Feature-Driven Development (FDD)
- [ ] Crystal
- [ ] Rational Unified Process (RUP)
- [ ] Adaptive software development (ASD)
- [ ] Lean and Kanban Software Development

17. In your opinion, which one of the Agile methods mentioned below is more effective with large and complex projects? *

- [ ] The XP method
- [ ] The Scrum method
- [ ] Dynamic Systems Development Method (DSDM)
- [ ] Feature-Driven Development (FDD)
- [ ] Crystal
- [ ] Rational Unified Process (RUP)
- [ ] Adaptive software development (ASD)
- [ ] Lean and Kanban Software Development

18. Adopting Agile methods in a complex project was the main reason behind the project's failure?*

- ◯ Yes
- ◯ No

19. What are the Agile practices used by your company so far? . choose more than one *

- [ ] Continuous integration
- [ ] System metaphor
- [ ] Code standard
- [ ] Scrum daily meetings
- [ ] Refactoring
- [ ] Small release cycles
- [ ] Code and design reviews
- [ ] Use of design patterns
- [ ] Other: _____

Section III The Success Factors of Agile Methods Adoption

20. In terms of organizational factors, please rate the following aspects of organizational factor *

	very important	Important	Neutral	Unimportant	Very Unimportant
The culture of Organizations	◯	◯	◯	◯	◯
Team structure	◯	◯	◯	◯	◯
Team organization	◯	◯	◯	◯	◯
Management support	◯	◯	◯	◯	◯
Maintain Agility	◯	◯	◯	◯	◯
Acceptance of Agile universal methodology	◯	◯	◯	◯	◯

21. In terms of the 'people' factor , please rate the following aspects *

	Very important	Important	Neutral	Unimportant	Very Unimportant
Customer-dominant issues					
Team members with High capabilities					
Practicing and learning					
Communication and arbitration					
Encouragement					
Project champion					

22. In terms of the technical factor, please rate the following aspects *

	Very important	Important	Neutral	Unimportant	Very Unimportant
Assigning essential features first					
Frequent delivery of software					
High competence for team and organizational issues					
The use of tool					
Correct integration testing					

23. In terms of the 'project' factor, please rate the following aspects *

	Very important	Important	Neutral	Unimportant	Very Unimportant
Project category					
Smaller- size teams					
Agenda					
Least changes of requirement					
Projects within advance hazard analysis performed					

24. In terms of the 'process' factor, please rate the following aspects. *

	Very important	Important	Neutral	Unimportant	Very Unimportant
Clarity					
Strong customer participation					
Efficient requirements-gathering method					

Section IV The Failure Factors of Agile Methods Adoption

25. In terms of the organizational factor, please rate the following aspects.*

	Very important	Important	Neutral	Unimportant	Very Unimportant
Absence of managemen support					
Too large an organization size					
Too traditional & Political an organizational culture					

26. In terms of the 'people' factor, please rate the following aspects. *

	Very important	Important	Neutral	Unimportant	Very Unimportant
Terrible customer relationship					
Absence of cooperation					
Absence of essential skill- set					

27. In terms of the technical factor, please rate the following aspects *

	Very Important	Important	Neutral	Unimportant	Very Unimportant
Absence of full set of right Agile practices					
Inadequacy of the technology and tools					

28. In terms of the 'process' factor, please rate the following aspects *

	Very Important	Important	Neutral	Unimportant	Very Unimportant
Absence of customer's presence					
Absence of tracking mechanisms during Agile progress					
Determine the role of customers					

Section V The Acceptance of Agile

29. Please select one of the scales that represents your opinion *

	Strongly Agree	Agree	Neutral	Disagree	Strong Disagree
The usage of Agile provides me with better control over my work					
Using Agile allows me to finish my tasks quickly					
Agile methods are used because they cope better with changing user requirements.					
Agile adoption allows us to achieve better quality.					
Agile methods are selected because of the type of the project.					
Agile is used because it helps in effort estimation (cost, schedule) for software development.					
Agile methods are used because they result in more customer satisfaction.					
Using the Agile methods help us to reducing the delivery schedules.					
The Agile methods are used to increase the return on investment.					

Section VI Comments

30. Please use this section to provide any additional comments you may have about the use of Agile methods in your work environment. Your comments help us to better comprehend your perspectives as a whole.