University of Southampton

Faculty of Engineering and Physical Sciences

Electronics and Computer Science

A Review on the Critical Success Factors of Agile Software Development: an Empirical Study

by

Abdullah Mohammed Aldahmash

Thesis for the degree of Doctor of Philosophy

September 2018

University of Southampton

Abstract

Faculty of Engineering and Physical Sciences

Electronics and Computer Science

Thesis for the degree of Doctor of Philosophy

A Review on the Critical Success Factors of Agile Software Development: an Empirical Study

by

Abdullah M. Aldahmash

Given the evolution and increasing usage of agile development practices and techniques, the successful implementation of agile development is crucial. Agile software development has become one of the most commonly used methodologies for developing software, and it promises to deliver many benefits. Nevertheless, the implementation of agile practices and techniques requires many changes that might be a challenge for organisations attempting to succeed with agile software development. The relevant literature presents a great deal of research which has studied the critical success factors (CSFs) of agile software development. This study aims firstly to review the literature related to agile software development in order to identify the CSFs of agile software development. With this in mind, one of the objectives of this study is to investigate those factors which contribute to the success of agile software development. This study also aims to explore the relations between these factors and to suggest a set of measurements which could be used to measure the success of agile software development projects. To achieve these objectives, this research has employed empirical research methodologies aiming to address the research objectives. All of the research methods employed in this study have received ethical approval from the ethical committee of the School of Electronics and Computer Science at the University of Southampton.

This research involved carrying out an exploratory study to investigate the identified success factors of agile software development. A web-based survey was distributed to agile practitioners in order to obtain their beliefs regarding the importance of the identified success factors. As a result, it was possible to order the CSFs of agile development by importance. Communication was found to be the most important success factor. The relations between the agile project's progress and the importance of these factors were explored. Using factor analysis, the inter-relations between the identified success factors were also investigated. The success factors were split into two components with the aim of developing a better understanding of said factors; the two resulting components were as follows: the organisational and people component, and the technical and project component.

This research, moreover, developed an instrument with which the success of agile development projects could be evaluated. The proposed instrument includes a list of questions and metrics to measure the success of agile development projects. Agile experts were interviewed to review the development of the proposed instrument. Following the feedback from the experts, the instrument

was amended. Once this stage had been completed, the instrument was used in three case studies; the aim of this was to seek a practical evaluation on whether the proposed instrument is valid which was confirmed and some suggestions on how it could be improved were obtained.

To summarise, this research attempted to recognise the CSFs and to understand their importance, how this varies through the agile project, and their interrelations to provide insights into these CSFs. Furthermore, this research developed and validated an instrument to measure and evaluate the success in addressing these CSFs during an agile software development project.

Table of Contents

Table of Tables

Table of Tables

Table of Figures

Dedicated to

My mother

For her love, prayers, supports, encouragements

Without which I could not accomplish anything in my life

Thank you for always being there for me

My Bride

For her love that made it worthwhile

My Sisters and Brothers

For their unconditional love

Research Thesis: Declaration of Authorship

Print name:	Abdullah M. Aldahmash

Title of thesis:	A Review on the Critical Success Factors of Agile Software Development: an Empirical Study

I declare that this thesis and the work presented in it are my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as:

Aldahmash, A., Gravell, A. M., & Howard, Y. (2017). Critical Success Factors of Agile Software Development: Review Study. 18th International Conference on Agile Software Development, XP 2017, Cologne, Germany, May 22-26, 2017. Proceedings of the XP17 Scientific Posters Session.

Aldahmash, A., Gravell, A. M., & Howard, Y. (2017). A Review on the Critical Success Factors of Agile Software Development. Systems, Software and Services Process Improvement: 24th European Conference, EuroSPI 2017, Ostrava, Czech Republic, September 6-8, 2017. Communications in Computer and Information Science, vol. 748. Springer. pp. 504-512.

Research Thesis: Declaration of Authorship

Aldahmash, A., Gravell, A. M., & Howard, Y. (2017). Using Factor Analysis to Study the Critical Success Factors of Agile Software Development. 10th International Conference on Computer Science and Information Technology (ICCSIT 2017), Florence, Italy, October 23-25, 2017.

Aldahmash, A., Gravell, A. M., & Howard, Y. (2017). Using Factor Analysis to Study the Critical Success Factors of Agile Software Development. JSW, vol. 12 no. 12, pp. 957-963.

Aldahmash, A., & Gravell, A. M. (2018). Measuring Success in Agile Software Development Projects: a GQM Approach. 13th International Conference on Software Engineering Advances ICSEA 2018, Nice, France, October 14-18, 2018.

Signature:		Date:	

Acknowledgements

I would like to express my thanks and appreciations to my supervisor Dr. Andy Gravell for his supports, encourages, feedbacks without which this research could not be completed. By working closely with Andy, I learned many things besides this dissertation.

Many thanks go to the organisations and individuals who participated in this research. Their participation contributed positively to the completion of this study. I would like also to thank my sponsor Ministry of Education, Kingdom of Saudi Arabia for funding my studies abroad.

Huge thanks go to my family and friends, for their patience, support, and love which made it worthy.

Definitions and Abbreviations

CSF Critical Success Factor

SDLC Software Development Life Cycle

ISD Information System Development

XP Extreme Programming

FDD Feature Driven Development

DSDM Dynamic Software Development Method

RQ Research Question

CI Confidence Interval

EV Extreme Values

EVA Exploratory Factor Analysis

CFA Confirmatory Factor Analysis

PCA Principal Component Analysis

PAF Principal Axis Factoring

KMO Kaiser-Meyer-Olkin

GQM Goals Questions Metric

PM Project Management

IS Information System

IT Information Technology

KBI Key Business Indicator

CMMI Capability Maturity Model Integration

SPICE Software Process Improvement and Capability Determination

PMO Project Management Office

SPI Schedule Performance Index

Chapter 1 Introduction

Agile software development matters have been progressively investigated in software engineering research. According to Sjøberg, Dybå and Jørgensen (2007), in the future, software engineering research should elaborate more on empirical studies. Such empirical research would lead to established scientific knowledge regarding how different software engineering methods, tools, and techniques are being practically used. In the case of agile development, empirical studies should enable a better understanding of how agile principles and practices are adopted. Important factors associated with agility, such as people, process, and organisational culture, are unlikely to be addressed without solid empirical research. This study will use empirical methodologies to answer the undermentioned research questions concerning the definition of success in agile software development and how this success can be understood and measured.

Although agile software development promises many benefits, it is also associated with difficulties and barriers, which the organisations attempting to implement agile practices need to be aware of. According to the latest State of Agile Report (State of Agile Report, 2018), the top three reasons for adapting agile development are to accelerate product delivery, enhance the ability to manage changing priorities, and to increase productivity. However, Boehm and Turner (2005) identified several barriers preventing the success of agile development in an organisation. These barriers are not only technical in nature, but also organisational and people-related. Being successful in implementing agile practices and techniques is the aim of many organisations and individuals, and numerous researchers have investigated this. Declaring an agile software project a success is difficult, and there is a need for more measurements with which agile success could be evaluated.

The critical success factors of agile software development have been studied and explored over the last decade or more. However, most previous research has dealt with agile development as one whole project. The nature of agile development projects is iterative, thus meaning that the development must be based on phases or iterations. As such, the success of agile software development is a result of the success of implementing and managing these phases and iterations. It can only be helpful for agile practitioners to know which success factors are more important than other factors during a specific time or situation throughout the agile software development life cycle. It is also vital to understand the inter-relations between the success factors and how these success factors are linked to each other. To the best of our knowledge, this is the first study which has attempted to analyse the success factors of agile software development based the agile project progress and to investigate the inter-relations between these success factors. To achieve said goals, an empirical investigation will be carried out to study the agile success factors in their real context.

This research also aims to weight the importance of the success factors of agile software development. Moreover, the present study will investigate how the importance of the success factors varies during agile development projects. In addition, the relationships between the CSFs of agile software development will be explored in this research using statistical techniques. It is hoped that this will provide a better understanding of agile development success and how it is possible to avoid the agile barriers or challenges.

Furthermore, this research aims to suggest measurements with which the success of agile software development projects could be evaluated. The proposed measurements have been validated by interviewing agile experts and applied in case studies. The aim is to gain a practical evaluation of the measurements and to improve the suggested measurements. It is hoped that by using such empirical methodologies, it is possible to develop a greater understanding of measuring success in agile software development.

1.1 Research Motivations and Objectives

Prior to starting this study, the researcher worked as a software engineer for five years. During this time, he worked on implementing agile practices and techniques in several software development projects. He noticed that agile principles are have been the mainstream when it comes to software development. Consequently, there will be a need for a rigorous definition on how to succeed in implementing agile practices and how that success could be assessed and measured. Nowadays, agile principles go beyond software development to many other fields. As stated in a report from the Boston Consulting Group (BCG), organisations should take advantage of agile practices that transform the software development so that they can deploy these practices in other businesses within various sectors, from finance and human resources, to marketing (BCG, 2017). This shows the potential of agile practices and points to the need for software engineering researchers to establish profound consciousness about success in implementing agile practices and techniques. Such consciousness will not only be helpful for software engineers, but also for many others behind the scenes.

Given how the use of agile methods and practices has evolved, the successful implementation of agile development is vital. Although agile methods and practices promise many benefits to the software development process, they also necessitate the implementation of many changes. The agile transformation process is not straightforward, and encompasses numerous barriers which can potentially hinder the successful transformation. Many organisations are attempting to implement agile practices and techniques. These organisations or individuals are facing a challenge when it comes to defining success in implementing agile practices and also when it comes to assessing and

evaluating their own implementation. Declaring success in implementing agile software development practices is difficult. Although many research studies have identified the success factors of agile software development, little is known about how these success factors are related to each other or how they could be measured. As such, there is a need to investigate the conditions or factors which contribute to the success of agile software development. It is also essential to explore the success measurements, and how it is possible to measure and tackle success in agile software development projects.

In light of this, the research objectives are as follows:

- To review the relevant literature and identify the success factors of agile software development.
- To investigate the differences in importance between the identified success factors, the relations between the agile project's progress and the success factors, and the inter-relations among the identified agile success factors.
- To employ a set of statistical tests with which the quantitative data can be analysed and comprehended.
- To suggest measurements which could be used to measure the success of agile software development projects.
- To validate the proposed measurements by employing empirical research methods.

Chapter 2 introduces the research questions which were defined to achieve the aforementioned objectives. In this dissertation, a research design (as shown in chapter 3) is followed to investigate the research questions with the aim of addressing the research objectives. The research questions which will be addressed throughout this thesis are:

RQ1. What are the differences in importance between the CSFs and what is the weight for each factor?

RQ2. Does each success factor of agile software development have the same importance during all phases or iterations of the agile project?

RQ3. Using factor analysis, how can the CSFs of agile development be grouped into a smaller number of categories?

RQ4. What is the appropriate instrument with which to measure the success of an agile software development project in an organisation?

The research questions were positively answered by the research conducted in this dissertation. Having answered the research questions and achieved the research objectives, this research made

a set of contributions to the field of agile software development. The contributions of this research are highlighted in the conclusion chapter (chapter 8).

1.2 Report Structure

The remainder of this report is structured as follows: Chapter 2 provides an overview of the relevant literature concerning agile software development and the success factors of agile software development; this chapter also identifies the research gap and the research questions. Chapter 3 describes the research methodology and the research methods used in the present study. The results and findings of the survey are presented in Chapter 4. Following that, these results are discussed in Chapter 5. In Chapter 6, the development of the proposed instrument is examined. Chapter 7 introduces the case studies and discusses their findings. Lastly, Chapter 8 presents the conclusion drawn from this research and puts forth directions for future work.

1.3 Peer-reviewed and Published Work

Some results of this dissertation have already been published, in the following peer reviewed conferences and journals:

Aldahmash, A., Gravell, A. M., & Howard, Y. (2017). Critical Success Factors of Agile Software Development: Review Study. *18th International Conference on Agile Software Development, XP 2017,* Cologne, Germany, May 22-26, 2017. Proceedings of the XP17 Scientific Posters Session.

Aldahmash, A., Gravell, A. M., & Howard, Y. (2017). A Review on the Critical Success Factors of Agile Software Development. *Systems, Software and Services Process Improvement: 24th European Conference, EuroSPI 2017,* Ostrava, Czech Republic, September 6-8, 2017. *Communications in Computer and Information Science*, vol. 748. Springer. pp. 504-512.

Aldahmash, A., Gravell, A. M., & Howard, Y. (2017). Using Factor Analysis to Study the Critical Success Factors of Agile Software Development. *10th International Conference on Computer Science and Information Technology (ICCSIT 2017),* Florence, Italy, October 23-25, 2017.

Aldahmash, A., Gravell, A. M., & Howard, Y. (2017). Using Factor Analysis to Study the Critical Success Factors of Agile Software Development. *JSW*, vol. 12 no. 12, pp. 957-963. (Extended paper).

Aldahmash, A., & Gravell, A. M. (2018). Measuring Success in Agile Software Development Projects: a GQM Approach. *13th International Conference on Software Engineering Advances ICSEA 2018,* Nice, France, October 14-18, 2018. (Invited to submit an extended version).

Chapter 2 Literature Review

This chapter discusses the software development process methodologies and the agile momentum. In section 2.1, an overview of the software development life cycle and a comparison between the traditional methodologies and the agile methodology are discussed. Following that, section 2.2 provides an overview of agile software development and from where it started. Section 2.3 reviews the different agile practices and methods and it compares the most common agile methods. In section 2.4, a review study on the critical success factors of agile development is discussed. Section 2.5 explores the relationships between the identified success factors and the agile principles. The research gap is discussed in section 2.6. Following that, the research questions which will be answered in this study are listed in section 2.7. Lastly, section 2.8 summaries the chapter.

2.1 Software Development Life Cycle

The concept of software development life cycle (SDLC) has been well developed during the early years of the field of software engineering. According to Davis, Bersoff, & Comer (1988), software development life cycle has been facilitated and implemented by many models. It started with (Royce, 1987) who introduced the waterfall model and later refined by (Boehm, 1988) with the spiral model. In 2001, Agile manifesto (Beck et al., 2001) introduced the agile principles to the software development community. The waterfall, spiral, and agile models are discussed below:

- Waterfall model: the traditional way of developing software. This is a sequential model, with every stage having to be completed before the start of the next stage. The waterfall model pays more attention to the documentation during the development (Royce, 1987). In the waterfall model, most challenges are related to requirement gathering and verification (Petersen et al., 2009).

- Spiral model: with this model, the entire development process is evolved from many iterations. Each iteration contains all five stages, from planning to coding; indeed, and at the end of each iteration, there should be an iteration delivery. The most important features of the spiral model are its risk-driven approach and the combination between prototyping and evolutionary development (Boehm, 1988).

- Agile model: this is an iterative development model with no rigid or agreed upon requirement. The agile model encourages changes in requirements during the development and supports the involvement of the users; indeed, it focuses on improving communication and collaboration between the stakeholders (Highsmith & Cockburn, 2001).

Differences between waterfall, spiral, and Agile Developments are listed in Table 2-1.

Table 2-1 Differences between waterfall, spiral, and agile development Adapted from (Leffingwell, 2007)

Software Development Methodology	**Waterfall Development**	**Spiral Development**	**Agile Development**
Management	Plan-Driven	Iteration- Driven	Result-Driven
Requirement	Defined up-front	Defined up-front each iteration	Emergent/ Evolve
Implementation/ Test	Code all the features and test later	Implementation and Test at the end of each iteration	Code and test in parallel
Delivery	Once at the end	At the end of each iteration	Continuous
Documentation	Comprehensive	Moderate	Limited

Agile Development is becoming the mainstream methodology in software development. In the coming section, agile development will be reviewed.

2.2 Agile Software Development

Agile practices have been frequently used and common since the late 1990s. With some agile principles even rooted before that, from the spiral model. The agile momentum in the software industry started with the Agile Manifesto. In 2001, a group of software practitioners introduced the Agile Software Development Manifesto (Beck et al., 2001). While there is no agreed upon definition of agile development in software engineering, most of the definitions emphasise the idea of speed and flexibility when it comes to responding to changes during the software development (Kettunen and Laanti, 2008). According to Conboy (2009), agility denotes “the continual readiness of an information system development (ISD) method to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment”.

The Agile Manifesto (Beck et al., 2001) introduced the following values in agile development:

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

While there is a value on the left side, the items on the right side are more important (Beck et al., 2001). According to agile principles, the focus should be on adding value rather than following the plan. Delivering working software to the users frequently and in a short period of time could add a value for the users (Dingsøyr et al., 2012). Agile development expects the software development team to deliver early and then gain feedback early, meaning it can make changes more easily, improve quality, and conduct constant testing. Agile development focuses on quality in design, with the design phase, as well as other phases, developed on an ongoing basis (Highsmith & Cockburn, 2001). The agile manifesto (Beck et al., 2001) introduced twelve principles of agile development, which are listed in Table 2-2.

Table 2-2 Agile Principles from Agile Manifesto (Beck et al., 2001)

Agile Principles
1. Customer satisfaction.
2. Welcome requirements changes, even late.
3. Within short timescale, deliver working software frequently.
4. Business people and development team should work closely during the development.
5. Motivated individuals in a supportive environment.
6. The most effective way of communication is face-to-face conversation.
7. The main measure of progress is the working software.
8. Sustainable development.
9. Attention to technical excellence and good design.
10. Simplicity.
11. Self-organised teams.
12. The behaviour of the team should be adjusted on how to be more effective.

The agile mindset supports the changes during the development rather than discouraging them. It also encourages feedback from the users as early as possible (Highsmith & Cockburn, 2001). Performance management and social factors are crucial to success in an agile environment. The development team, users, and top management should follow a result-driven approach and avoid being plan-driven during the agile development (Sheffield and Lemétayer, 2013). Agile practices address two of the toughest challenges facing business and technology nowadays: firstly, the need for an innovative approach in developing software and, secondly, the need for a work environment which is dynamic in responding to frequent changes (Highsmith & Cockburn, 2001).

The agile philosophy encourages the overlapping of roles and tasks within the development team. Such overlapping will likely lead to skills improvement among the team members. This overlapping of roles and redundancy of skills will probably enhance the team's ability to respond to changing requirements throughout the development project (Nerur & Balijepally, 2007).

Agile software development has many characteristics, all of which are identified in the literature. According to Dyba & Dingsoyr (2009) the key characteristics of agile development are collaboration and communication, embracing conflicts, and supporting creativity and innovation; managers are only facilitators, and the design and implementation processes are iterative and indivisible.

Cockburn & Highsmith (2001) claimed that the agile mindset places more emphasis on the people factor. A skilled team is crucial, and so each individual team member adds more value to the agility of the development process. Thus, agility not only reduces the documentation effort in a development project, but also focuses on building a highly-skilled team capable of using any technique or tool to achieve its objectives. Cockburn & Highsmith (2001) stated that "Agile processes are designed to capitalize on each individual and each team's unique strengths".

In addition to this, Boehm & Turner (2005) identified several barriers preventing the implementation of agile practices in a legacy organisation. They divided the barriers into three categories, the first of which is software development conflicts, which are linked to the process of developing the software itself. Since the agile methods and the traditional methods differ, the process of requirement gathering, designing, implementation... etc. will require new approaches which will, in turn, require new techniques, tools, and skills to develop software. The second category is related to business process conflicts. Because agile development is evolutionary or iterative, there will be tremendous changes in the way business processes are built or managed. The third category is people conflicts. According to Boehm & Turner (2005), people conflicts constitute the most critical barrier standing in the way of development. Agile practices require new skills and mindsets. The people factor is a vital aspect of agile movement, since agile relates to motivated individuals and flexibility in a supportive environment.

Gandomani et al. (2013) claimed that the obstacles in agile transformation originate from the organisational culture and structure. They suggested that organisations attempting to move to agility should pay attention to the efforts involved in moving from a process-centric model to a people-centric model. Unlike the traditional methodologies, agility relies on people and their innovation and creativity, rather than the processes and the technology (Cockburn and Highsmith, 2001).

Vijayasarathy & Turk (2008) conducted a survey which revealed that the use of agile development will lead to the ability to meet customer needs and to deliver quality software. Other identified benefits of using agile development include the increasing of flexibility in development, as well as low development costs, more frequent working software delivery, and more reusable code. Since many facets of software development exist in a rather complex domain, agile development could be seen as a technique with which to address this complexity (Pelrine, 2011).

Begel & Nagappan (2007), from Microsoft Research, conducted a survey with 487 respondents, all of whom were asked what they felt were the top benefits and problems associated with agile development. The top three benefits were the improvement of communication and coordination among the development team, faster delivery and release, and quicker response to changes/design flexibility. On the other hand, the top three problems were the difficulties with large-scale projects, many required meetings, and management buy-in, the latter of which means being strict with dates (Begel and Nagappan, 2007).

Although that agile model appears to become the mainstream model for software development, agile benefits and limitations are yet to be clearly agreed upon. There is a need for a clearer definition of agile development and its success factors and barriers so that the organisations who attempting to adopt agile development could benefit from adopting the agile model (Bustard et.al, 2013). According to the latest State of Agile Survey (State of Agile Report, 2018), the top three reasons for adopting agile were to accelerate product delivery, enhance the ability to manage changing priorities, and increase productivity. The reasons for adopting agile are shown in Figure 2-1.

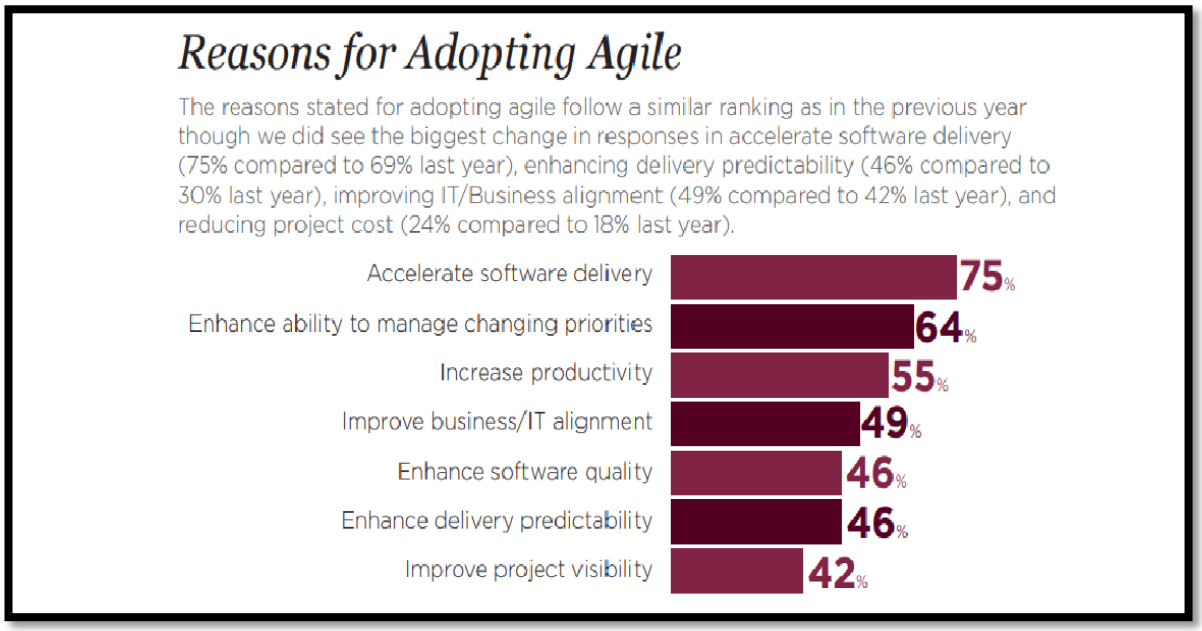


Figure 2-1 Reasons for Adopting Agile from —A State of Agile Survey 2018 (State of Agile Report, 2018)

According to Abrahamsson, Salo, Ronkainen, & Warsta (2017), most of the agile practices focus on customer involvement and developers empowerment. Agile practices also encourage changes and improve the responses to those changes. Therefore, adopting agile requires a culture change – a new method of communication and coordination. The traditional rigor control mechanisms within the organisations, such as the command/control management approach, should also be changed.

Turk, France, & Rumpe (2005) pointed out that many of the agile principles, ideas and approaches are built on assumptions which are probably not suitable for all software development projects. In such a case, waterfall, spiral and other SDLCs could be used. When an organisation finds out that some of the agile principles will not fit its case, these principles should be partially adapted, or simply ignored altogether. The worst scenario for an organisation is to try to enforce all of the agile principles. The organisation must be sure of those assumptions before going with agility. If this is not done, agile development will probably provide less value for the organisation.

Pikkarainen et al. (2012) studied the case of successful agile software development projects in three software companies in Finland. The study revealed the importance of management support and belief in agile. The study also found that organisations adopting agile methods need to empower the development team and tailor the agile methods and practices to fit the organisational needs.

2.3 Agile Development Practices

There are several agile development practices available today. These agile practices evolved and merged overtime. Software engineers sometimes tend to follow one practice or to combine more than one practice following their needs. The current most common agile practices are: Scrum, Extreme Programming (XP), Feature Driven Development, Crystal, and Dynamic Systems Development Method (DSDM). Each practice focus on specific values, but they share common principles (Campanelli and Parreiras, 2015). Based on Begel & Nagappan (2007), Scrum is the most common method used in practice, with 65% of respondents saying that they use Scrum. Moreover, the latest 2018 state of agile report from VersionOne indicate that Scrum is dominating the usage of agile practices with 56% (State of Agile Report, 2018). The most common agile practices and methods are discussed in the following sections.

2.3.1 Scrum

Scrum is an approach to implementing and managing any type of project. However, Scrum has always been associated with agile development in the software industry. Scrum should overcome the issue of unclear requirements with its iterative approach. Scrum will also improve team communication and users will have frequent feedback while the project progresses (Rising and Janoff, 2000). The Scrum life cycle consists of iterations (sprints), and is shown in Figure 2-2. Key Scrum principles are discussed below (Sutherland and Schwaber, 2017):

a) Product Backlog: a list of all the system's features prioritised based on their importance.

b) Sprints: the project consists of many sprints or phases, and each sprint has a fixed duration and Sprint Backlog.

c) Daily Scrum: a daily meeting to track the progress and address the issues; it is usually led by the Scrum Master.

d) Product Owner: the person who is representing the customers and the key stakeholders in the development team. He/she will be in charge of managing and prioritising the product backlog.

Scrum was always linked with small development team, but it could work for large teams as well. Scrum might be implemented with a combination of other available methods. The challenges of following Scrum is that it introduces new predefined roles and responsibilities for the development team such as product owner and Scrum master. Implementing these new roles might be a challenge for the team (Alqudah and Razali, 2017).

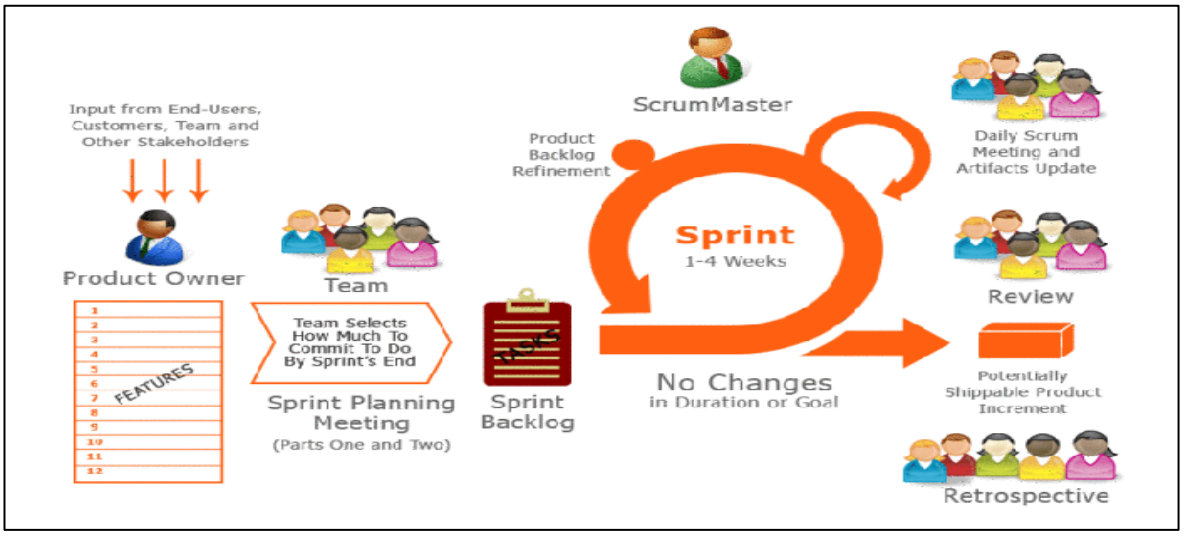


Figure 2-2 Scrum Life Cycle, (Source: http://www.agiletroop.com/product/life-cycle-of-scrum/)

According to Sutherland and Schwaber (2017), Scrum has three categories or groups of phases: Pregame, Game, and Postgame.

- Pregame group: this group essentially contains the planning and the architecture phases. During the planning phase, the first task is to create the backlog. The backlog is a list of features that need to be implemented during the project. An early estimation of time and cost is expected to be made at the end of the planning phase. The architecture phase pertains to a high-level design on how the backlog will be implemented.

- Game group: this consists of one phase, usually called the development phase or sprints phase. It is an iterative phase. Each iteration (sprint) duration should be constant. The outcome of each sprint is a new functional release which will be used to evolve the development of the system.

- Postgame phase (closure phase): the product is preparing for a release. The closure phase may include activities such as final testing, documentation, and training.

2.3.2 Extreme Programming (XP)

The extreme programming concept has developed from the issues associated with the rigorous development cycle of the traditional software development methodologies. XP developers should respond to the frequently changing requirement with more courage (Beck, 1999). Highsmith stated that the key values of extreme programming are: quality work, simplicity, courage, feedback and communication (Highsmith, 2002). According to Beck (2000), the life cycle of XP is divided into the following phases:

- Exploration phase: this phase includes initial requirement gathering and initiating the development of the user stories. An initial architectural modelling is supposed to be carried out during the exploration phase.
- Planning phase: during the planning phase, both the business users and the development team will start to estimate cost and time constraints for the development. Modelling architecture for the final product will be discussed during this phase.
- Iteration to Release phase: Beck considered this phase to be the primary effort of XP development (Beck, 2000). During the phase, many iterations will be completed before the final release. Each release will include modelling, implementation, testing activities and, finally, delivering the release.
- Production phase: this is a phase during which the project team must check if they are ready to go to production. Final system testing is conducted during this phase. The documentation process is also a big part of the phase.
- Maintenance phase: Beck (2000) stated that the maintenance phase is the normal state of any XP project, since XP development continues to evolve over time. Any issue encountered after moving to production, and even if a new feature needs to be implemented, will be tackled during the maintenance phase.

2.3.3 Feature Driven Development (FDD)

The main idea of this practice is to divide the development work into a list of features. The first step of FDD is to develop an overall model and then to build the features list. Following this, iterative work will be carried out to implement each feature. The iteration of a feature is divided into two phases, namely the design phase and implementation phase. There are many who argue that

feature-driven development is appropriate for the development of highly critical systems (Palmer and Felsing, 2001).

2.3.4 Crystal Methods

This is a family of methods, and each development project has to select one or more combined Crystal methods which fit the project needs. The most common method is the Crystal Clear method, which places emphasis on building an effective and efficient method of communication within the team. According to Cockburn (2004), crystal methods are appropriate for small teams and less critical systems. However, it is possible to combine Crystal methods with heavier methods such as XP or Scrum, for large or more complex systems.

2.3.5 Dynamic System Development Method (DSDM)

The origins of DSDM can be traced to the rapid application development (RAD) methodology. While many other software development methods involve adjusting time and resources to implement a fixed functionality of a system, the main idea of DSDM is to adjust the functionality of the system based on the available resources and time (Abrahamsson et al., 2003). DSDM embrace the following principles: focusing on the customer needs, user involvement, frequent releases rather than quality releases, and the testing and quality assurance aspects to be achieved by the iterative development (Anand and Dinakaran, 2016).

2.4 Critical Success Factors (CSF) of Agile Development

In this section, the success of software development projects will be defined. Following this, the CSFs of agile development will be identified from the literature.

2.4.1 Defining the Success of Software Development Projects

The last few decades have produced a great deal of research focused on identifying the factors that influence the success of software development. According to Leidecker & Bruno (1984), the critical success factors are those factors, conditions, variables and attributes that, when properly addressed, managed and sustained, have a huge impact on the success of the work. DeLone and McLean (1992) stated that a project is considered successful if it meets the traditional success measures of cost (delivering within estimated cost and effort), time (delivering on time) and scope (delivering the requirements). However, Bytheway (1999) claimed that unless a software project addresses organisational needs, it cannot be deemed successful. Shenhar, Dvir, Levy, & Maltz (2001) introduced a multidimensional concept of project success. The four measures of project

success are: project efficiency (meeting time, cost and scope constraints), customer impact or satisfaction, business or organisational success, and future preparation.

One of the common approaches to the project success is to consider the project as a successful project if it has met the time and budget constraints. However, this may not be valid for all the cases. There were projects with a delay in the schedule and with overrun budget, but it ended up with great business success. Thus, time and budget are not enough to measure the project success. Moreover, the stakeholders or the customers' satisfaction should be one of the measures to the project success (Shenhar et al., 2001).

Different perspectives can be held by different stakeholders and customers in the agile development project. What may satisfy one stakeholder does not necessary satisfy the remaining stakeholders involved in the agile project. This is where the value-obtained approach becomes vital (Kitchenham & Pfleeger, 1996). There should be always a value-driven approach when chasing for the satisfaction from the stakeholders.

According to Bytheway (1999), the software development project cannot considered to be a successful project if it does not address the organisational needs. The agile development of software within an organisation should always be aligned with the business strategy of the organisation and the organisational needs.

According to Patton (2008), the business value is something which benefits the organisation investing in the software by increasing the revenue, cutting costs, and improving the service. Rawsthrone (2004) reviewed the business value in agile projects and how it could be measured. He concluded that business value in agile projects is what the organisation is willing to pay for. Moreover, the business value should only be defined by the organisation and should be gauged using a specific measure (product, service, revenue, or cost) which addresses the organisation's needs. Alahyari, Svensson, and Gorschek (2017) reviewed how the value could be defined in agile software development projects and which value aspects considered most important in agile development context. They summarised that delivery time and perceived quality are the most important aspects of the value added in agile software development projects.

2.4.2 Critical Success Factors (CSFs) of Agile Development

According to Kitchenham (2004), one of the reasons for undertaking a systematic review is to summarize the existing evidence about a technology or a phenomenon. In this research, the reason for undertake a review study is to summarize the empirical evidence of the critical success factors of agile software development during the last ten years. The literature reveals number of studies

which have examined the success factors of implementing agile practices in an organisation. A review to search for the studies of agile CSFs in the last ten years (2006–2016) which used empirical methodologies to identify the success factors of agile development was conducted. So the inclusion criteria to include the studies in the review process are a) used an empirical methodology to identify the CSFs of agile development, b) dated from 2006 to 2016, c) did not focused on a specific domain (industry).

Following the inclusion criteria these studies (Chow & Cao, 2008; Stelzmann, Kreiner, Spork, Messnarz, & Koenig, 2010; Misra, Kumar, & Kumar, 2009; Sheffield & Lemétayer, 2013; Wan & Wang, 2010; Stankovic, Nikolic, Djordjevic, & Cao, 2013; Livermore, 2007; Kelle, Visser, Plaat, & Wijst, 2015) have been selected to identify the critical success factors in agile development.

Indeed, a number of success factors in agile development have been identified as a result of these studies. The criterion for selecting the critical success factors in this research is that the success factor being mentioned in at least two different papers of the selected eight papers. The identified critical success factors for agile development and the corresponding literature are listed in Table 2-3.

Table 2-3 CSFs of Agile Software Development

Success Factor	**Literature**
Delivery Strategy	(Chow and Cao, 2008) (Stelzmann et al., 2010)
Team Capability and Training	(Misra et al., 2009) (Sheffield and Lemétayer, 2013) (Chow and Cao, 2008) (Wan and Wang, 2010)
Agile Software Development Techniques / Practices	(Chow and Cao, 2008) (Stelzmann et al., 2010) (Wan and Wang, 2010)
Customer Involvement	(Misra et al., 2009), (Chow and Cao, 2008) (Stelzmann et al., 2010)
Project Management Process / Approach	(Chow and Cao, 2008) (Stelzmann et al., 2010) (Stankovic et al., 2013)
Organisational Culture	(Misra et al., 2009) (Sheffield and Lemétayer, 2013) (Wan and Wang, 2010)
Communication	(Stelzmann et al., 2010) (Kelle et al., 2015)
Top Management Support	(Livermore, 2007) (Kelle et al., 2015)

There are other success factors which we did not include in this research because they were mentioned only in one study. For example, Livermore (2007) indicated that the organisation size is a success factor for agile development. Another example, Stelzmann et al. (2010) claimed that the change response strategy is a success factor for agile development.

A taxonomy of the critical success factors is shown in Figure 2-3. This taxonomy of the success factors is driven from the classical approaches of project management (PMBOK 5t edition) and ISO21500 were the success factors classified into technical, process, and organisational categories. According to Dvir et al. (1998), projects success factors could be classified into four categories as follows:

I. Project initiation and pre-contract activities.

II. Project preparations, design policy, technological infrastructure, design methods.

III. Planning and control processes.

IV. Organisational and management environment.

In light of this, these categories proposed by (Dvir et al., 1998) could be labelled as Process, Technical, and Organisational categories. With categories I and III to be merged as Process. And category II could be labelled as Technical and category IV could be viewed as organisational category.

In the case of agile projects success factors. With the concentration of the agile manifesto values (Beck et al., 2001) on individuals and the people role in agile software development. There is a need to add a new category of the success factors of agile development which is people category. Therefore, this research categorized the eight identified success factors of agile development from the literature into Technical, Process, People, and Organisational categories Figure 2-3. In the coming sections, the identified success factors will be defined.

2.4.2.1 Delivery Strategy

In order to have an effective delivery strategy in agile development projects, two aspects must be assured. Indeed, there must be frequent delivery of working software and delivery of the most important features first (Chow and Cao, 2008). The agile software development should be driven by a prioritised requirements/features backlog. The most important features must be delivered faster and cannot be postponed (Stelzmann et al., 2010). Boehm & Turner (2005) argued that agile principles and practices involve scheduling many delivery releases within short time spans which will deliver value to the customers quickly. The delivery strategy should be clear between the project team, and each member will know his/her role in the strategy. The number of releases

delivered and the number of top prioritised features delivered could well represent a criterion with which to measure the effectiveness of the delivery strategy.

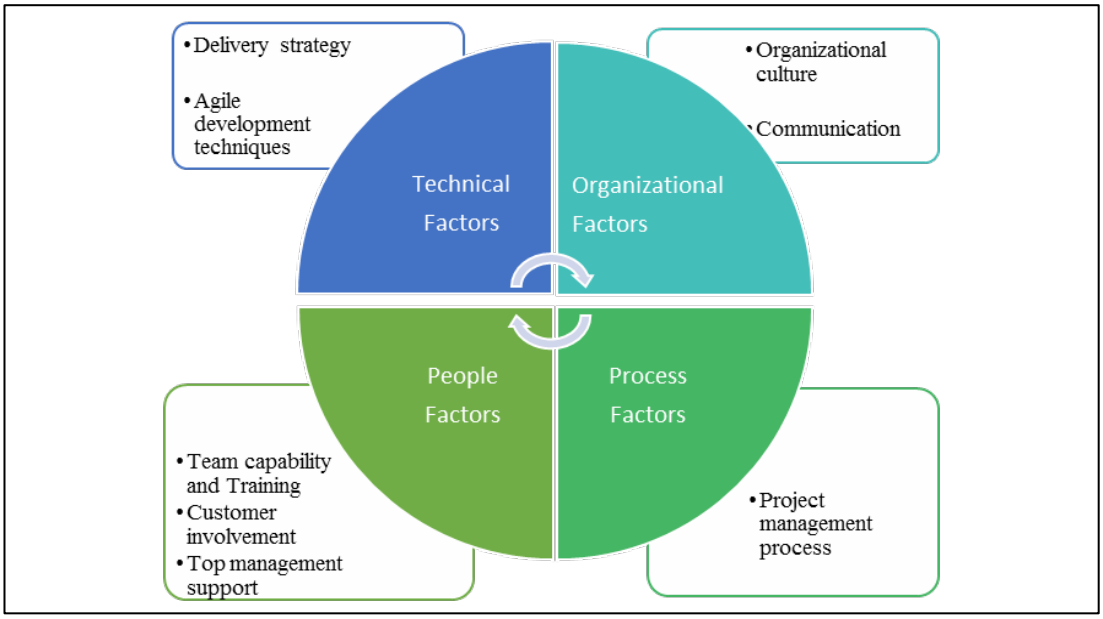


Figure 2-3 Agile Development Success Factors Taxonomy

2.4.2.2 Team Capability and Training

One of the agile values introduced by the agile manifesto (Beck et al., 2001) is the focus on individuals over processes. Agile development should be built on motivated individuals. Having the right people is essential for any project, and in agile projects it is even more important, since the project depends on the individuals' competency. There should be an emphasis on training and continuous learning during the agile development project (Misra et al., 2009). Cockburn & Highsmith (2001) argued that agile development focuses on the talent of the team. As such, the agile processes are designed to capitalise on each individual's strength. Sheffield & Lemétayer (2013) claimed that one of the most important factors for agility success is the empowerment of the project team. The success of agile development should not depend solely on the team's existing capabilities, but should also be linked to elaborating on the development of the team's skills and capabilities. In order to maximise the benefits of the team capability factor, agile development should be built on a talented team. Moreover, all necessary steps should be taken to ensure that the team has the training needed. This will lead to high levels of trust, and will ultimately result in success on the agility journey.

2.4.2.3 Agile Software Development Techniques

Before using an agile method or technique, it is essential that the team is familiar with the selected method or practice (Stelzmann et al., 2010). The organisation's legacy technology could have an

impact on the move to agility. Existing tools and infrastructure play an important role in the success of implementing agile software (Nerur, Mahapatra, & Mangalaraj 2005). Agile software development techniques and tools are built on an assumption that might not fit all projects (Turk, France, & Rumpe, 2005). Thus, the project team may need to combine many practices and techniques for the benefit of the development project. Selecting those practices and techniques is critical for the success of the agile development project.

2.4.2.4 Customer Involvement

Customer involvement is crucial during agile software development. One of the agile principles introduced by the agile manifesto (Beck et al., 2001) clearly states that customers should work closely with the development team. Delivering frequent releases of working software and welcoming the changes in the requirements depend heavily on the involvement of customers. The more involved customers are, the more satisfied they will be with the agile development. Customer collaboration and commitment are believed to have an impact on the successful implementation of agile software (Misra et al., 2009). Customer involvement also includes building an effective method of communication with the customer and the different stakeholders. According to the agile manifesto (Beck et al., 2001), face-to-face communication, if applicable, is considered the best approach of communication in agile development.

2.4.2.5 Project Management Process / Approach

The project management approach must be agreed upon by the project team members and be well defined from the beginning of the project (Stelzmann et al., 2010). The nature of agile projects usually tend to have an uncertainty at the early stages of the project since the requirements are not clear and supposed to be evolving throughout the project. The selection of an appropriate project management process contributes to the success of agile development in term of developing quality software (Chow and Cao, 2008). According to Sheffield & Lemétayer (2013), the project management aspects are considered to be an indicator of the agility in software development. The selected project management process should empower the project team and address the talents of the development team. The nature of agile projects which embrace changes and deliver many iterations requires a flexible project management approach.

2.4.2.6 Organisational Culture

Organisational culture can be defined as a set of factors and variables which may have an impact on the development of agile software. Since agility is a multi-facet concept, it will require the adoption of an agile culture (Iivari and Iivari, 2011). According to Wan & Wang (2010) three aspects of organisational culture may affect agile development, the first of which is the overtime culture

(overtime work is not suggested as per the agile manifesto principles (Beck et al., 2001)). This is followed by the culture of no trust in an organisation, and finally the lack of mutual collaboration culture. Strode, Huff, & Tretiakov (2009) investigated the relationship between organisational culture and the use of agile methods. The most important organisational culture factors were found to be: the organisation must be result-oriented, the leadership of the organisation must comprise a risk-taking, innovative team, and the organisation must be based on trust, loyalty and commitment.

2.4.2.7 Communication

Agile software development emphasizes the shift from traditional communication methods to informal, face-to-face communication. The agile practices improve the organisation's ability to communicate effectively. Indeed, Pikkarainen, Haikara, Salo, Abrahamsson, & Still (2008) conducted a study to explore the impact of agile practices on communication in software development. The study showed that agile methods improve informal and formal communication during the software development project. It also demonstrated that agility helps to improve internal communication between the development team and external communication with the stakeholders of the development project. Stelzmann et al. (2010) claimed that it is crucial to establish direct communication between customers and the development team. Direct communication is efficient because it has higher bandwidth than indirect communication. Agile development in a global context or distributed environment may hinder the communication, since agility places emphasis on face-to-face communication and daily meetings. Alzoubi & Gill (2014) argued that people differences and distance differences are the greatest challenges facing communication in agile global software development. It is suggested that the use of social technologies, such as Skype and Yammer, could enhance communication in agile global or distributed software development.

2.4.2.8 Top management Support

According to Livermore (2007) study, there was a significant relation between the support of top management and the success of agile projects. The management involvement and support is crucial to the success of agile projects. The leadership believe in the value created by the agility is essential to the success in agile development projects (Kelle et al., 2015).

2.5 The Relationship between the Identified CSFs and Agile Principles

The agile manifesto introduced twelve agile principles (Beck et al., 2001). There is a clear link between the identified critical success factors of agile software development and the agile

manifesto principles. The delivery strategy factor is linked to the third and the seventh principles, which are to deliver working software frequently, and working software is the main measure of progress. The team capability factor and top management support factor are linked to the fifth principle which state that agile development should be built on motivated individuals, with the support they need. The agile software development techniques factor is compatible with the sustainable development, simplicity, and technical excellence principles. The customer involvement factor is linked to the first and fourth principles, which are customer satisfaction and business users working closely with the development team. The same is applied to the project management process, with the self-organised team principle. The organisational culture factor is associated with welcoming changes and the effective team behaviour principles. The communication factor is linked to the sixth principle, which argues that face-to-face communication is the best method of communication. Table 2-4 presents the success factors and their links to agile manifesto principles.

Table 2-4 the Relationship between the Identified CSFs of Agile Development and Agile Principles

Success Factors	**Agile Manifesto Principles**
Delivery Strategy	3rd principle: deliver working software frequently. 7th principle: working software is the main measure of progress.
Team Capability and Training	5th principle: Motivated individuals, with the support they need and trust them to get the job done.
Top Management Support	
Agile Software Development Techniques /Practices	8th principle: sustainable development. 10th principle: simplicity. 9th principle: technical excellence.
Customer Involvement	1st principle: customer satisfaction. 4th principle: business users working closely with development team.
Project Management Process / Approach	11th principle: self-organised team.
Organisational Culture/ Environment	2nd principle: embracing changes. 12th principle: team's behaviour to be more effective.
Communication	6th principle: face-to-face communication.

Having reviewed the literature, the researcher identified research gap which will be discussed in the next section.

2.6 Research Gap

Over the last decade, many studies have researched and identified the success factors of agile software development. However, most of these studies did not differentiate the importance of these success factors, and how said factors vary in terms of their impact on the success of agile software development projects. As such, there is a need to study the success factors of each phase or iteration independently. Indeed, there may well be success factors that are more important

during a specific phase than the remaining project phases. The progress rate of an agile project may also increase or decrease the importance of some of the success factors.

There is also a need to study the differences in importance between the CSFs of agile software development. This will help those aiming to achieve successful agile software development projects to prioritise certain factors based on their importance and their contribution to the success of agile projects. To the best of our knowledge, and based on our review study of agile development success factors, there exist no previous studies which have attempted to investigate the differences in importance of agile development critical success factors.

The inter-relationships between the identified success factors of agile software development could be explored, and will be in the present study. Figuring out these inter-relationships will help for a better understanding of the CSFs and how they impact each other. For this purpose, factor analysis is conducted to understand the latent dimensions among the CSFs of agile development. Factor analysis will help in categorising these success factors accordingly.

Kunz, Dumke, & Zenker (2008) study concluded that not all the software traditional lifecycle metrics are suitable for agile software development. It was suggested that future work should focus on how the use of the traditional software measurement could be adapted to work with agile development or to develop new metrics for agile development. A recent study by (Padmini et al., 2015), reviewed a total of 22 software metrics and resulted with only 10 metrics that could be used in agile software development. Therefore, there is a need for more software metrics that could be used to measure the agile software development. Thus, this research will develop an instrument that could measure the success with agile projects in an organisation.

2.7 Research Questions

Having identified the research gap, a set of research questions will be defined with the aim of addressing the identified research gap. Throughout this thesis, the following research questions will be answered:

RQ1. What are the differences in importance between the CSFs and what is the weight for each factor?

RQ2. Does each success factor of agile software development have the same importance during all phases or iterations of the agile project?

RQ3. Using factor analysis, how can the CSFs of agile development be grouped into a smaller number of categories?

RQ4. What is the appropriate instrument with which to measure the success of an agile software development project in an organisation?

A summary of the research questions and the research methods which were employed to answer these questions is presented in Table 2-5.

Table 2-5 Summary of the Research Questions and Research Methods

Question	Method	Purpose	Answer
RQ1. What are the differences in importance between the CSFs and what is the weight for each factor?	An online questionnaire distributed to agile practitioners, and descriptive statistics analysis.	To identify the differences in importance between the agile success factors.	Section 4.2.1
RQ2. Does each success factor of agile software development have the same importance during all phases or iterations of the agile project?	An online questionnaire distributed to agile practitioners, and regression analysis.	To investigate the relationship between the agile projects progress and the importance of each success factor.	Section 4.3
RQ3. Using factor analysis, how can the CSFs of agile development be grouped into a smaller number of categories?	An online questionnaire distributed to agile practitioners, and factor analysis.	To explore the inter-relations between the agile success factors.	Section 4.4
RQ4. What is the appropriate instrument with which to measure the success of an agile software development project in an organisation?	Development of an Instrument.	To suggest measurements and metrics with which the success of agile software development projects could be measured.	Section 6.1

2.8 Post-study Review

Having completed this study, the researcher went back to review the literature of the CSFs for agile software development. This was aiming to check for the new studies that were published after 2016 to make sure that the identified CSFs is still relevant and to mention any new success factor if exist. Chiyangwa and Mnkandla (2017) have recently studied the CSFs of agile software development projects in South Africa. In their study, no new success factor has been identified. However, they classified the success factors into five categories: organisational, process, people, technological, and project. Their findings showed that the organisational culture is the dominant factor that indicates whether or not the individuals are practising agile successfully.

Kulathunga and Ratiyala (2018) reviewed the success factors of scrum method using a survey with 241 sample size. They identified a set of success factors and categorised them into five categories: organisational, people, process, technical, and project. No new success factors have been identified. Dhir, Kumar, and Singh (2019) reviewed the success and failure factors of the agile software development projects using a case study methodology. With regard to the success factors, the same eight success factors identified in this research have been identified. A new success factor has been identified which is documentation and how an accurate documentation and avoiding extra efforts contribute to the success of agile software development projects. However, it might be argued that the documentation process could be seen as a part of the agile techniques success factor. Shameem et.al (2017) did a systematic review of the critical success factors of agile development in global software development environment. The same identified eight success factors are still relevant and two new factors were introduced which are encouraging for project visibility and small team size.

2.9 Summary

In this chapter, the literature of the agile development and its methods and practices is reviewed. The critical success factors of agile software development are reviewed in this chapter. As a result of a systematic review, eight previous studies have been selected to identify the critical success factors of agile development. The selected studies identified many success factors for agile software development. Of which eight factors have been selected in this study because they were identified by more than one study. The selected success factors are: delivery strategy, team capability and training, agile software development techniques / practices, customer involvement, project management process / approach, organisational culture, communication, and top management support. The selected factors have been classified into a taxonomy of factors which include Technical, Organisational, Process and People categories. The relationships between the identified success factors and the agile manifesto (Beck et al., 2001) principles are explored in this chapter. Following that, the research gap and the research questions were discussed. The identified research questions will be addressed in the following chapters. Having completed the study, the researcher reviewed the literature again to search for new studies investigating the CSFs of agile software development that are published recently.

Chapter 3 Research Methodology

In this chapter, the research design of the present research is presented. This chapter discusses the research methodologies which will be used throughout this study. The chapter is separated into five sections. Section 3.1 illustrates the research design. Section 3.2 presents a general discussion of the quantitative and qualitative research methods. In section 3.3, the research methods which have been used in this study are discussed. The hypotheses were defined and the used data analysis techniques were briefed. Section 3.4 contains information pertaining to the ethical considerations. Following that, section 4.5 is a summary of the chapter.

3.1 Research Design

In order to answer the research questions which have been discussed in the previous chapter, a programme of research has been designed. A simple representation of the research design which has been followed throughout this study is illustrated in Figure 3-1. This figure gives an overview of the applied research methodologies.

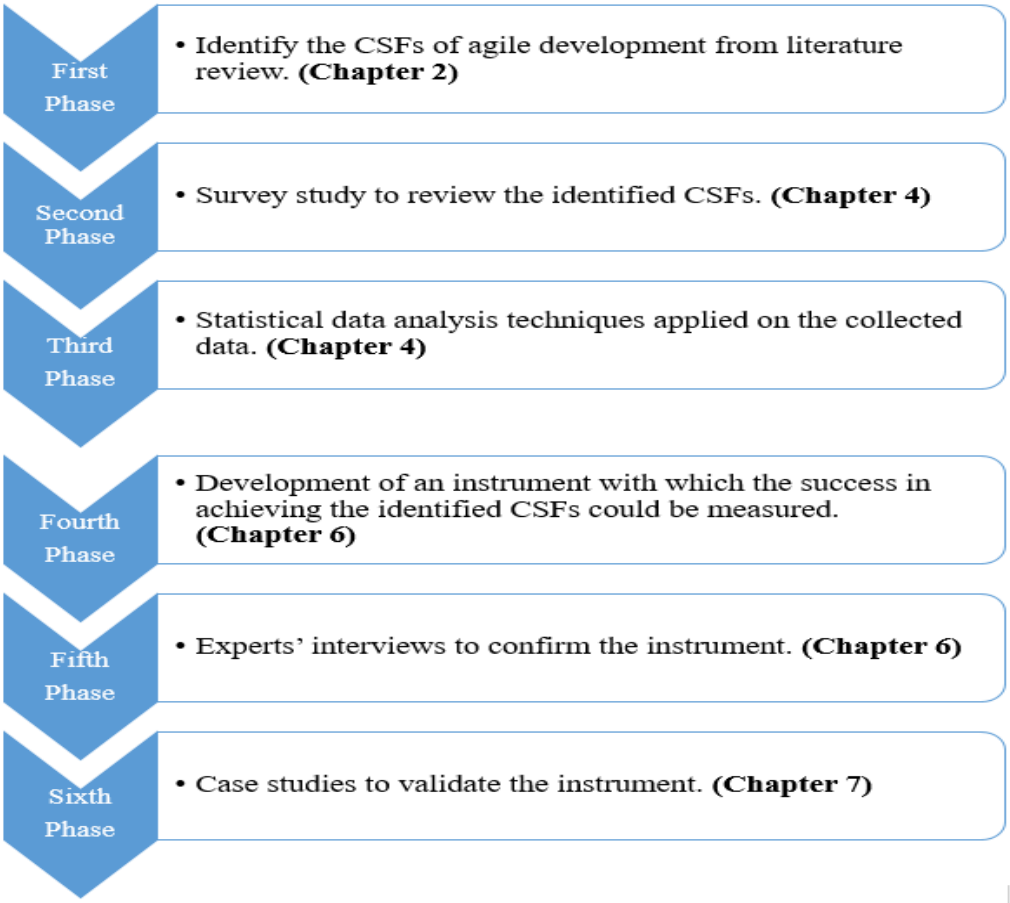


Figure 3-1 Research Design

3.2 Research Methods

According to Dybå, Kampenes, & Sjøberg (2006), software engineering empirical research is usually associated with weak statistical design and falls below the accepted norms. Given this revelation, it is clear that software engineering empirical studies should pay more attention to enhancing their statistical design.

A recent study by Jørgensen, Dybå, Liestøl, & Sjøberg (2015), which investigated software engineering experiment studies, indicated that bias in software engineering is quite common. To avoid such bias, they introduced a set of recommendations for software engineering researchers. These recommendations include avoiding low statistical power, avoiding complex design, accepting non-sophisticated results, and making the raw data available. Software engineering research tends to be biased, since many software engineers are in favour of certain tools or techniques. Some blind analysis techniques may well be used in the present study, if it is necessary to do so.

Kitchenham et al. (2002) introduced guidelines for empirical research in software engineering, thus indicating that it is important for the data collection process to report any measures of validity or quality control used on the collected data. They also found that response rate is an important factor which should be monitored and reported throughout the study. In the following two sections, a general background about the quantitative and qualitative research methods is given.

3.2.1 Quantitative Research Method

Quantitative research uses mathematical or numerical measurements to study a specific topic or issue, and the collected data are analysed using statistical techniques (Neuman, 2012). One way of collecting quantitative data can be through a set of questions in a questionnaire or survey. According to Pfleeger & Kitchenham (2001), a survey is “a comprehensive system for collecting information to describe, compare or explain knowledge, attitudes, and behaviour”. Thus, a survey is an instrumental process, which starts with careful planning and designing, through to analysing the data and reporting the results.

3.2.2 Qualitative Research Method

Qualitative research is mainly exploratory in nature, and aims to discover the targeted sample’s perceptions of a specific topic. It helps the researchers to understand the causes of a phenomenon by giving insights into said causes (Patton, 1990). Qualitative data can be collected through open-ended questions, interviews, or case study.

3.2.3 Mixed Methods

Mixed-method uses both quantitative and qualitative methods to provide the researcher with more choices to consider (Creswell and Clark, 2011). Usually, the qualitative method used for a deeper understanding of the results obtained by the quantitative method (Creswell, 2003). It is believed that by mixing the quantitative and qualitative methods, the reliability of the results could be enhanced (Teddlie and Tashakkori, 2010).

3.3 Research Methods Used in this Research

This research starts with a literature review aiming to identify the CSFs of agile software development. Following which, these CSFs were investigated to establish a better understanding of how they are related to each other and how they varies in terms of contributing to the success of agile software development. Moreover, this research aims to develop an instrument to measure the success of agile software development. To do so, a mix of quantitative and qualitative research methods will be used. The mixed-method approach has been chosen in order to have more variety of methods, tools with which the collected data could be analysed and the research questions could eventually be addressed.

In order to evaluate and capture the differences in the importance of the success factors of agile software development, quantitative methods were used. Indeed, we employed a web-based survey comprising a Likert scale questionnaire; this was also accompanied by information regarding the agile project, and the individuals/organisations' experience with agile development. The target population was agile software practitioners and users around the world. The survey was sent to agile software experts using groups and contacts on LinkedIn (the professional social media platform), while the Agile Alliance network was also used.

In order to develop an instrument with which the success of agile software development could be measured, qualitative methods were used. This study develops an instrument to measure the success of achieving the CSFs in an agile software development project. Following this, the developed instrument was reviewed by agile experts. The criterion to select those experts is that they had at least five years' experience with agile development. The followings sections give an overview of the used research methods and statistical tests in this research.

3.3.1 Survey Design

This research employed the same survey instrument as that used in Chow & Cao (2008) and Stankovic, Nikolic, Djordjevic, & Cao (2013). Additional questions were included in the survey to

capture the individuals and organisations' experience with agile development, the status and progress of the project, and the number of completed iterations in the agile projects; the purpose of this was to investigate how the progress of agile projects is related to the importance of the success factors. More details about the survey design will be discussed in section 4.1.

3.3.2 Hypotheses Defining

The survey was designed to collect data with which the following hypotheses could be tested. These hypotheses were defined to address the research questions RQ1 and RQ2. The results of testing these hypotheses will be reported in sections 4.3.2 and 4.4.

H_1. The importance of the CSFs of Agile software development varies during the agile development project depending on the number of completed iterations in the project (X):

H_1a. The importance of delivery strategy factor varies depending on X.

H_1b. The importance of team capability and training factor varies depending on X.

H_1c. The importance of agile software development techniques factor varies depending on X.

H_1d. The importance of customer involvement factor varies depending on X.

H_1e. The importance of project management process/approach factor varies depending on X.

H_1f. The importance of organisational culture factor varies depending on X.

H_1g. The importance of communication factor varies depending on X.

H_1h. The importance of top management support factor varies depending on X.

H_2. Agile software development projects are successful in terms of (a) Time, (b) Cost, (c) Scope, (d) Organisational needs and (e) Stakeholders' satisfaction.

3.3.3 Instrument Development

The proposed instrument was developed following the Goal-Question-Metric (GQM) approach. The development of the instrument is detailed in section 6.1.

3.3.4 Expert Interview

This study used an open-ended interviews methodology with agile experts to review the proposed instrument. The design of the experts' interviews and its results are available in section 6.2.

3.3.5 Case Study

In this study, the case study research methodology was employed aiming to validate the instrument. The case study design and results are discussed in Chapter 7.

3.3.6 Quantitative Data Analysis

In this section, the quantitative data analysis techniques which were used throughout this study will be briefed.

3.3.6.1 One-sample T-test

The statistical one-sample t-test was used to test the mean values of the distribution. The selected test value was 4, as this number is in the middle of the 7-point Likert scale used in the survey. The Likert scale ranged from 1 (strongly disagree) to 7 (strongly agree), with 4 being neutral. More details will be followed in section 4.3.

3.3.6.2 Factor Analysis

Factor Analysis is a statistical technique commonly used for data reduction and to discover the latent dimensions among a set of data. In other words, it is employed to classify multiple items into a smaller number of factors (Field, 2009). Factor analysis was used in this research in order to explain the inter-relations among the critical success factors of agile development, and to establish the underlying dimensions of these factors. The factor analysis is discussed in section 4.5.

3.3.6.3 The Confidence Intervals

Confidence intervals (CI) are very useful measurements, and thus it is recommended that they should be used and reported explicitly throughout the data analysis stage to enhance the interpretations of the results (Thompson, 2002). Determining how much confidence should be placed in a sample and the emerging results is a controversial decision. This depends on many aspects, including the nature of the study. For example, medical research tends to use high confidence levels, such as 95% or 99% confidence level, while social research might be more tolerant, with low confidence levels, such as 90%, 88%, and 85%. Nakagawa & Cuthill (2007) emphasised that reporting the effect size along with the confidence intervals is crucial and will eventually reduce the misinterpretations of the presented results.

In this research, we used a 95% confidence interval, with a 5% margin of error. The confidence intervals were reported throughout the research in order to allow for more accurate interpretation of the sample and the results. More information will be given in section 4.4.

3.3.6.4 The Significance Level

The significance level for a tested hypothesis indicates whether the probability of a finding (P-value) is less than or equal to a certain value. The significance level was used to determine whether or not the results are statistically significant. The most common values are 0.05 and 0.01, so if ($p < 0.05$, 0.01), then this result is considered statistically significant. In other words, if the p-value is less than or equal to the selected significance level, then the null hypothesis (H0) will be rejected (Rice, 1989). The significance level is usually denoted as α or alpha. Under certain circumstances, some researchers tend to use a large α, such as 0.1, to be sure that they detect any differences that occur. Conversely, others might use a small α, such as 0.01, to be sure that they detect only the differences that really exist.

The guidelines for empirical research in software engineering (Kitchenham et al., 2002) suggested using the customary significance level of 0.05. At the same time, the guidelines indicated that, in the case of multiple tests, the overall significance level of 0.05 should be adjusted for each independent test. For instance, if 10 tests require an overall significance level of 0.05, then the significance level for each test should be adjusted to 0.005 in order to ensure an overall significance level of 0.05 for the 10 tests. An alternative to adjusting the significance level is to not only report the positive results, but also the negative results, as well as the number of conducted tests (Kitchenham et al., 2002).

In this research, we chose a significance level of 0.05. This is considered a moderate significance level selection, and is compatible with the guidelines for software engineering research (Kitchenham et al., 2002). The significance level was determined prior to commencing the data analysis. The selection of the significance level will be detailed in section 4.3.1.

3.3.6.5 Missing Values and how to Deal with them

There are traditional ways of dealing with missing values (list-wise deletion, pair-wise deletion). Indeed, Acock (2005) reviewed the strategies related to working with missing values and suggested a set of recommendations. One of the presented recommendations was that the researcher should explain how and why cases are deleted from the analysis, and should also report the percentage of deleted cases. In this research, before starting any statistical testing, the strategy for dealing with missing values was determined, and information about the deleted cases was reported. Moreover, methods for dealing with extreme values (EV) were formulated before conducting any statistical tests, so as to ensure a better understanding of the data. The missing data will be list-wise deleted in this research. More information about the dealing with missing values is discussed in section 4.4.

3.3.7 Qualitative Data Analysis

This study collected qualitative data by employing experts' interviews and case studies methodologies. Qualitative data are usually a source of rich explanations and descriptions of a context whose the researcher seeking to explore. A precise qualitative data are more likely to lead to new outcomes and to help researchers to get beyond initial notions if rigors analysis techniques were applied (Miles & Huberman, 1994). The qualitative data are usually collected to understand a phenomenon or to seek for more insights regarding an investigated issue. Several qualitative data analysis techniques are available. One of which is the thematic analysis. According to Braun & Clarke (2006), thematic analysis helps in identifying and reporting themes within a set of data. These themes reveal patterns within the data and help understand the investigated phenomenon. As the expert's interviews and the case studies' interviews revolve around the proposed instrument. Thus, the themes will be the instrument's items. If the volume of the qualitative data is considerably large, Nvivo software might be used to facilitate the analysis of the qualitative data. More details about the analysing of the qualitative data are followed in the coming chapters. While the qualitative data analysis of the interviews is discussed in chapter 6. The qualitative data analysis of the case studies is discussed in chapter 7.

3.4 Ethical Approval

The survey used during this research was approved by the Ethical Committee of the School of Electronic and Computer Science at the University of Southampton. This ethical approval was assigned the reference number ERGO/FPSE/24345. More details are provided in Appendix B.

With regard to the experts' interviews and the case studies, the ethical approval was obtained from the same ethical committee at the University of Southampton. The ethical approval was given with the reference number ERGO/FPSE/31342. For more details, see Appendix D.

3.5 Summary

This chapter has discussed the research design which will be followed in this study. Moreover, a general discussion about the research methodologies was presented. Following which, this chapter explained the used research methods in the present study and forwarded justifications for selecting these research methods in this study. These methods were used aiming to address the research questions and motivations. In this chapter, the selected research methods were briefed and information on where they are detailed in this thesis was provided. The used statistical

techniques were introduced in this chapter and further details will be followed in the coming chapters. The ethical considerations for using these research methods was provided in this chapter.

Chapter 4 Results and Findings

In this chapter, the results and the findings obtained from the survey are explored. The chapter is organised as follows. Section 4.1 gives an overview about the design of the survey and the data collection process. Following this, section 4.2 presents the results of the survey. In section 4.3, the descriptive statistics analysis of the questionnaire data is reported. Section 4.4 then puts forth the regression analysis results, which compare agile project progress with the importance of the CSFs. Section 4.5 discusses the factor analysis findings discovered in this research. Section 4.6 presents the qualitative comments that were received in the survey. Lastly, section 4.7 summaries the chapter. As the volume of results is considerable, this chapter presents the results and findings while the discussion will be followed in chapter 5.

4.1 Survey Design and Data Collection

In this section, the survey design and the data collection process will be discussed.

4.1.1 Survey Design

A Likert scale questionnaire was used to collect information about how the views of the study participants differ regarding specific questions. According to Matell & Jacoby (1971), there is no optimal number of Likert scale items. Such a decision depends on the types of questions being investigated in the study. In this research, a 7-point Likert scale questionnaire was used to reflect how well the respondents comprehended the questions. The scale ranged from 1 (strongly disagree) to 7 (strongly agree) with 4 as (neutral).

The questionnaire was designed to capture the status of the agile projects, i.e. a completed project or an on-going project. The questionnaire was also formulated to measure the progress of the projects, identify how many iterations or phases have been completed, and establish how many iterations are remaining. The questionnaire mainly consisted of three parts: The first part comprised demographic information about the participants, the agile project, and the organisation. The purpose of collecting this demographic information was to capture the status of the organisations, and their experience with agile development. Certain questions addressed the size of the project and the location of the project, while other questions related to the individuals' role in the project and their experience with agile development.

The second part asked about the eight identified critical success factors of agile development; the aim of this was to weight the importance for each factor using a 7-point Likert scale. The third part

was related to the success measures and how agile development contributes to the success of a project in terms of time, cost, scope, organisational needs, and stakeholders' satisfaction; again, a 7-point Likert scale was used for this purpose.

At the end of the survey, participants were presented with a section for additional comments, where they were asked to provide their feedback and declare if they felt anything was missing.

4.1.2 Data Collection Process

The survey period lasted 11 weeks, so as the maximum number of responses could be gathered. Agile software development practitioners are the population of this survey. Google Forms platform was used to build the survey, the full version of which is available in Appendix A. Many reminders were sent to agile practitioners on LinkedIn and were posted on agile users groups on LinkedIn, so as to allow for more responses. The Agile Alliance network collaborated with the researcher in distributing the survey to the network members. The survey received 131 responses. These responses came from more than 28 countries around the world.

The collected data was stored on the researcher's desktop in a password-protected directory to guarantee its security. The survey data was also saved in a Microsoft excel file, in order to prepare it for the analysis. For data analysis, IBM SPSS version 24 was used. SPSS version 24 for windows 64 was downloaded, and all the necessary setups were completed on the researcher's machine.

Certain steps of data preparation took place before SPSS was used. These steps included changing the format of some of the columns and changing the names of the variables into more meaningful names. For example, the Likert scale questionnaire was decoded into integer numbers from 1 to 7, instead of strongly disagree to strongly agree. It was deemed that further data preparation steps could be needed during the data analysis stage.

4.2 Results of the Survey

The survey received 131 responses. The responses came from agile practitioners in more than 28 countries around the world. The countries of the respondents are listed in Table 4-1. In total, 32% of the responses pertained to completed agile projects, while 68% belong to on-going agile projects. With regard to the sizes of the projects, 52.67% were small projects with between 1 to 15 project team members. Large projects with more than 30 project team members accounted for 33.58% of the participating projects in the survey. The sizes of the agile projects are listed in Table 4-2.

Table 4-1 List of the Countries

Country	Number of responses	Percentage
Multiple Locations	43	32.82%
United States	25	19.08%
Saudi Arabia	14	10.68%
United Kingdom	9	6.87%
Other Countries	40	30.53%
Total	131	100%

Table 4-2 Size of Agile Projects

Project Size	Number of responses	Percentage
1-15 team members	69	52.67%
16-30 team members	18	13.74%
More than 30 team members	44	33.58%
Total	131	100%

In terms of the duration and length of the agile projects, 22.90% of the projects were short projects, whose duration ranged from 1 to 6 months. On the contrary, long projects with a duration of more than 18 months accounted for 29.77% of the projects reported in the survey. The project durations of the agile projects in the survey are reported in Table 4-3.

Table 4-3 Agile Projects Duration

Project Duration	Number of responses	Percentage
1-6 months	30	22.90%
7-12 months	40	30.53%
12-18 months	22	16.79%
More than 18 months	39	29.77%
Total	131	100%

In terms of the organisation's experience with agile development, the participating organisations reported the following level of experience with agile projects: 54.19% with 0-3 years, 29% with 3-6 years and 16.79% with more than 6 years' agile experience. On the other hand, the participants' experience with agile development were reported as follows: 38.93% with more than 6 years'

experience, 32.06% with 0-3 years, and 29% with between 4 and 6 years' experience. The levels of the participants and organisations' experience with agile development are listed in Table 4-4.

Table 4-4 Level of Experience with Agile Development

Years of experience	**Participant's experience** (%)	**Organisational experience** (%)
0-3 years	42 (32.06%)	71 (54.19%)
4-6 years	38 (29.00%)	38 (29.00%)
More than 6 years	51 (38.93%)	22 (16.79%)
Total	131 (100%)	131 (100%)

The roles of the survey participants in the agile projects varied from project managers, to customers or stakeholders. This showed the variety of the participants' backgrounds and roles concerning the agile software development projects. The different roles of the participants in the agile projects are shown in Table 4-5.

Table 4-5 Roles of Participants in the Agile Project

Role (Job Responsibility)	**Number of responses**	**Percentage**
Project manager (or scrum master)	62	47.32%
Organisation management	18	13.74%
Team leader	15	11.45%
Customer	2	1.52%
Team member	15	11.45%
Stakeholder	2	1.52%
Other	17	12.97%
Total	131	100%

4.3 Survey Data Analysis

In this section, the different quantitative data analysis techniques and statistical tests used in this study are reported, along with the results and findings which they generated.

4.3.1 The Critical Success Factors

The first research question (RQ1) to be addressed in this study asked: "What are the differences in importance between the CSFs and what is the weight for each factor?" In the questionnaire part of the survey (see Appendix A for the full version of the survey), the participants were asked to

evaluate the importance of each factor to the success of the agile project; the scale used ranged from strongly disagree (1) to strongly agree (7).

In order to answer the first research question (RQ1), the mean and the standard deviation were calculated for each success factor, based on the participants' responses to the success factor questions. The success factors of agile software development were ordered by importance, and the weight for each factor was reported. The ranking of the agile development success factors and a summary of the descriptive statistics for each success factor are presented in Table 4-6.

A one-sample two-tailed t-test was run to determine whether the means of the importance of CSFs in the sample were different to the mid-point of the sample. The sample was compared against the test value of 4, which stood in between 1 (strongly disagree) and 7 (strongly agree). A two-tailed t-test was performed and all the p-values were found to be statistically significant ($p < 0.05$).

Table 4-6 The Ranking of Agile Software Development Success Factors

Success Factors of Agile Development	**N**	**Mean**	**Std. deviation**	**Sig (2 -tailed)**
1. Communication	**131**	**6.28**	**1.21**	**< 0.001**
2. Customer involvement	**131**	**6.18**	**1.28**	**< 0.001**
3. Team capability and training	**131**	**6.07**	**1.31**	**<0 .001**
4. Top management support	**131**	**5.87**	**1.30**	**< 0.001**
5. Organisational culture	**131**	**5.81**	**1.44**	**<0 .001**
6. Agile software development techniques and practices	**131**	**5.57**	**1.44**	**<0 .001**
7. Delivery strategy	**131**	**5.43**	**1.56**	**< 0.001**
8. Project management process	**131**	**5.04**	**1.66**	**<0 .001**

Some could argue that because the data are ordinal and most likely will not be normally distributed, t-tests and other parametric tests like the regression cannot be used. Although there are tests to check the normality of the data such as K-S test and Shapiro-Wilk test, this step was not taken by the researcher in this stage. The using of parametric tests with ordinal data which came from a Likert Scale received criticisms even though much research collects data using rating scales. While many might question the using of t-tests and the parametric tests because the data are not normally distributed. Norman (2010) claimed that for t-tests, the assumptions of normality is of the distribution of means, not the data. Moreover, the Central Limit Theorem indicates that the means are approximately normally distributed for sample sizes greater than 5 or 10 per group regardless of the data distribution. Other scholars like Field (2009) claimed that for large sample size the

normality of the data should not be a concern since the distribution of the means will be approximately normal. In addition, there have been many studies which reassured that the correlations are robust with respect to non-normality such as Pearson (1931). According to Carifio and Perla (2008), the parametric tests could be used on ordinal data because of the robust nature of the data. And that the ratings generated from Likert Scale are appropriate to summarise using means and standard deviations and to use the parametric tests to analyse the data. There has been always a debate on using the parametric tests on data generated from Likert Scale. Nevertheless, the Likert Scale rating is very helpful and many researchers choose to use them. In this research, some parametric tests will be applied on the data following the justifications presented in the abovementioned studies: Norman (2010); Field (2009); Carifio and Perla (2008).

By calculating the mean and standard deviation and reviewing the results using a t-test, this gives us a value of p and it tells us that investigating of a significant level of 95% if $p < 0.05$ there is a significant difference. Nevertheless, it fails to determine the degree of the deference. Cohen (1992) introduced the idea of effect size (d) sometimes called (Cohen's d). Which aims to standardize the difference between the means and compare it to a pooled standard deviation. Cohen (1992) offered an interpretation of the value of the effect size as follows: 0.80 Large, 0.50 Medium, 0.20 small.

For example, the mean of communication factor was statistically different with low effect size comparing to the mean of customer involvement, team capability and training factors. The effect size was large with a value of $d= 0.85$ when comparing the mean of communication factor to the mean of project management factor. The same technique could be applied to all the eight factors if the effect size of the differences is to be calculated using Cohen's d. However, the research question RQ1 was not looking to calculate the differences in the importance. Indeed, RQ1 was aiming to investigate the differences and to report the order of the CSFs after applying the statistical tests.

A paired t-test was run to determine whether there was a statistically significant mean difference between the identified success factors of agile software development. Using a 95% confidence interval, the difference between the means is statistically significant if the t-test p-value < 0.05. To do so, eight paired t-test will be required to compare the mean of one factor with the means of the remaining factors. The paired t-test of comparing the mean of communication factor with the other success factors' means is shown in Table 4-7. The conducted paired t-test showed that communication's mean is statistically significantly different than the other success factors' means except for the team capability factor and the customer involvement factor where the p-value is >

0.05 as shown in Table 4-7. The paired t-tests for the remaining pairs concerning the other success factors are discussed in Appendix F.

Table 4-7 Paired t-test Results (Communication, Remaining CSFs)

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	Communication - Project management	1.244	1.724	.151	.946	1.542	8.263	130	.000
Pair 2	Communication - Delivery strategy	.855	1.678	.147	.565	1.145	5.830	130	.000
Pair 3	Communication - Agile software development techniques and practices	.710	1.321	.115	.482	.938	6.149	130	.000
Pair 4	Communication - Organisational culture	.473	1.521	.133	.210	.736	3.562	130	.001
Pair 5	Communication - Top management support	.412	1.122	.098	.218	.606	4.204	130	.000
Pair 6	Communication - Team capability and training	.214	1.347	.118	-.019	.447	1.816	130	.072
Pair 7	Communication - Customer involvement	.107	1.349	.118	-.126	.340	.907	130	.366

Using descriptive statistics and t-test, we addressed the first research question RQ1 concerning the differences in importance between the agile software development critical success factors.

4.3.2 The Success Measures

Here, it is important to allude to the previously mentioned hypothesis H2, which postulated that: "H2. Agile software development projects are successful in terms of (a) Time, (b) Cost, (c) Scope, (d) Organisational needs and (e) Stakeholders' satisfaction". To test this hypothesis, in Section 3 of the questionnaire, the participants were asked if the agile project was successful in terms of five measures, namely time, cost, scope, organisational needs, and stakeholders' satisfaction.

Table 4-8 The Success Measures of Agile Software Development Projects

Success Measures of Agile Development	**Mean**	**Std. deviation**	**Sig (2 -tailed)**
1. Achieving Stakeholders' Satisfaction	**6.09**	**1.06**	**< 0.001**
2. Addressing the Organisational Needs	**5.85**	**1.08**	**< 0.001**
3. Scope	**5.31**	**1.31**	**< 0.001**
4. Cost	**4.93**	**1.40**	**< 0.001**
5. Time	**4.91**	**1.51**	**< 0.001**

The mean and standard deviation were calculated for each success measure as per the received responses. A one sample two-tailed t-test was run on the sample to compare the means with the

midpoint of the scale (4) and using a significant level of 95% to check if there is a significant difference. The results of which are shown in Table 4-8. All the means were > 4.The p-value of the measures were all <0.05 which indicate that there is a significant difference comparing the mean of each measure to the value of four. Therefore, the null hypothesis should be rejected and hypothesis **H_2** to be accepted. All the values concerning the success measures are reported in Table 4-8 and the testing of the second hypothesis **H_2** to be followed below.

H_2a. Agile software development projects are successful in terms of the time constraint

Mean value of 4.91 > 4

P-value. 0.001 < 0.05

H_2a >>> Accepted

H_2b. Agile software development projects are successful in terms of the cost constraint

Mean value of 4.93 > 4

P-value. 0.001 < 0.05

H_2b >>> Accepted

H_2c. Agile software development projects are successful in terms of the scope constraint

Mean value of 5.31 > 4

P-value. 0.001 < 0.05

H_2c >>> Accepted

H_2d. Agile software development projects are successful in terms of addressing the organisational needs

Mean value of 5.85 > 4

P-value. 0.001 < 0.05

H_2d >>> Accepted

H_2e. Agile software development projects are successful in terms of achieving stakeholders' satisfaction

Mean value of 6.09 > 4

P-value. 0.001 < 0.05

H_2e >>> Accepted

4.4 The Differences in the Importance of the CSFs

Linear regression analysis was used to answer the second research question (RQ2), which asked: "Does each success factor of agile software development have the same importance during all phases or iterations of the agile project?" In order to investigate how the importance of the success factors varies during the agile development project, it was essential to test the H1 hypothesis: H1. The importance of the CSFs of agile software development varies during the agile development project depending on the number of completed iterations in the project (X).

The questionnaire was designed to measure the progress of the agile projects, and establish how many iterations or phases have been completed. Indeed, one particular question asked participants how many iterations or phases had been completed in the agile project (the survey is available in Appendix A).

To conduct the H1 hypothesis test, linear regression was used. A confidence level of 95% was employed, and the error rate accepted by the researcher α (alpha) was set at 0.05. When using the Anova test with 95% confidence, the regression Sig. must be less than 0.05 in order for the regression to be considered significant. The regression test was applied to the independent factor (number of completed iterations) and the dependent factor (the critical success factor). The Defining of the hypotheses were discussed in section 3.3. The null hypothesis and the alternative hypothesis were as follows:

H_0: The importance of the CSFs of Agile software development does not vary during the agile development project depending on the number of completed iterations in the project:

H_1: The importance of the CSFs of Agile software development varies during the agile development project depending on the number of completed iterations in the project:

Looking at the questionnaire data, it was clear that there were 11 responses which contained empty answers for the question of the number of completed iterations. These missing values had to be handled before conducting the linear regression analysis. The missing values were list-wised deleted from the analysis (see Table 4-9).

Table 4-9 Excluded Cases before the Regression Analysis

Case Processing Summary

	N
Total Cases	131
Excluded Cases	11

The variation of the importance of a success factor during the agile project will be accepted if the null hypothesis was rejected ($p < 0.05$). A linear regression test was conducted for each success factor with the independent factor (number of completed iterations). The regression data analysis resulted in only one factor, namely the project management process, which had a value of $p < 0.05$. Conversely, the other seven success factors yielded values of $p > 0.05$. Table 4-10 shows the p-values of the regression of each success factor, along with the number of completed iterations.

Table 4-10 shows the regression significance levels of the tested relationships between the importance of the success factors and the number of completed iterations. Table 4-10 also displays the type of the relation between the importance of the success factors and the number of completed iterations, in terms of whether there is a positive or negative relation, or there is no relation at all. It is important to remember that a positive relation means that the importance of the success factor increased when the agile project progressed, and a negative relation means that the importance of the factor decreased when the agile project progressed.

Table 4-10 The Regression Significance Level (p-values) for the CSFs

Success Factor	**Regression Sig.**	**Relation**
Project Management Process	0.001	Negative
Delivery Strategy	0.325	Negative
Organisational Culture	0.446	Positive
Customer Involvement	0.470	Positive
Communication	0.713	Positive
Agile Software Development Techniques and practices	0.714	Positive
Team Capability and Training	0.805	Positive
Top Management Support	1.00	No relation

In this section, we will present the linear regression test results for two of the success factors, namely the Project management process factor and the Organisational culture factor. The results of the linear regression tests of the remaining success factors are available in Appendix C.

4.4.1 Project Management Process Linear Regression Analysis

In the case of the project management factor, a negative relation was found between the importance of the project management factor and the number of completed iterations in the project. This means that when an agile project progressed, and the number of completed iterations increased, the importance of the project management factor decreased. Table 4-11 shows the

Anova test results of the linear regression analysis for the project management factor. In addition, Figure 4-1 illustrates the linear regression curve for the project management factor.

Table 4-11 Anova Test Results for Project Management Success Factor

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	30.695	1	30.695	11.911	.001
Residual	304.097	118	2.577		
Total	334.792	119			

The independent variable is Number of completed iterations (or phases) in the project.

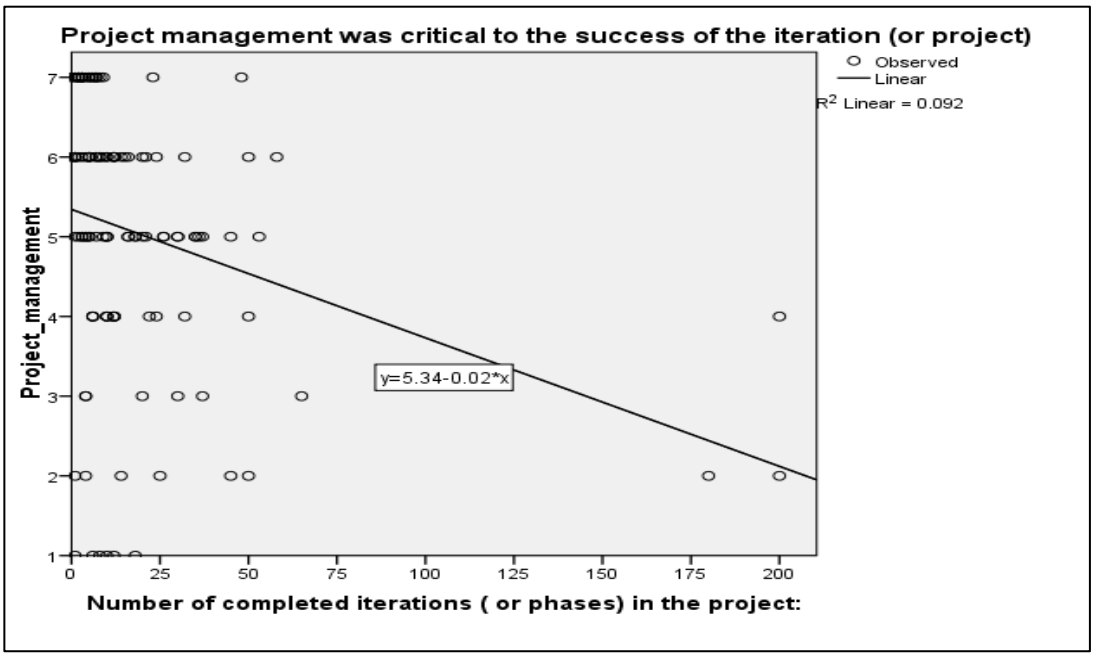


Figure 4-1 The SPSS Output of the Regression Curve for the Project Management Success Factor

4.4.2 Organisational Culture Linear Regression Analysis

With regard to the organisational culture factor, there was a positive relation between the importance of the organisational culture factor and the number of completed iterations in the project. This means that when the agile project progressed, and the number of completed iterations increased, the importance of the organisational culture factor increased. Table 4-12 shows the Anova test results of the linear regression analysis for the organisational culture factor. Moreover, Figure 4-2 illustrates the relation between the organisational culture factor and the number of completed iterations. The regression Sig. value for the organisational culture factor was 0.446,

which is not significant; however, the result is presented here regardless of this factor's non-significance.

Table 4-12 Anova Test Results for Organisational Culture Success Factor

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	1.138	1	1.138	.585	.446
Residual	229.453	118	1.945		
Total	230.592	119			

The independent variable is Number of completed iterations (or phases) in the project.

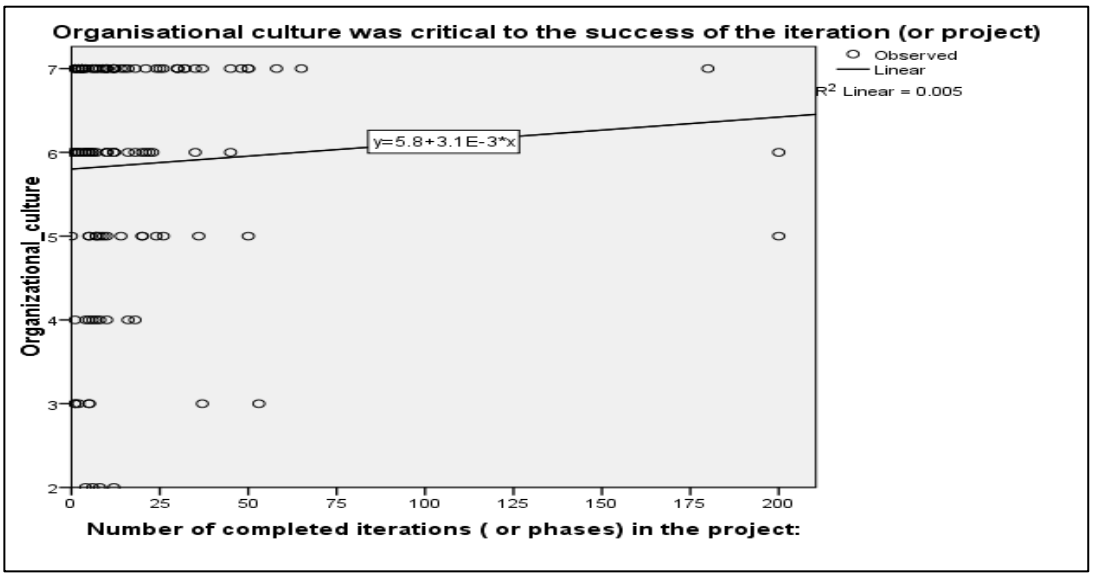


Figure 4-2 The SPSS Output of the Regression Curve for the Organisational Culture Success Factor

4.4.3 Linear Regression Analysis after Excluding the EV

The data used in the linear regression analysis, which came from the survey, comprised three responses where the completed iterations question was answered, as follows: 180, 200, and 200. These values may well be outlier values or extreme values (EV) that influenced the regression testing results. Thus, further linear regression analysis was conducted on the data after removing these extreme values.

The result of the regression testing after excluding the extreme values showed some changes in the regression Sig. values. Nevertheless, the project management factor remained the only factor to yield a significant regression ($p < 0.05$); indeed, this factor delivered a p-value of 0.011, which is less than the significance level of 0.05.

Table 4-13 shows the Anova test results of the linear regression analysis for the project management factor after excluding the EV. Moreover, Figure 4-3 illustrates the linear regression curve for the project management factor after excluding the EV. The results of the linear regression tests of the remaining success factors after excluding the EV are available in Appendix C.

Table 4-13 Anova Test Results for Project Management Success Factor after Excluding the EV

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	17.115	1	17.115	6.612	.011
Residual	297.654	115	2.588		
Total	314.769	116			

The independent variable is Number of completed iterations (or phases) in the project.

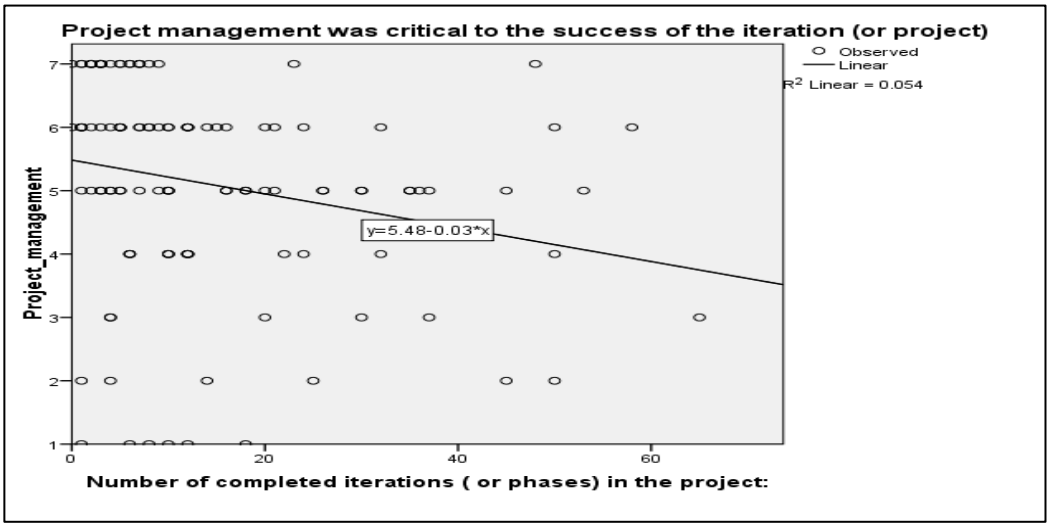


Figure 4-3 The SPSS Output of the Regression Curve for the Project Management Success Factor after Excluding the EV

H_1. The importance of the CSFs of Agile software development varies during the agile development project depending on the number of completed iterations in the project (X).

H_1a. The importance of delivery strategy factor varies depending on X.

P-value. 0.325 > 0.05 **H_1**a >>> Rejected

H_1b. The importance of team capability and training factor varies depending on X.

P-value. 0.805 > 0.05 **H_1**b >>> Rejected

H_1c. The importance of agile software development techniques factor varies depending on X.

P-value. 0.714 > 0.05 **H_1**c >>> Rejected

H_1d. The importance of customer involvement factor varies depending on X.

P-value. 0.470 > 0.05 **H_1**d >>> Rejected

H_1e. The importance of project management process/approach factor varies depending on X.

P-value. 0.001 < 0.05 **H_1**e >>> Accepted

H_1f. The importance of organisational culture factor varies depending on X.

P-value. 0.446 > 0.05 **H_1**f >>> Rejected

H_1g. The importance of communication factor varies depending on X.

P-value. 0.713 > 0.05 **H_1**g >>> Rejected

H_1h. The importance of top management support factor varies depending on X.

P-value. 1.00 > 0.05 **H_1**h >>> Rejected

4.5 Categorising the CSFs of Agile Software Development

Factor analysis was used in this study for the purpose of answering the third research question (RQ3), which asked: Using factor analysis, how can the CSFs of agile development be grouped into a smaller number of categories?

There exist two types of factor analysis, namely Exploratory Factor Analysis (EFA) and Confirmatory Factor Analysis (CFA). EFA is used when a study is conducted without prior assumptions, while CFA is used when prior hypotheses have been formulated (Yong and Pearce, 2013). Since the objective in this study was to determine a smaller number of categories that can represent the relations between the CSFs of agile software development without a pre-set hypothesis, EFA was used throughout this research.

At this point, it is fitting to refer to the study of Williams, Onsman, & Brown (2010), who introduced a 5-step exploratory factor analysis protocol. The first step involves checking the appropriateness of the data for EFA, while the second step entails deciding how to extract the factors; the third step relates to deciding how many factors to extract, and the fourth step involves selecting the rotation method. Finally, the last step comprises interpreting the results and labelling the factors. This protocol was followed in the present study to assist with the conducting of factor analysis.

In the past, any suggestions that there is a minimum sample size for conducting factor analysis have always been disputed. However, Osborne and Costello (2005) and Mundfrom, Shaw, & Ke (2005) concluded that this depends on many artefacts, such as the ratio of the number of variables to the number of participants. Mundfrom, Shaw, & Ke (2005) suggested that, in general, 100 is a reasonable minimum sample size for conducting factor analysis. Guadagnoli & Velicer (1988) concluded that, with a small sample size (< 150), the researcher should be careful and should only accept the variables with loadings of > 0.4; conversely, a larger sample size (> 300) should be sufficient for all the loadings values. According to Yong & Pearce (2013), the ratio of participants to variables should be at least 10:1.

4.5.1 Data Screening

In this study, it was essential to assess the appropriateness of applying factor analysis to the collected data. Before conducting factor analysis, variables that correlate highly with other variables should be excluded. Field (2009) suggested that variables with correlation r more than 0.9 should not be included in factor analysis. This can be checked by examining the correlation matrix. The same holds for low correlations; if the correlation matrix between the variables results in many correlations that are under 0.3, the researcher should not proceed with factor analysis (Williams et al., 2010). The data set in this study did not contain any variables with a very high correlation or many low correlations which needed to be excluded prior to performing the factor analysis.

Since the initial requirements of conducting factor analysis were met, the factor analysis was used as part of the data analysis stage. In this study, the factor analysis was applied to a sample of 131 participants, while the total number of variables was 8.

4.5.2 Factor Extraction

There are a number of factor extraction techniques, such as principal component analysis (PCA), maximum likelihood, and principal axis factoring (PAF). Selecting a factor extraction technique depends on the nature of the research. The principal components analysis technique is the default of many software tools (SPSS, SAS), and is the most common approach (Osborne & Costello, 2005). Furthermore, principal component analysis was used in this research because it accounts for the maximum of the total variance in the variables (Field, 2009).

Indeed, the principal component analysis (PCA) used in this study was applied to an 8-question questionnaire that explored the importance of the CSFs of agile projects. Each question presented the respondents with seven options, from strongly agree, to strongly disagree (7-point Likert scale). The applicability of PCA was assessed prior to the analysis.

The Kaiser-Meyer-Olkin (KMO) is a measure of sampling adequacy, and ranges from 0 to 1. According to Williams et al. (2010), a KMO with more than 0.50 should be sufficient for factor analysis. In this study, the KMO using the PCA yielded a value of 0.820, which is good or meritorious. Kaiser (1974) suggested that if the KMO is in the 0.80s, then the factoring of the variables is meritorious. Table 4-14 lists Kaiser's (1974) classification of KMO values. Bartlett's test of Sphericity Sig. should be < 0.05 to be considered statistically significant. Table 4-15 displays the KMO and Bartlett's test. Both the KMO measure score and Bartlett's test of Sphericity indicated that the data in this study was sufficient for factor analysis.

Table 4-14 Kaiser's (1974) Classification of KMO Values

KMO Measure	Meaning
KMO ≥ 9	Marvellous
0.8 ≤ KMO < 0.9	Meritorious
0.7 ≤ KMO < 0.8	Middling
0.6 ≤ KMO < 0.7	Mediocre
0.5 ≤ KMO < 0.6	Miserable
KMO < 0.5	Unacceptable

Table 4-15 SPSS Output of KMO and Bartlett's Test

KMO and Bartlett's Test

Kaiser-Meyer-Olkin Measure of Sampling Adequacy.		.820
Bartlett's Test of Sphericity	Approx. Chi-Square	327.855
	df	28
	Sig.	.000

The PCA revealed a number of components, the first two of which explained 45.77% and 12.83% of the total variance, respectively. The two factors (or components) combined explained 58.61% of the total variance. According to Osborne and Costello (2005), there are two techniques which are used to decide how many factors should be retained or extracted from factor analysis.

The first technique, usually called Kaiser's criterion, involves retaining all the factors with eigen-values greater than 1. This criterion was proposed by Kaiser (1960). The second technique is the scree test. The scree test includes examining the scree plot graph, and looking for the break point where the curve starts to flatten after it; all the points above that break point are then counted (not counting the point itself). The scree test is subjective and requires a judgement from the researcher (Williams et al., 2010); this is because, with the scree plot, it is sometimes hard to determine where the breaking point is.

Kaiser's criterion revealed two components (or factors) to be extracted. For the purpose of exploring all the other options of factor extraction, PCA was run using SPSS, and this determined 3 and 4 factors to be retained. The 3-factor extraction results with one factor contained only two variables, which is not recommended, as per Field (2009) and Osborne & Costello (2005). The same was true of the 4-factor extraction, which contained two factors that had only one variable loaded into them. Figure 4-4 and 4-5 show the total variance and the scree plot.

Total Variance Explained

	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings		
Component	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	3.662	45.777	45.777	3.662	45.777	45.777	2.755	34.438	34.438
2	1.027	12.835	58.613	1.027	12.835	58.613	1.934	24.175	58.613
3	.862	10.777	69.390						
4	.640	8.001	77.391						
5	.611	7.641	85.032						
6	.489	6.111	91.143						
7	.423	5.291	96.434						
8	.285	3.566	100.000						

Extraction Method: Principal Component Analysis.

Figure 4-4 SPSS Output of the Total Variance

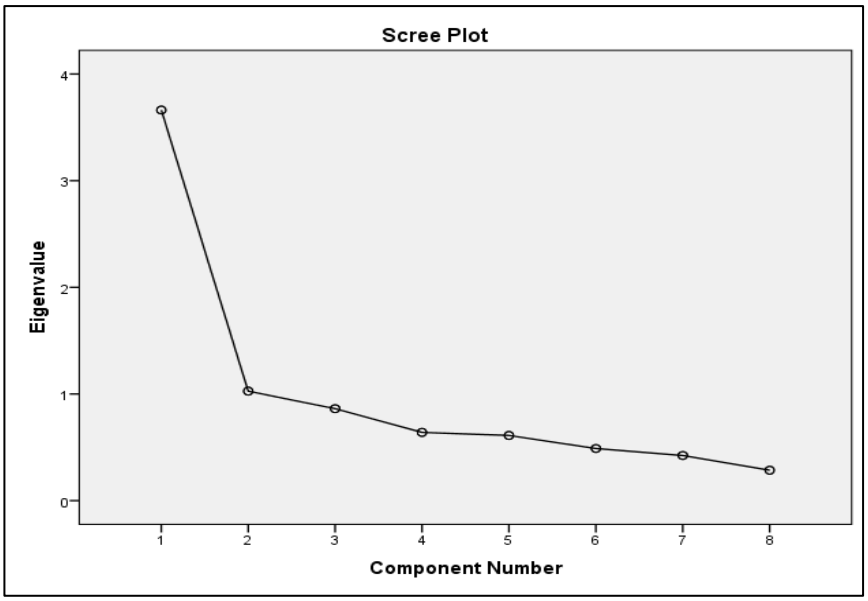


Figure 4-5 SPSS Output of the Scree Plot

In light of this, and following Kaiser's criterion, two components (or factors) were retained (extracted) for further investigation. The extracted components included: Component 1, which explained 45.77% of the total variance, and Component 2, which explained 12.83% of the total variance.

4.5.3 Factor Rotation

Having decided on the number of factor to be extracted, it was time to interpret the factors and their loadings. For this purpose, data rotation techniques were used. The aim of factor rotation is to simplify the data structure so as the variables can be better interpreted. There are two factor rotation techniques, namely the Orthogonal (or Varimax) technique, and the Oblique (or Oblimin) technique. With factor rotation, there is no technique which is expected to produce a substantially different outcome. This approach is just a way of simplifying the factorising, and producing results which are easier to interpret.

According to Field (2009), Varimax rotation provides results which are clearer and easier to interpret. Osborne and Costello (2005) also concluded that the Varimax rotation technique is the most common method, and again referred to the benefit of easy interpretation. Regardless of the rotation technique used, the objective is to produce more easily interpretable factoring (Williams et al., 2010).

In this research, the Varimax rotation technique was selected and applied to the two extracted components. The Oblique method was also examined for comparison. Of the two rotation techniques, the Varimax technique was used, since it is better suited to the two extracted

components and represents the variable loadings in a more interpretable way. Figure 4-6 shows the SPSS output of the component matrix and the rotated component matrix.

Following the eigen-values, two factors (or components) were extracted for further investigation. After rotation with the Varimax technique, the eight critical success factors (variables) of agile software development were loaded into two factors (or components). With the aim of representing how these variables were loaded into the two components, Figure 4-7 shows the SPSS output of the component plot in rotated space.

Component Matrix[a]

	Component	
	1	2
The delivery strategy was critical to the success of the iteration (or project)	.592	.390
Team capability and training were critical to the success of the iteration (or project)	.704	-.094
Agile software development techniques and practices were critical to the success of the iteration (or project)	.673	.226
Customer involvement was critical to the success of the iteration (or project)	.734	-.335
Project management was critical to the success of the iteration (or project)	.488	.671
Organisational culture was critical to the success of the iteration (or project)	.677	-.477
Top management support was critical to the success of the iteration (or project)	.757	-.149
Communication was critical to the success of the iteration (or project)	.744	.050

Extraction Method: Principal Component Analysis.

a. 2 components extracted.

Rotated Component Matrix[a]

	Component	
	1	2
The delivery strategy was critical to the success of the iteration (or project)	.251	.663
Team capability and training were critical to the success of the iteration (or project)	.626	.337
Agile software development techniques and practices were critical to the success of the iteration (or project)	.412	.578
Customer involvement was critical to the success of the iteration (or project)	.791	.159
Project management was critical to the success of the iteration (or project)	.002	.830
Organisational culture was critical to the success of the iteration (or project)	.828	.011
Top management support was critical to the success of the iteration (or project)	.701	.323
Communication was critical to the success of the iteration (or project)	.573	.477

Extraction Method: Principal Component Analysis.
Rotation Method: Varimax with Kaiser Normalization.

a. Rotation converged in 3 iterations.

Figure 4-6 SPSS Output of the Component Matrix and the Rotated Component Matrix

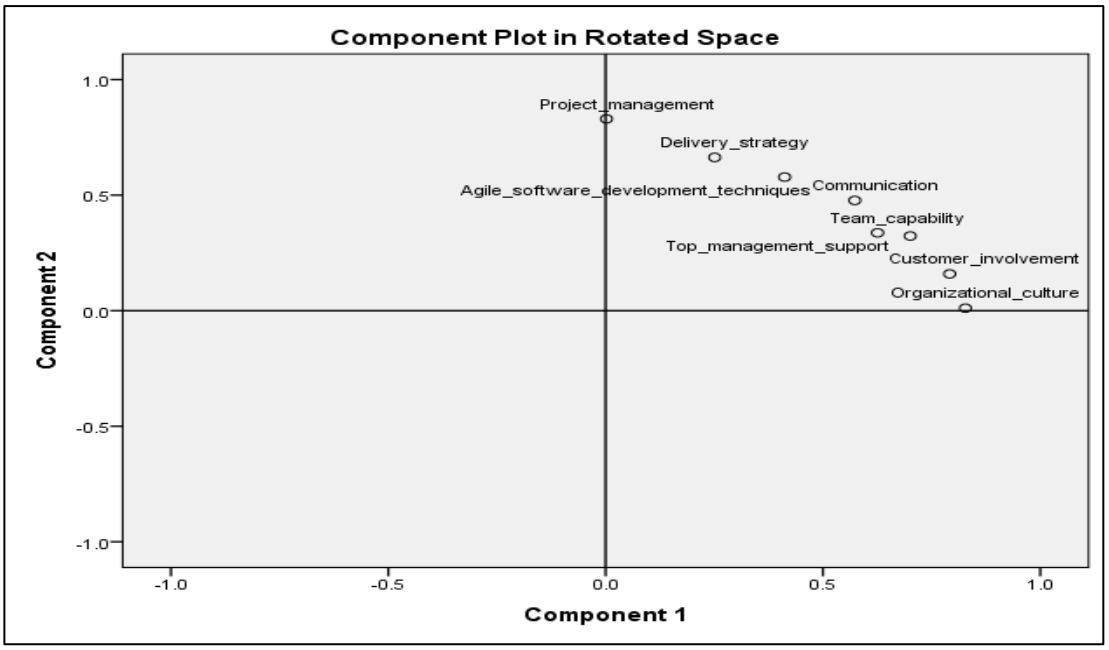


Figure 4-7 SPSS Output of the Component Plot in the Rotated Space

4.5.4 Factor Loading Cut-off

After deciding on the components (or factors) extraction technique, and the rotation technique, it was important to be able to interpret the components and their loaded variables; as such, we had to assign a factor loading cut-off point to the variable loadings.

Indeed, many past studies have investigated how to determine the factor loading cut-off point. Field (2009) suggested that, regardless of the sample size, a factor should be considered reliable if it has four or more loadings with 0.6 or above. On the other hand, Pituch and Stevens (2016) advocated using a factor loading cut-off of 0.4.

Tabachnick and Fidell(2007) suggested more rigid cut-offs: 0.32 as poor, 0.45 as fair, 0.55 as good, 0.63 as very good, and more than 0.71+ as excellent. Hair et al. (2010) provided a mapping of the factor loadings and the sample size needed. This held that a factor loading of 0.50 should be assigned to a sample of 120, while a factor loading of 0.45 should be assigned to a sample of 150, and so on.

In light of this, our sample size was 131; as such, according to Hair et al. (2010), the minimum factor loading that should be considered is 0.50. In this study, the minimum variable loading cut-off was determined as 0.50. The table that shows the rotated variable loadings after apply the cut-off is available in the SPSS. Table 4-16 is driven from the SPSS output with few formatting to display the components and their variables clearly.

Table 4-16 The Rotated Variable Loadings after Apply the Cut-off (factor loadings < 0.50 are excluded)

Variable Loading	Component 1	Component 2
Organisational culture	0.828	
Customer involvement	0.791	
Top management support	0.701	
Team capability and training	0.626	
Communication	0.573	
Project management process		0.830
The delivery strategy		0.663
Agile software development techniques and practices		0.578
Total % of Variance Explained	**45.77%**	**12.83%**

Principal component analysis was applied to the eight CSFs of agile software development. Following the eigen-values and Kaiser's criterion, two components (or factors) were extracted for further analysis. These two factors were then rotated for easier interpretation, and the cut-off point of the factor loading was determined as 0.50. The extracted components, after rotation with their variables and loadings, were:

Component 1: The Related Variables with their Loadings after the Rotation:

- Organisational culture = 0.828.
- Customer involvement = 0.791.
- Top management support = 0.701.
- Team capability and training = 0.626.
- Communication = 0.573.

Component 2: The Related Variables with their Loadings after the Rotation:

- Project management process = 0.830.
- The delivery strategy = 0.663.
- Agile software development techniques and practices = 0.578.

4.6 Survey Comments Analysis

The aim of this section is to present the comments about the agile success which were received by the participants of the survey. In terms of the reason for analysing these comments, it was essential to capture the perception of the participants in regard to the success of agile projects. The survey

received several comments, of which number of comments are selected for further analysis. The criterion of the selecting was to select all the comment which are related to the agile success factors and measures and to ignore the irrelevant comments. In this section, the results will be presented and the discussion of these comments will be followed in chapter 5.

4.6.1 Comments Concerning the Success Factors

With regard to agile success factors, the following comments were received from the participants of the survey:

“Managers training and involvement is key” Participant A. This was followed by a statement from Participant B “consideration for other Team members is the key success factor, especially for geographically spread teams.” In addition to this, Participant C stated that “close communication & frequent delivery can have effect on project's success.” Participant D added that “dedicated participants, and daily meeting are critical factors to reach the expected successful.” In addition to that, Participant E said that “agile/scrum training at the outset of all team members and stakeholders was the most important success factor.”

Participant F added 5 points as follows “1. The ability to self-organise and manage was a critical success factor at the team level. 2. The elimination of activities that provided limited value to the effort was a critical success factor (got rid of independent test, independent QA, watchers and lookers who did not help develop - only criticized) 3. The use of a seasoned agile coach was a critical success factor. 4. The capture and use of key agile metrics as a critical success factor. 5. The use of an agile method that was wedded to our internal management infrastructure (not pure but hybrid) was critical success factor.”

4.6.2 Comments Concerning the Success Measures

Of the received comments, the following are related to the success measures of agile software development projects:

Participant E concluded that “time, cost and scope constraints are not usually considerations when looking at an "agile" project. These apply in "upfront" planning projects, not "agile" ones. Agile projects focus on delivery of "business"/"user" value, in priority order, in an iterative sense.” Following that, Participant G stated “from what I have seen, it helps delivering the project, but time and scope and money wouldn't be easily defined.”

4.6.3 General Comments

Participant F added that “the need for product management is more important in agile than the need for project management.” Participant I concluded that “agile is about People and communication methods between them.” In addition to that, Participant J said that “a key component of successfully re-designing the look and process plus extending to new channels/delivery was to create the base product and premise of the larger, transformation project within existing constraints and ensuring that knowledge/information and decisions were shared, communicated, documented and understood.”

4.7 Summary

Chapter 4 has presented the results and the findings of the survey. The design of the survey and the data collection process were discussed. An overview of the survey’s results was introduced in this chapter. Following this, a descriptive statistical analysis of the questionnaire has been used to explore the differences in the importance among the CSFs of agile software development and among the success measures. Linear regression analysis was used to study the relationships between the agile projects’ progress and the importance of each success factor. Moreover, a principal component analysis was conducted to investigate the inter-relations between the eight identified success factors of agile development. The participants’ comments were analysed in order to obtain a deep understanding of their experience with agile development projects. In this chapter, the results have been introduced. The discussions of these results are followed in the next chapter.

Chapter 5 Discussions

This chapter highlights the observations from the results and findings presented in the last chapter. The literature revealed a set of critical success factors related to agile software development. Eight of these factors have been selected in this present research. The importance of these success factors and success measures of agile software development was explored by examining the results of the survey. The relationship between the agile project's progress and the importance of each success factor was assessed. Using factor analysis, the inter-relations between the CSFs of agile software development were investigated. In this chapter, Section 5.1 discusses the results and the findings related to the differences in importance between the CSFs of agile development. Section 5.2 examines the results of the linear regression analysis and the relationship between the agile project's progress and the importance of CSFs. In Section 5.3, focus switches to discussions and interpretations of the factor analysis. Section 5.4 analyses the results pertaining to the success measures of agile development. Section 5.5 summarises the chapter.

5.1 The Importance of the CSFs of Agile Development

To address the RQ1, agile success factors were ordered by their importance to the success of agile projects, as follows: communication, customer involvement, team capability and training, top management support, organisational culture, agile software development techniques and practices, delivery strategy, and project management process (see Table 4-6).

The results revealed that communication, customer involvement, and team capability were the most important success factors in agile development projects. This interpretation emerged after observing that the factors related to the people and the organisational culture were the most important factors in the agile projects. Moreover, it was found that people and organisational aspects were the most important challenges that need to be addressed in the agile project.

On the other hand, the project management process factor and the technical factors were found to be the least important between the CSFs. It is possible to interpret this result as meaning that, in agile development projects, the people and organisational factors matter the most.

The obstacles and challenges which emerge when moving into agility in the software development context are more related to the people and the organisational aspects. While the technical factors still contribute to the success of agile development, the people and organisational factors contribute more.

As stated by Participant I, “Agile is about people and communication methods between them”. This shows the importance of the communication factor and the people factors in the success of agile software development projects.

5.2 Agile Software Development Project Progress and the Importance of the CSFs

To answer the second research question RQ2, the hypothesis H2 on the relationship between the progress of agile projects and the importance of each success factor was tested in Section 4.3. The results of the regression analysis showed that, for some of the success factors, the importance of the factor increased whenever the agile project progressed. In contrast, for some of the success factors, the importance of the factor decreased whenever the agile project progressed.

The result for the project management process success factor yielded a significant relation (p-value ≤ 0.05) between the importance of the factor and the progress of the agile project. On the other hand, the results generated for the remaining seven success factors gave insignificant relations (p-value > 0.05).

The interpretation of the regression analysis results might be aided by observing that the importance of the project management process factor decreased when the agile project progressed. It was also possible to conclude that the importance of the delivery strategy factor decreased when the agile project progressed.

On the contrary, it was noticed that the importance of various factors, namely organisational culture, customer involvement, communication, team capability, and agile software development techniques, increased when the agile project progressed. With regard to the top management support factor, no relation was found between the importance of this factor and the agile project’s progress.

For those aiming to achieve a successful agile development project, it is important to notice that the project management process and delivery strategy factors are more important during the early stages of the agile development project. Top management support is needed to achieve the success with agile projects, regardless of the project’s status or progress.

When the agile project progresses, the importance of the team’s capability and training will increase. The same applies to the factors of communication, customer involvement, organisational culture, and agile software development techniques.

5.3 Factor Analysis Results Interpretation

To address the third research question RQ3, and using factor analysis techniques. A principal component analysis was conducted on the eight success factors of agile software development. This resulted in two extracted components (or factors), namely component 1 and component 2. The two extracted components (or factors) were interpreted and could be named as:

- Component 1: **Organisational and People Aspects.**
- Component 2: **Technical and Project Management Aspects.**

5.3.1 Component 1: Organisational and People Aspects

Component 1 indicates the importance of organisational aspects and people aspects in the success of agile software development. The highest loading in this component is the organisational culture variable, with loading of 0.828. This indicates that the organisational and people aspects are strongly affected by the organisational culture variable. This could also be supported by observing that the participants of the survey had a similar level of comprehension between the organisational culture and all of the organisational and people aspects combined. Thus, it seems that the people and organisational aspects of the success of agile software development projects can be understood by referring to organisational culture more than any other critical success factor. The organisational culture variable was followed by the variables of customer involvement and top management support, with 0.791 and 0.701 loadings, respectively. The team capability variable loaded into component 1 had a loading equal to 0.626. All the variables loadings on the two components were shown in Table 4-16.

It is hoped that this could help agile practitioners, as it will allow them to notice that, in agile software development projects, the organisational culture plays an important role when it comes to the people factors as well as the organisational factors. It was proven that success factors such as top management support, customer involvement, and team capability were defined and explained by the organisational culture success factor, more than any other success factor. The culture of the organisation where the agile project is implemented will have an impact on the people factors more than the technical and project management factors.

In contrast, this component had low loadings with the technical and project management variables. The project management variable yielded a 0.02 loading, while the delivery strategy variable yielded a 0.251 loading. This demonstrated the low importance of project management in explaining the organisational and people aspects in agile projects.

The communication variable was loaded into component 1 with a loading of 0.573. At the same time, the communication variable was also loaded into component 2 with a loading equal to 0.477 (below the chosen cut-off point). This showed that communication plays almost the same role when it comes to the variance of organisational and people aspects and the variance of technical and project management aspects. This could mean that, in the context of agile software development projects, organisational and people aspects influence the method of communication. At the same time, it could be said that technical and project management aspects influence the way the project team communicates during the agile project.

5.3.2 Component 2: Technical and Project Management Aspects

Component 2 shows the importance of the project management variable in explaining the variance across the technical success factors in agile development projects. The project management variable was loaded into component 2 with a loading equal to 0.830. This suggested that the project management variable is highly correlated with the technical facets of agile development. Furthermore, the project management variable can be used to describe the technical aspects of agile development projects. The delivery strategy variable was loaded into factor 2 with a 0.663 loading. The variable of agile software development techniques and practices was loaded with 0.578 loading.

On the other hand, the organisational culture variable was loaded with -0.477. The same was true for the variables of customer involvement, top management support, and team capability and training, all of which were loaded with negative values. Field (2009) and Pituch & Stevens (2016) concluded that negative variables indicate that these variable does not play role in explaining the variance of the component. This suggested that none of these variables play a role in explaining the variance of the technical and project management aspects. Moreover, the negative loadings suggested that these variables explain the variance in the other component.

The technical aspects of agile software development are influenced by the project management success factor more than any other success factors. This should help agile practitioners when they are selecting the project management approach for agile projects. The selected project management approach will influence the technical aspects of agile projects, such as the delivery strategy. Project management approach should be aligned with the technical aspects of the chosen agile technique or practice. Figure 5-1 illustrates the two extracted components and their items.

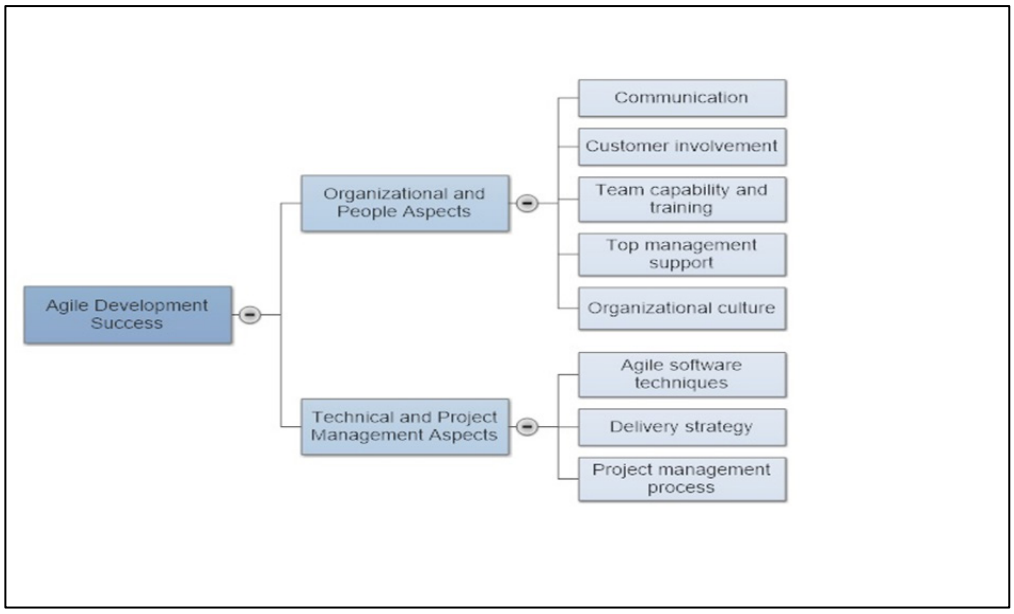


Figure 5-1 The Two Extracted Components and their Items

5.4 Success Measures of Agile Software Development

The H2 hypothesis has been tested in Section 4.3.2, with the results showing that all the five measures can be accepted as success measures of agile software development. The success measures were ordered as follows, with the most supported first: achieving stakeholders' satisfaction, addressing the organisational needs, scope, cost, and time. (See Table 4-8).

According to the participants of the survey, the most supported measure of agile software development projects success is achieving stakeholders' satisfaction. This will help agile practitioners, who are measuring the success of agile projects, to focus on the satisfaction of the stakeholders and the organisational needs more than any other measures. These results could indicate that, in the case of agile development projects, time and cost measures are not enough to measure the success of the project. Thus, measuring the success with agile projects is behind the time, cost, and scope traditional measures. Organisations attempt for a successful implementation of agile software projects should align the success by satisfying the customers and addressing the organisational needs.

As stated by Participant E in the survey, "time, cost and scope constraints are not usually considerations when looking at an agile project. These apply in upfront planning projects, not agile ones. Agile projects focus on delivery of business/user value, in priority order, in an iterative sense". Furthermore, Participant G concluded, "from what I have seen, it helps delivering the project, but time and scope and money wouldn't be easily defined". This shows the need for always linking to the customers and the organisational needs when measuring the success in agile projects. These

results show that agile projects not necessarily will be successful with the time measure. It was the lowest supported measure as per the responses received in the survey. Those who attempting for implementing agile practices should know that even that agile software development is promising in terms of satisfying the customers and meeting the organisational needs, it might fail when it comes to meeting the time constraint. Therefore, for cases where timing matter the most, agile software development might not be the appropriate choice.

Following the observations from the literature review and the empirical findings discussed in this chapter, it can be concluded that the measures of agile project success are ordered as follows: achieving stakeholders' satisfaction, addressing the organisational needs, scope, cost, and time. Figure 5-2 illustrates the agile software development success factors and measures identified in this study. The order of the success factors and measures in Figure 5-2 is based on their importance as per discussed in this chapter. Future work might work on using structural equation modelling techniques such as confirmatory factor analysis to establish the latent relations between these factors and measures (as shown in Figure 5-2) and to confirm that the model is correct.

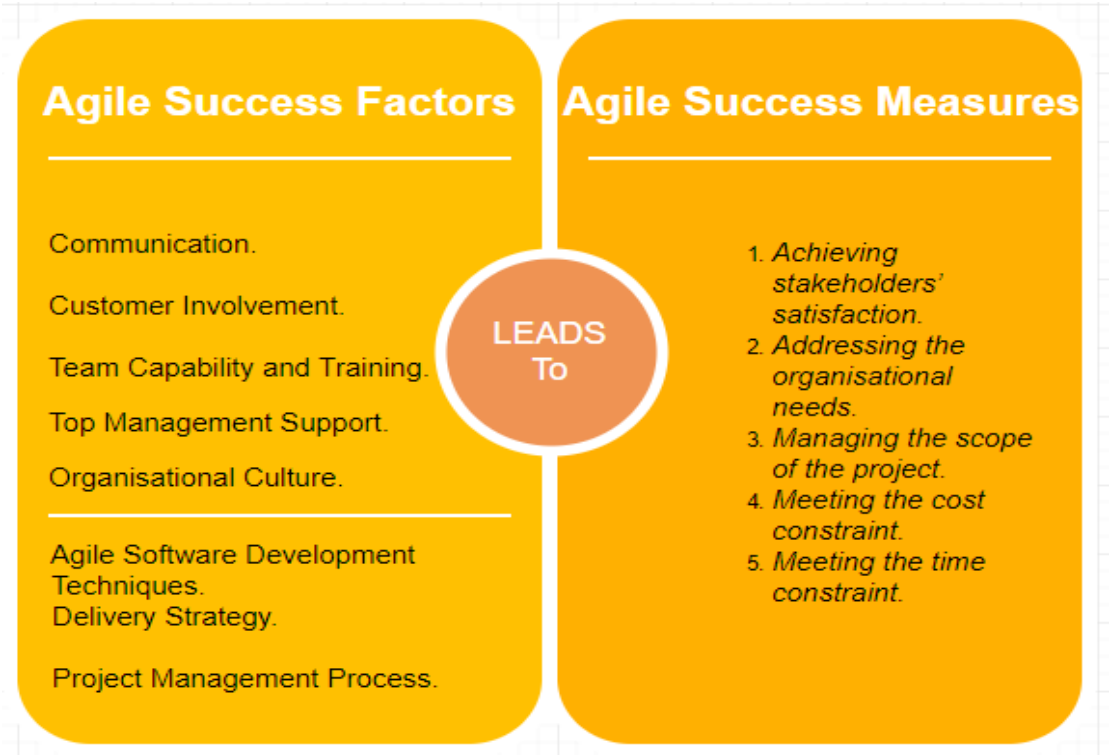


Figure 5-2 Success Factors and Measures of Agile Software Development

The rest of this thesis will be focusing on how the identified CSFs could be measured in an agile software development project. It is hoped that by developing rigorous measurements of the CSFs, agile practitioners will be able to evaluate their own performance on the identified success factors. The success with these CSFs will secure a successful implementation of agile in the organisations aiming to do so. In the following chapter, an instrument will be proposed with which a set of metrics

will be associated with every success factor of agile software development. The proposed instrument will be able to provide an overall score reflecting the success of an agile project in an organisation and independent scores for each success factor. By using the proposed instrument, agile practitioners will be able to evaluate their agile projects and will be able to know on which factors they are doing fine and on which factors they need to improve. In the two coming chapters, the developed instrument will be evaluated by agile experts and will be applied in case studies. The aim of these two steps is to validate the proposed instrument and to seek for a practical evaluation of the instrument, with which the instrument could be enhanced and refined for practical usage and acceptance from the agile practitioners' community.

5.5 Summary

The results and the findings drawn from the survey were explored in this chapter. The order of the success factors, which was determined by analysing the questionnaire data, was discussed. In addition to this, the chapter put forth the results of the regression analysis, which sought to establish the relationship between the importance of the success factors and the agile progress. This chapter introduced the interpretations drawn from the factor analysis of the CSFs of agile software development. In this chapter, the success measures for agile software development projects results were discussed. By the end of this chapter, the first three research questions were addressed. These research questions were concerning more understanding of the concept of the success in implementing agile software development projects. The next and the last research question RQ4 is concerning how the success with agile projects could be measured and evaluated. In the next chapter, the development of an instrument to measure the success of agile software development projects will be discussed.

Chapter 6 The Development of the Instrument

In this chapter, the development of the proposed instrument is discussed. Indeed, the instrument was developed by following the Goal-Question-Metric (GQM) approach and was subsequently validated through experts' interviews. The first section discusses the development of the instrument, providing information about the GQM approach; following this, the identification of the goals, questions, and metrics is explained. The scoring of the instrument is described as well. The second section examines the experts' interviews which were carried out to validate the proposed instrument. This chapter discusses the expert interview design and the methodologies employed to prepare these interviews. Following this, the results of these interviews are discussed. The analysis of the results is organised based on the goals of the proposed instrument, from the first goal to the last goal. The third section presents a review of the instrument after the experts' interviews and specifies how the instrument was revised. Lastly, the fourth section is a summary of the chapter. A full version of the instrument is presented in Appendix E.

6.1 Instrument Development

In this section, the development of the proposed instrument is discussed. The first section presents an introduction to the approach selected to develop the instrument. Following which, the defining of the instrument's goals is examined. Lastly, the scoring of the instrument is described.

6.1.1 Goal-Question-Metric Approach

The instrument will be developed by using the Goal-Question-Metric (GQM) approach. The GQM approach was proposed by Basili and Weiss (1984) with the aim of introducing a systematic way of defining goals that could easily be refined into questions and linked to metrics. The GQM approach has three levels: the conceptual level (Goal), the operational level (Question), and the quantitative level (Metric).

The goals are usually defined for specific purposes from a certain perspective and for a given objective. Therefore, the usage of the GQM approach will help to ensure that the measurements in question are defined for the aim of achieving specific goals (Gray & MacDonell, 1997). The questions are used to describe the approach to achieving the goals. The metrics are set based on data linked to each question, with the aim of answering said questions; indeed, the metrics could be objective or subjective. Defining goals is beneficial in terms of being able to focus on the important aspects.

Writing questions will make the goals more specific and will suggest the relevant metrics (Basili et al., 1994).

Using the GQM approach to develop software measurements is associated with many benefits, among which are improving the software product quality, enhancing software processes, and increasing the team cooperation (Birk, Van Solingen, & Jarvinen, 1998). The GQM-based measurement approach will help in avoiding irrelevant measurements through regular feedback and by involving the project team in defining measurements that are linked to the agreed-upon goals (Latum et al., 1998). A simple representation of the structure of the GQM approach is illustrated in Figure 6-1.

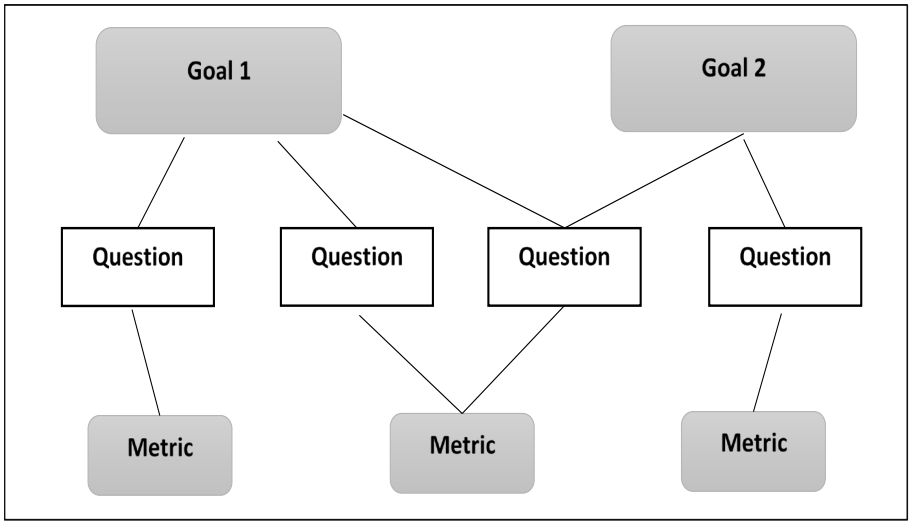


Figure 6-1 Representation of the Structure of the GQM Approach

In the real-world software engineering context, defining the right metrics and measures is the biggest challenge when it comes to implementing the GQM approach. Using goal-oriented measuring is believed to facilitate the process of defining the metrics in software engineering projects (Wang & He, 2003). The GQM is a systematic approach used to represent and combine a set of high-level goals into measurements. The result of implementing the GQM approach is the specification of a set of metrics addressing a particular set of goals and rules for interpreting these results (Solingen & Berghout, 2001).

According to Solingen (2014), the GQM approach is more relevant for agile software projects with many short-cyclic iterations, given the nature of such projects. The GQM approach will provide measurements with clear purposes and goals; indeed, this will result in saving time when developing the measurement, which is one of the agile development objectives. Berander and Jönsson (2006) introduced an extended GQM approach, in which the goals and questions are developed by a GQM-workshop. In these workshops, the goals and questions are incrementally

refined. Following this, a process of prioritising and categorising the questions is accomplished. The aim is to reach a small prioritised number of questions that are agreed upon from different perspectives in the organisation before defining the metrics.

Basili et al. (2007) introduced GQM+, which is a new measurement approach built on the well-established and widespread GQM approach. GQM+ provides a mechanism to link the software measurements to higher-level business goals. In GQM+, the goals are divided into two categories: business goals for the entire organisation, and software goals for the process of developing software in the organisation. Before defining the goals for the GQM approach, these two types of goals should be linked to define the measurement goals which will in turn be used to define the questions and metrics.

Using the GQM approach to define and classify the software measurement is a complicated process, and defining the metrics is the most challenging step. At the conceptual (goals) level, the software goals should be as specific as possible in order to facilitate the defining of the quantitative (metrics) level. At the metric level, the focus should be on finding the data to answer the questions and the metrics which could be used to answer them quantitatively (Yahaya, Abidin, Ali, & Deraman, 2013).

In order to define the metrics, the two following aspects are important. Firstly, the measurement should not be defined for the sake of measuring only. It should be explicitly linked to a defined goal. Secondly, the interpretations of the collected data can only be sufficient if they are made by the people whose knowledge is represented in the design of the metrics (Parviainen, Jarvinen, & Sandelin, 1997).

According to Lavazza, Frumento, and Mazza (2015), defining and evaluating success measures for software development projects involves several tasks. It is important to start with measures that represent the project's achievement. The measures should be precisely defined, feasible, and represent the different viewpoints of the team involved in implementing the GQM approach.

Since the introduction of the GQM approach in 1984, it has attracted significant attention. Several studies have been found in the literature concerning the usage of the GQM approach in software development. Some of the identified studies (e.g. Berander & Jönsson, 2006; Latum et al., 1998; Lavazza et al., 2015) used empirical methodologies to evaluate the adoption of the GQM approach in the real software development projects context by conducting case studies. Furthermore, Solingen (2014) advocated the adoption of the GQM approach for defining software measurements in agile software development projects.

Heidenberg et.al (2013) introduced a metrics model following the GQM approach to measuring the impact of agile transformation in software development organizations. Their model focused on measuring the business value, lead-time, and efficiency of the agile software development transformation. It was suggested that more measurements are necessary to assess the agile software development status in organizations attempting for agile transformation. Olszewska et.al (2016) have attempted to provide a quantitative measurement of the impact of agile transformation in software development organizations. They proposed a model with quantitative metrics following the GQM approach. The proposed model consists of one goal, four questions, and eight metrics following the structure of the Goal Question Metric (GQM) approach. It was suggested that future work might concentrate on providing qualitative metrics with which the status of agile software development could be evaluated and measured. In this research, the proposed instrument will use a mix of qualitative metrics and quantitative metrics to measure the success of agile software development projects.

In this study, the GQM approach will be followed to build an instrument that could measure the success of agile software projects. The goals will be driven by the success factors of agile software development which have been identified in the literature and which have been proven as a result of the previously-analysed survey (see Chapter 4). Achieving these goals will contribute to the success of agile software development projects. It is hoped that, by implementing the GQM approach, it will be possible to define a list of metrics which could measure the success of an agile software development project in an organisation. Each goal will be divided into items. These items will be in question form, and will be associated with metrics which could be used to measure the defined items.

6.1.2 Defining the Goals

The goals of the instrument will be driven by the CSFs identified in the literature (see Table 2-3). As suggested by the GQM approach, the eight identified CSFs will be rewritten as a set of goals. The two success factors of agile practices and techniques, as well as the project management approach, will be merged into one goal. This is because these two factors pertain to a selection process of available agile techniques and PM approaches. Moreover, to avoid the replication of having two goals related to a selection process, the goal will be to appropriately select these available agile practices and techniques and PM approaches.

Organisational culture is a soft factor which is difficult to measure and contains many aspects which overlap with other success factors such as communication and top management support factors, as acknowledged by Sheffield and Lemétayer (2013). As such, in the developed instrument, the

organisational culture factor will not be an independent goal. Instead, the organisational culture factor will be included in the first, second, third, and fourth goals. The goals of the proposed instrument are therefore listed as follows:

1. Improve the communication throughout the agile project.
2. Increase the customer involvement during the agile project.
3. Improve the training of the agile project team members.
4. Increase the support from top management in the agile project.
5. Enhance the delivery strategy.
6. Appropriate selection of agile techniques, practices, and project management (PM) approach.

6.1.3 Identifying the Questions and the Associated Metrics

Following the GQM approach, each goal will be associated with a set of questions and a set of metrics with which the defined questions could be answered. At the outset, these questions and metrics will be proposed by the researcher to construct the first version of the instrument. Following this, the first version of the instrument is reviewed by agile experts with the aim of validating the proposed instrument. The first version and the final version of the proposed instrument are both available in Appendix E. The experts' review and the modifications which were made to the proposed instrument are discussed in the coming sections.

6.1.4 Instrument Scoring

The proposed instrument followed a scoring scale with which the success of agile software development projects could be measured. The scoring is set to be used as an indication of how the participants of the instrument are doing and how they could achieve the defined goals of the proposed instrument, and ultimately achieve success with agile software development projects. With regard to the scoring of the instrument, the final score will range from 0 to 6, whereby 6 is the highest score. The final score is a result of totalling the scores of the six goals, each goal's score range from 0 to 1, whereby 1 is the highest score for each goal. The score of each goal is a result of summing of the scores for each question (0 to 1) dividing by the number of questions in that specific goal. This means that each question has the same weight when calculating the goal's score. And eventually, every goal of the six goals have the same weight when calculating the final score of the instrument.

6.2 Instrument Validation

The proposed instrument will be validated through experts' interviews. By using the experts' interviews research method, the researcher will likely be able to investigate the subject in depth and will eventually understand the investigated phenomenon accurately (Berg, 2004).

With regard to the design of the experts' interviews, qualitative research methods were used. These qualitative methods made it possible to confirm the development of the proposed instrument. These participants were all experts in agile software development. The participants were only considered to be experts if they had at least five years' experience with agile practices and techniques.

6.2.1 Experts Interview Piloting

The aim of piloting is to test the tool before it is employed in the study. As such, the pilot test will help in simulating the purposes for which the instrument is intended (Leeuw et al., 2012). In this study, piloting sessions were conducted to test the interview questions. The piloting involved five computer science PhD students at the University of Southampton. The goal of this was to ensure that the interview questions were comprehensible. They provided comments which pertained to the language of some of the questions, and to some complicated sentences. This made it easy to edit these questions in the instrument according to their comments.

6.2.2 Experts Interview Sample Size

When it comes to conducting experts' interviews, it is important to interview an appropriate number of experts, as this will help in obtaining reliable results (Banerjee et al., 2009). The experts sampling should be carried out according to non-random criteria, and the experts' selection should be directly related to their experience and knowledge of the investigated field (Bhattacherjee, 2012). With regard to the minimum sample size, scholars have yet to reach an agreed-upon minimum sample size in terms of how many experts should be interviewed. Nevertheless, according to Grant and Davis (1997), the majority of research has recommended a range of 3-20 experts to be interviewed. Marshall et al. (2013) examined the interview sample sizes of 83 information systems qualitative studies in leading information systems journals. They concluded that the number of conducted interviews is usually correlated with cultural factors such as the proceedings, number of authors, and world region. One suggestion is related to the use of the saturation method (Marshall et al., 2013). The saturation point is achieved when no new data can be generated (Guest et al., 2006).

According to Mason (2010), PhD students usually tend to interview a larger sample than they need to, just to be safe and to feel confident when it comes to their examination. Mason (2010) advocated the concept of saturation and stated that researchers should refrain from interviewing more people when they have reached the saturation point and no more new data is likely to be generated.

As stated above, scholars are yet to agree on an exact number of experts to be interviewed in order to validate an instrument. However, both Nielsen (2000) and Virzi (1992) claimed that, for usability validation, five is the right number of participants to be interviewed. With regard to this point, Guest et al. (2006) stated that the saturation point could be generally reached with 12 participants. Therefore, the present study will aim to hold interviews with 5-12 experts in the field of agile software development.

6.2.3 Expert Interview Design

This research applied semi-structured interviews encompassing open-ended questions and closed-ended questions to review the proposed instrument. The purposes of these interviews were: firstly, to review and confirm the proposed instrument's goals, questions, and metrics, and secondly to suggest any other questions and metrics that need to be considered when measuring the success of an agile software development project. The experts' interview process comprised many steps, which were as follows:

- Emails were sent to experts briefing them on the research and the objectives of the instrument. In the email, the experts were also asked to identify their preferred date and time for the interview.

- Depending on where the experts lived, some interviews were conducted online via Skype and Zoom, which are video calling applications. In contrast, other interviews were conducted on a face-to-face basis.

- Prior to the start of the interviews, all participants were requested to read the participant information sheet, following which they were asked to sign the consent form and return it by email.

- The participants were shown the instrument, and then had the opportunity to ask for further explanation if needed. This lasted approximately 5-10 minutes.

- Following this, the experts were asked about each goal in the instrument, starting with the first goal and ending with the last one. The experts were also asked about each item in the instrument and whether they felt that any additional item(s) needed to be added to the instrument.

- In the last part of the interview, the participants were asked to answer open-ended questions concerning how the instrument could be improved. This allowed the researcher to ensure that, according to the opinions of the interviewed experts, different aspects of agility were addressed in the proposed instrument. This also made it possible to confirm whether or not, according to the experts' interviewee responses, additional items were needed.

- The interviews were recorded and lasted approximately in the range of 35-60 minutes.

- The interviews were voice-recorded and summarised by the researcher using a pen and notebook. However, one participant refused to have his voice recorded, and that interview was hence not recorded. Instead, hand-written notes were taken as a record.

- The interviews were conducted over a period of five weeks during the months of January and February 2018.

6.2.4 Experts Interviews Data Reporting

Generally speaking, reporting the results of a study is a procedure which involves presenting the findings of said study in a way that makes it possible for those who are interested in the study to understand what the findings are. There are some differences between reporting the results of quantitative studies and those of qualitative studies.

Qualitative data reporting is commonly built around themes and general observations. To support the reporting, direct quotes from the interviews are usually included. With regard to qualitative data analysis, it is suggested that the results be reported so that general themes can be identified (Rubin & Rubin, 2011). According to Livingstone and Bovill (2013), presenting the qualitative data is challenging. Moreover, the researcher should always be careful when deciding which data to present in the report. The selection of the qualitative data to be reported should be linked with the aim of developing a better understanding of the investigated phenomenon and the identified research questions. In this study, the reporting of the interviews' qualitative data will focus on the data concerning the investigated instrument and the interviewees' comments and feedback on how to improve the proposed instrument.

6.2.5 Ethical Approval

Prior to conducting the interviews, obtaining the ethical approval from the ethics committee at the University of Southampton was considered. The ethical approval was given on the date of 15/12/2017: the reference number is ERGO/FPSE/31342.

6.2.6 Experts Demographic Information

At the outset, invitations were sent by email to 28 agile software development experts asking if they wanted to participate in the interviews. All of the contacted experts had at least five years of experience with agile software development; this meant that they were judged to be capable of reviewing the instrument. Of these 28 experts, 15 stated that they were happy to participate, although 2 later decided not to participate. Following this, all of the remaining 13 experts were interviewed. The interviews were recorded and later summarised. These participants came from various countries, including the US, Saudi Arabia, the UK, and France. The industries represented by the interview participants included, but were not limited to: the Education sector, the Finance and Banking sector, and the Information Technology & Software sector. The interviewed experts also represented different types of organisations, which ranged from small organisations with only 15-30 employees, to large multinational organisations. A breakdown of the demographic information pertaining to the interviewed experts is shown in Table 6-1.

Table 6-1 Breakdown of the Demographic Information of the Experts

Variable		**Frequency**	**%**
Country	France	2	15.38
	Saudi Arabia	2	15.38
	Czech Republic	2	15.38
	India	2	15.38
	UK	2	15.38
	US	1	7.69
	Australia	1	7.69
	Turkey	1	7.69
Sector	IT & Software	5	38.46
	Consultancy	4	30.76
	Finance and Banking	2	15.38
	Education	2	15.38
Job Title	Agile Coach	5	38.46
	Scrum Master	3	23.07
	CEO	2	15.38
	CTO	1	7.69
	Team Leader	1	7.69
	Engineer	1	7.69
Experience with Agile (years)	5-10	6	46.15
	11-15	4	30.76
	+15	3	23.07

These agile experts were interviewed aiming to validate the proposed instrument. The analysis of the interviews are detailed in the following section.

6.2.7 Interviews Data Analysis

The aim of conducting the interviews with the agile experts was to review and validate the proposed instrument and its ability to measure the success of agile software development. Semi-structured interviews were employed, which encompassed closed-ended and open-ended questions. At the start of the interviews, experts were asked closed-ended questions which concerned the experts' background, years of experience with agile software development, organisation profile, and sector. The results of these questions can be found in table 6-1.

Following this, the proposed instrument was briefly introduced and shown to the experts. The participants were given the chance to ask for any clarification, if needed. As soon as the experts had understood the proposed instrument, open-ended questions followed, asking the experts about each item in the instrument. In fact, the interview questions covered the first goal in the instrument right through to the last goal. The experts were given the chance to express their thoughts about each question and metric in the instrument. At the end of each goal, the experts were asked if they could think of any new questions or metrics that could be added under the corresponding goal in the instrument in order to facilitate better measurement of success within agile software development. The purpose of the open-ended questions in the interviews was to ensure that the proposed instrument could accurately measure the success of agile software development; moreover, according to the interviewed experts, the proposed instrument covered the aspects required to measure the success of agile software development. At the end of each interview, the participants were asked if they could think of suggestions on how to improve the proposed instrument.

After the data collected from the experts' interviews was analysed, certain results were achieved, as listed in the coming sections. With regard to the criterion that was followed to select the quotes from the interviews to be reported in the thesis, this was to report the quotes which were related to the purpose of the instrument and the investigated phenomenon, as recommended by Rubin and Rubin (2011) and Livingstone and Bovill (2013). Indeed, in the present study, the purpose was to measure the success of agile software development projects. The quotes from the experts are listed in the coming sections in no particular order.

6.2.7.1 General Feedback

A great deal of feedback was received from the interviewed experts concerning the point of who will answer the instrument's questions and how those individuals will be selected. Expert J and Expert K stated that "Identifying who will answer the questions is vital. Different perspectives could be obtained by different project's stakeholders regarding these questions. It is fitting to assess the

organisations in terms of how they could select the individuals who will answer the instrument's questions". On the same point, Expert F stated that "some questions should be anonymously answered. Otherwise, they will be afraid of encountering the top management and customers, or criticising them." Indeed, Expert H concluded that "having the authority to communicate with different stakeholders and having the authority and the information to answer these questions is important".

Expert G suggested restructuring the goals so as to have one main goal; this expert also suggested that all the remaining goals should be sub-goals of the main goal: "think of having one goal which is, for instance, to measure the success of agile software development projects. And then, all the remaining goals are sub-goals of this main goal". Expert G also provided some hints regarding the data collection during the case studies, stating that "it might be a good idea to capture some more qualitative inputs from the participants of the case studies. For example, for each scale metric, if they select a value, try to ask them why. This qualitative data will help you during the analysis and will generate a better understanding of the investigated issues".

Moreover, Expert L stated that "the main concern that I have is using the term project. In my experience, in agile we do not usually do projects. The project is a temporary task that will end. However, agility is more of a continuous work than a project. The focus is on building self-organising teams that will work on many projects. I recommend changing the purpose of the instrument from measuring the success of agile projects to measuring the success of the agility of a team or an organisation". Another piece of general feedback about the proposed instrument was given by Expert C, who stated that "planning is a very important aspect in agile projects and it contributes to the success. Make sure to have a goal or questions which are related to planning aspects, as I couldn't see any".

Expert J gave his thoughts on the five-point Likert scale metric which was used in the instrument, opining that "for the five-point Likert scale metric and the questions of how often, this is really very open and could be problematic. You may think of a new metric and be as specific as you possibly can be. This metric, in my opinion, is very open and needs to be reconsidered maybe". With regard to the percentile metric which was used in the instrument, Expert G put forth his opinion, stating that "for the percentile metrics, you need to decide how you will value the percentages. What will this valuing be based on? Are you going to make a comparison of the cases to see which is more successful? You need to have benchmarks. It is a brilliant idea that you mention you are going to use the ISO/IEC 15504 process assessment rating scale. If this is the case, make sure that this rating is presented to the participants of the case studies".

Both Expert J and Expert K, who worked in the same organisation, expressed their interest in using the proposed instrument: “the proposed instrument does overlap with some of our existing KPIs that we are using in our organisation and we will be interested in the instrument once it is ready to use or maybe merge it into our existing KPIs for the agile team”.

6.2.7.2 First Goal of the Instrument

The first goal of the proposed instrument is to improve the communication throughout the agile project. Under this goal, six questions and two metrics were proposed. Expert H put forth her opinion regarding the measuring of communication in agile projects, stating that “in my opinion, it is always the personnel aspects that may hinder the communication. For example, in pair programming practice sometimes two developers can’t work together. Psychology issues might make the communication difficult. Try a question that could measure these psychologies and personnel concerns”. For the purpose of clarifying the questions, Expert D suggested “give examples e.g. of what you mean by centralised repositories, so as to be clear with participants. Try to be specific regarding what you mean by informal communication. Give examples”.

An additional comment was received from Expert A regarding the first question: “very good question but what about own-developed platforms. In our case, we developed our own communication platform and we are using it in agile teams”. Expert C put forth his view on the importance of communication in agile teams, stating that “communication is essential for agile success; we use an automatic notification, whereby a system tells the team whenever a team member has done his/her job. In my experience, it improves the communication process. We also use GitHub and employ a Customer Relationship Management (CRM) system for code sharing and knowledge sharing which I believe improves the communication. For global teams, the communication is really a challenge and it is the most important aspect of success with agile in global development teams”.

Expert E stated that “the first communication way should be always face-to-face if it is possible. The communication during the agile project should not only be limited to sending emails and requests. Indeed, agile teams should also embrace discussions between the business users and the developers”. In addition to this, Expert B stated that “for question 2, be specific on what you mean by development’s aspects. For question 3, I believe even though that scrum encourages the daily meetings, what matters the most is the value behind these meetings, not only whether or not the team practice the daily meetings”. Expert I told of how communication in the agile environment is different: “the question of informal communication is really important and to the point of agile culture. How the team is communicating changing requirements, how they communicate risk aspects between the different stakeholders, and how they track this communication as well.

Tracking the communication is very important in agile projects. This is particularly so with business users and different stakeholders, since the agile project will involve many stakeholders and will foster the communication between them. As a result, tracking the communication in agile projects is a challenge".

6.2.7.3 Second Goal of the Instrument

The second goal of the instrument pertains to increasing the customer involvement. Under this goal, five questions and two metrics were proposed. Expert L expressed his thoughts on question 1 and question 2, stating that "I think measuring the involvement should not only be associated with attending the meetings. Instead, you might ask what is the percentage of the availability of the customers or customer representatives (e.g. product owner)?". Expert L also continued to give his feedback regarding the fourth question of the second goal: "be more specific. What do you mean by commitment here exactly? I would advise you to make sure that whoever is going to answer this question really understands the point".

Expert I stated "think of adding these questions to measure the involvement of the customers as I see them as important. Are business users able to understand or learn agile practices and techniques to work with the development team? Do the customers understand the technical debt and the technical details of the agile development and software development life cycles? This will increase their involvement as it will help them communicate with the development team". Moreover, Expert A stated that "these questions are good for the aim of measuring the customer's involvement in an agile project. If I could think of an additional question, it would be: How easy is it for the customer to express his/her needs?". Expert K also gave a statement regarding the difficulties in identifying the customers in agile projects: "the questions of customers' involvement is hard to answer in many cases based on my experience. The term customer is quite fuzzy. Sometimes you will have many different types of customers in the same agile project. Some of them are involved while some of them are isolated. You might think of a way to be more specific".

Expert G forwarded some suggestions: "with regard to questions 4 and 5, some rewording would be helpful. It is not clear enough to me what you mean by commitment and quality of customer involvement. Try to be clear and give examples; that will make it easy for the participants to understand these questions". Expert E also stated that "from my experience, customers are usually not interested in planning meetings. On the contrary, they are interested in the sprints demos. Moreover, keep in mind the changing of the requirements and how this affects the customers' involvement". Expert C expressed his view on how to be effective in measuring the involvement of the customers: "in my opinion, the most important part of an agile project where customers need to be involved is when the team is conducting the retrospectives. Are the customers involved in the

feedback loops and retrospectives? The retrospectives time is the time when the agile team will evaluate the progress and then act upon that. Indeed, it is a vivid time during which the involvement of customers will be key".

6.2.7.4 Third Goal of the Instrument

The third goal of the proposed instrument is to improve the training of the agile project team members. Under this goal, four questions and two metrics were proposed. Expert D provided suggestions to improve the accuracy of the questions: "I would suggest you reword the first question and use the term available training resources instead of available training. For the fourth question, try to give examples of self-training to ensure that the participants understand it clearly". In the same manner, Expert M stated that "the fourth question is a good question, especially in the agile context. The self-training is crucial to the success of agility. An additional question could be: Do you think having the agile certifications or scrum certifications will help in improving the training and the capabilities of the team?".

In terms of the responsibility of training in agile teams, Expert E indicated that "the team members should do assessments and they should ask for the needed training. Training in agile culture is the responsibility of the team, not the organisation. Therefore, I would advise that these questions be answered by the team members, not by the management of the organisation". The same point was raised by Expert A, who stated that "my suggestion on the training questions is to have two groups of respondents to the questions: team managers and team members. Because views will vary between these two groups". Moreover, Expert L stated that "the agile principles emphasise learning and continuous learning, not only training programmes. It is a good idea for you to capture that in the self-training. This is because the continuous training and learning will eventually lead to improvement of the team's capability and competences". Expert B also commented on the same manner: "the self-training and the learning process is very important in agile projects. The instrument may also ask about how often the management promote self-training to the team? Or how often they encourage the team to continue learning". Expert I expressed thoughts on how to increase the support from the leadership towards the training: "I was thinking of how agile training for the leadership of the organisations could increase their support and belief in agile projects? But I can see that the instrument covered this point in the fourth goal".

6.2.7.5 Fourth Goal of the Instrument

The fourth goal of the instrument is related to increasing the support from top management. Under this goal, six questions and three metrics were proposed. With regard to the support from top management, Expert F stated that "in question 3, the instrument is asking about the events initiated

by top management. Yet, I think this is problematic; what about if no events need to be initiated? The questions are very good and will measure the support of top management. But it might be a good idea for these questions to be anonymously answered. Otherwise, they will be afraid of encountering the top management". On the same point, Expert I stated that "regarding the first question, if the top management need to be attending, then you can measure their attendance. Same for the second question; the instrument should measure the cancelled meetings by the top management, not for various reasons that may occur during the project". In the same manner, Expert J stated that "for question 1 and question 2, try to capture not only the meetings. Think of measuring all aspects of the support. Attending the meetings only does not necessarily mean supporting the team".

Expert H drew conclusions on how the team could get the most from top management: "in agile, the project team will need to have direct access to the top management and the decision makers in the organisation. Such access could be measured by asking them to rate their access to the top management and the decision makers". Expert B expressed his thoughts on rewording some of the questions: "my suggestion is that for question 4, make the term commitment more specific. And for question 5 and question 6, I think the term challenges instead of issues will make the questions clearer". Expert A described one challenge that faces the top management in supporting agile projects: "in my experience, the fewer events you have initiated, the more efficient the agile team is. Indeed, particularly in multiple-location agile projects, the top management will face challenges in supporting the distributed teams and managing the different time zones".

Expert G provided his thoughts on question 3, stating "how are you going to measure the average? It is really hard for them to calculate the average and to be accurate on that. Think of using the always, often etc. five-scale metric that you are using in other questions and reword question 3 upon that". Expert G also continued to comment on question 4, stating that "it is again the term commitment. It could be understood differently without giving examples of what you are trying to measure here. Why not use the term overall support? And give examples of overall support means. For instance, budget, time, resources, etc." Expert E put forth his opinion on what support from top management in agile projects should be: "the most kind of support agile teams will need is the empowerment of the project team by the top management. Trust the team; more delegation of responsibilities and decision making should be given to the agile team".

6.2.7.6 Fifth Goal of the Instrument

The fifth goal of the proposed instrument is to enhance the delivery strategy during the agile project. Under this goal, five questions and five metrics were proposed. Expert B stated the following: "for question 1, the velocity will estimate the work required to complete a story point.

Yet, you are here not measuring the delivery but measuring the velocity of a team which is out of your scope". Expert G highlighted the same point: "for question 1 and question 2, if you are trying to enhance the delivery, then you shouldn't care much about the work required to finish story points. You may keep question 2 since the cycle time makes sense for enhancing the delivery. But I do believe that question 1 should be dropped". Expert I opined that "question 1 does not make any sense unless you want to compare two teams. Which is not the aim of the instrument. For question 2, make sure that the cycle time is calculated until the story point is delivered to the customer. In my experience, some teams end the cycle time once the development team has completed all the required work, so make sure the participants of the case studies are aware of this practice". Expert C emphasised the same idea: "the velocity of a team is not linked to how they deliver the working software. Make sure the story point cycle time is defined and agreed upon across the team. And they need to understand the question in order to answer it correctly".

With regard to the estimation, and how agile teams could estimate the work required for each story point as well as the time estimation for delivery, Expert H suggested the following: "accurate estimation could be reached if the team is working together for a long period of time. You could ask and measure how this point is affecting the estimation". Expert D expressed his thoughts on question 6, stating "I have not used the schedule progress index SPI before. Make sure that the participants of the case studies know the SPI. Otherwise, I believe the velocity could be used to answer this question". Expert E reinforced the point of getting rid of question 1 and also suggested new questions to be added to this goal: "question 1 will not measure the delivery of a story point; rethink it. An additional aspect that could speed up the delivery in agile teams is the number of automated tests. The ratio of manual tests to automated tests in the sprint, and how many of the story points are covered by automated tests". Expert L commented on question 4: "I have a concern regarding the fourth question. Why are you asking about releases that are on time? Remember the term on-time is not an agile term. Since changing requirements are expected, try to redesign the question or just remove it from the instrument".

6.2.7.7 Sixth Goal of the Instrument

The sixth goal of the proposed instrument is the appropriate selection of agile techniques, practices, and project management approach. Under this goal, six questions and two metrics were proposed. Expert L stated that "for question 1, it is good to consider the knowledge of the team when selecting the agile practice. The needs of the organisation should also be considered". With regard to the second question, Expert F questioned "why are you valuing adjusting the agile techniques more than following a specific technique? In the case of immature teams, I think sticking off the shelf will be the right choice in this case. Try to know the status of the team before valuing this question".

With regard to the second question, Expert G also stated: “If you ask yes/no question, you can’t have or option, so reword the question to be a one option question”.

Expert M, Expert L and Expert F all suggested that questions 3 and 5 be combined into one question and questions 4 and 6 as well. Expert L stated that “Question 4 and question 6 are quite similar, so you might need to rewrite them or maybe combine them into one question. The instrument in question 3 and question 4 asked about if the team were to conduct retrospectives to evaluate the selection, which is a very important point. Then you are repeating the point again in question 5 and question 6. No need for that”. This was reinforced by a statement from Expert A: “conducting retrospectives is very important for the agile team. It will help them evaluate their selection of the agile practices and their usage of selected practices. I think question 3 and question 4 are very important”.

With regard to the suggested additional questions, Expert C felt that it would be a good idea to ask “what is the influence of people (top management and customers) in selecting the agile techniques? Are they being given the choice to select these techniques and practices?”. Expert E also suggested some points that should be asked about in relation to this goal: “the point of how to improve the feedback cycles aspect by retrospectives, assessments, or evaluations. How to involve the customers and top management in all the feedback cycles. And how to encourage them to participate in these feedback cycles? What does the team do to automate the feedback cycles to improve the collaboration across the project team and with the customers and top management?”.

6.3 Instrument Review

This study applied semi-structured interviews encompassing open-ended questions and closed-ended questions to review the proposed instrument. The aims of these interviews were: firstly, to review and confirm the proposed instrument’s goals, questions, and metrics. Secondly, to suggest any other questions and metrics that need to be considered when measuring the success of an agile software development project. The experts’ interviews resulted in many suggestions for modifications to the proposed instrument. Following the received feedback, the instrument was revised and amended based on the feedbacks from the interviewed experts. These modifications ranged from some language and editing notes to additional questions and metrics to be added. The first version and the final version of the instrument after the validation from the interviewed experts are shown in Appendix E. The final version of the instrument consists of 6 goals, 30 questions, and 7 metrics.

6.4 Summary

This chapter focused on the process of developing and validating the proposed instrument. A literature review on the GQM approach was introduced in this chapter. This was followed by a discussion on the structure of the GQM approach and how the instrument will be built. The defining of the goals and how they were linked to the previously-identified success factors (identified in Chapter 2) was pointed out. Following this, the scoring of the instrument was discussed. The instrument was confirmed by applying semi-structured interviews with agile experts to review the proposed instrument. The criterion based on which the experts were selected was that each expert must have at least five years of experience with agile software development. Prior to conducting the experts' interviews, the interview plan itself was subjected to piloting to ensure the suitability of the questions to be put to the interviewees. The required size of the sample of experts to be interviewed was discussed, and it was found that at least 5-12 experts needed to be interviewed. In this study, 13 agile experts were interviewed to review and confirm the proposed instrument. This chapter also discussed the interview design and how the researcher contacted the experts. The reporting of the collected data during the interviews and the ethical considerations were then mentioned. The experts' background, as well as information about their organisations, roles, and experiences with agile were presented. The results and discussions of the data collected from the experts' interviews were examined in this chapter. The results were organised based on the goals of the instruments. After completing the experts' interviews, the instrument was revised based on the received feedback. The final version of the proposed instrument will be used in three different agile projects at three different organisations, as detailed in the next chapter.

Chapter 7 Case Studies

This chapter discusses the three conducted case studies. The reason behind conducting these case studies is to observe the practical use of the proposed instrument and to establish how the proposed instrument can be improved. A brief recap on how the proposed instrument was developed and validated is presented in the first section. Following this, the second section highlights the design of the case studies. Ethical considerations are discussed in the third section. The next section pertains to the first case study, and presents information about the organisation and the results and discussion. In the fifth section, the second case study is discussed, following which the third case study is presented in the subsequent section. The seventh section in this chapter discusses the observed practices which have an impact on the agile projects in the case studies. The two following sections present the evaluation of the proposed instrument from the participants and the researcher. The final section summarises the chapter.

7.1 Final Version of the Instrument

The proposed instrument was developed through several phases. By reviewing the literature, the critical success factors of agile software development were identified. Following this, the instrument was developed from these identified CSFs while employing the GQM approach. Once this had been accomplished, the instrument was reviewed by 13 agile experts, the aim being to validate and confirm the proposed instrument. Based on comments from the experts' interviews, the instrument was revised. The development of the proposed instrument is discussed in the previous chapter, while the final version of the instrument is provided in Appendix E. The case studies were conducted to examine the practical usage of the proposed instrument and to seek the participants' evaluation and suggestions, which could be used to enhance the proposed instrument.

7.2 Case study Design

Yin (2013) stated that a case study is helpful when 'why' or 'how' questions are being investigated in relation to a phenomenon over which the researcher has no control. The area of software engineering comprises many aspects, as does the software development carried out by people and organisations (Runeson and Höst, 2009). In particular, to investigate an agile development project, which is a people-centric development, the social questions must be answered fairly, together with the technical, process and organisational questions. Runeson and Höst (2009) provided a set of guidelines for conducting and reporting case studies in software engineering. These guidelines will be followed in the present study while reporting the case studies.

It is important to identify the context, the unit of analysis, and the goal prior to conducting a case study (Runeson and Höst, 2009). In this study, three agile software development projects in three different organisations were the context of the case study. Where the three units of analysis are the agile software development project in each case. The goal of the case studies was to use the proposed instrument to measure the success of the agile software development projects.

7.3 Ethical Consideration

The case studies conducted in this research were accepted by the ethical governance body of the School of Electronics and Computer Science at the University of Southampton. Ethical approval was granted with the reference number ERGO/FPSE/31342. All the participants were asked to sign a consent form prior to their participation in the case studies. More information about the ethical consideration can be found in Appendix D.

7.4 Organisation A

In this section, a brief overview of organisation A will be presented. Following this, the results of the instrument and the interviews will be detailed. The discussion will be structured based on the six goals of the proposed instrument. Following this, the participants' evaluation after using the instrument is discussed. Finally, suggestions to improve the instrument from organisation A are listed.

7.4.1 Organisation Profile

This is a government agency with more than 10,000 employees. The head office is located in Riyadh, and there are many branches across Saudi Arabia serving the citizens. This agency is responsible for implementing the rules and regulations and for planning for the labour market and dealing with workers' issues, complaints, and safety. Throughout the present thesis, this agency will be referred to as organisation A. The case study was hosted by the Information Technology IT department. The main point of contact with organisation A was the IT project management office (PMO) director. Indeed, he is responsible for the IT projects and was contacted about the instrument and the intention to study the status of agile software development projects in organisation A. During the case study, a total of 9 representatives from different departments at organisation A were met. All of these representatives were involved in the agile project. Said representatives included: development team members, quality assurance and testing team members, business analysts, customer representatives, and project managers.

7.4.2 The Results and Discussions

Organisation A scored 4.4 on a scale from 0 to 6 and independent scores for the instrument's items were presented to the organisation. A results meeting was held, during which the researcher briefed the attendees on the instrument scoring and how these scores could be interpreted. Following this, the participants of the meeting were given the chance to ask for clarifications or explanations, if needed. A detailed discussion of each goal and its score will be put forth in the following sections. A breakdown of the organisation's results is shown in Table 7-1. Figure 7-1 illustrates how the scores of the instrument's items varies for organisation A.

Table 7-1 The Instrument's Result for Organisation A

Instrument Item	**Score (0-1)**
Communication Goal	0.60
Customer Involvement Goal	0.80
Training Goal	0.65
Top Management Support Goal	0.93
Delivery Goal	0.66
Agile Practices, Techniques, PM Goal	0.76
Overall Score	**4.4**

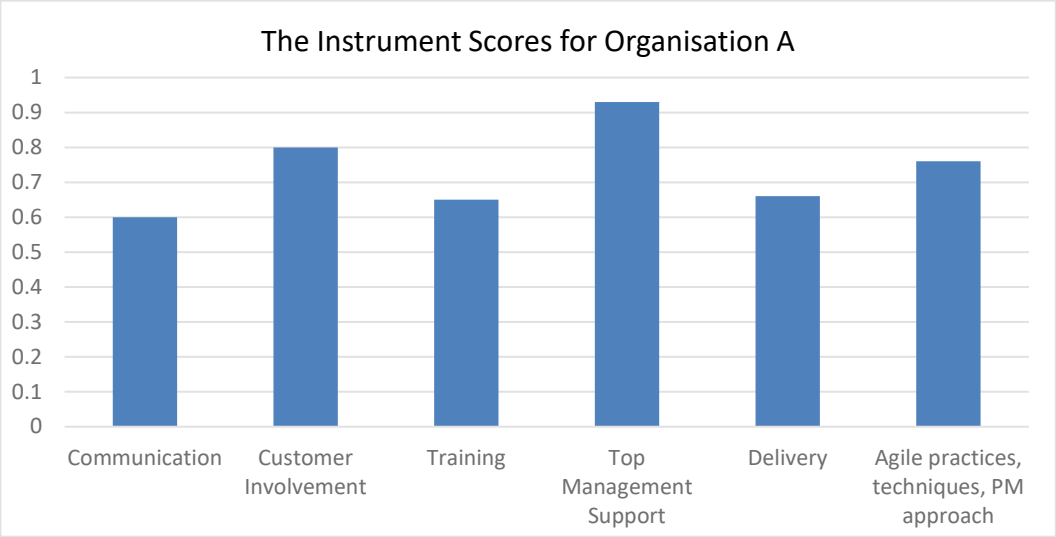


Figure 7-1 The Instrument Scores for Organisation A

7.4.2.1 Communication

With regard to communication, organisation A scored 0.60, which was the lowest score for organisation A among the other items of the instrument. This was due to three issues. The first is that organisation A does not use any platform or tool for communication. Emails are the only

available channel of communication across the team. The second issue is the absence of the daily meetings practice. The participants from organisation A admitted that daily scrum meetings would be beneficial for them. However, it is not possible for such meetings to be held every day, because the team is spread across two different locations and is too big to be involved in one meeting. The third issue is that there is no usage of a task board or smart board which is visible to all team members; indeed, this would increase the communication between the team members. As stated by a team member, "we kind of expected this result. We have many issues with the communication, not only with the agile projects, but also the whole communication process across the organisation".

As per the feedback from the participants, it can be concluded that the communication in organisation A is mainly formal communication, with face-to-face communication rarely occurring. The culture in organisation A encourages the employees to communicate formally using emails as the main method. This culture may be due to the fact that the organisation is a government agency and such employees tend to be more bureaucratic than private sector employees. The rigorous policy which is followed in this government agency leads to the employees preferring emails, since they are recorded and the employees want to protect themselves if any conflict occurs. Indeed, as stated by the project manager, "employees prefer to use emails for communication since the nature of our business is critical and they want to be always on the safe side". Retrospectives for knowledge sharing are used in organisation A, and this helps the team to communicate appropriately and enables the team members to remain constantly updated about the changes.

7.4.2.2 Customer Involvement

The score for the customer involvement item was 0.80. As stated by the PMO director, "overall, the customers are involved well in our projects and this is also reflected in the results using this instrument. It is always the case with IT projects in our organisation that business users and customers are interested. I think this is because there is a common agreement in our organisation about the importance of IT projects. This is due to many previous IT projects which transformed our organisation and the digitalisation efforts of our organisation, which has a tendency to digitalize the legacy processes. So, it creates an interest in technology and IT projects in our organisation".

The customers in this agile project at organisation A are split across two locations. Nevertheless, there are involved in the project. They participate in the planning meetings, and demos, and they respond quickly to the development queries. They help in resolving the issues which are facing the project team. In organisation A, the projects are usually initiated by business departments rather than the IT department. According to the project manager, "we see the IT as a service provider only. The business departments are the ones who start the projects and who own the data. And they

should be involved heavily during the implementation of these projects". The latter suggestion would allow the customers to be involved in the IT projects from the early stages until they go live. According to the project manager, agile projects hinder the involvement of customers because of the frequent meetings that are required. On the other hand, agile projects increase customer involvement when it comes to the frequency of delivery and the fostering of communication. Organisations attempting to be agile should pay attention to the many meetings required and how to avoid customers becoming distracted because of these meetings.

7.4.2.3 Training

The score for the training and team capability item at organisation A was 0.65 on a scale from 0 to 1. There were two main barriers that impacted the training in at organisation A. Firstly, the organisation is undergoing a transformation effort which involves many projects and initiatives. This means that the employees are very busy and involved in many tasks. The management tends to delay the training programmes because of the need for employees on the numerous projects. Secondly, there is a belief that, in agile, not much training is required, since it is based on skilled individuals. While the first barrier could be understandable and will only last for a certain period of time, the second barrier is an incorrect notion about agile. As time passes, the skilled individuals will need to update and increase their capabilities to avoid a knowledge gap that may occur after a period of time.

With regard to self-training, the answer to the question of how often self-training is practiced at organisation A was "seldom". The development team were asked why they are not investing in self-training to improve their skills. A team member answered, "we are having a busy schedule which does not give us time for self-training. Additionally, the culture of government organisations and public workers does not appreciate the improvement of skills as there are no sufficient reward programmes that encourage the employees to improve their capability compared with the private sector workers". Organisation A needs to recognise that the agility is a journey and not only a set of busy projects. Moreover, the organisation needs to give the employees more time for training and knowledge improvement. The rush and the eventful schedule might be reconsidered, as the organisation needs more time to invest in improving the employees' capability and knowledge.

7.4.2.4 Top Management Support

The top management support item scored 0.93, which was the highest score for organisation A among all items. The top management appreciate the digital transformation and aim to digitalise the organisation. Indeed, this creates support for the IT projects. According to the participants from organisation A, the top management support this agile project and are involved when necessary. In

the case of conflicts or development issues, they are there to help and to facilitate. As stated by the project manager, “we receive huge support from the top management. They support us and they provide us with all the required tools to achieve success. I think this is because the management have seen the many benefits of previous IT projects. Then, they believe that technology is critical to the organisation and they have pledged to support the IT projects and especially the agile projects”. It is assumed that the previous success of IT projects and the previous usage of new technologies at organisation A have given rise to the management’s interest in this agile project. Organisations which are in the early stages of digital transformation or agility efforts should know that in order to succeed with agile projects, support from top management is needed. Agile software development practices will require major changes in the way the organisations manage the projects, and these changes will need support from the management if they are to be successfully implemented. Agile practitioners should know that if they are doing well at their organisations and are successful in meeting the organisational needs, the top management will be supportive of them.

7.4.2.5 Delivery Strategy

The score for the delivery item for organisation A was 0.66. There was a major issue with the delivery in organisation A. This issue is that the sprint deliverables are not always delivered at the end of the sprint. On the other hand, the deliverables of the release are taken seriously. As stated by the project manager, “it was often that the delivery of some of the story points moved from one sprint to another. But once it comes to the delivery of the release, we work hard to deliver. This is because the release is a milestone and the project’s owners are expecting the release to be delivered. In contrast, the sprint’s delivery is more an internal manner that we used to be relaxed with”. It was also noticed that there were many frequently-planned deliverables at the start of the project, as advocated by Scrum. Yet, while the project was progressing, some deliverables were cancelled or postponed. The researcher was informed that cancelling or postponing the deliverables has always been discussed with the customers.

According to the responses of the participants from organisation A, the team acknowledged that agile principles advocate the early and frequent delivery of working software. Nevertheless, during the project they tended to reschedule some of the deliverables, as these might have been overestimated at the start of the project. It is important that organisations implementing agile projects ensure that such a rescheduling of the delivery is always coordinated with the customers. Customer satisfaction and the organisational needs should be the drivers of any changes in the schedule of the deliverables rather than only the development team conditions. While frequent

delivery is recommended by agile principles, the team should not overestimate its ability and plan for many deliverables that it cannot fulfil.

7.4.2.6 The Selection of Agile Practices, Techniques, and PM Approach

The selection of agile practices and PM approach item scored 0.76 for organisation A. The organisation follows the Scrum methodology, while also employing a mix of practices and techniques from other methodologies such as XP and Kanban. The company follows the agile project management framework which was proposed by the project management institute (PMI) and which is available in the Guide to the Project Management Body of Knowledge (PMBOK). According to the participants, organisation A is always considering the team's capabilities and familiarities when selecting these practices and techniques. Yet, the customers' needs are not considered when selecting the agile practices, techniques and the PM approach. When the researcher asked the participants about the reasons behind this, a team member answered: "customers are not constant. They are changing frequently from one project to another. We cannot change our practices regularly aiming to meet the customers' needs. We do consider our capabilities when selecting the practices; we think this is important and doable since the capabilities of the team are not changing frequently".

Indeed, organisations attempting to carry out successful agile projects should focus on the customers' needs and how to satisfy them, as per the agile values and principles. Therefore, the project team should work to change the employed practices and techniques in order to meet the customers' needs. Some of these practices may not require an enormous amount of effort and time. In that case, the project team should change said practices in order to align them with the customers' needs. If changing these practices and techniques would require more effort and result in delaying of the project, the decision should be taken by coordinating with the customers and bearing in mind the organisational needs.

With regard to organisation A, it seldom conducts retrospectives to review the selection of the agile practices and techniques during a project. As stated by the project manager, "we are busy during the project and we don't have the time to review our selected practices and change them if necessary. We do usually conduct such retrospectives by the end of the project". Regarding when to conduct the practices review retrospectives, this is a decision that the project team needs to take. Nevertheless, the project team should be sure that all the team members have the chance to give their feedback regarding the selected agile practices and the PM approach.

7.4.2.7 Final Thoughts and Suggestions

After finishing their experience with the instrument, the participants from organisation A were asked, during a meeting which they all attended, to evaluate the usage of the instrument. The PMO director stated that "the majority of the questions were understandable and easy to answer. Yet, to answer some of the questions we needed explanations from the researcher. Generally, the instrument is precise and to the point when it comes to agile principles. As we discussed during the meeting, we think that the instrument should measure the reporting during the project, and how the project team are reporting to customers, and management. The reporting is very important in the context of agile projects". The participants from organisation A agreed that the reporting matter is very important in their case and that was missed in the proposed instrument. They suggested that the instrument should measure the reporting mechanism and whether it delays or brings forward the progress of the project. According to the participants from organisation A, an appropriate reporting during the project to the customers would increase their involvement, while appropriate reporting to the top management would bring them into the picture and would motivate them to support the project.

7.5 Organisation B

In the present section, the profile of organisation B is briefly discussed. Following this, the results of the proposed instrument and the interviews will be presented. The discussion will be structured based on the items of the proposed instrument. Lastly, the participants' evaluation of using the instrument and their suggestions for improving the instrument will be examined.

7.5.1 Organisation Profile

This company was established in 2003 as a start-up company and specialises in business intelligence, data integration, and data warehouse solutions. Currently, the company has more than 150 employees across 3 offices in 3 different countries. The portfolio of the company is growing gradually and it has implemented several projects on big data, data governance, and mobile business intelligence. The company's goal is to become the market leader in providing data management services in Saudi Arabia, thus helping its clients to turning raw data into strategic assets. Throughout the present thesis, this company will be referred to as organisation B. According to one of the two co-founders of organisation B whom I met, "the agile track was a strategic track that we chose. From the early beginning, we wanted the company to be an agile company, where the agile principles are embraced. This is due to the fact that our business model is based on multiple projects that we should deliver to our clients; so, the delivery is a vital point in our business

model. Moreover, these clients assume that we should be at the cutting edge of technology and also on the way we manage the software development which is an agile method". During the case study, a total of six participants from organisation B were met.

7.5.2 The Results and Discussions

The results for the instrument and the total scores were presented to the organisation B representatives who attended the meeting. At the outset, the researcher started to brief the organisation on the results, what these results mean, and how the results could be interpreted. Following this, the participants were given the chance to ask for explanations of the results. Organisation B scored 3.9 in total and the independent scores for each goal were shared with the organisation. A breakdown of the organisation's results is shown in Table 7-2. Figure 7-2 illustrates how the scores of organisation B varies.

Table 7-2 The Instrument's Result for Organisation B

Instrument Item	**Score (0-1)**
Communication Goal	0.53
Customer Involvement Goal	0.76
Training Goal	0.45
Top Management Support Goal	0.50
Delivery Goal	0.93
Agile Practices, Techniques, PM Goal	0.76
Overall Score	**3.9**

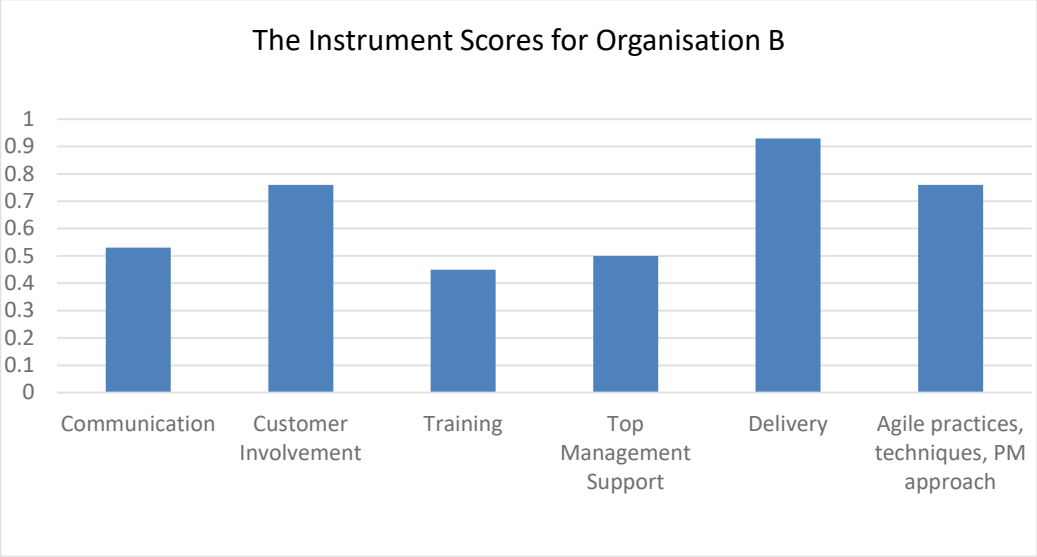


Figure 7-2 The Instrument Scores for Organisation B

7.5.2.1 Communication

When the organisation was asked about its communication difficulties in this agile project and why the score was only 0.53, the first issue raised was the lack of a communication platform or tool. As stated by a participant, “currently, we do not use any platform or tool to facilitate the communication. Emails are the main way to communicate across the team members”. Another aspect that might hinder the communication in organisation B is the absence of the daily stand-up meeting practice. In addition, there is no centralised repository which can be used for knowledge sharing and as a communication enabler during this agile project. Indeed, the project manager commented as follows: “I think we might consider using the centralised repositories in the future projects as they could help us to increase the agility of the team. We are aware of the daily stand-up practice but it is difficult to follow this in our case”.

With regard to the informal communication which is recommended by the agile manifesto and which is measured in the proposed instrument, the participants from organisation B agreed that informal communication is always used when applicable. When the participants were asked why this was the case, they responded with the following reasons: “we work in the same place, and the team is quite small, which makes face-to-face communication easy. And finally, the culture side; we think that the culture here in Saudi Arabia embraces the informal communication and face-to-face communication more. This is aligned with what agile principles advocate for the communication”.

The team was asked about the reasons why communication was difficult during this project, with a team member answering: “we faced some struggles in communication during this project. Especially communication with customers and the different stakeholders in the project. This is because stakeholders do not understand the technical details which sometimes hinders the communication”. This is a common issue that many technical projects are facing. The technical gap between the development team and the stakeholders should be minimised. This could be achieved by implementing two approaches: firstly, the development team should simplify the technical details and try to use simple terminologies that can be understood by non-technical stakeholders. Secondly, non-technical stakeholders should make some effort to understand the technical details. This would allow them to realise the constraints and to understand their needs and the organisational needs.

7.5.2.2 Customer Involvement

Organisation B scored 0.76 for the customer involvement item in their agile project. The participants were asked about the instrument’s assessment of customer involvement and about their justification of the result. The main reason which increased customer involvement, according

to the participants, was a practice followed by organisation B. This practice involves sending some of the project team members to the customer's location and letting them work with the customer in the same place. Said practice, according to the project manager, encourages the customer to become more involved and creates good relationships between the customers and the development team. A team member stated: "our way of implementing the projects is to work closely with the customers. In the case where a customer is in a different location, we do send some of the development team members to work there with the customer in the same location. This practice helps us in engaging the customers more and in encouraging them to respond to the development queries".

The customers' representative whom I met concluded that "being with the development team in the same place was really helpful. Additionally, the customers came to this project from a failure experience with an agile project. This project was implemented before, but it failed and the organisation stopped it. When the new project started, I think the customers were involved more in order to avoid the previous experience and they understood that their involvement with the development team is important". This statement could be interpreted as follows: if the development team is working with the customer in the same place, the involvement will be increased. Moreover, customers' experience with previous agile projects will increase their awareness of the importance of their involvement to the success of the agile projects.

7.5.2.3 Training

The training goal was measured in organisation B and scored only 0.45, which was the lowest reading of the instrument pertaining to organisation B. The co-founder of organisation B stated that "our philosophy of training is a little bit different. We hire highly-skilled employees and we trust them to practise self-training and to increase their knowledge. This is because our nature of work is very unstable and we work on many projects at the same time with a shortage of staff. We are weighing up this philosophy and it may change, as the company is now more stable". It was a strategic choice from the organisation to not spend more money on training and rather to hire highly-skilled employees. This strategy paid off in terms of saving time and staying within budget, which allowed organisation B to implement more projects and to serve more clients. On the other hand, and from the perspectives of the development team and the project manager, eventually the organisation will need to change its strategy regarding training; indeed, a skills gap has already started to appear, which will be overcome through more training programmes to be provided to the employees, and especially the technical team.

The self-training concept, which is recommended by agile principles, was followed in organisation B. However, this is not enough for large teams. After a period of time, and while the team is busy

with implementing new projects and running the business of the organisation, there emerges a need for training programmes to solve the skills gap which will be created over time. After the results of the instrument were shown to organisation B, the management stated that they might reconsider their strategy regarding training. They said that the organisation might explore some agile training that is needed for the team, along with technical training that is needed for the development team.

7.5.2.4 Top Management Support

The top management support item scored 0.50. The investigated agile project in organisation B was, according to the participants of the interviews, considered a mid-size project. Indeed, the project manager stated the following: “as you can see, we are suffering in this project from weak top management involvement and support. This is due to two reasons: firstly, this project is a bottom-up rather than a top-down project. This project was initiated by mid management in our company and the top management was not so much interested. Secondly, this project is a mid-size project in terms of scale and budget, and obviously will not capture the interest of the top management in our organisation”. According to the interviewees, this kind of project does not normally involve top management and will not capture their interest. On the contrary, a co-founder and managing director of the organisation stated that “the success of every project in our company is in the interest of the top management and it should always be like that”. Such a conflict of views between the project team and the top management indicates that there is an issue which needs to be fixed.

There should be agreement between the project team and the top management regarding how the support from the management should be and how to achieve it. The project team should understand that the top management are busy and will not be available most of the time. The project team should also work on how they present their project and ought to align it with the organisational strategy and objectives, which is in the interest of the top management. On the other hand, the top management should know that every project matters. Even if it is a small project in terms of scale and budget, the successful implementation of small projects will contribute positively to the culture of the organisation and to the experience of the team members.

One particularly interesting suggestion for the instrument was raised during the meeting with organisation B. It was suggested that the instrument should measure the conflict between the team and the management regarding how they see the support. The researcher responded to this suggestion, stating that the instrument could be filled individually and that some statistical tests could be applied to the response sample to calculate the average of the responses. Otherwise, the instrument might be filled during a workshop which invites all the stakeholders, during which they

could agree on to answer the questions on the instrument, which are related to the top management support.

7.5.2.5 Delivery Strategy

The delivery goal achieved the highest score on the instrument for organisation B, with a score of 0.93. This came from the organisation's long commitment to delivering frequently and efficiently. According to the project manager, "the most important benefit from being agile in software development is to be able to deliver more. Many aspects contribute positively to our ability to deliver. One aspect is that our management always stress the point that delivery should always be achieved and the customer are satisfied. Another aspect is related to the estimation process. During the estimation of the story points and the sprints backlogs, we plan for buffer and we aim to give the team more time. One point that might help the development team to keep these buffers for themselves is when the customers are not familiar with the technical details and they cannot assume what these tasks might require. Which is the case with us in this project".

As described by the project manager, there were two points which had an impact on the delivery in this agile project. The first point is that the organisation's management always put pressure on the delivery and they understand that adopting agile practices should allow them to deliver more. The second point is that the development team allow themselves buffers and a more tolerant estimation of the delivery in the project. The second point is only possible due to the non-technical background of the customers, who are happy with the estimation and the delivery respectively.

The team followed the Scrum methodology in terms of defining a backlog for each release and for each sprint. Although they used terminologies which differed from the Scrum terms, Scrum was clearly followed when defining the story points and for the delivery at the end of each iteration (sprint). The score was 0.93, which is an excellent score, and both the customers and the project team were happy about the delivery in this project. Nevertheless, it is recommended that organisation B not only seek to achieve more deliveries, but also more efficient delivery. Furthermore, the organisation should also be more realistic about the estimation, not only to help the development team with more time, but also to help both the customers and the team to achieve the success of the agile project in the shortest possible time. Certainly, this would lead to efficient delivery and more agility in the organisation.

7.5.2.6 The Selection of Agile Techniques, Practices, and PM Approach

With regard to the appropriate selection of agile techniques, practices, and PM approach goal, the score for organisation B was 0.76. The organisation used many practices from the Scrum methodology while some practices from the XP methodology were also employed. A development

team member stated the following: “we don’t follow a specific methodology or technique. We plan always to use a mix of practices which are available to us”. Organisation B is conducting retrospectives to discuss the selection of agile practices to ensure that they are aligned with the needs of the project and the organisational needs. These retrospectives, according to the project manager, help the organisation to tailor the agile transformation to its needs and to select the practices that it needs while also avoiding any practices that might create no value for the organisation.

The proposed instrument measures if an organisation considers the current knowledge and capabilities of the team when selecting agile techniques, practices and PM approach. At organisation B, the answer in this regard was “seldom”. According to the project manager, “if we are going to consider the knowledge of the team and the organisational needs during the selection of agile practices, it will be a complicated process to choose the practices”. Organisation B needs to understand that, even if involving the team in the selection process seems to complicate said process, it will nevertheless lead to the resolving of issues that will be created when selecting a practice or technique which the team are not familiar with. This kind of selecting will require more time and effort from the team to become familiar with the selected technique. Another important point here is that the organisation should always select agile practices and PM approach in order to carry out a successful project and meet the organisational needs, not in order to be more agile.

7.5.2.7 Final Thoughts and Suggestions

With regard to the evaluation of the instrument, and according to the participants from organisation B, the instrument was easy to fill in and no major clarifications were needed from the researcher. The questions were well written and easily understood. As stated by one team member in the meeting, “the questions were clear and we felt confident in answering them accordingly. One issue that we faced is that we don’t use Scrum. The instrument used many terms from Scrum, such as sprint and burn-down. So, there was a need to change these terms into terms we are familiar with so as to make the instrument clear for all the project team members”. The project manager followed up with: “I thought that the instrument was easy to fill in from our side, apart from one question which wasn’t really clear to us. In this case, the input from the researcher was used. This question was the fourth question on the delivery goal which was related to scheduling the performance index (SPI) metric. This we do not use in the project, but with further discussion we were able to answer the question eventually”.

7.6 Organisation C

In this section, the profile of organisation C will be presented. Following this, the results of the proposed instrument will be presented. The results and discussion will be structured based on the six items of said proposed instrument. Finally, the participants' evaluation of using the instrument and their suggestions on how to improve it are discussed.

7.6.1 Organisation Profile

This is a company with more than 2,000 employees and whose headquarters is located in Riyadh, Saudi Arabia. Its main business includes providing E-services, IT solutions and technology consultancy for both the private and public sector. It was established as a research and development company specialising in information technology in 1986. It has maintained a substantial growth and aims to continue this growth since the economy of Saudi Arabia is growing and the needs for E-services solutions are emerging. The company is now providing E-services, customised IT solutions, training solutions, governmental support services and consultations. It started as a governmental research and development body but underwent privatisation, and is now a private listed company. Throughout the present thesis, this company will be referred to as organisation C. During the case study, 10 participants from the organisation were interviewed.

7.6.2 The Results and Discussions

Organisation C scored 5.1 on a scale from 0 to 6, and the independent scores for each goal were presented to the organisation. A meeting was held, which participants from organisation C attended. Firstly, the researcher started to brief the attendees on the instrument scoring and how to interpret these results. Following this, the attendees were given the chance to ask for clarifications of the results, if needed. Further discussion on the results for each goal is presented in the coming sections. A breakdown of the organisation's results is presented in Table 7-3. Figure 7-3 shows how the instrument's scores for organisation C varies.

7.6.2.1 Communication

Organisation C used a platform to foster the communication between the different stakeholders of the agile project. As stated by the project manager, "we use the IBM Jazz tool as a communication channel between all the stakeholders involved in the project". This tool helped in facilitating the communication and also in bringing all the stakeholders into one place. With regard to using task boards, the participants from organisation C agreed that this is a practice recommended by Scrum. Nevertheless, they reported that this practice is rarely followed at organisation C. Informal

communication was always the method of communication when applicable. The daily stand-up meetings practice was followed whereby every day at 10 a.m., the team would meet for 15 minutes to discuss the project. The daily stand-up meeting was an informal meeting and attendance was not mandatory if some team members were busy. Organisation C was following Scrum, as stated by the project manager: “we use Scrum with a few customisations to fit our needs”.

Organisation C was using a tool to enable knowledge sharing for them. The tool is called Syncplicity, and is a file sharing service tool. A team member commented as follows: “we use Syncplicity for knowledge sharing which is a tool that enables us to achieve stable and convenience knowledge sharing across the project team”. A stable knowledge sharing throughout the project has helped the team to all be on the same page and will eventually help the team to communicate effectively. One of the senior management team from organisation C stated that “all these communication tools and more tools are available in our organisation, not only for agile projects, but also for all our employees. This is because we believe that the communication is very important within any organisation and such tools will help facilitate the communication”.

Table 7-3 The Instrument's Result for Organisation C

Instrument Item	**Score (0-1)**
Communication Goal	0.86
Customer Involvement Goal	0.72
Training Goal	0.90
Top Management Support Goal	0.83
Delivery Goal	0.87
Agile Practices, Techniques, PM Goal	0.92
Overall Score	**5.1**

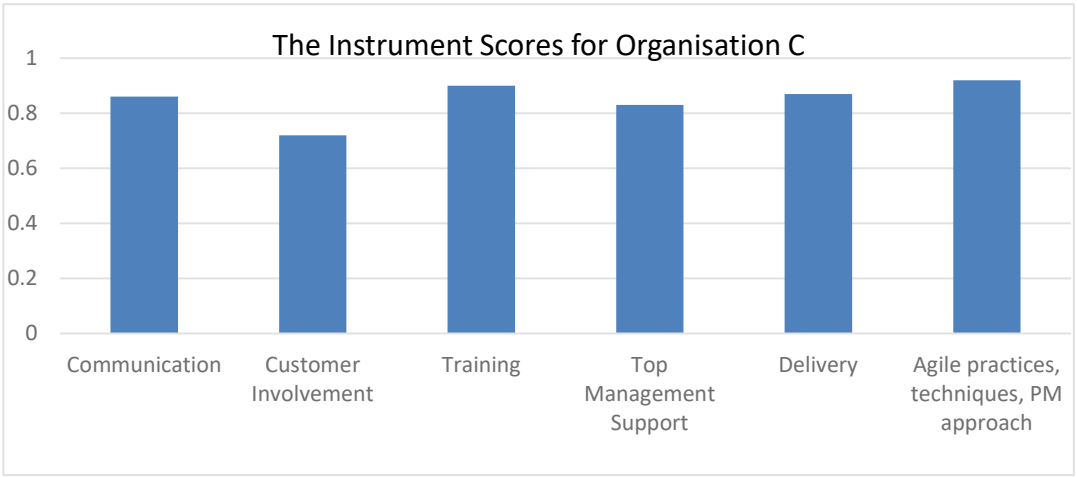


Figure 7-3 The Instrument Scores for Organisation C

7.6.2.2 Customer Involvement

On customer involvement, organisation C scored 0.72, which was the lowest score among the instrument's six items. During the results meeting, which involved representatives of the project team members, customers, and management of the organisation, the researcher asked why the customer involvement item scored the lowest. The project manager responded, stating that "if I could think of the reason behind this, it would be: the structure of our organisation, where the first point of contact between us and the customers is the account management team. The account manager who is responsible for the customers will act as the channel between us and the customers. This means that the customers do not communicate with the project team, but communicate with their account manager instead".

This issue of the account managers isolating the customers from the development team was an issue faced by many projects in organisation C. The organisation will implement a new structure by July 2018. The new structure is customer-centric, whereby a customer will have access to all of the employees who worked on the projects for this particular customer. A team member commented in this regard: "the role of account manager will only be before the start of the project (e.g. negotiations, offers, presentations). Once the project is agreed on between organisation C and the customer, a virtual team (a team which will be terminated by the end of the project) will be created. This team involves delegates from the development team, business analysts, and project managers. And the customer will contact the organisation directly through this team only". It is hoped that this new structure in organisation C will involve the customers more in the project and will eventually make said customers satisfied. Being a customer-centric organisation is the aim behind this new structure, and it will certainly help the organisation to be more agile.

7.6.2.3 Training

The score for the training item on the instrument was 0.90 for organisation C. This indicates that there exist training programmes which are provided to the employees. It also indicates that, according to the interviewees from organisation C, the training helps to improve the capability of the team and helps the organisation to achieve successful agile projects. As stated by the project manager, "our company values the training and every employee has the right to have at least two training programmes each year. Of course, this will be done by coordinating with his/her manager and it is subject to work circumstances". Organisation C has internal and external training programmes available to the employees. There is also a rewarding mechanism based on the evaluation of the employee to determine whether he/she will be eligible for the training.

With regard to self-training, organisation C was aware of the importance of self-training to enhance the capabilities of the individuals. In this regard, a team member stated that “the self-training concept is very important, especially in the context of agile. The issue we are facing is that there is no time. The team are always busy with work and spending some time for training or learning something new is difficult”. Generally speaking, the score of 0.90 is a good score and the interviewees were happy with the training. Organisation C might need to consider more time for the employees to practice self-training and more awareness of the importance of self-training. The managers need to pay attention to agile training and how this training will help their employees and help the organisation in its attempt to be an agile organisation.

7.6.2.4 Top Management Support

The top management support item scored 0.83. The project manager stated the following: “the management are always there to support. Although they do not bother with the details, they know the importance of the project and they support us in case further support from them is needed”. At organisation C, there is a belief in agility, and they want the company to always be agile. A senior manager from organisation C stated that “we see ourselves as an agile company, and we support any agile initiative as long as it is compatible with the organisational needs”. This was followed by a statement from one of the development team: “I think we have support from top management. But, I think from my experience, the support depends on the project. Big and important projects usually have the interest of top management more than small projects”.

The management of organisation C need to know that the success of every project is in the interest of the organisation, even if it is just a small-scale project. Indeed, the success of small projects will create a success story and will contribute to the culture of the organisation. The project team should work on how to present the project to the management, all the while avoiding the technical details that may disrupt their interest in the project. Overall, there was sufficient support from top management at organisation C, which was shown by the score; actually, this support contributes to the success of the project, as per the responses from the participants in this case study.

7.6.2.5 Delivery Strategy

With regard to the delivery item, the score of organisation C was 0.87. The project manager stated that “one of the most important reasons for us to follow agile practice is the ability to deliver fast and to deliver efficiently”. Several Key Business Indicators (KBIs) were used in organisation C to measure the delivery of each project. A team member opined that “the delivery deadlines are treated seriously in our company. Especially if the delivery is for external clients and if the client is an important client to us”. At organisation C, the importance of the delivery strategy is clear.

Consequently, the organisation has decided to adopt agile practices and techniques with the aim of being able to deliver more and to deliver as early as possible in an incremental way. Achieving a stable delivery strategy allows organisation C to engage its clients more and to build good relations with said clients. Moreover, it helps the organisation in its attempts to successfully adopt agile practices.

It was noticed that, at organisation C, the delivery of the release was always achieved on time. This was because of the management pressure and the defined KBIs on the release level that need to be reached. However, the delivery at the end of each sprint was not fully achieved. As mentioned during the meeting with the project team, some story points and deliverables might be postponed and moved from one sprint to another. However, at the end of each release, the planned story points for that release should be delivered. This helps organisation C to achieve a relaxed delivery of the sprints and strict delivery of the releases. According to the interviewees from organisation C, this assists them in satisfying their clients and meeting their internal KBIs and organisational needs.

7.6.2.6 The Selection of Agile Practices, Techniques, and PM Approach

The item pertaining to the selection of agile practices, techniques, and PM approach on the instrument was the highest item for organisation C, with a score of 0.92. The project manager was asked why he thought this item scored the highest on the instrument. He answered as follows: “although we seem to follow Scrum, we always select any practices and techniques from different methodologies that will help us. Selecting these practices and techniques is crucial to the success of the agile software development project in our organisation”. With regard to the project management approach, organisation C is following the agile project management framework, which was proposed by the project management institute (PMI) and which is available in the Guide to the Project Management Body of Knowledge (PMBOK® Guide). As stated by the project manager, “the project management approach was compatible with agile and it worked very well in our case and helped us to be more agile”.

Many retrospectives were held, whereby the project team discussed the selection of agile practices. During these retrospectives, the potential practices and techniques were discussed and the project team evaluated their selection of said practices. It is believed that continuous retrospectives during which the selection of agile practices will be assessed is a helpful measure that agile project teams should consider. Aligning the project management approach with the agile practices and removing any conflict that may appear is an important aspect when it comes to carrying out a successful agile project. Customer satisfaction and the organisational needs should be taken into consideration when selecting agile practices, techniques and PM approach. The same is also true for the project team’s familiarity with the selected practices, since this will save time and effort and will result in

better comprehension of the selected practices. Following one agile methodology or practice does not necessarily mean that the organisation must fully follow this methodology, although a decision on what to follow and what to avoid should be taken.

7.6.2.7 Final Thoughts and Suggestions

As stated by the project manager, “the instrument was easy to fill in, except for three questions which needed a clarification from the researcher. Some terminologies were different to what we normally use, but it was okay”. It was expected that some differences in the terminology of agile practices would be faced. This is due to the lack of unified terminologies in agile techniques and practices. Organisation C was asked to change the terminologies in the instrument to suit its case. Following this, the participants were asked if they could think of suggestions on how to improve the instrument. They raised two main suggestions: the first of which, was that the instrument should provide hints on how to answer the questions by giving examples. This will help the participants to clearly understand the question and then answer it. Secondly, they recommended that the instrument should provide some assessments on how they could improve the agility of their organisation based on their score and how they could do better. It was stated that this would help them to know their weaknesses and how they could improve them practically, as well as how to fruitfully adjust agile practices to achieve their needs.

7.7 Observed Practices

In this section, the practices which were followed in the investigated cases will be presented. The practices will be divided into two categories. The first section concerns the positive practices which, according to the participants of each case, have contributed to the success of agile projects. The second section pertains to the practices which, according to the interviewees, have hindered the agile projects in the case studies.

It is important to emphasize that these practices have been reported by the participants. The impacts of said practices are based on the researcher’s own judgement. Said impacts are driven by the scores of the organisations on the proposed instrument. These observed practices have impacted on how the studied organisations are managing their agile software development projects. The practices that contributed positively to the success of the agile software development projects in the studied cases are listed in Table 7-4.

With regard to the negative practices, these practises were reported by the participants from each case, while the impacts are based on the researcher’s own judgement. With regard to the third practice, namely the lack of agreement on the top management support, it is important to mention

that not all of the interviewees from organisation B agreed that this practice exists. Thus, there was a conflict of views on this practice between the participants of organisation B. It was decided that this practice would be included here for two reasons: firstly, the organisation's score on top management support was only 0.50, which indicates that there is a problem when it comes to the support from top management at organisation B. The second reason is that the researcher's own observation indicates that the third practice is followed at organisation B. The practices that hinder the success of agile software development projects in the studied cases are listed in Table 7-5

Table 7-4 The Observed Practices in the Case studies that impacted positively on the Agile Software Development Projects

Practice	Impact	Implemented at
1. Sending a group from the development team to work with the customer at the same place.	Increased customer involvement in the project.	Organisation B
2. Following the agile project management framework proposed by the PMI.	Aligned the project management approach with the agile practices and principles.	Organisations A, C
3. Defining and using measurements and KBIs to keep track of the delivery during the project.	Assured the frequent delivery of working software during the project.	Organisation C
4. Unifying the communication tools and platforms between all the different stakeholders of the project.	Simplified the communication and brought all the stakeholders of the project into one place.	Organisation C
5. Continuous reporting to the top management and the customers.	Preserved the interest of the top management and the customers.	Organisation A

7.8 Participants' Evaluation of the Instrument

At the end of each case study, the participants were asked to evaluate the instrument and to provide suggestions on how to enhance it. These suggestions ranged from notes about the language and the used terminologies, to deeper insights. The instrument was developed with no intention to be used as a stand-alone instrument. It was intended that the researcher will be there with the organizations when filling the instrument. Nevertheless, the participants' evaluation were somehow focusing on how the organization could solely be using the instrument. This indicates that some future work might concentrate on how to make the instrument more stand-alone instrument. In this section, the suggestions on how to improve the proposed instrument will be discussed. These suggestions will be helpful for those seeking to develop tools which can be used to measure the success of agile software development projects. A list of the suggestions on how to improve the instrument from the participants of the case studies is presented in Table 7-6.

Table 7-5 The Observed practices in the Case Studies that Hinder the Agile software Development Projects

Practice	Impact	Implemented at
1. The lack of communication tools and the usage of emails as the only communication channel.	Neglected communication efforts resulted in a usual delay of the development aspects.	Organisations A, B
2. The tendency to delay training programmes because the project team members are busy working on the project.	With time, a gap of knowledge and skills will be created and more training programmes will be needed.	Organisations A, B
3. The lack of agreement on how the support from top management should be between the project team and the top management.	Led to a decline in the involvement of the top management in the project.	Organisation B
4. Insufficient attention paid to scheduling of sprints deliverables (the deliverables being moved from one sprint to another within the same release) and to taking the release deliverables seriously.	Postponed the frequent delivery of working software which is leading to a decline in customer involvement.	Organisation A

7.9 Researcher's Evaluation of the Instrument

After completing the three case studies, the researcher evaluated the practical usage of the proposed instrument and how it could be improved. The investigated research question RQ4 was concerning to the development of an instrument with which the success of agile software development project could be measured. At this point, this research question has been addressed and the evaluation of the instrument is set to improve the proposed instrument and to put forth the direction for future work. A list of suggestions which should be considered was compiled by the researcher, with the aim of enhancing the usage of the instrument based on the experience from the three case studies. Some of the researcher's suggestions overlapped with some of the organisations' suggestions. The researcher's own suggestions to improve the proposed instrument are as follows:

- A recent study (Clarke et.al, 2018) reviewed the terminologies of agile methodologies and how they differ from one methodology to another. For instance, the term sprint in Scrum is equivalent to iteration, while the term sprint retrospective is equivalent to review meeting. According to Clarke et.al (2018), Scrum has achieved success in agile software development, although Scrum's distraction from the terminologies used in agile software development is possibly an undesirable aspect of the approach. This issue of terminological differences was faced in the case studies with the proposed instrument. The terminologies and notions used in the instrument

need to be more familiarised and unified. Alternatively, the instrument could be accompanied by an appendix containing more explanations of the used terminologies, while their equivalent terms in other methodologies could also be provided.

Table 7-6 Case Studies Participants' Evaluation of the Instrument

Suggestion	Reason	From
Some of the used terminologies pertain to a specific methodology such as Scrum. The instrument should make the terminologies very clear, so that the questions to be clear and concise. Alternatively, the instrument should leave it up to the participants to use their own familiar terminologies.	This will make it easy for participants to answer the questions. Additionally, it will eliminate any sort of possible misinterpretations which would affect the results of the instrument and which may not be what the participants intend to answer.	Organisation B
The proposed instrument should measure the reporting during the project, as well as how the project team are report to customers, and the top management. The instrument should also measure whether the reporting mechanism delays or brings forward the progress of the project.	An appropriate reporting during the agile project will increase customer and top management involvement in the project. The reporting in agile projects is important and contributes to the failure or success of the project; in fact, this was missed in the proposed instrument.	Organisation A
The proposed instrument should provide hints on how to answer the questions. This could be achieved by giving examples or by further explaining of the questions to be attached as an appendix to the instrument.	This will help the participants to clearly understand the questions and the purposes of these questions, thus allowing them to answer the instrument's questions accurately.	Organisation C
The proposed instrument should provide a set of assessments on how the organisations could improve their score after finishing the instrument. These assessments will help the organisations to improve the status of their agile software development projects.	This will help the organisations to know their weaknesses and how they could improve them so as to be more success with agile. It could also work as a guideline or a roadmap to achieving the success with agile projects.	Organisation C

- During the case studies, it was noticed that some of the participants found it difficult to answer some of the questions using the provided corresponding metric, and further explanation by the researcher was needed. Some future work may focus on revising the instrument to ensure that all of the employed metrics are clear to whomever uses the instrument. This could be achieved by introducing new metrics or by paying more attention to explaining the used metrics and presenting them clearly.

- In order to encourage the practitioners to use the instrument and to spend their time responding to the instrument's questions, the instrument should clearly explain how these

individuals will benefit from it. Such an explanation could take the form of a list of recommendations on how they could improve their agile projects and to show them their weaknesses and strengths based on their scores on the instrument.

- The final score of the instrument might be linked to some of the well-established technology process assessment models. For example, the ISO/IEC 15504 Software Process Improvement and Capability Determination (SPICE) has a process assessment model whose processes range from level 0 (incomplete) to level 5 (optimising). In this case, each item on the instrument will be a process which could be evaluated using the ISO/IEC 15504 model. According to (Lami and Falcini, 2009), ISO/IEC 15004 could be aligned to agile principles and many insights could be obtained using this process assessment model for measuring in an agile context. Future work on the instrument might focus on how to map the instrument scoring to this assessment model, the aim being to achieve a better acceptance of the scoring of the proposed instrument.

7.10 Summary

In this chapter, the three conducted case studies were discussed. Firstly, the context of the case studies was defined. The goal of conducting these case studies and the unit of analysis were also then defined. In this study, three agile software development projects in three different organisations were studied, the aim being to use the proposed instrument to measure the success of these projects. Each case study was presented alongside its results on using the instrument and the discussions and explanations of its scores for the investigated agile project. Following this, the practices observed in these organisations and the impact of said practices on the agile project were discussed. These practices were divided into two categories: positive practices which help the success of agile projects, and negative practices which hinder the agile projects.

One of the objectives of conducting the case studies was to seek evaluations and suggestions from participants which could be used to improve the proposed instrument. These evaluations and suggestions were presented in this chapter. Having completed the three case studies, and working with these organisations to use the instrument, the researcher gained a further understanding of the practical usage of the proposed instrument. This, in turn, allowed him to provide his own evaluation of the instrument and how it could be enhanced. The researcher's evaluation was discussed in this chapter. The conducted case studies discussed in this chapter made it possible to evaluate the instrument and to finally compile a list of suggestions on how to improve the instrument. These case studies also underlined topics which could be explored further in future work.

Chapter 8 Conclusion and Future Work

This chapter provides a summary of the present research and puts forth a conclusion based on the results and findings. In this chapter, the main contributions of the research are outlined and listed. This chapter also highlights the limitations of the research. The chapter then puts forth the directions for future work. This chapter is organised as follows. Section 8.1 presents the conclusion of the research. Section 8.2 recaps the contributions of this research. In section 8.3, discussion switches to the limitations of the current study. Section 8.4 examines potential future work. Section 8.5 reviews the related work. In section 8.6, an acknowledgment is provided. Finally, section 8.7 presents concluding remarks.

8.1 Conclusion

The last decade and more has produced a great deal of research focused on identifying the factors which influence the success of agile software development. This study aimed to conduct a review of research dated in the period of (2006-2016) concerning the critical success factors of agile software development. Eight previous studies were selected since they used empirical methodologies to validate the CSFs. The selected studies identified many success factors for agile software development. Of these factors, eight were selected in this research because they were identified by more than one study. The eight factors were delivery strategy, team capability and training, agile software development techniques, customer involvement, project management process, organisational culture, communication, and top management support.

Following this, it was important to explore the differences in the importance between the identified success factors and how the importance of these factors changed throughout the agile project phases. To achieve said goal, a survey was sent to agile practitioners so as to capture their experience with agile projects. The survey was designed while bearing in mind the purpose of evaluating the success factors' importance and the changes in the importance of these factors depending on the agile project's progress. The results from the survey were analysed using different statistical techniques.

The findings from the survey made it possible to order the identified CSFs of agile software development as follows: communication, customer involvement, team capability and training, top management support, organisational culture, agile software development techniques, delivery strategy, and project management process. The relations between the agile project's progress and the importance of the success factors were explored. This research found that the importance of

the success factor of project management significantly decreased whenever the agile project was progressing. Conversely, the results of the remaining factors were insignificant when testing the relationships between the agile project's progress and the importance of these success factors.

Factor analysis was applied to the CSFs of agile software development, the aim being to investigate the inter-relations among the CSFs. A principal component analysis technique was selected and resulted in the success factors being grouped into two components. The first component, termed organisational and people aspects, contained the following success factors: communication, customer involvement, team capability and training, top management support, and organisational culture. The second component is technical and project management aspects, contains the following success factors: delivery strategy, project management process, and agile software development techniques. The categorising of the success factors created an improved understanding of how these factors are related to each other.

Having investigated the critical success factors of agile software development, this research attempted to suggest measurements which could be used to assess and evaluate the successful implementation of agile software development projects. In this research, an instrument containing a set of metrics was developed following the GQM approach. The developed instrument was used to evaluate how the organisations succeed in meeting the success factors of agile software development. To validate the proposed instrument, interviews were conducted with 13 agile experts. During the experts' interviews, the instrument was shown and the experts' amendments were reflected in the final version of the instrument. Following this, the instrument was applied in three case studies. The aim of conducting the case studies was to obtain an evaluation of the usage of the instrument and to confirm its validity. The case studies allowed the researcher to observe practices that have an impact on the success of agile development projects. The case studies also made it possible for the researcher to identify a set of potential refinements to enhance the proposed instrument.

During this study, the researcher endeavoured to create empirical evidence which could make a contribution to the success of agile software development. A mixed methodology of quantitative and qualitative research was employed in this study. Several data collection methods were used. By the end of this study, the researcher had used a variety of research methods which helped him to understand the process of planning and producing scientific research. This study mainly contributes towards understanding the concept of successful agile development and how the success of implementing agile practices and techniques could be assessed and measured using the proposed instrument.

8.2 Contributions of the Research

The investigated research questions (see Section 1.1) were positively answered. Having answered the research questions and achieved the research objectives, this research made a set of contributions to the field of agile software development. In this section, the contributions made by this research will be reinforced. The main contributions of this research are recapped and listed below:

- Identifying the differences in the importance of agile software development success factors.
- Investigating the relationships between the progress of the agile project and the importance of the success factors.
- Categorising the success factors into two components using factor analysis.
- Proposing and validating of an instrument which could be used to evaluate the success of agile software development projects.

Taken together, this research identified the CSFs and investigated their importance, how the importance varies through the agile project, and their interrelations to provide insights to understand these CSFs. Furthermore, this research suggested a validated instrument to measure and evaluate the success in addressing these CSFs during an agile software development project.

8.3 Limitations of the Study

This study is not free of shortcomings. Indeed, this research takes the first step in identifying the CSFs of agile development and in developing a measurement instrument to evaluate the success of implementing agile software development projects. Despite this, however, more studies will be needed in order to create and validate precise measurements of agile software development success. This study opens the door for potential future work to improve the proposed instrument.

This research employed a survey with 131 responses to collect data regarding the CSFs of agile development. If time and accessibility constraints did not exist, a larger sample would have generated more confidence in the obtained findings. Indeed, this limitation might be resolved in future work if more data can be collected from a larger sample. The survey data came from Likert Scale items which contain non-interval data. Consequently, we assumed that the intervals between the points are approximately equal. Although that the applied statistical tests could be used with non-interval data as discussed in section 4.3.1, some might argue that this is a limitation of the study.

With regard to investigating how the importance of the CSFs varies among the agile project iterations, the survey asked the participants to report their progress on the iterations. This means that the survey investigated the importance in a cross-sectional manner. If the researcher had more time and resources, this could be changed to a longitudinal study to investigate how the importance varies from one iteration to another.

A component analysis was conducted in this research. A larger scale of data could confirm these components and be used to develop a structural equation model. During this study, self-reported data were collected by using survey and interviews. Such self-reported data might contain a potential source of bias which the researcher needs to be aware of and to recognise it as a limitation. With regard to investigating agile success, many factors are related to people, organisations, and culture. These factors are subjective. In the present research, the subjectivity was reduced as much as possible; however, as is the case with any human effort, it is difficult to remove subjectivity entirely.

The three case studies conducted in this research were all in one country which is Saudi Arabia. There might be some cultural causes that impact the obtained findings. It is a possible weakness of this research which could be addressed if the same instrument were used in a case study at organisations from different countries and different cultures. The validity of the research is discussed in the following sections:

8.3.1 Construct Validity

This research employed different data collection methods such as survey and interviews. It is possible that some participants might not understand the questions accurately. This show a limitation of the research and its findings. During the case studies, some of the instrument's items might not be very clear for some of the participants which allow for possible misinterpretation.

8.3.2 Internal Validity

This research conducted experts' interviews in which the researcher assumed that the participants had the experience with agile because they said they had and that they are actually using agile development. Despite the fact that this research stated that it relies on self-reporting data. This might be seen as a potential limitation of the results drawn from these interviews.

8.3.3 External Validity

The external validity and the generalising of the findings in this research could be improved if a larger sample size of the survey's participants could be achieved. Possible bias from participants toward agile development would have threatened the external validity of the research. It is also possible that the interviewed experts have a bias toward agile or that they understand agile differently. With regard to the case studies, the external validity could be enhanced if more cases from different countries were conducted. This might be the case with any empirical research methodologies. The researcher should be aware of this and to report it as a possible weakness of the research.

8.3.4 Conclusion Validity

This research employed qualitative techniques and used measurements which are subjective. This has benefits surely but it is important to acknowledge the fact that accuracy is hard to enforce. Consequently, the conclusions achieved as a result of these techniques could be inaccurate. The subjectivity of these techniques and measurements is one of the limitations of this research. The quantitative data collected by the survey were assumed to be approximately normally distributed and the intervals between the choices in the survey to be approximately equal in order to calculate the means and the standard deviations. This was assumed in order to apply the statistical tests. It has been debatable that such data could be used or not. Hence, it could be seen as a limitation of the findings obtained from the survey.

8.4 Future Work

This research has raised additional questions while answering the predefined research questions. This shows that the area of the success of agile software development is still in need of more research and that are yet many research gaps to be addressed. In this section, directions for future work will be highlighted. It is anticipated that other researchers might benefit from suggestions of potential future work which could be built on the results and findings obtained in this study.

Having recognised the differences in the importance between the CSFs of agile software development, there are now further questions which must be answered, as follows: What are the circumstances (circumstances of the organisations and perhaps of the individuals) that increase or decrease the importance of each CSF dependently? How could the organisations be aware of such circumstances? Indeed, it may be fitting for future work to focus on the results of the factor analysis in this study, which categories the factors into two components. More statistical tests might be applied with additional data sources to understand the latent reasons behind this categorising.

These reasons will contribute towards an understanding of how these success factors affect each other within the two components introduced in the present research. Indeed, future work could conduct the same survey again but add some qualitative questions to be answered by the participants regarding the CSFs. These qualitative data will make it easier to interpret the results of the linear regression and factor analysis. Moreover, future work might work on a larger scale of data to confirm the component analysis and to develop a structural equation model to confirm these components.

With regard to the proposed instrument, future work may focus on an extended version of the proposed instrument to reflect the evaluation received from the participants of the case studies. Future work might also shift the focus into how to assess the organisations to improve their agile development projects based on their score from the instrument. While the proposed instrument is successful in measuring the status of agile projects, it does not provide guidelines on how to improve the status of the agile development project. Although, providing guidelines to improve agile development projects was not one of the objectives of the present research; such guidelines, however, would be helpful and would encourage the organisations to use the instrument as per the feedback received during the case studies. Future work might revise the instrument to make sure that it could be used as a stand-alone instrument (to be used independent of the researcher) and to ensure that all of the metrics employed are clear for potential users. Moreover, future work may concentrate on how the resulting process of the instrument could be enhanced, such enhancement might be achieved by linking the scores of the instrument to more widely-accepted assessment models in the industry such as ISO/IEC 15504 SPICE model and the Agile Maturity Model introduced by Gartner (Norton, 2008). In this way, the scores of the instrument will gain more acceptance from the agile practitioners. Moreover, this will allow the organisations to compare their scores with other competitors by employing benchmarking.

8.5 Review of Related Work

This dissertation has been produced in three years. By the time this research is completed, there might be similar research conducted. In this section, a review of the related work will be discussed.

The research conducted in this thesis focused on the CSFs of agile software development and how to obtain a better understanding of these CSFs. Furthermore, to propose measurements with which the claim of agile success could be measured. There is other research which investigated the CSFs and worked on measuring the success in agile software development. The related research and how they are similar to or different to this research will be highlighted below.

The success in implementing agile practices was a concern for many researchers since the agile practices emerged, with the aim of obtaining agile practices benefits and mitigating their shortcomings. Some of these researchers looked into the success of agile software development by identifying the critical success factors and how to achieve these factors such as Chow and Cao (2008). While others investigated the success by identifying the risks of failure associated with agile development and how to avoid them such as Boehm and Turner (2003). It is just another way of defining the success with agile development. Both two ways helped in the understanding of the success in implementing agile practices.

Nerur, Mahapatra, & Mangalaraj (2005) identified the challenges associated with implementing agile software development in organisations. It is stated that those seeking to success with agile software development should be aware of these challenges. On the contrary, Cohn (2010) reviewed the success factors with agile development and proposed guidelines on how to succeed in adopting agile practices. These guidelines were categorised into three categories: Individuals guidelines, team guidelines, and organisation guidelines. These guidelines are useful for organisations at the early stages of agile adopting, but it does not provide an assessment or evaluation to measure the success. Similarly, the eight studies identified in the literature (section 2.4.2) investigated the success factors of agile development.

With regard to the related work pertaining to measuring of the success in agile development, Hartmann and Dymond (2006) stated that software measurements inherited from traditional software development methodologies might hinder organisations shifting to agile development. However, agile teams should work to adjust these measurements to align it with the new concepts introduced by agile methods or to develop new measurements which could be used with agile software development. Patel and Ramachandran (2009) developed agile maturity model (AMM). The AMM model is mainly built on the CMMI model tailored with some amendments to reflect agile practices. The main contribution of this model is to map the agile practices with the levels of the maturity model. While this model might be used to assess the status of agile practices in organisations, the success of the agile software development projects is lacking. The influence of CMMI on the agile models proposed in the literature is clear. Many of the proposed models are maturity models. Conversely, the instrument developed in this research focuses on measuring the success with agile development not on the maturity of doing agile development.

Stojanov, Turetken, & Trienekens (2015) developed a maturity model to provide a structural approach to increase the chance of success with agile software development. The model is working to define stages and to assess the organisations how to gradually improve the agile adoption from one stage to another via a suggested path. This model is based on the implemented agile practices

and how each practice is mature. While this model could be used to assess the practices independently, it fails to assess the whole agile project success. On the other hand, the developed instrument in this thesis assesses the whole agile project success and assess the success in addressing each success factor, but not each agile practice.

Soundararajan, Arthur, & Balci (2012) proposed a framework for assessing agile software development. The objectives, principles, and practices (OPP) framework provides linkages among the organisation's objectives and agile principles from the agile manifesto and the implemented agile practices. The OPP framework highlights the indicators for assessing each implemented agile practice. This framework provides a high-level assessment of agile practices and how these practices are linked with the organisational objectives.

Fontana et.al (2018) conducted a systematic review on the agile development maturity models and they compared these models aiming to develop a model which could evaluate the adoption of agile practices and techniques in an organisation. They found and reviewed fourteen models which assess the agile maturity, of which six models were introduced in the last four years. The most of these available models are built on a combination between agile principles and Capability Maturity Model Integration (CMMI). Fontana et.al (2018) recommended that future works should focus on empirical validation of these measurements which this research already achieved. While our instrument focused on measuring the success of implementing each success factor, these models were designed to measure the maturity of each agile practice independently. Both ways could be used to indicate the success in implementing agile software development projects.

Chita (2018) suggested that the activity theory, which is usually used in social science, could be used to assess the factors for successful agile software development implementation, since the agile success is built on organisational, cultural, and social factors. They developed a model following the activity theory for successful agile adoption and they indicated that their on-going research will validate this model by conducting a case study.

Laanti (2017) introduced a framework which could assess the agile transformation in large software development organisations. The framework classified the status of the organisations into five categories from beginner organisations to world-class organisations. While the framework assesses the agile adoption in the organisations, it lacks details on how organisations could evaluate their own adoption and how they could improve their agile transformation.

The findings obtained as a result of this dissertation and of the abovementioned emerging related work focused on understanding how the organisations could succeed in implementing agile practices. Furthermore, how this success could be measured and assessed. This show the needs for

more research that could work to validate these measurements and to introduce them to be used by agile software development practitioners.

8.6 Acknowledgment

Huge thanks and appreciations go to the organisations and the individuals that have been involved in this research answering the survey, actively participate in the interviews, and giving the researcher the ability to observe their agile projects. For their participation, experience, feedback, and knowledge without which this research could not be completed.

8.7 Final Remarks

Although empirical research tends to be challenging, it can create valuable findings when it is applied properly. The researcher found that statistical analyses are very powerful and promise to yield valuable results if the statistical tests are selected carefully and always aligned with the research questions and objectives. It was found that, in order to encourage organisations and individuals to take part in scientific research, researchers need to exert more effort to make their participation valuable not only for the research, but also for the participants themselves. Employing such an approach could lead to more engagement from the industry and further participation from practitioners. This would in turn increase the confidence of empirical studies in both the scientific and industrial communities.

List of References

Abrahamsson, P., Salo, O., Ronkainen, J. & Warsta, J. (2002) Agile software development methods: Review and analysis, *VTT publication* 478, Espoo, Finland.

Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. (2003). New directions on agile methods: a comparative analysis. In *25th International Conference on Software Engineering.* IEEE. pp. 244–254.

Acock, A. C. (2005). Working with missing values. *Journal of Marriage and Family*, *67*(4), pp. 1012–1028.

Agile State Report. (2018). 12th Annual State of Agile Report, State of Agile Report 2018, accessed May 06, 2018, https://explore.versionone.com/state-of-agile/versionone-10th-annual-state-of-agile-report-2.

Alahyari, H., Svensson, R. B., & Gorschek, T. (2017). A study of value in agile software development organizations. *Journal of Systems and Software*, *125*, pp.271-288.

Alqudah, M., & Razali, R. (2017). A comparison of scrum and Kanban for identifying their selection factors. In *6th International Conference on Electrical Engineering and Informatics (ICEEI),* Langkawi, Malaysia, IEEE, pp. 1-6.

Alzoubi, Y. I., & Gill, A. Q. (2014). Agile Global Software Development Communication Challenges: a Systematic Review. *Proceedings Pacific Asia Conference on Information Systems, PACIS 2014.* pp. 20-32.

Anand, R. V., & Dinakaran, M. (2016). Popular Agile Methods in Software Development: Review and Analysis. *International Journal of Applied Engineering Research*, *11*(5), pp. 3433-3437.

Banerjee, A., Chitnis, U. & Jadhav, S., (2009). Hypothesis testing, type I and type II errors. *Industrial psychiatry Journal, 18*(2) pp. 127-131.

Basili, V. R. (1992). *Software modeling and measurement: the Goal/Question/Metric paradigm* (technical report).

Basili, V. R., & Weiss, D. M. (1984). A methodology for collecting valid software engineering data. *IEEE Transactions on software engineering, 10*(6), pp.728-738.

Basili, V., Heidrich, J., Lindvall, M., Munch, J., Regardie, M., Trendowicz, A. (2007). GQM^+ Strategies--Aligning Business Strategies with Software Measurement. In *First International Symposium on Empirical Software Engineering and Measurement, 2007. ESEM 2007*. pp. 488–490.

Basili, V.R., Caldiera, G., Rombach, H.D. (1994). Goal Question Metric Paradigm. In *Encyclopedia of Software Engineering, 1*, pp. 528-532.

Beck, K. (1999). Embracing change with extreme programming. *Computer*, *32*(10), pp. 70–77.

Beck, K. (2000). *Extreme programming explained: embrace change*. addison-wesley professional.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., and others. (2001). Manifesto for agile software development.

References

Begel, A., & Nagappan, N. (2007). Usage and perceptions of agile software development in an industrial context: An exploratory study. In *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*. pp. 255–264.

Berander, P., Jönsson, P. (2006). A goal question metric based approach for efficient measurement framework definition. In Proceedings of *the 2006 ACM/IEEE International Symposium on Empirical Software Engineering*, pp. 316–325.

Berg, B. L. (2004). *Qualitative research methods for the social sciences*. 5th Edition, Pearson.

Bhattacherjee, A. (2001). Understanding information systems continuance: an expectation-confirmation model. *MIS Quarterly*, Vol. 25, No. 3, pp. 351–370.

Bhattacherjee, A. (2012). Social science research: Principles, methods, and practices. *Textbooks Collection 3, University of South Florida.*

Birk, A., Van Solingen, R., Jarvinen, J. (1998). Business impact, benefit, and cost of applying GQM in industry: an in-depth, long-term investigation at Schlumberger RPS. *Proceedings of Fifth International Software Metrics Symposium. Metrics.* IEEE, pp. 93-96.

Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, *21*(5), pp. 61–72.

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

Boehm, B., & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE Software*, *22*(5), pp. 30–39.

Boston Consulting Group. (2017). Taking Agile Way Beyond Software. [online] Available at: https://www.bcg.com/publications/2017/technology-digital-organization-taking-agile-way-beyond-software.aspx [Accessed 5 Sep. 2018].

Braun, V. & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*(2), pp.77–101.

Bustard, D., Wilkie, G., & Greer, D. (2013). The maturation of agile software development principles and practice: Observations on successive industrial studies in 2010 and 2012. In *20th IEEE International Conference and Workshops on the Engineering of Computer Based Systems (ECBS 2013),* IEEE, pp. 139-146.

Bytheway, A. J. (1999). Guest Editor's Introduction: Successful Software Projects and How to Achieve Them. *IEEE Software*, *16*(3), pp. 15.

Campanelli, A. S., & Parreiras, F. S. (2015). Agile methods tailoring–A systematic literature review. *Journal of Systems and Software*, *110*, pp.85-100.

Carifio, L., & Perla, R. (2008). Resolving the 50 year debate around using and misusing Likert scales. *Medical Education*, *42*(12), pp. 1150–1152.

Chau, P. Y. K. (1996). An empirical assessment of a modified technology acceptance model. *Journal of Management Information Systems*, *13*(2), pp. 185–204.

Chita P. (2018). Agile Software Development – Adoption and Maturity: An Activity Theory Perspective. In: Garbajosa J., Wang X., Aguiar A. (eds) Agile Processes in Software Engineering and Extreme Programming. XP 2018. Lecture Notes in Business Information Processing, vol 314. Springer, Cham, pp. 160-176.

Chiyangwa, T. B., & Mnkandla, E. (2017). Modelling the critical success factors of agile software development projects in South Africa. *South African Journal of Information Management*, *19*(1), pp.1-8.

Chow, T., & Cao, D.B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, *81*(6), pp. 961–971.

Clarke, P., O'Connor, R. V., & Yilmaz, M. (2018). In search of the origins and enduring impact of Agile software development. In *Proceedings of the 2018 International Conference on Software and System Process ICSSP 18,* ACM. pp. 142-146.

Cockburn, A. (2004). *Crystal clear: a human-powered methodology for small teams*. Pearson Education.

Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer*, *34*(11), pp. 131–133.

Cohen, J. (1992). Quantitative Methods in Psychology A power primer. *Psychological bulletin*, *112*, pp. 1155-1159.

Cohn, M. (2010). *Succeeding with agile: software development using Scrum*. Pearson Education.

Conboy, K. (2009). Agility from first principles: Reconstructing the concept of agility in information systems development. *Information Systems Research*, *20*(3), pp. 329–354.

Costello, Anna B. & Jason Osborne (2005). Best practices in exploratory factor analysis: four recommendations for getting the most from your analysis. *Practical Assessment Research & Evaluation, 10*(7), pp. 1-9.

Creswell, J. W. and Clark, V. L. (2011). *Designing and Conducting Mixed Methods Research*. SAGE Publications.

Creswell, J. W. (2003). *Research design - Qualitative, Quantitative and Mixed Methods Approaches*. Second Edi. SAGE Publication.

Davis, A. M., Bersoff, E. H., & Comer, E. R. (1988). A strategy for comparing alternative software development life cycle models. *IEEE Transactions on Software Engineering*, *14*(10), pp. 1453–1461.

Davis, F. D. (1985). *A technology acceptance model for empirically testing new end-user information systems: Theory and results* (phd thesis). Massachusetts Institute of Technology.

Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1989). User acceptance of computer technology: a comparison of two theoretical models. *Management Science*, *35*(8), pp. 982–1003.

References

Dhir, S., Kumar, D., & Singh, V. B. (2019). Success and Failure Factors that Impact on Project Implementation using Agile Software Development Methodology. In *Software Engineering Advances in Intelligent Systems and Computing, 731*, Springer, pp. 647-654.

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, *85*(6), pp. 1213–1221.

Dvir, D., Lipovetsky, S., Shenhar, A., & Tishler, A. (1998). In search of project classification: a non-universal approach to project success factors. *Research Policy*, *27*(9), pp. 915–935.

Dybå, T., & Dingsøyr, T. (2009). What do we know about agile software development? *IEEE Software*, *26*(5), pp. 6–9.

Dybå, T., Kampenes, V. B., & Sjøberg, D. I. K. (2006). A systematic review of statistical power in software engineering experiments. *Information and Software Technology*, *48*(8), pp. 45–755.

Field, A. (2009). *Discovering statistics using SPSS*. Sage publications.

Fontana R.M., Albuquerque R., Luz R., Moises A.C., Malucelli A., Reinehr S. (2018). Maturity Models for Agile Software Development: What Are They? In: *European Conference on Software Process Improvement EUROSPI 2018*. Communications in Computer and Information Science, vol 896. Springer, Cham, pp. 3-14.

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), pp. 620-625.

Grant, J. & Davis, L., (1997). Selection and use of content experts for instrument development. *Research in nursing & health, 20*(3), pp. 269-274.

Gray, A., MacDonell, S.G. (1997). GQM++ a full life cycle framework for the development and implementation of software metric programs. In *Proceedings of ACOSM'97 Fourth Austrailian Conference on Software*, IEEE, pp. 22–35.

Guadagnoli, E., & Velicer, W. F. (1988). Relation to sample size to the stability of component patterns. *Psychological Bulletin*, *103*(2), pp. 265-275.

Guest, G., Bunce, A. & Johnson, L. (2006). How Many Interviews Are Enough? An Experiment with Data Saturation and Variability. *Family Health International*, *18*(1), pp.59–82.

Hair, J. F., Black, W. C., & Babin, B. J. (2010). *Multivariate Data Analysis: A Global Perspective*. Pearson Education.

Hartmann, D., & Dymond, R. (2006). Appropriate agile measurement: using metrics and diagnostics to deliver business value. *Agile Conference AGILE2006,* pp. 134-139.

Heidenberg, J., Weijola, M., Mikkonen, K., & Porres, I. (2013). A metrics model to measure the impact of an agile transformation in large software development organizations. In *International Conference on Agile Software Development XP 2013*, Springer, Berlin, Heidelberg, pp. 165-179.

Highsmith, J. A. (2002). *Agile software development ecosystems*. Addison-Wesley Professional, Pearson.

Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *Computer*, *34*(9), pp. 120–122.

Iivari, J., & Iivari, N. (2011). The relationship between organizational culture and the deployment of agile methods. *Information and Software Technology*, *53*(5), pp. 509–520.

Jørgensen, M., Dybå, T., Liestøl, K., & Sjøberg, D. (2015). Incorrect results in software engineering experiments: How to improve research practices. *Journal of Systems and Software, 116,* pp. 133-145.

Kaiser, H. F. (1960). The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, *20*(1), pp. 141–151.

Kaiser, H. F. (1974). An index of factorial simplicity. *Psychometrika*, *39*(1), pp. 31–36.

Kelle, E. Van, Visser, J., Plaat, A., & Wijst, P. Van Der. (2015). An Empirical Study into Social Success Factors for Agile Software Development. *2015 IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering*, pp. 77–80.

Kettunen, P., & Laanti, M. (2008). Combining agile software projects and large-scale organizational agility. *Software Process: Improvement and Practice*, *13*(2), pp. 183–193.

Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, *33*, pp. 1–26.

Kitchenham, B. A., Pfleeger, S. L., Pickard, L. M., Jones, P. W., Hoaglin, D. C., El Emam, K., & Rosenberg, J. (2002). Preliminary guidelines for empirical research in software engineering. *IEEE Transactions on Software Engineering*, *28*(8), pp. 721–734.

Kitchenham, B., & Pfleeger, S. L. (1996). Software quality: the elusive target. *IEEE Software*, *13*(1), pp. 12–21.

Kloppenborg, T. J., Manolis, C., & Tesch, D. (2009). Successful project sponsor behaviors during project initiation: an empirical investigation. *Journal of Managerial Issues*, *21*(1), pp. 140–159.

Kulathunga, D., & Ratiyala, S. D. (2018). Key Success Factors of Scrum Software Development Methodology in Sri Lanka. *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, *45*(1), pp.234-252.

Kunz, M., Dumke, R. R., & Zenker, N. (2008). Software metrics for agile software development. In *19th Australian Conference on Software Engineering (ASWEC 2008)*, IEEE. pp. 673–678.

Laanti, M. (2017). Agile transformation model for large software development organizations. In *Proceedings of the XP2017 Scientific Workshops*, ACM, pp. 19.

Lami, G., & Falcini, F. (2009). Is ISO/IEC 15504 Applicable to Agile Methods? In *International Conference on Agile Processes and Extreme Programming in Software Engineering XP 2009,* Springer, pp. 130-135.

Lavazza, L., Frumento, E., Mazza, R. (2015). Defining and evaluating software project success indicators: A GQM-based case study. In *10th International Joint Conference on Software Technologies (ICSOFT)*. pp. 1–12.

Leeuw, E. D., Hox, J., & Dillman, D. (Eds.). (2012). *International Handbook of Survey Methodology*. Routledge.

References

Leidecker, J. K., & Bruno, A. V. (1984). Identifying and using critical success factors. *Long Range Planning*, *17*(1), pp. 23–32.

Livermore, J. A. (2007). Factors that impact implementing an agile software development methodology. In *Proceedings of 2007 IEEE SoutheastCon,* Richmond, VA,USA. pp. 82–86.

Livingstone, S., & Bovill, M. (2013). *Children and their changing media environment: A European comparative study*. Routledge.

Marshall, B., Cardon, P., Poddar, A., & Fontenot, R. (2013). Does sample size matter in qualitative research?: A review of qualitative interviews in IS research. *Journal of Computer Information Systems*, *54*(1), pp. 11-22.

Mason, M. (2010). Sample size and saturation in PhD studies using qualitative interviews. *Forum: qualitative Social Research Sozialforschung, 11*(3), pp. 1-19.

Matell, M. S., & Jacoby, J. (1971). Is there an optimal number of alternatives for Likert scale items? Study I: Realibility and Valdity. *Educational and Psychological Measurement*, *31*(3), pp. 657–674.

Miles, MB. & Huberman, MA. (1994). *Qualitative Data Analysis: An Expanded Sourcebook*. Second edition. Thousand Oaks, CA: Sage Publications.

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

Mundfrom, D. J., Shaw, D. G., & Ke, T. L. (2005). Minimum sample size recommendations for conducting factor analyses. *International Journal of Testing*, *5*(2), pp. 159–168.

Nakagawa, S., & Cuthill, I. C. (2007). Effect size, confidence interval and statistical significance: a practical guide for biologists. *Biological Reviews*, *82*(4), pp. 591–605.

Nerur, S. ., & Balijepally, V. (2007). Theoretical reflections on agile development methodologies. *Communications of the ACM*, *50*(3), pp. 79–83.

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, *48*(5), pp. 72–78.

Neuman, W. L. (2012). *Basics of social research: Qualitative and quantitative approaches*. Second Edition, Pearson.

Nielsen, J. (2000). Why you only need to test with 5 users: Alertbox. Retrieved Feb 01, 2018 from http://www.useit.com/ alertbox/20000319.html.

Norman, G. (2010). Likert scales, levels of measurement and the "laws" of statistics. *Advances in health sciences education*, *15*(5), pp. 625-632.

Norton, D. (2008). The Current State of Agile Method Adoption. *Gartner Research Report*.

Olszewska, M., Heidenberg, J., Weijola, M., Mikkonen, K., & Porres, I. (2016). Quantitatively measuring a large-scale agile transformation. *Journal of Systems and Software, 117*, pp. 258-273.

Padmini, K. V. J., Bandara, H. M. N. D., & Perera, I. (2015). Use of software metrics in agile software development process. In *Moratuwa Engineering Research Conference (MERCon),* pp. 312–317.

Palmer, S. R., & Felsing, M. (2001). *A practical guide to feature-driven development*. Pearson Education.

Parviainen, P., Jarvinen, J., Sandelin, T. (1997). Practical experiences of tool support in a GQM-based measurement programme. *Software Quality Journal, 6*(4), pp.283–294.

Patel, C., & Ramachandran, M. (2009). Agile maturity model (AMM): A Software Process Improvement framework for agile software development practices. *International Journal of Software Engineering, IJSE*, *2*(1), pp. 3-28.

Patton, I. (2008). Ambiguous business value harms software products. *IEEE Software*, *25*(1), pp. 50-51.

Patton, M. Q. (1990). *Qualitative evaluation and research methods.* SAGE Publications, Inc.

Pearson, E. S. (1931). The analysis of variance in the case of non-normal variation. *Biometrika*, *23*(1/2), pp. 114– 133.

Pelrine, J. (2011). On Understanding Software Agility—A Social Complexity Point Of View. *Cognitive-Edge.Com*, *13*, pp. 26–37.

Petersen, K., Wohlin, C., & Baca, D. (2009). The waterfall model in large-scale development. In *International Conference on Product-Focused Software Process Improvement*, Springer, Berlin, Heidelberg, pp. 386-400.

Pfleeger, S. L., & Kitchenham, B. A. (2001). Principles of survey research: part 1: turning lemons into lemonade. *ACM SIGSOFT Software Engineering Notes*, *26*(6), pp. 16–18.

Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., & Still, J. (2008). The impact of agile practices on communication in software development. *Empirical Software Engineering*, *13*(3), pp. 303–337.

Pikkarainen, M., Salo, O., Kuusela, R., & Abrahamsson, P. (2012). Strengths and barriers behind the successful agile deployment—insights from the three software intensive companies in Finland. *Empirical Software Engineering*, *17*(6), pp. 675–702.

Pituch, K., & Stevens, J. (2016). Applied Multivariate Statistics for the Social Sciences. 6th Edition, New York: Routledge.

Rawsthorne, D. (2004), Managing the Work in an Agile Project. Net Objectives, Bellevue, http://www.netobjectives.com/files/resources/downloads/ManagingTheWork.pdf, (Accessed 02/08/2016).

Rice, W. R. (1989). Analyzing tables of statistical tests. *Evolution, 43*(1), pp. 223–225.

Rising, L., & Janoff, N. S. (2000). The Scrum software development process for small teams. *IEEE Software, 17*(4), pp. 26-32.

Royce, W. W. (1987). Managing the development of large software systems. In *Proceedings of the 9th international conference on Software Engineering*, IEEE Computer Society Press, pp. 328-338.

References

Rubin, H. J., & Rubin, I. S. (2011). *Qualitative interviewing: The art of hearing data*. Sage.

Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering, 14*(2), pp. 131–164.

Shameem, M., Kumar, C., Chandra, B., & Khan, A. A. (2017). Systematic Review of Success Factors for Scaling Agile Methods in Global Software Development Environment: A Client-Vendor Perspective. In *Software Engineering Conference Workshops (APSECW), 2017 24th Asia-Pacific,* IEEE, pp. 17-24.

Shaw, M. (2002). What makes good research in software engineering? *International Journal on Software Tools for Technology Transfer*, *4*(1), pp. 1–7.

Sheffield, J., Lemétayer, J. (2013). Factors associated with the software development agility of successful projects. *International Journal of Project Management*, *31*(3), pp.459–472.

Shenhar, A. J., Dvir, D., Levy, O., & Maltz, A. C. (2001). Project success: a multidimensional strategic concept. *Long Range Planning*, *34*(6), pp. 699–725.

Sjøberg, D. I. K., Dybå, T., & Jørgensen, M. (2007). The Future of Empirical Methods in Software Engineering Research. In *2007 Future of Software Engineering,* IEEE, pp. 358–378.

Soundararajan, S., Arthur, J. D., & Balci, O. (2012). A methodology for assessing agile software development methods. *Agile Conference AGILE2012,* pp. 51-54.

Stankovic, D., Nikolic, V., Djordjevic, M., & Cao, D.B. (2013). A survey study of critical success factors in agile software projects in former Yugoslavia IT companies. *Journal of Systems and Software*, *86*(6), pp. 1663–1678.

Stelzmann, E., Kreiner, C., Spork, G., Messnarz, R., & Koenig, F. (2010). Agility meets systems engineering: a catalogue of success factors from industry practice. In *European Conference on Software Process Improvement EUROSPI 2010*, pp. 245–256.

Stojanov, I., Turetken, O., & Trienekens, J. J. (2015). A maturity model for scaling agile development. *41st Euromicro Conference on Software Engineering and Advanced Applications (SEAA),* pp. 446-453.

Strode, D. E., Huff, S. L., & Tretiakov, A. (2009). The impact of organizational culture on agile method use. In *42nd Hawaii International Conference on System Sciences,* IEEE, pp. 1–9.

Sutherland, J., & Schwaber, K. (2017). The Scrum guide. the definitive guide to Scrum: The rules of the game. *ScrumGuides.com*.

Tabachnick, B. G., & Fidell, L. S. (2007). Using multivariate statistics. 5th Edition, Boston, MA, : Allyn & Bacon/Pearson Education.

Teddlie, C. and Tashakkori, A. (2010). Overview of Contemporary Issues in Mixed Methods Research. *In Sage handbook of mixed methods in social & behavioral research*, pp. 1–44.

Thompson, B. (2002). What future quantitative social science research could look like: Confidence intervals for effect sizes. *Educational Researcher*, *31*(3), pp. 25–32.

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

Van Latum, F., Van Solingen, R., Oivo, M., Hoisl, B., Rombach, D., Ruhe, G. (1998). Adopting GQM based measurement in an industrial environment. *IEEE software, 15*(1), pp.78–86.

Van Solingen, R. (2014). Agile GQM: Why Goal/Question/Metric is more Relevant than Ever and Why It Helps Solving the Agility Challenges of Today's Organizations. In *Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA)*, IEEE. p. 271.

Van Solingen, R., Berghout, E. (2001). Integrating goal-oriented measurement in industrial software engineering: industrial experiences with and additions to the Goal/Question/Metric method (GQM). In *Software Metrics Symposium, 2001. METRICS 2001.* IEEE. pp. 246–258.

Vijayasarathy, L., & Turk, D. (2008). Agile Software Development: A survey of early adopters. *Journal of Information Technology Management*, *19*(2), pp. 1–8.

Virzi, R. A. (1992). Refining the test phase of usability evaluation: How many subjects is enough? *Human factors, 34*(4), pp. 457-468.

Wan, J., & Wang, R. (2010). Empirical Research on Critical Success Factors of Agile Software Process Improvement. *Journal of Software Engineering and Applications*, *3*(12), pp. 1131-1140.

Wang, Y., He, Q. (2003). A practical methodology for measurement deployment in GQM. In *Canadian Conference on Electrical and Computer Engineering CCECE 2003*, IEEE. pp. 1329–1332.

Williams, B., Onsman, A., & Brown, T. (2010). Exploratory factor analysis: A five-step guide for novices. *Australasian Journal of Paramedicine*, *8*(3), pp. 1-13.

Yahaya, J.H., Abidin, Z.N.Z., Ali, N.M., Deraman, A. (2013). Software ageing measurement and classification using Goal Question Metric (GQM) approach. In *Science and Information Conference (SAI)*, IEEE, pp. 160–165.

Yin, R. K. (2013). *Case study research: Design and methods*. Sage publications.

Yong, A. G., & Pearce, S. (2013). A beginner's guide to factor analysis: Focusing on exploratory factor analysis. *Tutorials in Quantitative Methods for Psychology*, *9*(2), pp. 79–94.

Appendix A The Survey

Critical Success Factors of Agile Software Development Projects Survey

This survey has been approved by the Ethical Committee of the School of Electronic and Computer Science at the University of Southampton under the Ethics reference number: ERGO/FPSE/24235

What is the research about?

This survey aims to evaluate the critical success factors of agile development based on phase or iteration level. This study aiming to explore the relationship between the status and the progress of agile projects and the agile success factors.

Why have I been chosen?

This survey will be posted to many agile experts and developers groups on Linkedin. If you decided to take part your participation is appreciated.

What will happen to me if I take part?

You will be asked to answer the survey questions on a web form at your convenience.

Are there any benefits in my taking part?

This research is not designed to help you personally, but your feedback will help us evaluating the critical success factors of agile software development.

Are there any risks involved?

No.

Will my data be confidential?

Yes. Any data will be stored will not be linked to your name or to your organisation's name, you will not be asked to provide your name or any personal data. Your data and that of other participants will be stored and used on secure systems. All the data gathered during this study will be completely destroyed at the end of the study.

How long it will take?

This survey designs to take approximately 5-8 minutes.

What happens if I change my mind?

You have the right to terminate your participation in the research, at any stage, you do not need to give any reasons, and without your legal rights being affected. Your data will be deleted directly if you decide to withdraw at any time.

What happens if something goes wrong?

Should you have any concern or complaint, contact me (a.aldahmash@soton.ac.uk)

Otherwise please contact the FPSE Office (ergopse@soton.ac.uk) or any other authoritative body such as the Research Integrity & Governance Team (rgoinfo@soton.ac.uk).

Where can I get more information?

For further details, please contact me Abdullah Aldahmash a.aldahmash@soton.ac.uk

---

By clicking on to the next section, you have read and understood the Participant Information and you are agreeing to take part in this survey. Also note that you may withdraw at any time by not completing the survey, in which case your data will not be processed.

* Required

Section 1of 3 – Demographics

Thank you very much for agreeing to spend your time to complete this survey. If you have been involved previously or you are currently involving with more than one agile project, please pick one that was most relevant with regard to critical success factors of such a project. In this survey, the term iteration is equal to the term Sprint in case of Scrum.

1. Size of the project (number of project team members)

Mark only one oval.

0-15

16-30

More than 30

2. Length of the project (months)

Mark only one oval.

1-6

7-12

12-18

More than 18 months

3. Location of the project: (Country, NA if it is global or multi location project)

4. The project status: *

Mark only one oval.

On-going project

Completed project

5. Number of planned iterations (or phases) in the project:

6. Number of completed iterations (or phases) in the project:

7. Length of the iteration (days): (NA if the iteration length is not fixed)

8. Your job responsibility in the project:

Mark only one oval.

project manager (Scrum Master)

team leader

team member

customer

organisation management

stakeholder

other

9. Your level of experience with agile projects (years):

Mark only one oval.

0-3

4-6

More than 6 years

10. The organisation's level of experience with agile projects (years):

Mark only one oval.

0-3

4-6

More than 6 years

Section 2 of 3: Success factors of the agile project

This section includes seven success factors of agile software development. The following questions are trying to study the CSFs related to your current iteration (or the whole project in case of completed projects). Please responses to each of the following statements regarding the critical success factors of agile projects:

11. The delivery strategy was critical to the success of the iteration (or project)

Mark only one oval.

Strongly disagree

Disagree

Somewhat disagree

Neutral

Somewhat agree

Agree

Strongly agree

12. Team capability and training were critical to the success of the iteration (or project)

Mark only one oval.

Strongly disagree

Disagree

Somewhat disagree

Neutral

Somewhat agree

Agree

Strongly agree

13. Agile software development techniques and practices were critical to the success of the iteration (or project)

Mark only one oval.

Strongly disagree

Disagree

Somewhat disagree

Neutral

Somewhat agree

Agree

Strongly agree

14. Customer involvement was critical to the success of the iteration (or project)

Mark only one oval.

Strongly disagree

Disagree

Somewhat disagree

Neutral

Somewhat agree

Agree

Strongly agree

15. Project management was critical to the success of the iteration (or project)

Mark only one oval.

Strongly disagree

Disagree

Somewhat disagree

Neutral

Somewhat agree

Agree

Strongly agree

16. Organisational culture was critical to the success of the iteration (or project)

Mark only one oval.

Strongly disagree

Disagree

Somewhat disagree

Neutral

Somewhat agree

Agree

Strongly agree

17. Top management support was critical to the success of the iteration (or project)

Mark only one oval.

Strongly disagree

Disagree

Somewhat disagree

Neutral

Somewhat agree

Agree

Strongly agree

18. Communication was critical to the success of the iteration (or project)

Mark only one oval.

Strongly disagree

Disagree

Somewhat disagree

Neutral

Somewhat agree

Agree

Strongly agree

Section 3 of 3: Measures of success of the agile development project

This section includes aspects of your perceived level of success and the success measures of the agile software development). The following questions are trying measure the success of your current iteration (the whole project in case of completed projects).Please responses to each of the following statements:

19. The project (or iteration) was successful in terms of meeting the time constraint

Mark only one oval.

Strongly disagree

Disagree

Somewhat disagree

Neutral

Somewhat agree

Agree

Strongly agree

20. The project (or iteration) was successful in terms of meeting the cost constraint

Mark only one oval.

Strongly disagree

Disagree

Somewhat disagree

Neutral

Somewhat agree

Agree

Strongly agree

21. The project (or iteration) was successful in terms of meeting the scope constraint

Mark only one oval.

Strongly disagree

Disagree

Somewhat disagree

Neutral

Somewhat agree

Agree

Strongly agree

22. The project (or iteration) was successful in terms of addressing the organisational needs

Mark only one oval.

Strongly disagree

Disagree

Somewhat disagree

Neutral

Somewhat agree

Agree

Strongly agree

23. The project (or iteration) was successful in terms of achieving stakeholder satisfaction

Mark only one oval.

Strongly disagree

Disagree

Somewhat disagree

Neutral

Somewhat agree

Agree

Strongly agree

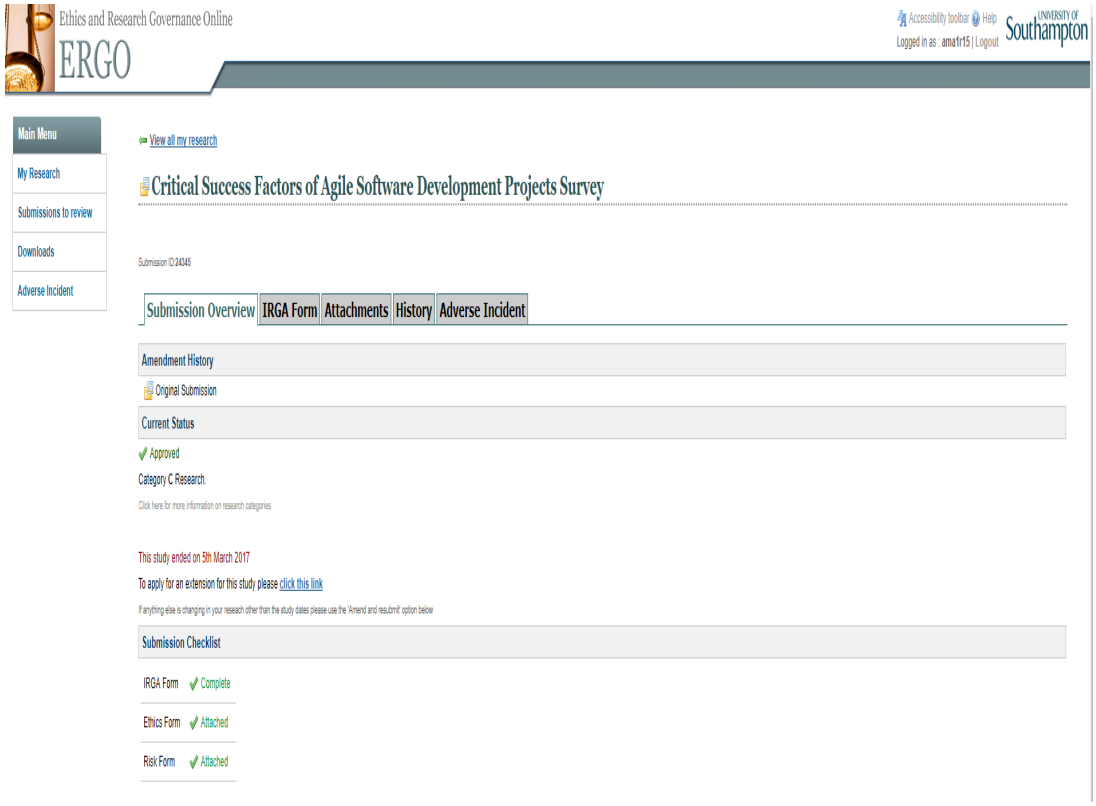
Additional comments

In this section please provide any further comment which has not been covered in the survey and you think it is important.

24. Please enter any additional comments or thoughts here:

Appendix B Ethical Approval of the Survey

The survey used during this research was approved by the Ethical Committee of the School of Electronic and Computer Science at the University of Southampton. This ethical approval was assigned the reference number 24345.



Consent Information

Ethics reference number: **ERGO/**FPSE**/24345**	Version: 1	Date: 2016-11-28
Study Title: Critical Success Factors of Agile Software Development Projects Survey		
Investigator: Abdullah Aldahmash		

I have read and understood the Participant Information (version 1 dated 2016-11-28) and have had the opportunity to ask questions about the study.

I agree to take part in this study.

I understand my participation is voluntary and I may withdraw at any time and for any reason.

By clicking on to the next page, you are agreeing to take part in this survey. Also note that you may withdraw at any time by not completing the survey, in which case your data will not be processed.

Participant Information

Ethics reference number: **ERGO**/FPSE/**24235**	Version: 1	Date: 2016-11-28
Study Title: Critical Success Factors of Agile Software Development Projects Survey		
Investigator: Abdullah Aldahmash		

What is the research about?

This survey aims to evaluate the critical success factors of agile development based on phase or iteration level. This study aiming to explore the relationship between the status and the progress of agile projects and the agile success factors.

Why have I been chosen?

This survey will be posted to many agile experts and developers groups on LinkedIn. If you decided to take part your participation is appreciated.

What will happen to me if I take part?

You will be asked to answer the survey questions on a web form at your convenience.

Are there any benefits in my taking part?

This research is not designed to help you personally, but your feedback will help us evaluating the critical success factors of agile software development.

Are there any risks involved?

No.

Will my data be confidential?

Yes. Any data will be stored will not be linked to your name or to your organisation's name, you will not be asked to provide your name or any personal data. Your data and that of other participants will be stored and used on secure systems. All the data gathered during this study will be completely destroyed at the end of the study.

What happens if I change my mind?

You have the right to terminate your participation in the research, at any stage, you do not need to give any reasons, and without your legal rights being affected. Your data will be deleted directly if you decide to withdraw at any time.

What happens if something goes wrong?

Should you have any concern or complaint, contact me (a.aldahmash@soton.ac.uk)

Otherwise please contact the FPSE Office (ergopse@soton.ac.uk) or any other authoritative body such as the Research Integrity & Governance Team (rgoinfo@soton.ac.uk).

Where can I get more information?

For further details, please contact me Abdullah Aldahmash a.aldahmash@soton.ac.uk

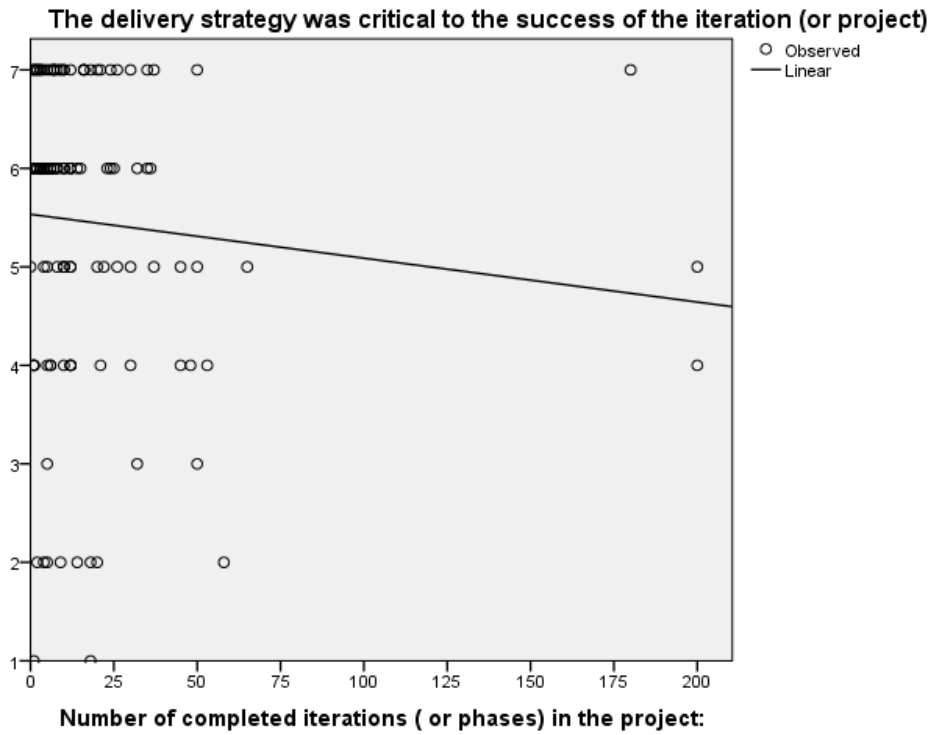
Appendix C Linear Regression Analysis of the CSFs

Delivery Strategy Factor Linear Regression Analysis:

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	2.345	1	2.345	.976	.325
Residual	283.355	118	2.401		
Total	285.700	119			

The independent variable is Number of completed iterations (or phases) in the project:.

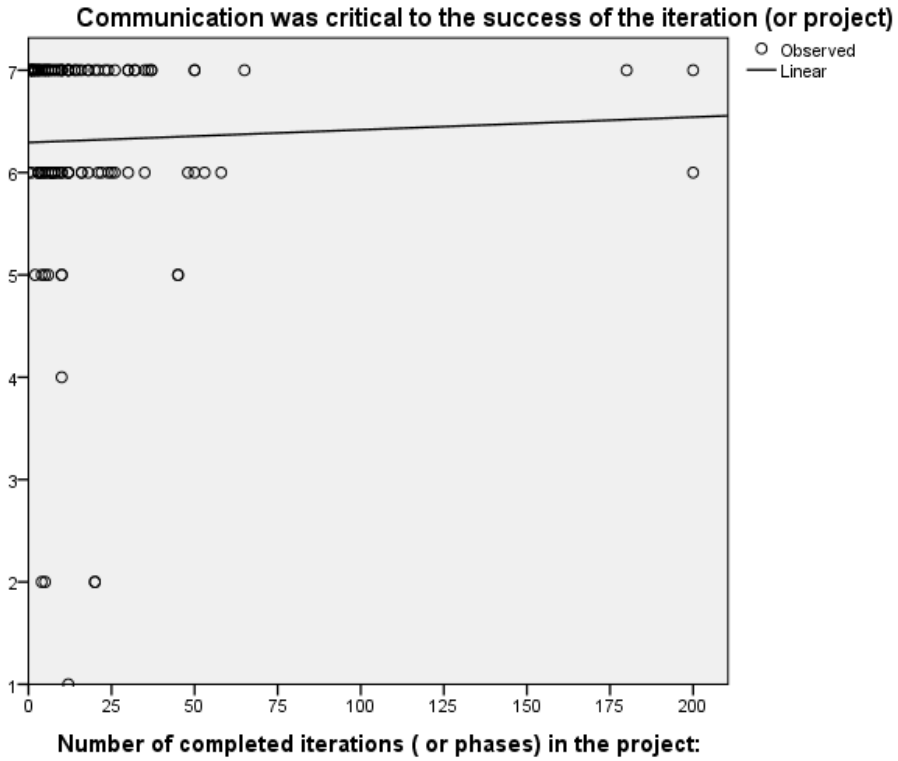


Communication Factor Linear Regression Analysis:

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	.182	1	.182	.136	.713
Residual	157.784	118	1.337		
Total	157.967	119			

The independent variable is Number of completed iterations (or phases) in the project:.

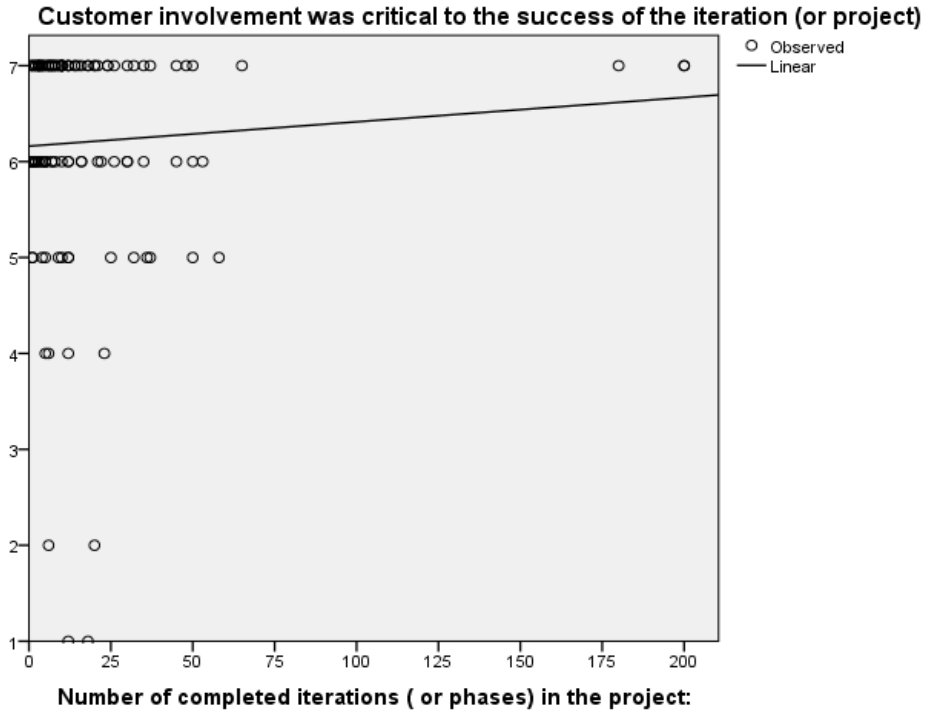


Customer Involvement Factor Linear Regression Analysis:

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	.760	1	.760	.524	.470
Residual	171.031	118	1.449		
Total	171.792	119			

The independent variable is Number of completed iterations (or phases) in the project:.

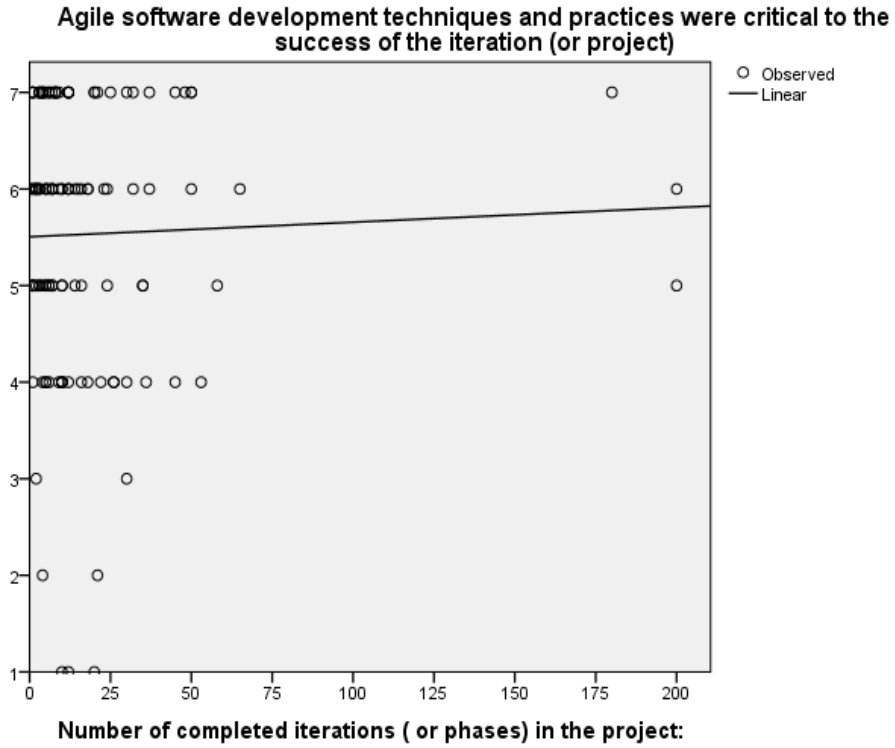


Agile Software Development Techniques Factor Linear Regression Analysis:

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	.270	1	.270	.135	.714
Residual	235.596	118	1.997		
Total	235.867	119			

The independent variable is Number of completed iterations (or phases) in the project:.

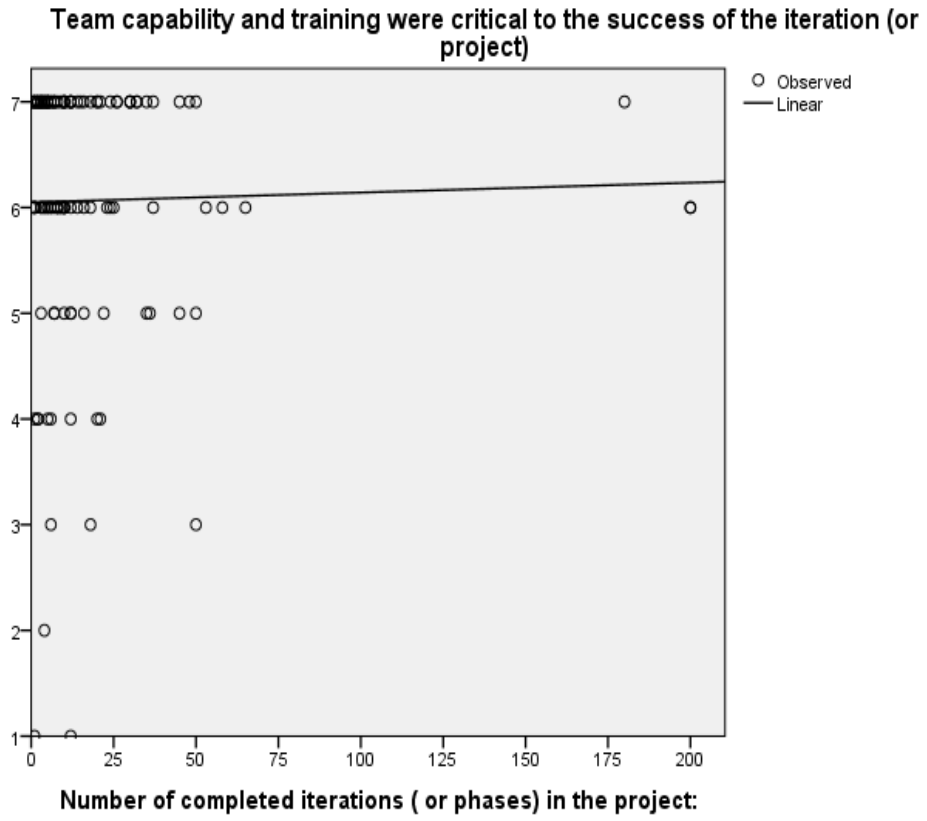


Team Capability and Training Factor Linear Regression Analysis:

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	.101	1	.101	.061	.805
Residual	195.365	118	1.656		
Total	195.467	119			

The independent variable is Number of completed iterations (or phases) in the project:.

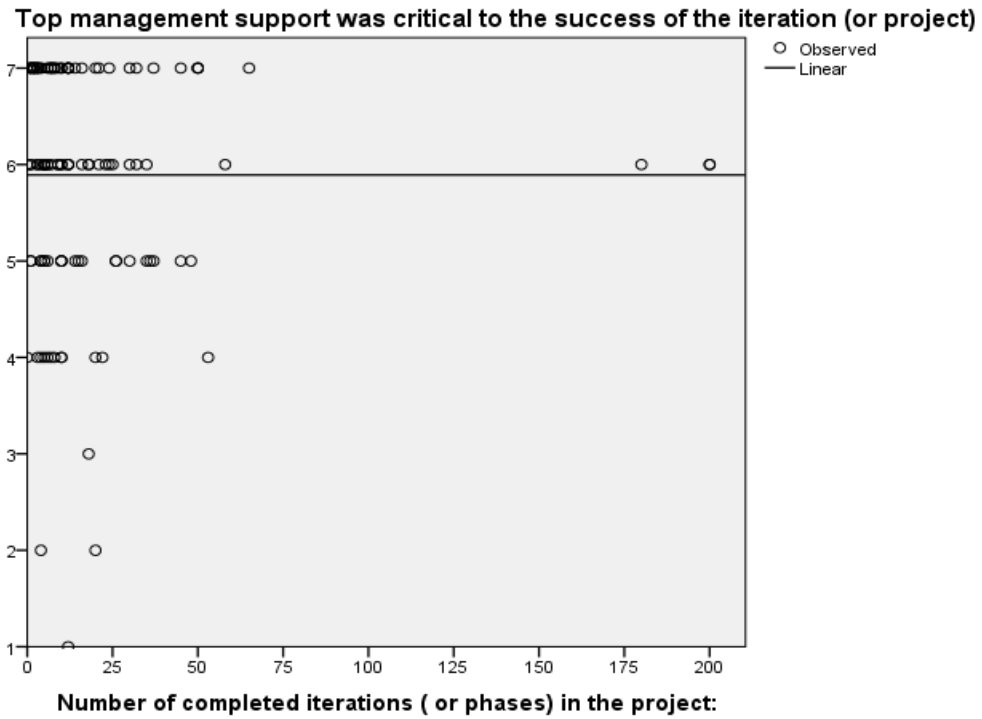


Top Management Support Factor Linear Regression Analysis:

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	.000	1	.000	.000	1.000
Residual	181.592	118	1.539		
Total	181.592	119			

The independent variable is Number of completed iterations (or phases) in the project:.

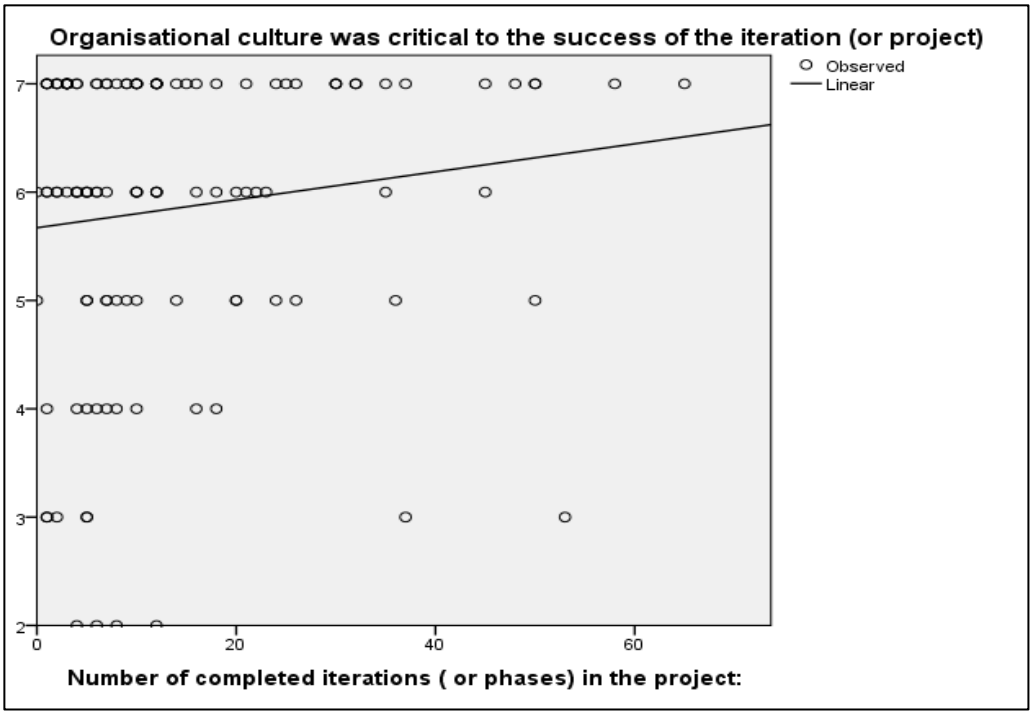


Organizational Culture Factor Linear Regression Analysis after Excluding the EV:

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	4.012	1	4.012	2.055	.154
Residual	224.518	115	1.952		
Total	228.530	116			

The independent variable is Number of completed iterations (or phases) in the project.

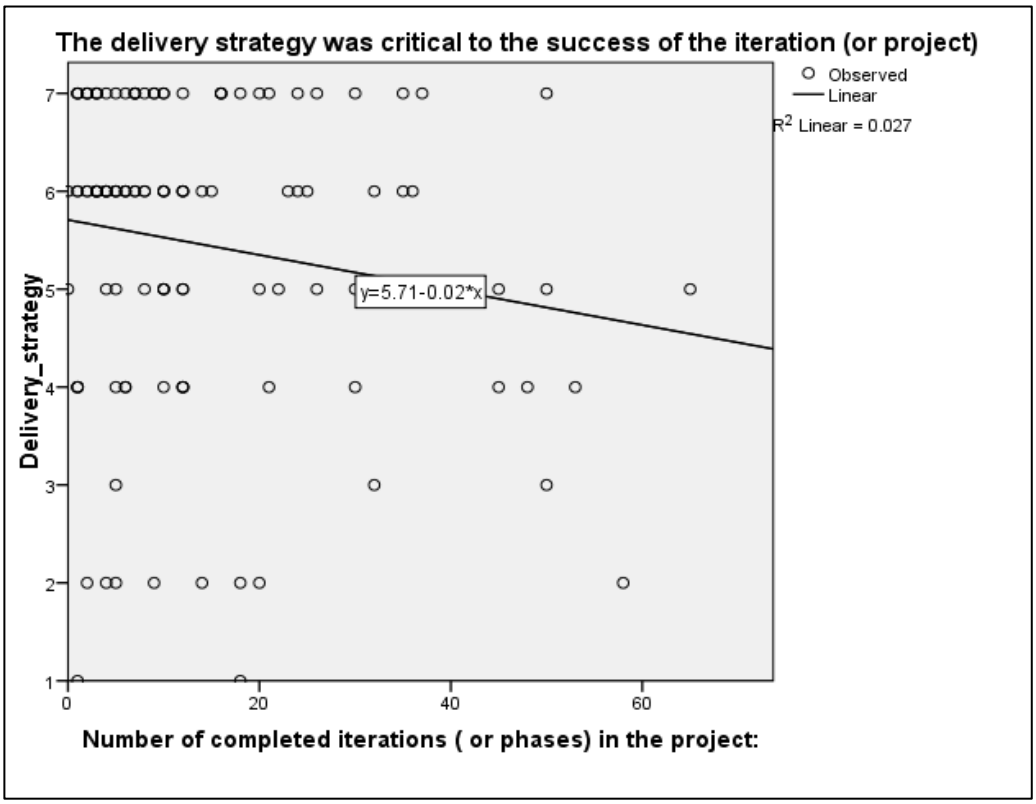


Delivery Strategy Factor Linear Regression Analysis after Excluding the EV:

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	7.700	1	7.700	3.240	.074
Residual	273.292	115	2.376		
Total	280.991	116			

The independent variable is Number of completed iterations (or phases) in the project.

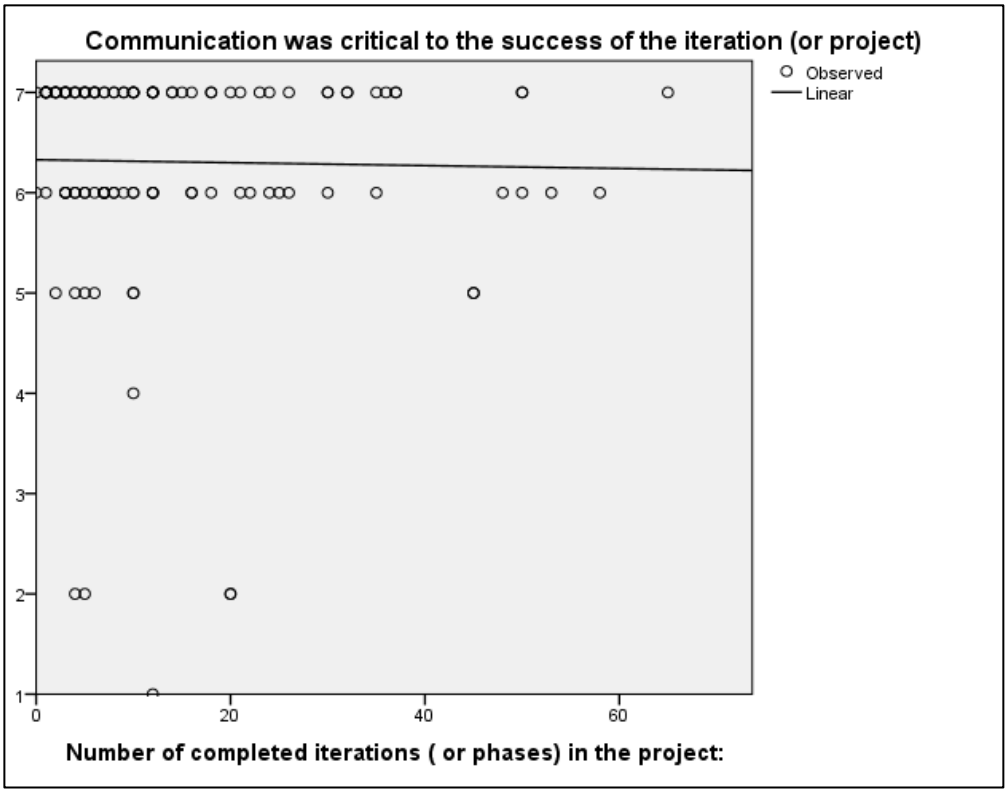


Communication Factor Linear Regression Analysis after Excluding the EV:

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	.050	1	.050	.037	.848
Residual	156.873	115	1.364		
Total	156.923	116			

The independent variable is Number of completed iterations (or phases) in the project.

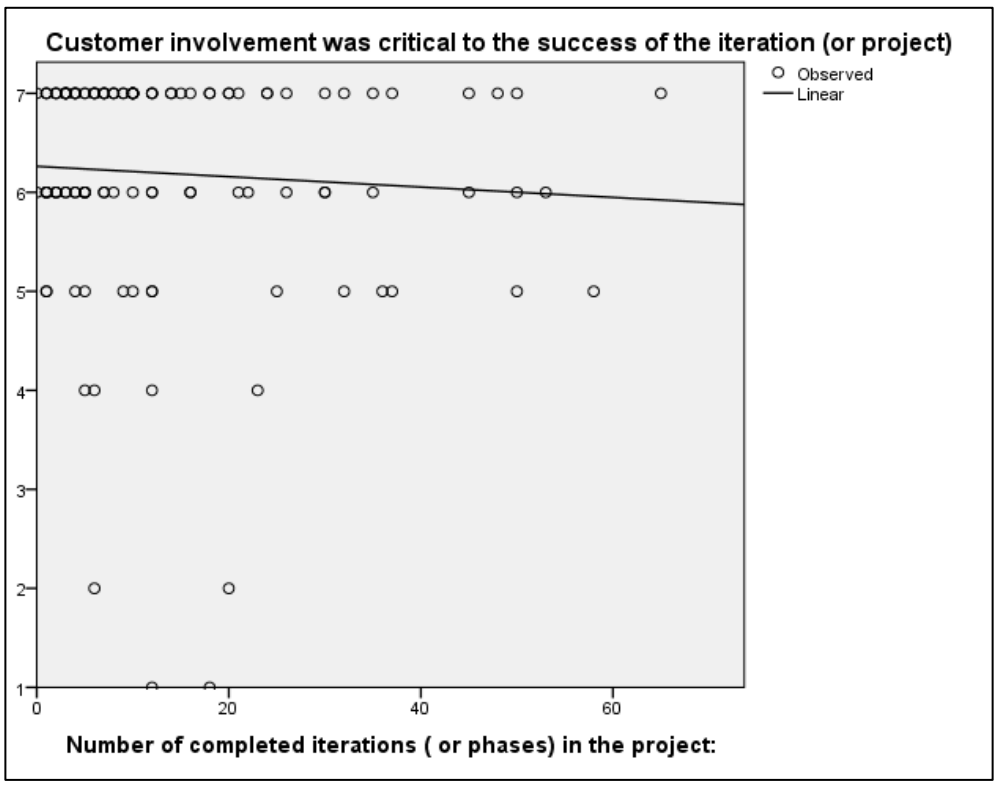


Customer Involvement Factor Linear Regression Analysis after Excluding the EV:

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	.657	1	.657	.447	.505
Residual	169.206	115	1.471		
Total	169.863	116			

The independent variable is Number of completed iterations (or phases) in the project.

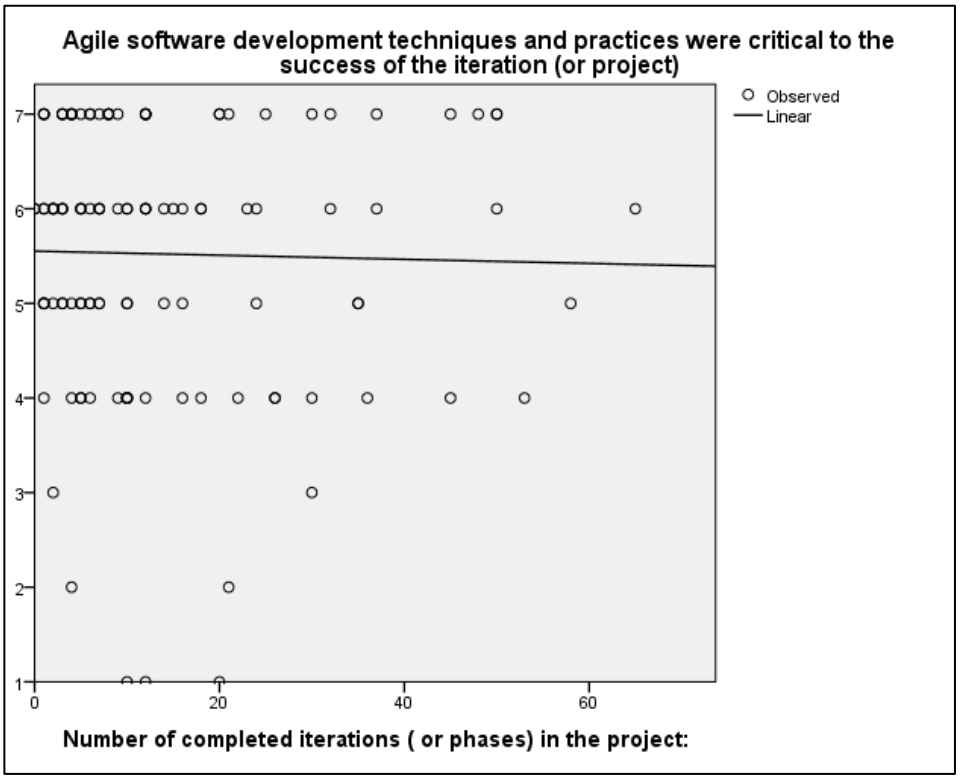


Agile Software Development Techniques Factor Linear Regression Analysis after Excluding the EV:

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	.113	1	.113	.056	.814
Residual	233.084	115	2.027		
Total	233.197	116			

The independent variable is Number of completed iterations (or phases) in the project.

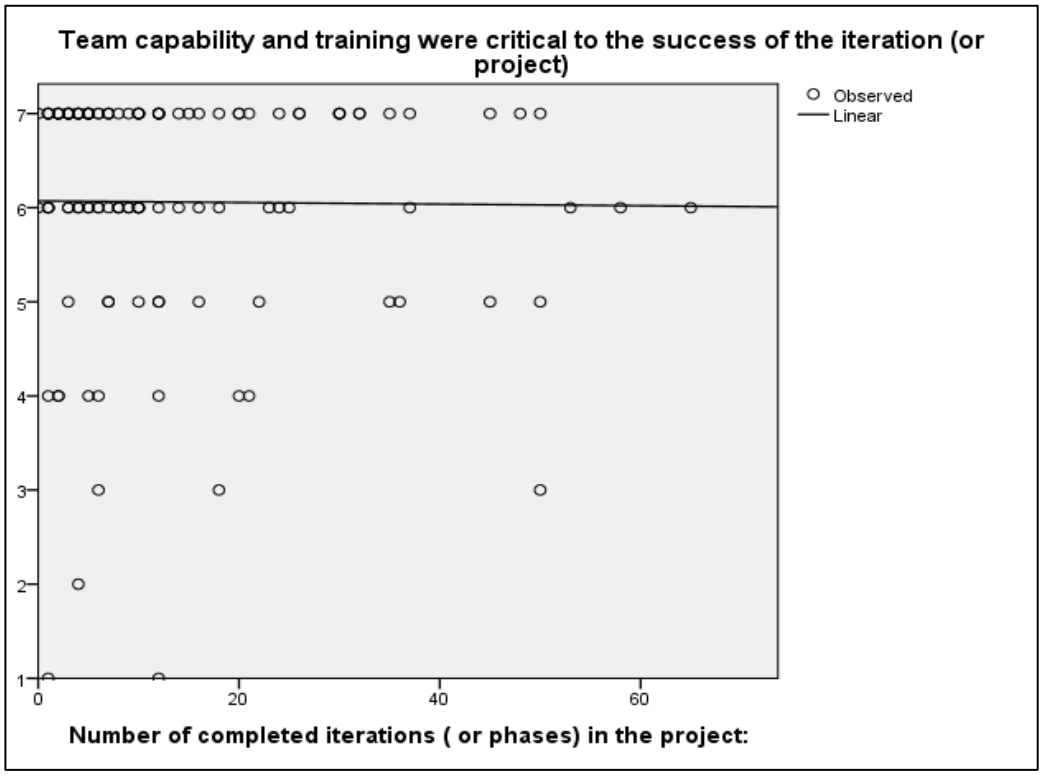


Team Capability and Training Factor Linear Regression Analysis after Excluding the EV:

ANOVA

	Sum of Squares	df	Mean Square	F	Sig.
Regression	.017	1	.017	.010	.919
Residual	194.564	115	1.692		
Total	194.581	116			

The independent variable is Number of completed iterations (or phases) in the project.

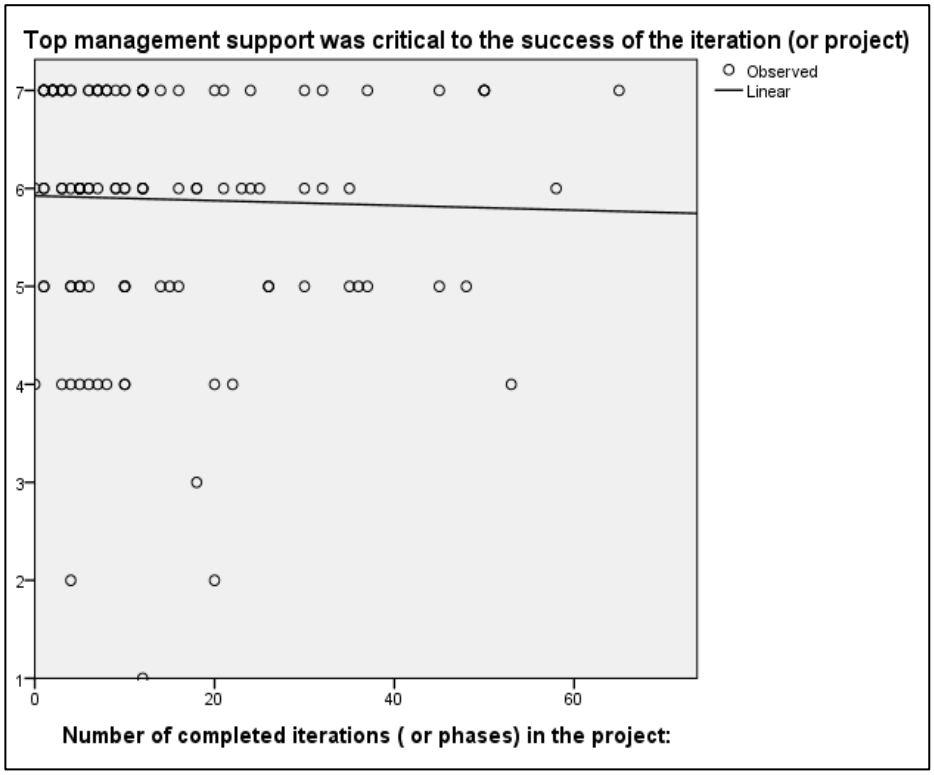


Top Management Support Factor Linear Regression Analysis after Excluding the EV:

ANOVA

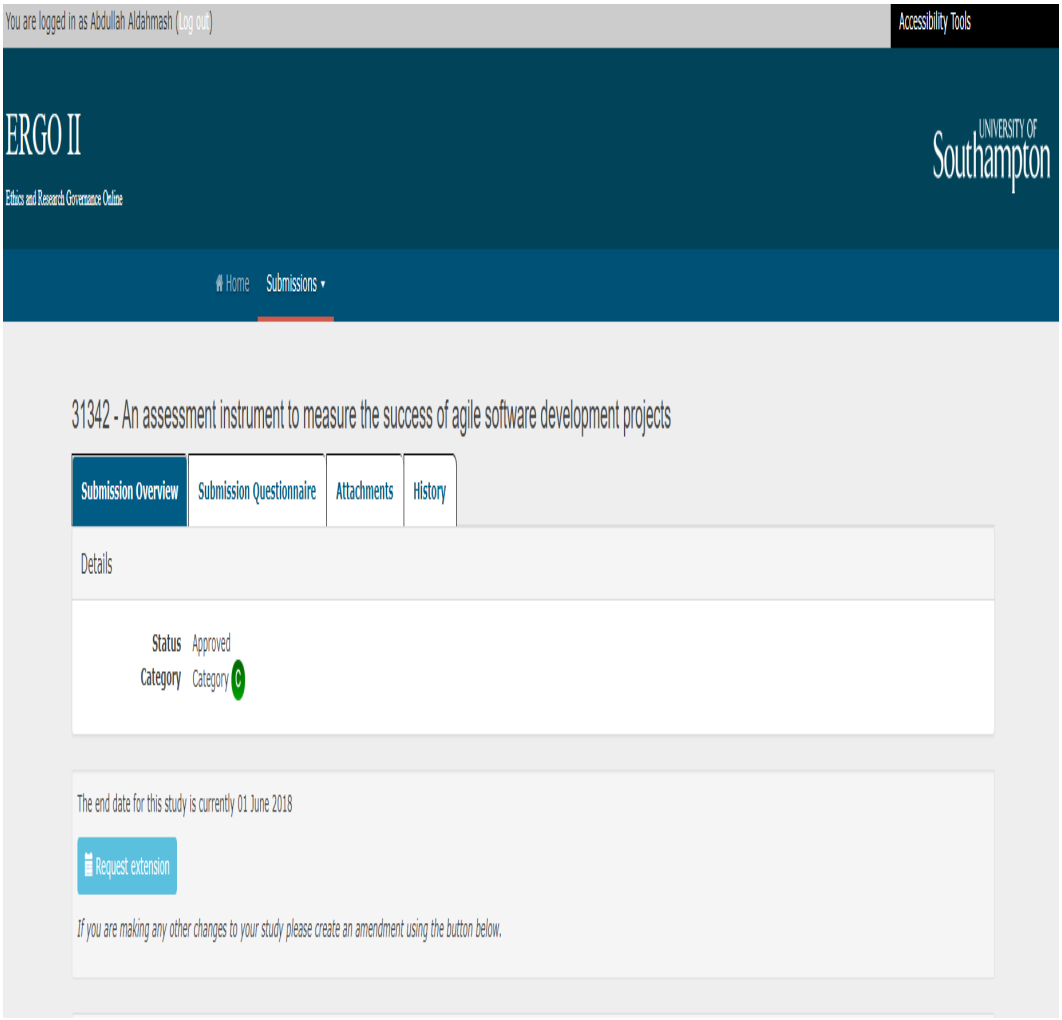
	Sum of Squares	df	Mean Square	F	Sig.
Regression	.141	1	.141	.089	.766
Residual	181.415	115	1.578		
Total	181.556	116			

The independent variable is Number of completed iterations (or phases) in the project.



Appendix D Ethical Approval of the Experts Interviews and the Case Studies

The experts' interviews and the case studies conducted during this study were approved by Ethical Committee of the School of Electronic and Computer Science at the University of Southampton. This ethical approval was assigned with the reference number 31342.



Amendment History

Original Submission 31342 (Created 29/11/2017)

User Uploaded Documents

	Title	Document Type	Original File name	Uploaded	Size
Current Version					
1.	Ethics_Form_Version 1	Ethics Form	ethicsformversion1.doc	07/12/2017 13:03:32	75 Kb
2.	Risk_Assessment_Version_1	Risk Assessment	riskformversion1.doc	07/12/2017 13:05:51	189 Kb
3.	Participant_Information_Version_1	Participant Information Sheet	participantinformationversion1.doc	07/12/2017 13:04:20	43 Kb
4.	Consent_Form_Version_1	Consent Form	consentforms.docx	07/12/2017 13:04:53	32 Kb
5.	Data_Collection_Method_Version_1	Interview Topic Guide	datacollectionmethodversion1.docx	07/12/2017 13:08:30	16 Kb
6.	Dpa_Plan_Version_1	Debriefing Statement	dpaplanversion1.docx	07/12/2017 13:07:31	19 Kb

Checklist

Submission Questionnaire ✓

Coordinators

Abdullah Aldahmash (ama1r15 A.Aldahmash@soton.ac.uk)
Andrew Gravell (gravell amg@ecs.soton.ac.uk) supervisor

Appendix D Ethical Approval of the Experts Interviews and the Case Studies

Consent Form for the Experts Interview (Group one)

Ethics reference number: **ERGO/**FPSE**/31342**	Version: 1	Date: 2017-12-04
Study Title: An assessment instrument to measure the success of agile software development projects.		
Investigator: Abdullah Aldahmash		

Please initial the box(es) if you agree with the statement(s):

I have read and understood the Participant Information (version 1 dated 2017-12-04) and have had the opportunity to ask ☐

I agree to take part in this study. ☐

I understand my participation is voluntary and I may withdraw at ☐

I agree to my voice being recorded during my participation in this study ☐

Data Protection

I understand that information collected and recorded during my participation in this study is completely secured and it will be stored on a password-protected computer and that this information will only be used for the purpose of this study and it will be immediately destroyed at the end of the research.

Name of participant (print name)……

Signature of participant……..

Date……

Consent Form for the Case Studies Participants (Group two)

Ethics reference number: **ERGO/**FPSE **/31342**	Version: 1	Date: 2017-12-04
Study Title: An assessment instrument to measure the success of agile software development projects.		
Investigator: Abdullah Aldahmash		

Please initial the box(es) if you agree with the statement(s):

I have read and understood the Participant Information (version 1 dated 2017-12-04) and have had the opportunity to ask ☐

I agree to take part in this study. ☐

I understand my participation is voluntary and I may withdraw at ☐

Data Protection

I understand that information collected and recorded during my participation in this study is completely secured and it will be stored on a password-protected computer and that this information will only be used for the purpose of this study and it will be immediately destroyed at the end of the research.

Name of participant (print name)……

Signature of participant……..

Date……

Appendix D Ethical Approval of the Experts Interviews and the Case Studies

Participant Information

Ethics reference number: **ERGO/**FPSE**/31342**	Version: 1	Date: 2017-12-04
Study Title: An assessment instrument to measure the success of agile software development projects.		
Investigator: Abdullah Aldahmash		

What is the research about?

This study aims to develop an instrument that could assess with measuring the success in agile software development projects.

Why have I been chosen?

This study aiming to present the developed instrument to many agile experts in order to capture their feedback about the instrument. Following that, the developed instrument will be used as a case study in organizations which running software projects following agile practices or methods. If you decided to take part your participation is appreciated.

What will happen to me if I take part?

You will be interviewed with open-ended questions at your convenient time to discuss the developed instrument.

Are there any benefits in my taking part?

This research is not designed to help you personally, but your feedback will help us measuring the success of agile software development projects.

Are there any risks involved?

No.

Will my data be confidential?

Yes. Any data will be stored will not be linked to your name or to your organisation's name, you will not be asked to provide your name or any personal data. Your data and that of other participants will be stored and used on secure systems. All the data gathered during this study will be completely destroyed at the end of the study.

What happens if I change my mind?

You have the right to terminate your participation in the study, at any stage, you do not need to give any reasons, and without your legal rights being affected. Your data will be deleted directly if you decide to withdraw at any time.

What happens if something goes wrong?

Should you have any concern or complaint, contact me (a.aldahmash@soton.ac.uk) or contact the project supervisor (amg@ecs.soton.ac.uk)

Otherwise please contact the FPSE Office (ergopse@soton.ac.uk) or any other authoritative body such as the Research Integrity & Governance Team (rgoinfo@soton.ac.uk).

Where can I get more information?

For further details, please contact me Abdullah Aldahmash a.aldahmash@soton.ac.uk

Appendix E First and Final Version of the Instrument

The First Version of the Proposed Instrument

1st Goal: Improve the communication throughout the agile project

Goal		Question	Metric
Purpose	Improve	Q1. Rate your use of a communication platform across the team (e.g. Slack, Jira)?	• Very Good • Good • Acceptable • Poor • Very Poor
Issue	the Communication	Q2. How often the team is sharing and communicating development's aspects?	• Always • Often • Sometimes • Seldom • Never
Object	project team members and customers	Q3. Rate the team practice of daily meetings (physical or virtual) where the team set together to discuss the project progress (this is equivalent to daily scrum practice and pair programming practice in XP)?	• Very Good • Good • Acceptable • Poor • Very Poor
Viewpoint		Q4. Rate your use of centralized repositories to enable documents and knowledge sharing throughout the project?	• Very Good • Good • Acceptable • Poor • Very Poor
		Q5. How often the team have access to task boards (or smart boards) to communicate with co-located members and video conferences capabilities to communicate with different-located members?	• Always • Often • Sometimes • Seldom • Never
		Q6. What is the percentage of informal communication to the whole communication in the project?	Percentage
		Qn.	Relevant software metric

2nd Goal: Increase the customer involvement during the agile project

Goal		Question	Metric
Purpose	Increase	Q1. What is the percentage of planning meetings at the beginning of the sprint where the customer is present?	Percentage
Issue	the customer involvement	Q2. What is the percentage of end of sprints demos where the customer is present?	Percentage
Object	customers	Q3. Rate the response time from the customer to development queries?	• Very Good • Good • Acceptable • Poor • Very Poor
Viewpoint		Q4. Rate the commitment of the customer in the project?	• Very Good • Good • Acceptable • Poor • Very Poor
		Q5. Rate the quality of customer involvement and input in the project?	• Very Good • Good • Acceptable • Poor • Very Poor
		Qn.	Relevant software metric

3rd Goal: Improve the training of the agile project team members

Goal		Question	Metric
Purpose	Improve	Q1. Rate the available trainings in covering all aspects needed by the project team members?	• Very Good • Good • Acceptable • Poor • Very Poor
Issue	The training	Q2. Rate the appropriateness of the contents of the training received?	• Very Good • Good • Acceptable • Poor • Very Poor
Object	Training programs, and agile project team members	Q3. Rate the commitment of the project team members toward the available training programs?	• Very Good • Good • Acceptable • Poor • Very Poor
Viewpoint		Q4. How often did the project team members practice self-training?	• Always • Often • Sometimes • Seldom • Never
		Qn.	Relevant software metric

4th Goal: Increase the support from top management in the agile project

Goal		Question	Metric
Purpose	Increase	Q1. What is the percentage of meetings where the top management were involved (support meetings) to all of the meetings?	Percentage
Issue	The support from top management	Q2. What is the percentage of support meetings that were cancelled or postponed?	Percentage
Object	top management	Q3. How many in average events (meetings, emails, requests, etc.) did top management initiate or propose per phase (sprint)?	Average
Viewpoint		Q4. Rate the level of commitment of the top management in the project?	• Very Good • Good • Acceptable • Poor • Very Poor
		Q5. Rate the role of top management support in facilitating development issues?	• Very Good • Good • Acceptable • Poor • Very Poor
		Q6. Rate the role of top management support in expediting development issues?	• Very Good • Good • Acceptable • Poor • Very Poor
		Qn.	Relevant software metric

5th Goal: Enhance the delivery strategy

Goal		Question	Metric
Purpose	Enhance	Q1. What is the average volume of work required to deliver a story point?	Story point Velocity
Issue	the delivery strategy	Q2. How long it takes to deliver a story point?	Story point cycle time
Object	Story points	Q3. How much of the sprint's (or iteration) planned story points actually delivered by the end of the sprint?	Sprint Burndown
Viewpoint		Q4. What is the percentage of planned to delivered story points in the project?	Epic and Release Burndown
		Q5. What is the percentage of releases that are released on-time?	Percentage
		Q6. What is your schedule efficiency (how fast you are progressing against the rate of progress planned)?	Schedule Performance Index (SPI)
		Qn.	Relevant software metric

6th Goal: Appropriate selection of agile techniques, practices, and project management PM approach.

Goal		Question	Metric
Purpose	Appropriate	Q1. Do the current knowledge and capabilities of the team are considered when selecting agile techniques, practices and PM approach?	• Always • Often • Sometimes • Seldom • Never
Issue	Selection of agile techniques, practices and PM approach	Q2. Do the team use an existing agile method "off the shelf" or adjust it to suit their needs?	Yes/No
Object	agile techniques, practices, and PM approach	Q3. How often do the team conduct retrospectives (sprint review) to discuss the selection of agile techniques, practices and PM approach?	• Always • Often • Sometimes • Seldom • Never
Viewpoint		Q4. How often do these sprint reviews lead to a change in agile techniques, practices and PM approach?	• Always • Often • Sometimes • Seldom • Never
		Q5. How often do the team consider the effectiveness of the current agile techniques, practices and PM approach?	• Always • Often • Sometimes • Seldom • Never
		Q6. How often are changes made to these agile techniques, practices and PM approach?	• Always • Often • Sometimes • Seldom • Never
		Qn.	Relevant software metric

The Final Version of the Proposed Instrument

1st Goal: Improve the communication throughout the agile project

Goal		Question	Metric
Purpose	Improve	Q1. Rate your use of the ready communication platforms across the team (e.g. Slack) or your own developed platform?	• Very Good • Good • Acceptable • Poor • Very Poor
Issue	the Communication	Q2. How often the project team is sharing and communicating development's aspects?	• Always • Often • Sometimes • Seldom • Never
Object	Agile project stakeholders	Q3. Rate the team practice of daily meetings (physical or virtual) where the team sit together to discuss the project progress (this is equivalent to daily stand-up scrum practice and pair programming practice in XP)?	• Very Good • Good • Acceptable • Poor • Very Poor
Viewpoint		Q4. Rate your use of centralized repositories to enable documents and knowledge sharing throughout the project?	• Very Good • Good • Acceptable • Poor • Very Poor
		Q5. How often the team have access to task boards (or smart boards) to communicate with co-located members and video conferences capabilities to communicate with different-located members?	• Always • Often • Sometimes • Seldom • Never
		Q6. How often do you communicate informally (face to face communication) during the project when it is possible?	• Always • Often • Sometimes • Seldom • Never

2nd Goal: Increase the customer involvement during the agile project

Goal		**Question**	**Metric**
Purpose	Increase	Q1. How often do the customers attend the meetings (planning meetings, demos, and retrospectives) when they are requested to do so by the project team?	• Always • Often • Sometimes • Seldom • Never
Issue	the customer involvement	Q2. Rate the customers' participation in planning meetings, demos, retrospectives and how they contribute to the success of these events?	• Very Good • Good • Acceptable • Poor • Very Poor
Object	customers	Q3. Rate the response time (e.g. how fast they are) from the customers to development queries?	• Very Good • Good • Acceptable • Poor • Very Poor
Viewpoint		Q4. Rate the commitment and the support of the customers in the project toward resolving development issues and difficulties?	• Very Good • Good • Acceptable • Poor • Very Poor
		Q5. How often do the customers express their needs to the project team, or suggest improvement for enhancing the project to the team?	• Always • Often • Sometimes • Seldom • Never

3rd Goal: Improve the training of the agile project team members

Goal		Question	Metric
Purpose	Improve	Q1. Rate the available training resources in covering all aspects needed by the project team members?	• Very Good • Good • Acceptable • Poor • Very Poor
Issue	The training	Q2. Rate the appropriateness of the contents of the training received by the project team?	• Very Good • Good • Acceptable • Poor • Very Poor
Object	Training programs	Q3. Rate the participation (e.g. attending, supporting, and facilitating) of the project team members in the available training programs?	• Very Good • Good • Acceptable • Poor • Very Poor
Viewpoint		Q4. How often did the project team members practice self-training (e.g. watching learning videos, attending webinar, etc.)?	• Always • Often • Sometimes • Seldom • Never

4th Goal: Increase the support from top management in the agile project

Goal		Question	Metric
Purpose	Increase	Q1. How often are the top management involved in planning meetings, demos, and retrospectives when they are requested to be there?	• Always • Often • Sometimes • Seldom • Never
Issue	The support from top management	Q2. Rate the role of top management support toward the success of the attended planning meetings, demos, and retrospectives during the project?	• Very Good • Good • Acceptable • Poor • Very Poor
Object	top management	Q3. How often do the top management initiate or propose events (meetings, emails, requests, etc.) whenever it is necessarily to do so?	• Always • Often • Sometimes • Seldom • Never
Viewpoint		Q4. Rate the role of top management support in facilitating development issues?	• Very Good • Good • Acceptable • Poor • Very Poor
		Q5. Rate the role of top management support in expediting development issues?	• Very Good • Good • Acceptable • Poor • Very Poor
		Q6. Rate the overall support (budget, time, resources, etc.) from top management in the project?	• Very Good • Good • Acceptable • Poor • Very Poor

5th Goal: Enhance the delivery strategy

Goal		**Question**	**Metric**
Purpose	Enhance	Q1. How long it takes to deliver a story point?	Story point cycle time
Issue	the delivery strategy	Q2. How much of the sprint's (or iteration) planned story points actually delivered by the end of the current sprint?	Sprint Burndown
Object	Story points	Q3. What is the percentage of planned to delivered story points in the current release?	Release Burndown
Viewpoint		Q4. What is your schedule efficiency (how fast you are progressing against the rate of progress planned)?	Schedule Performance Index (SPI)

6th Goal: Appropriate selection of agile techniques, practices, and project management PM approach.

Goal		Question	Metric
Purpose	Appropriate	Q1. How often are the current knowledge and capabilities of the team are considered when selecting agile techniques, practices and PM approach?	• Always • Often • Sometimes • Seldom • Never
Issue	Selection of agile techniques, practices and PM approach	Q2. How often are the needs of the customers and top management considered when selecting agile techniques, practices and PM approach?	• Always • Often • Sometimes • Seldom • Never
Object	agile techniques, practices, and PM approach	Q3. Do the team use an existing agile method "off the shelf" without adjusting it to suit their needs?	Yes/No
Viewpoint		Q4. How often do the team conduct retrospectives (sprint reviews) to discuss the improvement of the selection of agile techniques, practices and PM approach?	• Always • Often • Sometimes • Seldom • Never
		Q5. How often do these retrospectives (sprint reviews) lead to a change in agile techniques, practices and PM approach?	• Always • Often • Sometimes • Seldom • Never

Appendix F Paired T-test Results

Paired T-Test comparing the difference in the means between Customer Involvement factor and the other CSFs

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	Customer involvement - Project management	1.137	1.872	.164	.814	1.461	6.954	130	.000
Pair 2	Customer involvement - Delivery strategy	.748	1.670	.146	.459	1.037	5.126	130	.000
Pair 3	Customer involvement - Agile software development techniques and practices	.603	1.562	.137	.333	.873	4.418	130	.000
Pair 4	Customer involvement - Organisational culture	.366	1.314	.115	.139	.594	3.192	130	.002
Pair 5	Customer involvement - Top management support	.305	1.324	.116	.077	.534	2.640	130	.009
Pair 6	Customer involvement - Team capability and training	.107	1.242	.109	-.108	.322	.985	130	.326
Pair 7	Customer involvement - Communication	-.107	1.349	.118	-.340	.126	-.907	130	.366

Paired T-Test comparing the difference in the means between Team Capability and Training factor and the other CSFs

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	Team capability and training - Project management	1.031	1.831	.160	.714	1.347	6.442	130	.000
Pair 2	Team capability and training - Delivery strategy	.641	1.589	.139	.367	.916	4.619	130	.000
Pair 3	Team capability and training - Agile software development techniques and practices	.496	1.561	.136	.226	.766	3.638	130	.000
Pair 4	Team capability and training - Organisational culture	.260	1.460	.128	.007	.512	2.034	130	.044
Pair 5	Team capability and training - Top management support	.198	1.536	.134	-.067	.464	1.479	130	.142
Pair 6	Team capability and training - Communication	-.214	1.347	.118	-.447	.019	-1.816	130	.072
Pair 7	Team capability and training - Customer involvement	-.107	1.242	.109	-.322	.108	-.985	130	.326

Paired T-Test comparing the difference in the means between top Management Support factor and the other CSFs

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
					95% Confidence Interval of the Difference				
		Mean	Std. Deviation	Std. Error Mean	Lower	Upper			
Pair 1	Top management support - Project management	.832	1.832	.160	.515	1.149	5.198	130	.000
Pair 2	Top management support - Delivery strategy	.443	1.665	.145	.155	.731	3.044	130	.003
Pair 3	Top management support - Agile software development techniques and practices	.298	1.429	.125	.051	.545	2.385	130	.019
Pair 4	Top management support - Organisational culture	.061	1.334	.117	-.170	.292	.524	130	.601
Pair 5	Top management support - Communication	-.412	1.122	.098	-.606	-.218	-4.204	130	.000
Pair 6	Top management support - Customer involvement	-.305	1.324	.116	-.534	-.077	-2.640	130	.009
Pair 7	Top management support - Team capability and training	-.198	1.536	.134	-.464	.067	-1.479	130	.142

Paired T-Test comparing the difference in the means between Organisational Culture factor and the other CSFs

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	Organisational culture - Agile software development techniques and practices	.237	1.718	.150	-.060	.534	1.577	130	.117
Pair 2	Organisational culture - Delivery strategy	.382	1.825	.159	.066	.697	2.394	130	.018
Pair 3	Organisational culture - Project management	.771	2.029	.177	.420	1.122	4.349	130	.000
Pair 4	Organisational culture - Communication	-.473	1.521	.133	-.736	-.210	-3.562	130	.001
Pair 5	Organisational culture - Customer involvement	-.366	1.314	.115	-.594	-.139	-3.192	130	.002
Pair 6	Organisational culture - Team capability and training	-.260	1.460	.128	-.512	-.007	-2.034	130	.044
Pair 7	Organisational culture - Top management support	-.061	1.334	.117	-.292	.170	-.524	130	.601

Paired T-Test comparing the difference in the means between Agile Software Development Techniques and Practices factor and the other CSFs

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	Agile software development techniques and practices - Delivery strategy	.145	1.733	.151	-.154	.445	.958	130	.340
Pair 2	Agile software development techniques and practices - Project management	.534	1.837	.160	.217	.852	3.330	130	.001
Pair 3	Agile software development techniques and practices - Communication	-.710	1.321	.115	-.938	-.482	-6.149	130	.000
Pair 4	Agile software development techniques and practices - Customer involvement	-.603	1.562	.137	-.873	-.333	-4.418	130	.000
Pair 5	Agile software development techniques and practices - Team capability and training	-.496	1.561	.136	-.766	-.226	-3.638	130	.000
Pair 6	Agile software development techniques and practices - Top management support	-.298	1.429	.125	-.545	-.051	-2.385	130	.019
Pair 7	Agile software development techniques and practices - Organisational culture	-.237	1.718	.150	-.534	.060	-1.577	130	.117

Paired T-Test comparing the difference in the means between Delivery Strategy factor and the other CSFs

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	Delivery strategy - Project management	.389	1.825	.159	.074	.705	2.441	130	.016
Pair 2	Delivery strategy - Communication	-.855	1.678	.147	-1.145	-.565	-5.830	130	.000
Pair 3	Delivery strategy - Customer involvement	-.748	1.670	.146	-1.037	-.459	-5.126	130	.000
Pair 4	Delivery strategy - Team capability and training	-.641	1.589	.139	-.916	-.367	-4.619	130	.000
Pair 5	Delivery strategy - Top management support	-.443	1.665	.145	-.731	-.155	-3.044	130	.003
Pair 6	Delivery strategy - Organisational culture	-.382	1.825	.159	-.697	-.066	-2.394	130	.018
Pair 7	Delivery strategy - Agile software development techniques and practices	-.145	1.733	.151	-.445	.154	-.958	130	.340

Paired T-Test comparing the difference in the means between Project Management factor and the other CSFs

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower	Upper			
Pair 1	Project management - Communication	-1.244	1.724	.151	-1.542	-.946	-8.263	130	.000
Pair 2	Project management - Customer involvement	-1.137	1.872	.164	-1.461	-.814	-6.954	130	.000
Pair 3	Project management - Team capability and training	-1.031	1.831	.160	-1.347	-.714	-6.442	130	.000
Pair 4	Project management - Top management support	-.832	1.832	.160	-1.149	-.515	-5.198	130	.000
Pair 5	Project management - Organisational culture	-.771	2.029	.177	-1.122	-.420	-4.349	130	.000
Pair 6	Project management - Agile software development techniques and practices	-.534	1.837	.160	-.852	-.217	-3.330	130	.001
Pair 7	Project management - Delivery strategy	-.389	1.825	.159	-.705	-.074	-2.441	130	.016