THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

AGILE SOFTWARE PROJECT EVALUATION BY USING PARTIAL LEAST SQUARES STRUCTURAL EQUATION MODELING (PLS SEM) APPROACH IN THE VIEW OF CRITICAL SUCCESS INDICATORS' FAILURE RESEARCH

Master's Thesis

HARUN ÇALIŞKAN

ISTANBUL, 2016

THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
COMPUTER ENGINEERING (ENGLISH - THESIS)

AGILE SOFTWARE PROJECT EVALUATION BY USING PARTIAL LEAST SQUARES STRUCTURAL EQUATION MODELING (PLS SEM) APPROACH IN THE VIEW OF CRITICAL SUCCESS INDICATORS' FAILURE RESEARCH

Master's Thesis

HARUN ÇALIŞKAN

Supervisor: Assoc.Prof. Dr. Dilek KARAHOCA

İSTANBUL, 2016

THE REPUBLIC OF TURKEY
BAHCESEHIR UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
COMPUTER ENGINEERING (ENGLISH – THESIS)

Name of the thesis: Agile Software Project Evaluation by Using Partial Least Squares Structural Equation Modeling (PLS SEM) Approach in the View of Critical Success Indicators' Failure Research
Name/Last Name of the Student: Harun Çalışkan
Date of the Defense of Thesis: 20/05/2016

The thesis has been approved by the Graduate School of Natural and Applied Sciences.

Assoc.Prof. Dr.Nafiz ARICA
Graduate School Director

I certify that this thesis meets all the requirements as a thesis for the degree of Master of Arts.

Ph.D. Tarkan AYDIN
Program Coordinator

This is to certify that we have read this thesis and we find it fully adequate in scope, quality and content, as a thesis for the degree of Master of Arts.

Examining Comittee Members	Signature
Thesis Supervisor Assoc.Prof. Dr. Dilek KARAHOCA	----------------------------------
Thesis Co-supervisor Prof. Dr.Adem KARAHOCA	---------------------------------
Member Prof. Dr. İbrahim PINAR	----------------------------------

ACKNOWLEDGEMENTS

I want to sincerely thank to my supervisors, Prof.Dr. Adem KARAHOCA and Assoc.Prof.Dr. Dilek KARAHOCA, for all their help, suggestions and valuable feedback that they've provided during my thesis work.

I would like to thank to my company, NETAŞ Techno-Academy program, and my directors too, Gönül KAMALI and Mesut ŞEN, for funding my studies; I really appreciate for the opportunity that I'm given and especially for their confidence in me.

Special thanks to my colleagues and teammates, Şenay DEMİREL and Seydi MİHMANLI, for their motivation, support and guidance all through this study.

Finally, I would like to thank to my parents and my dear wife, Seren ÇALIŞKAN, for their full support and patience during my studies and for all my life.

İSTANBUL 2016 Harun ÇALIŞKAN

ABSTRACT

AGILE SOFTWARE PROJECT EVALUATION BY USING PARTIAL LEAST SQUARES STRUCTURAL EQUATION MODELING (PLS SEM) APPROACH IN THE VIEW OF CRITICAL SUCCESS INDICATORS' FAILURE RESEARCH

Harun Çalışkan

Computer Engineering

Thesis Supervisor: Assoc. Prof. Dilek Karahoca

May 2016, 43 Pages

The thesis deals with the agile software development methodology and the critical factors and indicators that lead an agile project to a failure.

One of the software development metholodogies, agile software development (ASD) is an approach for the innovative path that anticipates the demand for flexibility and targets faster and less complicated delivery of the completed project. Agile software development focuses on keeping code simple, testing frequent, and delivering functional parts of the application once and as soon as they are available. The goal of agile software development is to construct small pieces of customer approved applications while the project moves along, instead of delivering the whole product or application once the project is ended.

Technology companies have been trying to apply agile software development methodology in their projects; however, agile exercises are known very little about how effective and efficient they are when compared to the traditional methodologies and what their success or failure factors are. In order to reduce or eliminate project failures which cause money, time and labor loss, project indicators need to be examined which lead to the failure.

The purpose of this dissertation is to identify criticial success factors in agile software development methodology and specifically focus on failure factors and indicators to conclude their significance of relationship and impact so that the possible failures are determined, predicted and exterminated in advance.

The study started by searching the liteature (published in papers, articles and technical reports) to determine the failure factors of agile projects in a multi-dimensional view of failure factors and indicators which were classified into four classifications: organizational, people, process and technical. In addition to that, each and every failure factor was decomposed into a group of sub failure factors and indicators in which this classification helped in obtaining a multi-dimensional view of failure factors that made them more viable.

The data were collected through an online survey and the data collection process yielded useful information for the dimensions defined. The data was analyzed using Partial Least Squares Structural Equation Modeling (PLS SEM) to propose an approach to evaluate the adherence of these failure factors in agile projects. The proposed approach is intended to be a preliminary step to change and improve the adherence of these failure factors in agile projects going forward.

Keywords: Agile Software Development, Critical Success Factor, Failure Indicator, Partial Least Squares Structural Equation Modeling (PLS SEM)

ÖZET

KISMİ EN KÜÇÜK KARELER YAPISAL EŞİTLİK MODELLEMESİ (PLS SEM) KULLANARAK ÇEVİK YAZILIM GELİŞTİRME PROJELERİNDE KRİTİK BAŞARISIZLIK FAKTÖRLERİNİN ANALİZİ

Harun Çalışkan

Bilgisayar Mühendisliği

Thesis Supervisor: Assoc. Prof. Dilek Karahoca

Mayıs 2016, 43 Sayfa

Bu tez çalışması ile çevik yazılım geliştirme metodolojisinin ve çevik yazılım projelerinin başarısız olmasına sebep olan kritik faktör ve indikatörlerin incelenmesi amaçlanmıştır.

Yazılım geliştirme yöntemlerinden biri olan çevik yazılım geliştirme yöntemi, esneklik hususundaki ihtiyacı öngören, tamamlanan ürünün dağıtımı konusuna müşterinin faydasına olacak şekilde katkı sağlayan, yenilikçi ve yaratıcı bir metodolojidir. Çevik yazılım geliştirme yöntemi, yazılım kodunu basit tutmayı, devamlı bir test sürecinin olmasını ve çalışan yazılım parçalarının tamamlanır tamamlanmaz teslim edilmesini esas almaktadır. Çevik yazılım geliştirme yöntemi ile, proje sonunda tek bir büyük parçadan oluşan uygulamanın teslim edilmesi yerine, proje süreci devam ederken, müşteri tarafından onaylanan daha küçük uygulama parçalarının yaratılması ve müşteriye sürekli sunulması hedeflenmektedir.

Bir süredir, teknoloji firmalarının, çevik yazılım geliştirme yöntemini projelerinde uygulamak istemesine karşın, verimlilik acısından çevik yöntemi geleneksel yöntemlerden üstün kılan özellikleri, başarı ve başarısızlık faktörleri ile ilgili yeterli bilgi birikimi bulunmamaktadır. Bu nedenle, proje başarısızlıklarını azaltmak, dolayısıyla, para, zaman ve efor kaybını önlemek için, başarısızlığa neden olabilecek proje indikatörlerinin incelenmesi gerekmektedir.

Bu tezin amacı, çevik yazılım geliştirme yöntemindeki kritik başarı faktörlerini incelemek ve özellikle, başarısızlık faktörleri ve indikatörleri üzerine odaklanarak, bunların başarısızlık üzerindeki ağırlıklarını saptamak ve olası bir başarısızlığın öngörülüp, bununla ilgili önlem alınabilmesini mümkün kılmaktır.

Bu çalışma, geniş bir literatür taraması ile başlamış, (makaleler, teknik raporlar, vb.) araştırma sonucunda başarısızlığa sebep olan faktörler belirlenmiştir. Bu faktörler, literatürde, genel olarak, organizasyonel, insan, süreç ve teknik faktörler olmak üzere, dört farklı kategoride sınıflandırılmaktadır. Ayrıca, her başarısızlık faktörü için bir grup alt faktör belirlenmektedir. Böyle bir sınıflandırma ile başarısızlık faktörlerine çok boyutlu bir bakış açısı sağlanmıştır.

Gerekli olan veri, web üzerinden yapılan bir anket aracılığı ile toplanıp, belirlenen kategorilerdeki genel eğilimin saptanması amaçlanmıştır. Toplanan veri, kısmi en küçük kareler yapısal eşitlik modellemesi (PLS SEM) ile analiz edilmiş ve başarısızlık faktörlerinin çevik yazılım projeleriyle bağlantisi ve ağırlıklı etkisini belirleyen bir yaklaşım sunulmuştur. Önerilen yöntem, başarısızlık kriterlerinin hali hazırdaki durumlarını yansıtırken, bunların etkilerini değiştirmek ve geliştirmek için başlangıç adımı olma özelliği de taşımaktadır.

Anahtar Kelimeler: Çevik Yazılım Geliştirme Süreci, Kritik Başarı Faktörleri, Başarısızlık İndikatörleri, Kısmi En Küçük Kareler Yapısal Eşitlik Modellemesi

CONTENTS

TABLES

FIGURES

ABBREVIATIONS

ASD	:	Agile Software Development
AVE	:	Average Variance Extracted (Average Communality)
CB SEM	:	Covariance-based Structural Equation Modeling
CSF	:	Critical Success Factor
GDP	:	Gross Domestic Product
GoF	:	Goodness of fit
IT	:	Information Technology
LV	:	Latent variable
MV	:	Manifest variable
PLS	:	Partial Least Squares
PLS SEM	:	Partial Least Squares Structural Equation Modeling
PLS PM	:	Partial Least Squares Path Modeling
SEM	:	Structural Equation Modeling
U.S.	:	United States

SYMBOLS

The coefficient of determination (R-square)	:	R^2
Value of manifest variables (MV)	:	MV_i, MV_j

1. INTRODUCTION

The importance of the software in modern world is inevitable and it drives the economic and social activities in an increasing trend which the development of the software significantly contributes to the world economy. The total value of the software product industry has grown, for example, between 1997 and 2012, it increased from 1.7 percent of GDP (Gross Domestic Product) to 2.6 percent and software business directly contributed about $425 billion to U.S. GDP, and in addition to that, employment in the software industry increased almost 40 percent for the same period (from 1.4 million to 2.3 million); and the earnings of these workers on average is almost three times higher than for all other private-sector U.S workers (Shapiro 2014, p. 2).

Whilst software has become critical for business, economy and social activities, the process and management of software development has also been in trouble with the problems which are mostly encountered owing to the fact that software development is complex and costly, multiple stakeholders, teams of experts are involved and complex problems, multiple systems and innovative technologies are being integrated. Moreover, software projects have a high failure rate. The Portland Business Journal (2008) miserably indicates that most researches conclude between 65 and 80 percent of IT projects fail to meet their objectives, and also run significantly late or cost far more than planned.[1] Similary, based on a research by Bloch et al. (2012) with the collaboration of Oxford University, almost 45 percent of large scale IT projects (more than $15 million) exceeds their budget. Moreover, 7 percent of these large IT projects fails to be terminated on time and 56 percent of delivered projects results in the failure of satisfying the expectations from the projects.[2] Dr. Paul Dorsey (2000) summed it up "This is a catastrophe. As an industry we are failing at our jobs". Therefore, criticality, cost, complexity, and an excessively high failure rates require to choose the most

---

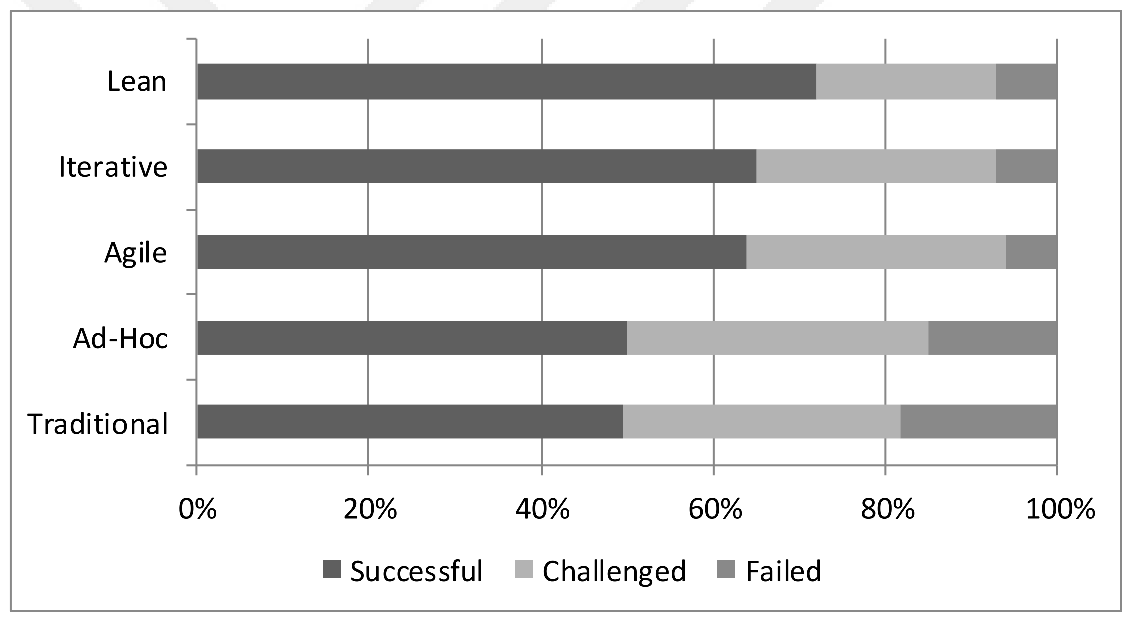
[1] Portland Business Journal, 2008. *Why do most IT projects fail? It's not because of technology.* http://www.bizjournals.com/portland/stories/2008/10/20/smallb4.html [accessed 02 May 2016]

[2] Bloch, M., Blumberg, S., & Laartz, S., 2012. *Delivering large-scale IT projects on time, on budget, and on value*, http://www.mckinsey.com/business-functions/business-technology/our-insights/delivering-large-scale-it-projects-on-time-on-budget-and-on-value [accessed 29 May 2016].

appropriate methodologies for developing software systems to avoid failures as much as possible.

Based on the Ambysoft (2013) IT Project Success Rates Survey with 173 respondents, different software development paradigms were compared and revealed that on average lean, agile and iterative approaches are statistically almost the same in terms of failure rates, and analogously, traditional (waterfall) and ad-hoc (no defined process) strategies are also statistically similar regarding failure rates.[3] This has been depicted in Figure 1.1.

Figure 1.1: Comparing different software development techniques



Source: Ambysoft Scott Ambler Associates, 2013. *IT Project Success Rates Survey*, http://www.ambysoft.com/surveys/success2013.html [accessed 02 May 2016]

As opposed to the waterfall methodology which is described as a sequential design process that includes each stages of software development cycle such as project initialization, analysis, implementation, testing and maintenance are finished before the developers proceed with consecutive steps. This requires that the project outcome and project plan needs to be carefully set in the beginning and followed accordingly.

[3] Scott Ambler Associates, 2014. *Lean and agile software development is more successful than waterfall*. http://scottambler.com/backup_muse/lean-and-agile-software-development-is-more-successful-than-waterfall.html [accessed 02 May 2016]

However, the agile recommends an incremental way that developers commence with a simple project design and study on small units which are targeted to complete during weekly and monthly sprints including project prioritization, test execution, bugs identification and resolution and customer feedback incorporation before the next sprint is continued.

Agile software development started to become popular in the late 1990s, and is now one of the most preferred software development methodologies to be used by organizations, project managers and developers. Agile methodology was proposed inside of 'Agile Manifesto' (Beck and et. al. 2001), which has the principles by means of individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, responding to change over following a plan.

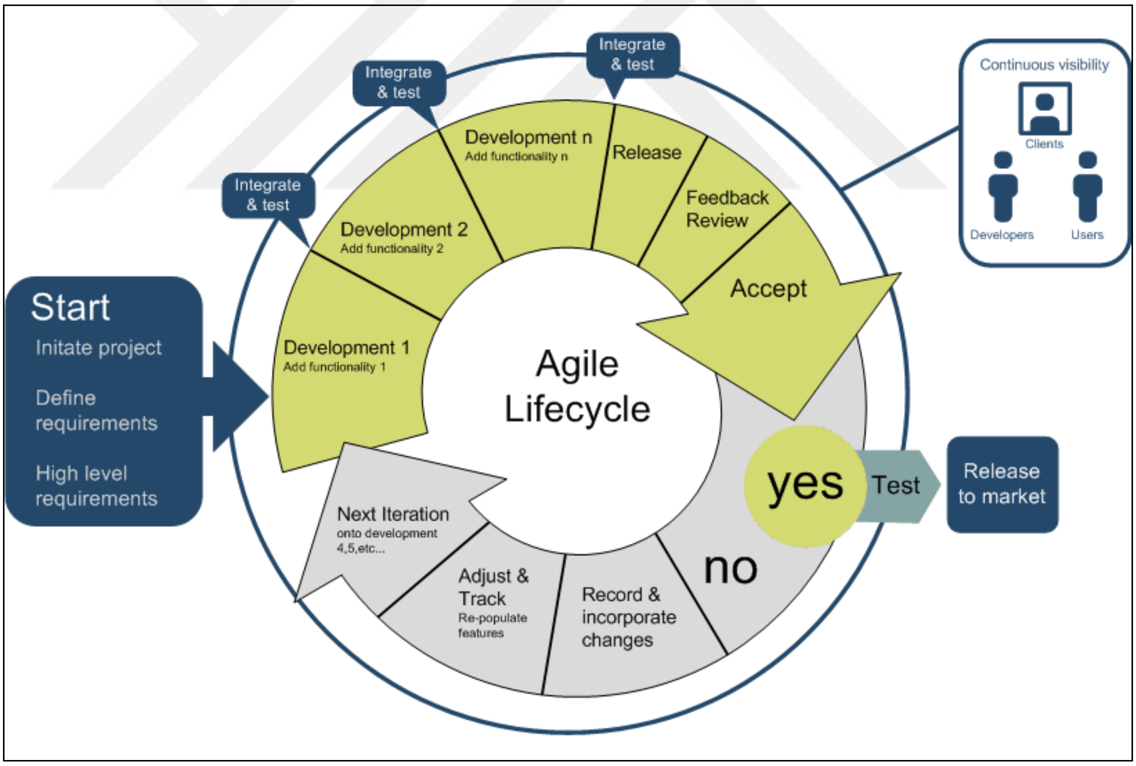
Agile is an aggregate basis of software development which is based on flexible coding, adaptive planning, continuous delivery and testing, collaboration between individuals and among cross-functional teams. Agile software development teams aim to deliver a working application with each and every sprint and to demonstrate it to customer or related people at the end of each sprint. The relationship and communication between team members are more important than using a development tool and pre-defined process. According to agile development process, team spirit drives a project to success. Agile method advocates that the relationship between team and customer is as important as the relationship between team members. Instead of deeply documentation, agile software development prefers working code which is tested periodically. When team member merges their own code into pre-merged code block, agile requests that the test procedures should start automatically, thus, successful code blocks would be ready. The most important characteristic of the agile software development is that the customer can change requirements anywhere during project moves along. Besides, the customer should participate on the project in every phase. As a result, developers and customers should work together by giving feedbacks during the project phase.

The comparison between agile software methodology and traditional (waterfall) plan-driven methodology has been explained in Table 1.1 and the details of agile software methodology has been briefly depicted in Figure 1.2.

Table 1.1: Agile versus plan driven methods

Plan-Driven Methods	Agile Software Methods
Using tools and processes	Focusing individuals and relations
Complete documentation	Functional software
Contact negotiation	Collaboration with the customer
Plan, which must be followed	Changeable plan

Figure 1.2: Agile software development cycle



Source: Software Design Consultants, 2011. *Application development agile.* http://sdc.net.au/services/application-development.aspx [accessed 10 May 2016]

As the projects have high failure rates which are considered to be vital in business life with the loss of money, time and resources, the thesis focuses on agile software metholodogy and aims to determine and deeply investigate the critical factors and indicators which will potentially lead an agile project to fail in order to diminish or avoid the failures in return.

The Critical Success Factor (CSF) approach to determine and evaluate an organization’s performance was first introduced by Rockart (1979, p. 87) and then became well-formed by Rockart and Crescenzi (1984, p. 7). Critical Success Factor is explained by areas to some degree that will ensure positive competitive performance in departments, organizations and for all individuals if the outcome is satisfactory. Critical success factors are key domains where each and every task is needed to be performed by using the right methods and processes in order to achieve and improve the accomplishment and manager’s goals (Nasehi 2013, p.37). This approach has been used to determine the main factors and indicators in the reason of the agile failures. Lack of executive or management support, insufficient experience or skillset of engineers, weak customer relations, inadequate/no involvement of customer and/or strong organizational resistance to the change or new agile methodologies are considered to be prominent failure factors and indicators in agile development. The failure factors has been identified and compiled into four dimensions, which are organizational, people, process and technical, in which each of four dimensions is subsequently separated into more detailed sub-factors.

A survey was conducted among 172 agile professionals, project managers, developers, executives, gathering survey data from different local or global projects and companies. The results are mapped into the failure factors pre-defined and the relationship and effects of these factors and indicators on the failure of agile projects are analyzed and briefly explained via Partial Least Squares Structural Equation Modeling (PLS SEM). The delivery and the quality were chosen as indicators to evaluate the effects of failure factors and indicators in agile projects.

The value of this research is firstly to demonstrate a real life reference from agile specialists worked on different projects and companies and applied agile as software development methodology. In addition to that, failure factors and indicator have been identified and presented with the significance and impacts on the agile projects by all means to guide the companies, project managers and developers appropriately with a feasible solution to decrease or avoid the drawbacks in agile projects and deliver more successful projects or products.

The rest of the thesis is structured as follows:

Chapter 2 Literature Review introduces relevant concepts and research into agile software methodology and critical success factors in the view of failure indicators briefly described in terms of relationship and effect along with PLS SEM analysis.

Chapter 3 Data and Method describes the data gathering and analysis method.

Chapter 4 Findings introduces the outcome of the data analysis presented.

Chapter 5 Discussion and Conclusion presents a summary of the research and findings of the study and concludes the thesis with a report. The significance of this research is examined and recommendations for further research end the chapter.

2. LITERATURE REVIEW

This literature review aims to figure out what previous studies have been done in the area of agile software development methodology and critical factors and indicators that lead the agile projects to fail by searching through books, papers, articles and business reports.

2.1 BACKGROUND

The critical failure factors and indicators are determined to be in a multi-dimensional way, in which they are grouped into four dimensions; organizational, people, process and technical, and moved another step forward to identify sub-factors in each and every of four dimensions.

In this research, due to its value for resulting with strong relationship between variables, the quantitative method has been chosen. For this reason, the participant’s feedbacks and ideas have been crucical to be evaluated in various aspects in this methodology via the research questions that have been formed based on findings from experimental research and literature review. Similarly, the drawbacks, issues and challenges are considered to be the basic focus of this research and that’s why empirical research has been essential among participants to briefly explain the connection. Moreover, previous theories will be considered as a base for the failures and indicators in agile projects and the criticality of these failures and indicators along with the reasons of these problems will be measured through the quantitative data collected via a survey (Nasehi 2013, p. 28). The data collected has been analyzed and evaluated through Partial Least Squares Structural Equation Modeling (PLS SEM) chosen from literature review.

2.2 LITERATURE SEARCH

The literature review around agile software development and critical failure factors and indicators have been listed below.

Agile software development methodology has become popular as it fastly adapts to change and the communication among all participants is effective that also draws the customer onto the team and eventually it provides rapid, incremental delivery of software (Pressman 2009).

Mannila (2013) emphasizes that, as 'Agile Manifesto' (Beck and et. al. 2001) states, individuals and communications are respected in agile development methodology because the ultimate goal is to have a well-operating team consists of skilled people and experts (Cohn 2007, p. 21) and the triumph of a software development project mostly depends on the skills and abilities of each team member. Also, at the end of each sprint, a stable, working and incrementally improved version of a product or application is targeted (Cohn 2007, p. 22). The amount of documentation do not have so much value unless if they contribute to the operational version of a product.

The factors and indicators that cause agile project failures have been reviewed and they are typically based on case studies or surveillances of agile projects and exercises. Vijayasarathy and Turk (2008) designate that lack of training, unfamiliarity with agile approaches, lack of managerial support and interest, resistance from individuals, teams or organization itself are considered to be some of the factors that lead agile projects to fail. Having said that, in parallel to the success factor analysis, Chow and Cao (2008, pp. 962-964) discuss failure factors in four dimensions, namely; organizational, people, process, and technical (Tanner and Willingh 2014) .

The data collection methods in the literature have been reviewed. Data collection process has been defined as gathering the necessary data and information and preparing them for the analysis. There exist different data collection methods, which are simply categorized as qualitative or quantitative. In the qualitative approach, the data can be gathered by facilitating interviews or real life observations; however, for quantitative approach, questionnaire (survey) is a usual way to collect the data to be analyzed further (Nasehi 2013, p. 28).

Chaw and Cao (2008, p. 965) used the quantitative method to gather data via an online survey which was was formed of demographic data collection and 7-point Likert scale questions. The target audience was members of Agile Alliance. Firstly, five members of the target population tested and validated the content and provided their feedback to enhance the survey and then the survey was spread to 83 group coordinators of Agile Alliance user groups and 60 contact people of corporate members of the agile. The survey period lasted for 6 weeks and a total of 408 people responded and 109 projects were submitted with comprehensive data.

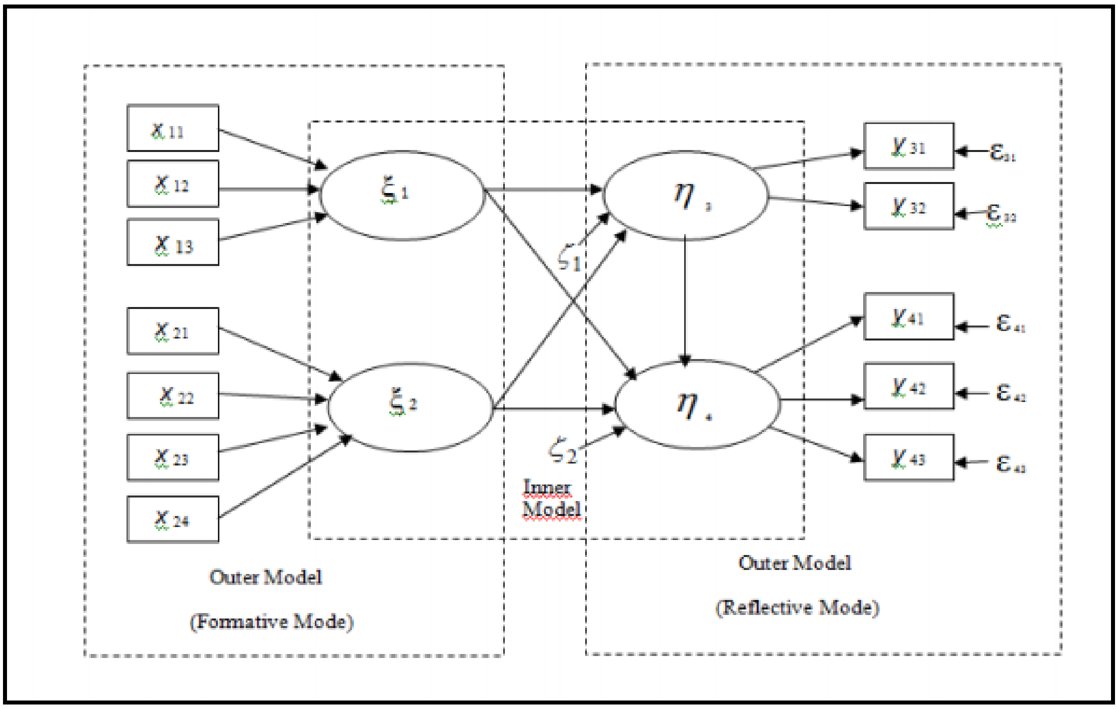
As another example, Stankovic et.al. (2013) collected the data in the study by using an online survey in the form of 7-point Likert scale. The survey was spread to the target audience consisting of managers, developers and experts in former Yugoslavia IT companies. There existed four sections in the survey including demographic or personal data, success factors, insight of success, additional notes and feedbacks. After one-month survey period, 23 complete responses were collected (Stankovic et.al. 2013, pp. 1666).

In this study, feedbacks of customers, managers, developers and testers and all other individuals that were involved in agile projects have been collected in order to have a detailed view with regards to the evaluation of agile failure factors, therefore, a quantitative way of distributing survey was the best option of gathering the data.

The data analysis methods have been reviewed in the literature. The variables that represent measurements obtained from the survey and associated with individuals, companies, situations are analyzed by the application of statistical methods (multivariate analysis). Partial Least Squares Sequential Equation Modeling (PLS SEM) has become a viable alternative to the more popular covariance-based SEM (CB SEM) with its distinctive methodological features and exponentially increasing trend of usage. Specifically, PLS SEM has several advantages over CB SEM in many situations commonly encountered in social sciences research, for example, when sample sizes are small, the data are nonnormally distributed, or when complex models with many indicators and model relationships are estimated (Hair et.al. 2014).

PLS SEM is formally defined by two different linear equations which are the inner model (or structural model) and the outer model (or measurement model). The inner model describes the relationships between latent variables, while the outer model defines the relationships between a latent variable and its manifest variables (observed variables). If a latent variable does not appear as a dependent variable, it is named as an exogenous variable, otherwise, it is an endogenous variable. The combination of inner and outer models leads to a comprehensive partial least squares model (Jamil 2012). Figure 2.1 shows the structure of a PLS SEM model.

Figure 2.1: The structure of partial least squares structural equation model



Source: Jamil, J., 2012. *Partial Least Squares Structural Equation Modelling with incomplete data.*

In this study, four dimensions, organization, people, process, technical, have been mapped to latent variables and failure factors and indicators have been mapped to manifest variables and PLS SEM model analysis has been conducted.

2.3 CRITICAL FAILURE INDICATORS IN AGILE

Lack of knowledge and involvement in agile methods, company culture in conflict with fundamentals of agile methodology, lack of manager or executive support, external force to follow traditional waterfall processes, a wider organizational or communications problem, reluctance of the team to follow agile, insufficient training are considered to be prominent failure factors and indicators in agile development.[4]

Table 2.1 depicts the failure factors and indicators to be analyzed in four dimensions, organizational, people, process and technical (Chow and Cao 2008).

Table 2.1: Failure factors and indicators in agile

Dimension	Main Failure Factor	Sub-Failure Factor
Organizational	Management Commitment	1. Absence of executive sponsorship 2. Absence of management support
	Organizational Environment and Culture	3. Organization is multi-regional and too large 4. Organizational principles excessively political 5. Organizational culture traditional or outdated 6. External pressure to follow traditional waterfall process 7. Unsuitable facility/working environment 8. Locally distributed teams instead of co-location 9. Team sizes are too large
People	Knowledge and Experience	10. Insufficient experience 11. Lack of the required skill set 12. Insufficient project management proficiency
	Team behavior	13. Absence of teamwork 14. Resistance from teams/individuals 15. Weak customer relations 16. Demotivation of team members/team

---

[4] Cunningham, L. , 2015. *8 reasons why agile projects fail.* https://blogs.versionone.com/agile_management/2015/04/09/8-reasons-why-agile-projects-fail/ [accessed 03 May 2016]

Process	Requirements and Planning	17. Imprecise project scope, requirements 18. Inaccurate project planning
	Customer role	19. Vague customer role 20. Absence of customer presence
	Tracking Tools	21. Absence of agile progress tracking methods/systems
Technical	Project, Technology and Tools	22. Unsuitable technology and tools 23. Diversion from coding standards 24. Lack of code review/inspections 25. Insufficient test cases/test coverage 26. Lack of tester in the team (developer is the tester) 27. Lack of technical or customer facing documentation 28. No/Long delivery cycles 29. Unrealistic/short design estimates 30. Insufficient training 31. Absence of developer involvement in prioritization 32. Absence of risk analysis, lessons-learned (retrospective)

Source: Chow T., Cao D., 2008. *A survey study of critical success factors in agile software projects.*

2.3.1 Organizational Factors

Agile implementation is tightly coupled with the organization culture and hierarchy. Any resistance or defense to the agile methodology in the organization may lead to the failure of the projects. Organizational factors are categorized as follows which would potentially cause the failure of an agile project.

2.3.1.1 Absence of management support

The support and commitment of executives and managers are critical in the success of agile development and lack of this deeply causes project failures. If there is no manager or executive support, the team tends to hide from the management in case their effort and project could be terminated ultimately. Lack of executives and management support will bound the visibility into the team’s success and provide inadequate support for acceptance (Tabaka 2015).

Even though organizational culture change over time, once they become set up, they stubbornly oppose to change. In order to facilitate the change and transformation of an organization to embrace and follow agile metholodogies, executives influence the organization most and they need to shape the behavior they want their management team and individuals to demonstrate, reflect the behaviour themselves what they want them to adopt, and help them digest how and why they are crucial in the changing organization. Teams need to be encouraged to take self-initiatives, decisions and cope with the outcome accordingly and this is achievable as long as there exists a strong management and executive level commitment (Cunningham 2015).

2.3.1.2 Organizational environment and culture

Successful agile development in any organization demands workplace cultural transformation. Existing culture, its willingness and ability to apply and improve deeply influence the failure or success of agile methodology and implementation (Ramaraju 2014).

In agile methodology, the manager is in charge of enabling the teams doing the work with their full capacity and ability to create value for customers and eliminate any obstacles that may be faced along the way. The manager trusts in the judgement of the teams in touch with customers as to what work needs to be done and also trusts in the talents doing the work to understand how to do the work in the right way. Agile is outside-in, neither top-down nor bottom-up and the primary focus is on delivering value to customers. The customer is simply the boss, not the manager (Denning 2015).

However, the role of the manager in traditional management is the opposite in which the manager identifies what needs to be done, tells the employee what to do, and then ensures the employee completes the work based on the instructions. The role of the employee is to follow the directions as told and to trust the judgment of the manager to ensure that the right work is being done in the right way. The primary goal of the organization is to make money and the manager is the ultimate boss (Denning 2015).

In organizations where there is a fundamental belief in the effectiveness of the top-down "the manager is the boss" approach where the politics are vital (traditional or outdated organizations) and mostly large organizations, it's hard to implement agile effectively. There is ongoing resistance between the different aims and approaches. Therefore, when implementation of agile is limited to the team level, it risks being inadequate and not operating properly, producing little if any improvement for the organization (Denning 2015).

2.3.2 People Factors

People are heart of the organization and their experience, skillset and their attitude (support or resistance) to essential agile requirements obviously impact the lifetime of the agile implementation in an organization. People factors are categorized below which may cause a failure in agile implementation.

2.3.2.1 Lack of the required skillset

Agile implementations require teams to be equipped with the essential talent and skills to implement and deliver the best solution, including both technical and business skills.

Ideal agile team brings the appropriate number of people together with the required skills to accomplish particular and diversified work tasks that each of them should be assessed respectively. The team is required to have all necessary skillset to cover the project. Preferably every team member must have sufficient skills in every discipline enclosed, otherwise agile implementation would probably fail as the project deadline and delivery might be affected by the lack of expertise or training requirements needed for this purpose (Roberts 2014).

2.3.2.2 Weak customer relations

The customers within an agile environment have critical roles as they are paying not for the work but also for the users of the system and project deliveries (Koch 2005). The customers can be part of the agile team's organization internally or part of an organization outside of the agile team's organization externally.

Within the agile metholodogy and agile teams, the customer has a dynamic and more interactive role and is responsible for providing comprehensive information to the developers. They also prioritize and select the particular requirements for each iteration and take initiative to determine the estimated times. They confirm the expected quality whether a deliverable has met and work closely with the developers to provide any feedback with regards to the development or communicate any changes. The customer is eventually responsible for any final decisions on the project scope and timelines (Koch 2005).

The criticality of customer involvement is apparent to avoid developer assumptions and biases for the design and final product. Any ignorance of the customer role in the agile implementation or weak relationships with the customer along the way could have led to rejection of the deliverable.

2.3.2.3 Absence of team work

The mindset and the attitude of the individual team members in the way of everyone sharing a team objective and a team success are considered to be one of the key drivers of a successful agile team (Roberts 2014). Communication plays a critical role in the implementation of agile methodologies and the communication between developers, operations, support, customers and management is required to be fast, honest, direct and effective in which agile looks for ‘How can I help you here?’ attitude, rather than the ‘That’s not my problem’ attitude (Ghahrai 2015).

2.3.2.4 Resistance from teams or individuals

If the members of a team keeps identifying themselves by function or if there is a team member with strong personality who insists on keeping his/her position at the top, the resistance tends to happen that causes an observed loss of identity or control without a doubt. In order to remedy these impediments, executive leadership's inspiration on the management team and the culture, detailed training, and team-level mentoring are essential. Otherwise, reluctance of the team to follow agile will cause failures in the projects (Cunningham 2015).

2.3.3 Process Factors

Process factors are categorized as below which may cause a failure in agile implementation.

2.3.3.1 Imprecise project scope and inaccurate planning

Agile methodology benefits from clear definition of scope and objectives, even though details are allowed to arise during the development. That's the nature of software that agile projects expect requirements not to be complete and to be changing and instead of reacting this, agile projects allow requirements to arise and change (Waters 2007).

The ongoing churn and expansion of the requirements, tied with poor prioritization, makes it hard to deliver the most important functionality on time or schedule. This demand for ever-increasing functionality leads to delays, quality problems, loss of focus and eventually leads to a failure of agile projects (Wiegers 2013).

2.3.3.2 Absence of customer or vague customer role

In agile methodology, teams incorporate the customers by having a person who is authorized to determine the user stories, set priorities and to provide feedbacks or answer questions in order to refine the real project requirements. In an ideal situation, the customer is co-located with the agile team, even though a better approach is to locate the team with the customer. It is especially important for an agile project due to the hands-on approach needed. That's why the customer should be located with the agile team in order to facilitate communication and become fully dedicated to the project which will reduce the chances of the failure. [5]

2.3.3.3 Absence of agile process tracking methods

The use of agile tools provides a clear view to the customer on what the exact priorities are and what they can expect at the end of the sprint. Besides they measure the productivity and resource usage along the way, absence of or inappropriate tools would cause the failure of the agile methodology as the teams are lacking the ability to track of the project and productivity (Erickson 2013).

2.3.4 Technical Factors

Existing agile methods could benefit from using a more processed approach across the entire implementation process in terms of development, production and delivery phases. The main benefits of adopting such an approach include improved communication of the requirements and better support for feedback and progress tracking.

Some organizations or teams use agile principles so strictly that they don’t see the true aim behind the exercise. Alternatively, some teams tailor agile models in a way that entirely lost any agile meaning, for instance, the daily stand-up meeting turns into a status meeting instead of a re-planning meeting; the sprint planning meeting is treated as a story assigment meeting, where developers move away from being creative problem solvers; and agile behaves very similarly to the traditional waterfall approach and this puts companies in danger to believe that agile exercises are no different from any other common practices (Singh 2013).

Agile methodology suggests less or no documentation or ability to track progress and it comes to the reality that available agile tools are important so that they help track both

[5] Directions, 2015. *The Customer’s Role in Agile Project Management.* http://blog.directionstraining.com/event/the-customers-role-in-agile-project-management?doing_wp_cron=1462315355.3164238929748535156250 [accessed 02 May 2016]]

team and individual productivity, allow faster and reliable software creation, give the ability to respond to change, use resources and help drive customer collaboration. In addition to this, agile is originated on simplicity and the tools used should reflect this. Absence of or unsuitable tools would cause the failure of the agile methodology as the teams are lacking the ability to track of the project and productivity (Erickson 2013).

3. DATA AND METHOD

In this section, the method for collecting and then analyzing the data will be briefly described.

3.1 DATA COLLECTION

Data was gathered with the use of an online survey that was spread to the target audience consisting of executives, managers, developers and customers in Turkey IT companies. The purpose of this survey was to validate the impact of key failure factors on the continued usage of agile methods on real life projects.

The survey was made up of three sections:

The Section 1 was on personal data and information about the respondent (age, gender, experience, job title, role in agile teams) and the agile projects that were participated into in terms of the size and complexity of the projects and teams. Additionally, the Section 1 was focused on personal influencers related to the respondent’s perception or belief on the teams that they worked in or are working with currently on agile projects.

The Section 2 was on the failure factors contributing the agile projects to fail based on the four dimension, organizational, people, process, technical, that were extracted from the literature review. To measure the identification and seriousness of failure factors and the insight of failure, a five point (5-point) Likert scale was used.

The Section 3 was reserved for additional notes and comments, where respondents were encouraged to enter any feedback or thought on a free text area.

The survey included 64 questions in total including validity and cross-check questions. The section 1 had 14 questions to gather demographic data and personal information and perception or belief about the respondents on their agile projects and experience.

The Section 2 had 50 questions to identify and measure the failure factors and indicators leading the agile projects to fail.

Table 3.1 shows the structure of the survey and questions, corresponding references to these questions and the items or dimensions covered. Also, “see. Appendix-1, Table-1” for the survey questions that were asked to the applicants.

Table 3.1: Survey questions, corresponding references and covered items

Covered item	Questions	Details	Adopted from
SECTION 1: Company Data and Personal Information			
1.1. Personal Information	Q1-9	Aims to gather personal information of the respondents, such as age, gender, experience, job title, agile role, the size and complexity of the projects and teams they are involved in.	(Senapathi and Srinivasan 2013).
1.2. Personal Influencers	Q10-14	Aims to gather personal influencers related to the respondent’s perception or belief on the projects and teams.	(Senapathi and Srinivasan 2013).
SECTION 2: Agile Methodology Failure Factors			
2.1. Organization Dimension	Q15-26	Aims to gather respondent’s feedback or belief on organizational factors such as management	(Abrahamsson et al. 2002). (Darwish and Rizk 2015). (Misra et al. 2009).

		commitment, organizational environment and culture, etc.	(Tanner and Willingh 2014). (Worren 2010).
2.2. People Dimension	Q27-37	Aims to gather respondent's feedback or belief on people factors such as knowledge and experience, required skillset, team behavior, resistance from the team, etc.	(Chow and Cao 2008). (Mannila 2013). (Martin 2003). (Sidky et al. 2007). (Worren 2010).
2.3. Process Dimension	Q38-41	Aims to gather respondent's feedback or belief on process factors such as requirement and planning, customer role and involvement, tracking tools, etc.	(Chow and Cao 2008). (Mannila 2013). (Martin 2003).
2.4. Technical Dimension	Q42-64	Aims to gather respondent's feedback or belief on technical factors such as technology used, coding, testing, design estimates, delivery cycles, retrospectives, training, etc.	(Abrahamsson et al. 2002). (Chow and Cao 2008). (Darwish and Rizk 2015). (Jugdev and Muller 2005). (Mannila 2013). (Martin 2003). (Sidky et al. 2007).

SECTION 3: Additional Comment			
	Free format text area	Aims to gather additional feedback or thought on the survey or agile practices.	(Senapathi and Srinivasan 2013).

As part of the trial of the survey to test content accuracy, readibility and understandability, 10 people provided their feedback to improve the survey and the feedback was integrated before the online survey was spread to and shared with the applicants.

Survey period lasted 2 months and a total of 172 people (124 male, 48 female) responded to the online survey. The average years of experience of the respondents in software development was 6.4 years and the average years of agile experience was 3.3 years. The average number of agile projects involved in by the respondents was 9.6.

Table 3.2 displays the breakdown of the roles of the respondents on agile projects submitted, while Table 3.3 displays the size and length of the projects respectively.

Table 3.2: Agile roles in survey

Agile Role	**Frequency**	**Percentage**
Designer	81	47.09%
Scrum Master	28	16.28%
Project Manager	27	15.70%
Tester	22	12.79%
Business Analyst	11	6.40%
Agile Coach	3	1.74%

Table 3.3: Agile project size and length in survey

Project (Size, Length)	Frequency	Percentage
Small (project length of 3–6 months/10–20 people)	79	45.93%
Medium (project length of 6 months-1 year/20-30 people)	39	22.67%
Large (project length of more than one year/30+ people)	28	16.28%
Very small (project length less than 3 months/less than 10 people)	26	15.12%

3.2 DATA ANALYSIS METHOD

The questions on the Section 2 of the survey intended to measure the failure factors and indicators of agile development methodology and their impact to the agile projects. 5-point Likert scale was used to codify the questions to be analyzed using PLS SEM model. The respondents' feedback and insight was simply evaluated by asking to which extent they agree or disagree with a particular statement. The 5-point Likert scale was chosen in the survey to be formed with the statements; Strongly Disagree, Disagree, Neither agree or disagree, Agree, Strongly Agree and they were codified by assigning 5 to the highest statement 'Strongly Agree' and assigning 1 to the lowest statement 'Strongly Disagree'. This was logically applied to the positive (affirmative) guided questions, however, for a few negative guided questions (e.g. developers don't need to be experienced with the required skillset?), the reverse mapping was applied in which 5 point was assigned to 'Strongly Disagree' and 1 point was assigned to 'Strongly Agree'.

Figure 3.1: Five point likert scale used in the survey

Strongly Disagree 1	2	3	4	Strongly Agree 5
○	○	○	○	○

Source: Sauro J., 2011. *How to interpret survey responses*

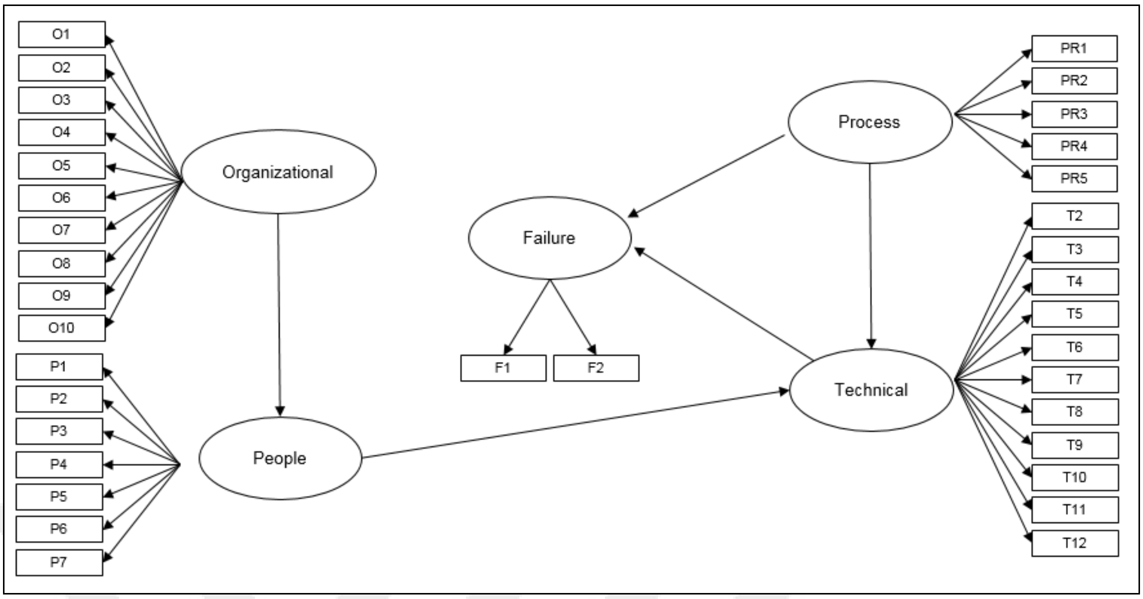
Then, the questions were associated and mapped to the main failure factors in pre-defined four categories (dimension), organizational, people, process, technical, and to the sub-failure factors in each of them. For instance, question 17 aimed to figure out people's idea and feedback on the impact of absence of management support under organization dimension; similarly, questions 13, 28, 52 were linked to the insufficient experience under people dimension. In the case of multiple questions logically associated with the same main factor and same sub-failure factor, in order to reflect all answers for more accuracy, the average (arithmetic mean) of all these answers was evaluated and assigned to the failure indicator (e.g. 4.3 point).

The same evaluation was applied to all questions in the survey, Section 1.2 Personal Influencers and Section 2 Agile Development Methodology Failure Factors, for all respondents (172) in the survey.

The data collected from the survey was decided to be analyzed using PLS SEM or PLS Path Modeling (PLS PM) based on the literature review. It is a statistical method for modeling complex relationships (structural equation models) among latent variables and manifest variables (observed variables).
PLS SEM (or PLS Path Modeling) was also used to display the model in a graphical format using what is called a path diagram that represents in a visual way the relationships stated in the model (Sanchez 2013).

In this study, the failure factors and indicators that were pre-defined in four dimensions and measured by using online web survey were presented in PLS PM diagram to construct the relationship between these main failure factors (organization, people, process, technical) and their sub-failure factors and their impact on the agile project failures (Tenenhaus et.al. 2004). Figure 3.2 depicts the initial construction of PLS PM diagram that leads agile projects to fail.

Figure 3.2: Initial construction of PLS PM diagram



In the PLS SEM modeling, 5 latent variables (LV) were defined in which 4 of them were defined as main failure factors and indicators, organizational, people, process, technical, and last latent variable, defined as failure, was main target variable (described as a combination of quality and delivery) which identified the agile projects to fail.

Each latent variable (concepts that cannot be directly measured) was linked to one or more manifest variables (MV) that were evaluated as shown in Table 3.4. For instance, for the People latent variable (LV), 7 manifest variables were:

P1. Insufficient experience

P2. Lack of the required skill set

P3. Insufficient project management proficiency

P4. Absence of teamwork

P5. Resistance from teams or individuals

P6. Weak customer relations

P7. Demotivation of team members/team

Table 3.4: Latent variables and associated manifest variables

Latent Variable	Manifest Variable	Description
O (Organization)	O1	Absence of executive sponsorship
	O2	Absence of management support
	O3	Organization is multi-regional and too large
	O4	Organizational principles excessively political
	O5	Organizational culture traditional or outdated
	O6	External pressure to follow traditional process
	O7	Inability to embrace the failure
	O8	Unsuitable facility/working environment
	O9	Locally distributed teams instead of co-location
	O10	Team sizes are too large
P (People)	P1	Insufficient experience
	P2	Lack of the required skill set
	P3	Insufficient project management proficiency
	P4	Absence of team work
	P5	Resistance from teams/individuals
	P6	Weak customer relations
	P7	Demotivation of team members/team
PR (Process)	PR1	Imprecise project scope and requirements
	PR2	Inaccurate project planning
	PR3	Vague customer role
	PR4	Absence of customer presence
	PR5	Absence of agile progress tracking methods
T (Technical)	T2	Unsuitable technology and tools
	T3	Diversion from coding standards
	T4	Lack of code review/inspections
	T5	Insufficient test cases/test coverage
	T6	Lack of tester in the team
	T7	Lack of documentation
	T8	No/Long delivery cycles

	T9	Unrealistic/short design estimates
	T10	Insufficient training
	T11	Absence of developer involvement in prioritization
	T12	Absence of risk analysis, lessons-learned
F (Failure)	F1	Code quality
	F2	Frequent code delivery

In this study, each and every manifest variable (MV) questions were transformed into new normalized values on a 0-100 scale in which the minimum value of MV was 0 and its maximum value was equal to 100 (Tenenhaus et.al. 2004, p. 161). The initial value of a manifest variable (MV_i) ranked from 1 to 5, was transformed into a new normalized manifest variable (MV_j) explained in the 'Equation 3.1' below.

$$MV_j=(MV_i-1)*(100/4) \qquad \textbf{(3.1)}$$

For example, if MV_i had the value of 4, its new value was 75 in 0-100 scale, and if MV_i had the value of 3.3, its new value was calculated as 55 respectively (Tenenhaus et.al. 2004, p. 161).

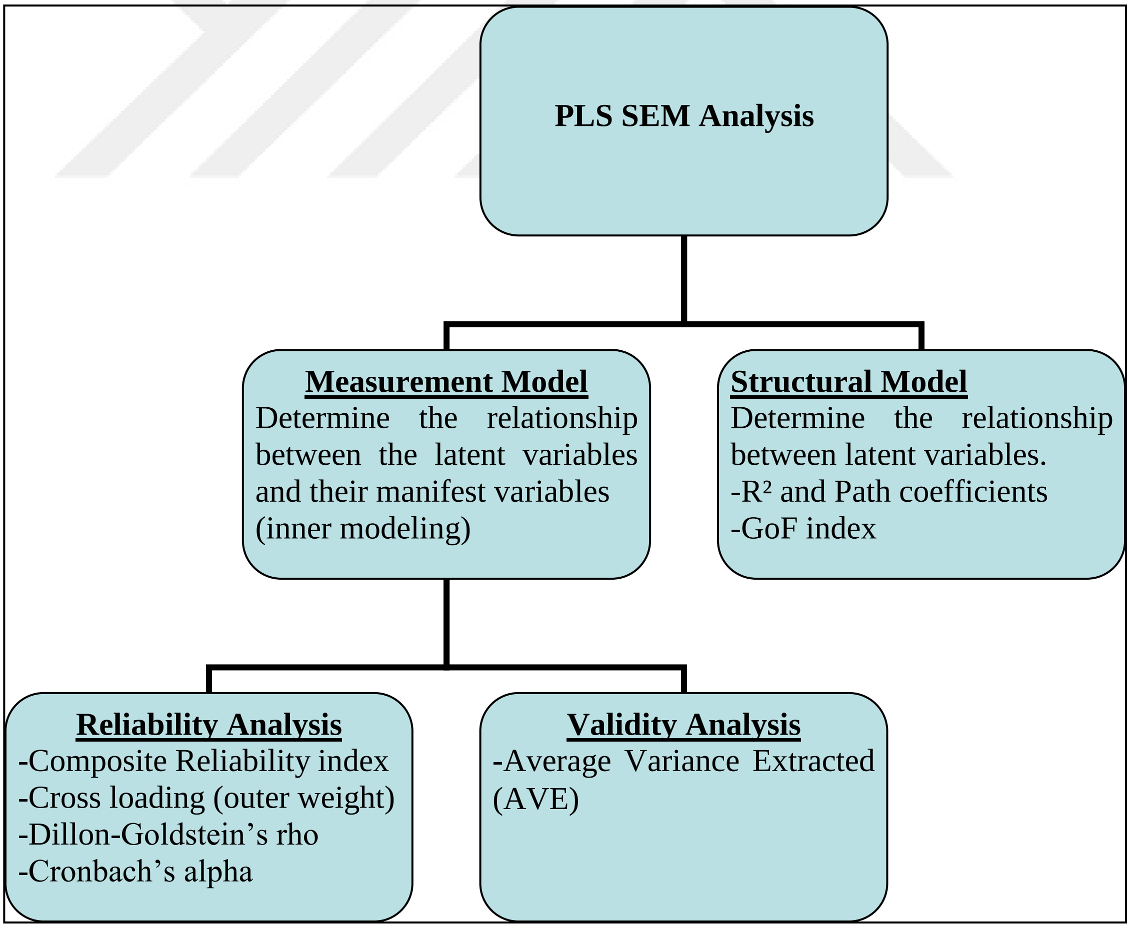
The detailed investigation and findings from PLS SEM Modeling will be covered in the next chapter.

4. FINDINGS

In this section, the outcome of the PLS SEM modeling on the defined critical failure factors and indicators contributing the agile projects to fail will be described. The data modeling and analysis has been performed using XLSTAT software program delivered by Addinsoft.

PLS SEM modeling has been analyzed in the view of measurement model (reliability and validity analysis) and structural model analysis. The Figure 4.1 depicts the high level view of PLS SEM modeling, measurement and structural model analysis and the used metrics to explain them respectively.

Figure 4.1: High level diagram of PLS SEM modeling



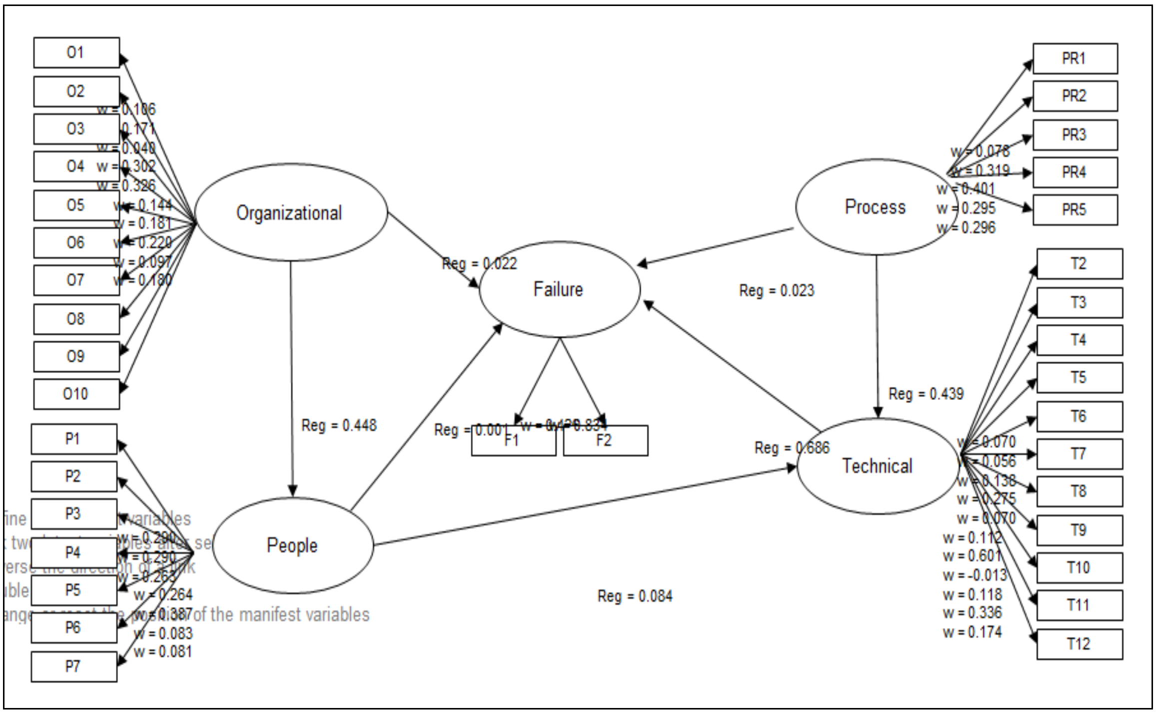
4.1 MEASUREMENT MODEL: RELIABILITY AND VALIDITY

The measurement model of SEM simply aims to determine the relationship between the latent variables and their observations which are called manifest variables. In the model, multiple manifest variables make it possible to assess reliability and consistency. If the the correlation between multiple manifest variables of a given latent variable is higher, it means more consistency or reliability of the indicators (or manifest variables).

In order to evaluate the reliability analysis, composite reliability and outer weights (loadings) are used for each and every latent variable to assess the consistency of the indicators (manifest variables).

The following Figure 4.2 depicts the first result of the initial PLS SEM diagram evaluated on the latent variables and associated manifest variables from the real life survey to measure the impact and factors that lead the agile projects to fail.

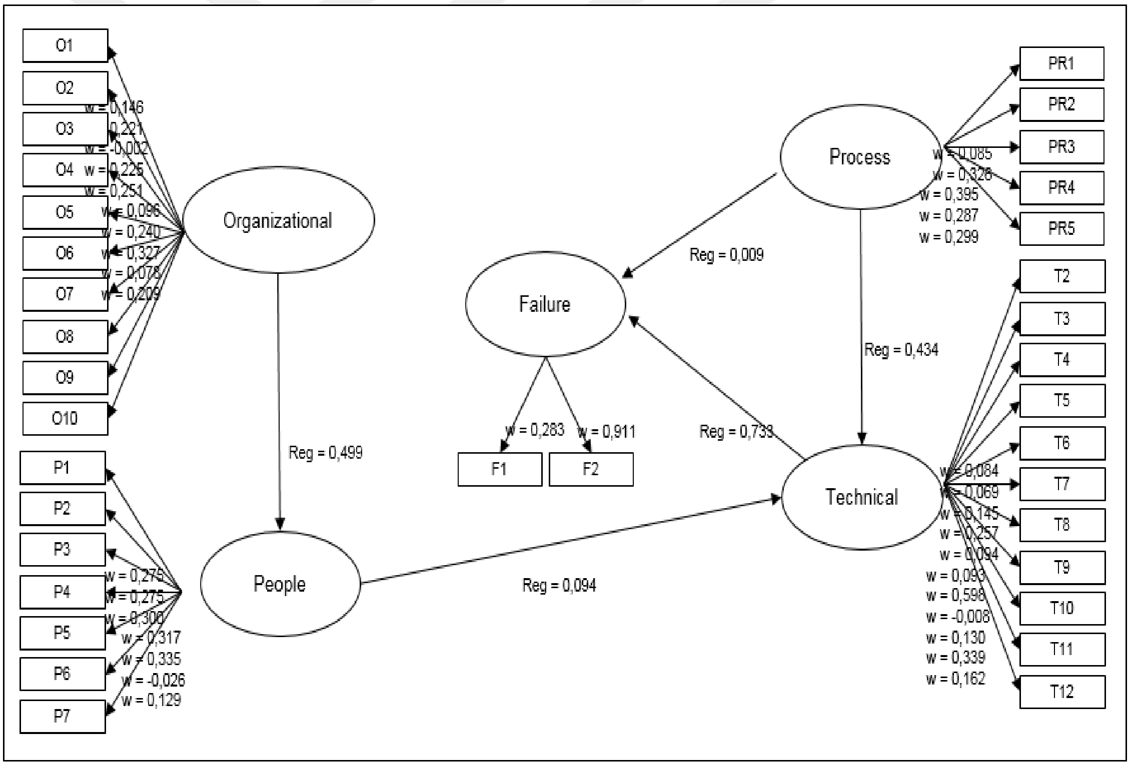
Figure 4.2: Initial PLS SEM diagram on agile failure factors



In the first PLS SEM diagram in which all latent variables were connected to the Failure latent variable, the execution of the model did not produce satisfactory results to continue with the model (GoF value was 0.302, R^2 value of the Failure latent variable was 0.502, AVE values of all latent variables were lower than 0.5). Even the model did not yield better results once the lower weighted manifest variables were dropped.

Thus, the model has been modified in the following Figure 4.3 (also shown in Figure 3.2) that reveals better evaluation and satisfactory outcome on the latent variables and associated manifest variables to measure the impact and factors that lead the agile projects to fail.

Figure 4.3: Modified PLS SEM diagram on agile failure factors



On PLS SEM Measurement modeling and realiability analysis, Composite Reliability indexes were investigated firstly which assess the internal consistency of a measure. Based on the initial Cross Loadings table (shown in Table 4.1), some manifest variables of corresponding latent variables had lower weights (less than or around 0.5); thus, they were dropped from the initial model. For the Organization latent variable, O1, O2, O3, O7, O9, O10 were removed. Similarly, for the People latent variable, P3, P6, P7 were dropped and for the Process latent variable, only PR1 was removed since other manifest variables were very higher than 0.5. For the Tehnical latent variable, T2, T3, T4, T5, T6, T7, T9, T10, T12 were dropped as their values were very lower than 0.5.

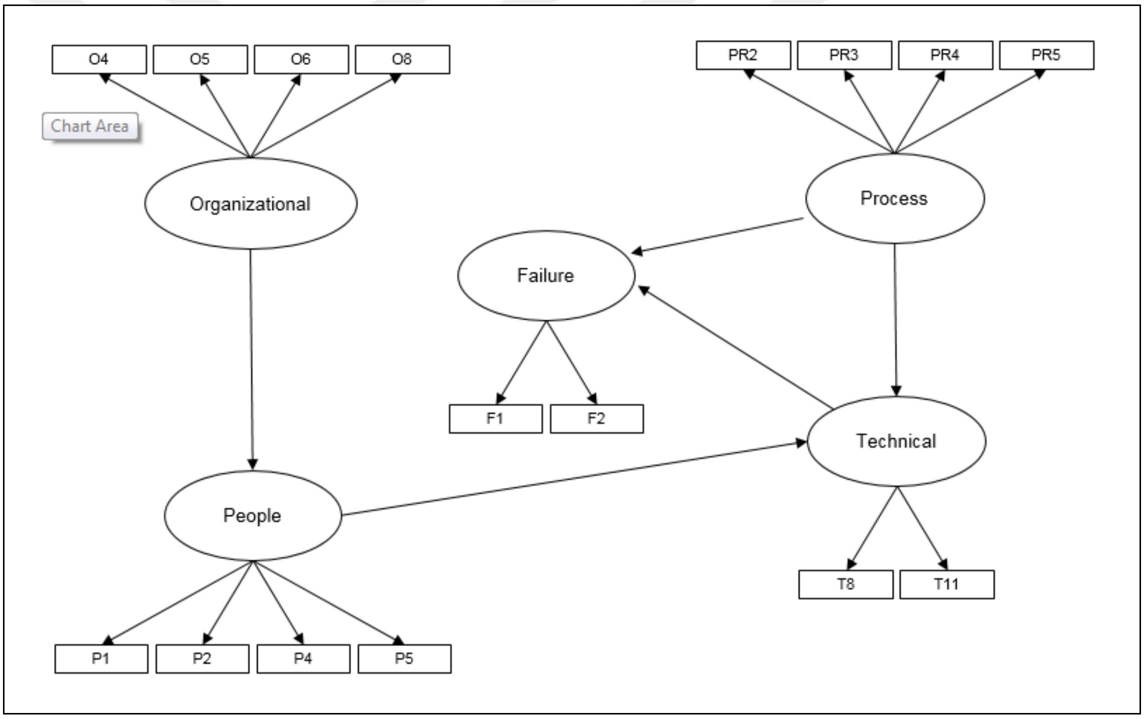
Table 4.1: Cross loadings of manifest variables

Latent variable	Manifest variables	Cross loadings
Organizational	O1	0,519
	O2	0,392
	O3	0,051
	O4	0,698
	O5	0,618
	O6	0,534
	O7	0,517
	O8	0,655
	O9	0,305
	O10	0,453
People	P1	0,728
	P2	0,728
	P3	0,510
	P4	0,638
	P5	0,624
	P6	0,023
	P7	0,276
Process	PR1	0,443
	PR2	0,766
	PR3	0,768
	PR4	0,727
	PR5	0,671
Technical	T2	0,251
	T3	0,251
	T4	0,362
	T5	0,456
	T6	0,216

	T7	0,271
	T8	0,764
	T9	-0,003
	T10	0,401
	T11	0,547
	T12	0,321

After repeatedly removing the low weights from the latent variables, Figure 4.4 displays the redefinition of the model.

Figure 4.4: Redefinition of PLS SEM diagram



Latent variables are represented by blocks which needs to be unidimensional and this is the reason why composite reliability of these latent variables (blocks) needs to be verified. There exist two different measures to test and verify the unidimensionality of the latent variables (blocks) in PLS SEM structure, Dillon-Goldstein's rho and Cronbach's alpha (Balzano and Trinchera 2010, pp. 61-62). Chin (1998) states that Dillon-Goldstein's rho is a better indicator than Cronbach's alpha since it uses loadings and other results from the model itself instead of the correlations between the pre-defined manifest variables. If the value is greater than 0.7, the latent variable (block) is reflected to be homogenous (Werts et.al. 1974).

According to the Composite Reliability indexes (explained in Table 4.2), Dillon-Goldstein's rho results were greater than 0.7 and the first Eigenvalues of each latent variable were bigger than the others, thus each and every latent variable block consisted of manifest variables is verified to be unidimensional. In other words, the reliability values of this model were satisfactory and moderately affected the model. If the Cronbach's alpha values were examined, the most effective variable (the highest score) was found to be the organizational variable, which was 0.805.

Table 4.2: Composite reliability table

Latent var.	Dim.	Cronbach's alpha	D.G. rho (PCA)	Cond. number	Critical value	Eigenvalues
Organizational	4	0,805	**0,877**	+Inf	1,333	2,595
						0,789
						0,616
						0,000
People	4	0,698	**0,817**	+Inf	1,333	2,221
						1,094
						0,685
						0,000
Process	4	0,720	**0,827**	3,197	1,000	2,181
						1,276
						0,329
						0,213
Technical	2	0,273	**0,733**	1,173	1,000	1,158
						0,842
Failure	2	0,297	**0,740**	1,192	1,000	1,174
						0,826

In order to evaluate the validity analysis, converged validity was used which reflected to which extent the manifest variables (measurements) were related to the latent variable (construct). Table 4.3 demonstrated the weights of the relations between each and every manifest variable and its own latent variable, and the average communality (AVE) index to which extent each LV (latent variable) described its own MV (manifest variables). Average Variance Extracted (AVE) assesses convergent validity (Fornell and Larcker 1981) and if the value of AVE is equal to 0.5, the latent variable can describe more than 50% of variance of its variables (Götz et al. 2010). Since this index was higher than 0.5 for each and every latent variable in the model, it was concluded that converged validity was established and all the latent variables were influental at describing their own manifest variables (Balzano and Trinchera 2010, pp. 62-63).

Table 4.3: Weights and average communalities

Latent variable	Manifest variables	Normalized Outer weights	Average Communality (AVE)
Organizational	O4	0,337	0,630
	O5	0,379	
	O6	0,140	
	O8	0,383	
People	P1	0,306	0,519
	P2	0,306	
	P4	0,312	
	P5	0,478	
Process	PR2	0,307	0,541
	PR3	0,433	
	PR4	0,316	
	PR5	0,297	
Technical	T8	0,818	0,571
	T11	0,460	
Failure	F1	0,260	0,556
	F2	0,922	

The normalized weight evaluates the impact of the corresponding manifest variable in measuring the latent variable score and the standardized loadings (Balzano and Trinchera 2010, p. 63). Based on the scores in the model, it was obvious that, for example, the manifest variable T8 (No/Long delivery cycles) was the most crucial contributor in evaluating the Technical LV. Similarly, the P5 MV (Resistance from teams or individuals) was dominantly the driver in evaluating the People LV and the PR3 MV (Vague customer role) was the most significant contributor in the Process LV. For the Organizational LV, the two manifest variables were directly linked to O5

(Organizational culture traditional or outdated) and O8 (Unsuitable facility/working environment) manifest variables almost equally.

4.2 STRUCTURAL MODEL

After completing the measurment model along with reliability and validity analysis, the structural model was evaluated which explained the relationship between latent variables. The key metrics and evaluation criteria for the structural model are R^2 values and path coefficients.

Firstly, the entire model was evaluated by using all data from the survey. Then the data was filtered out by agile experience of the respondents' and the filtered data was tested if there was any difference betwen experienced and inexperienced people in terms of agile critical failure factors.

4.2.1 Entire Model Analysis By Using All Data In The Survey

Table 4.4 demonstrated the results of the structural model estimates for the model using all data from the survey.

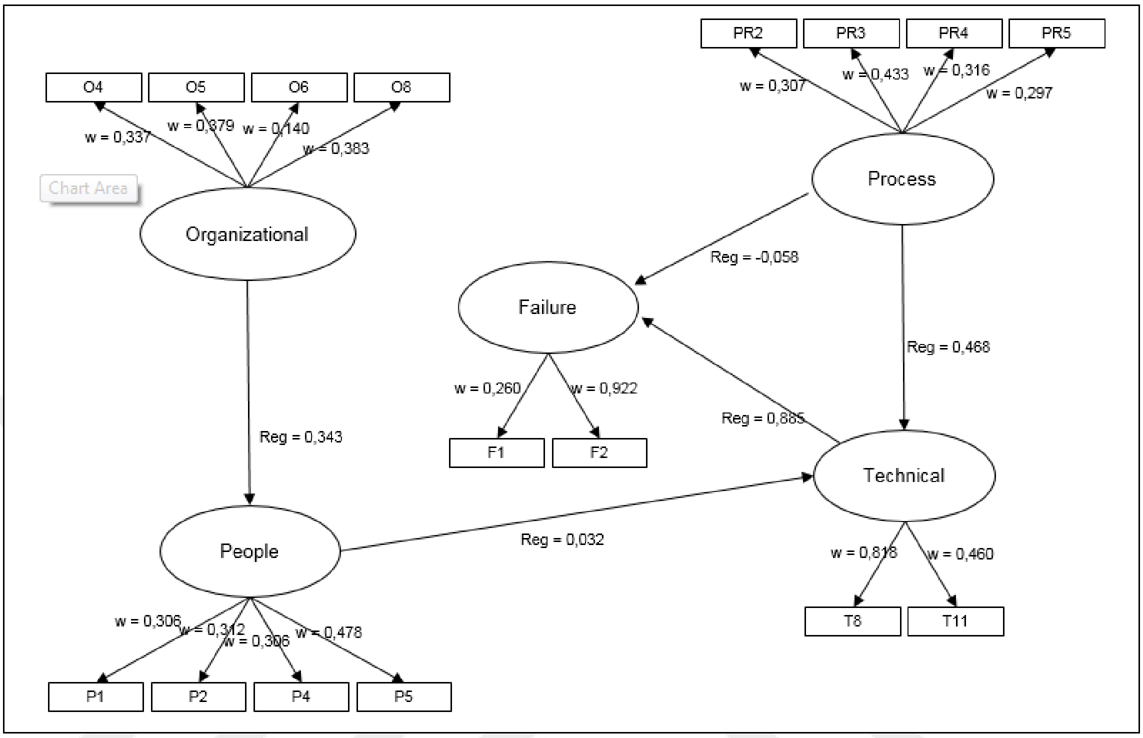
Table 4.4: The result of structural model assessment and R^2 values

Latent variable	Type	R^2	Adjusted R^2	Mean Communalities (AVE)	D.G. rho
Organizational	Exogenous			0,630	0,869
People	Endogenous	0,118	0,118	0,519	0,811
Process	Exogenous			0,541	0,824
Technical	Endogenous	0,232	0,228	0,571	0,718
Failure	Endogenous	0,737	0,736	0,556	0,684
Mean		0,363			

The R^2 values of endogenous (dependent) latent variables were 0,118 (People), 0,232 (Technical) and 0,737 (Failure) as illustrated in Table 4.4.

The path coefficients of the entire model was shown in Figure 4.5 below.

Figure 4.5: Path coefficients of the entire model



In order to explain the critical failure factors and indicators in the model using Failure latent variable, Table 4.5 showed the correlation coefficients connecting the Technical and Process latent variables. With regards to the path coefficients on Table 4.5, failure mainly depends on Technical factors (path coefficient= 0.885) while Process factors have negative and lower effects (path coefficient= -0,058).

Table 4.5: The failure LV

	Technical	Process
Correlation	0,857	0,368
Path coefficient	0,885	-0,058
Correlation * path coefficient	0,759	-0,021

In Table 4.6, R^2 which was defined as a coefficient of determination was 0.737 and could be considered to be substantial (Hair et al. 2011).

Table 4.6: R^2 on the failure LV

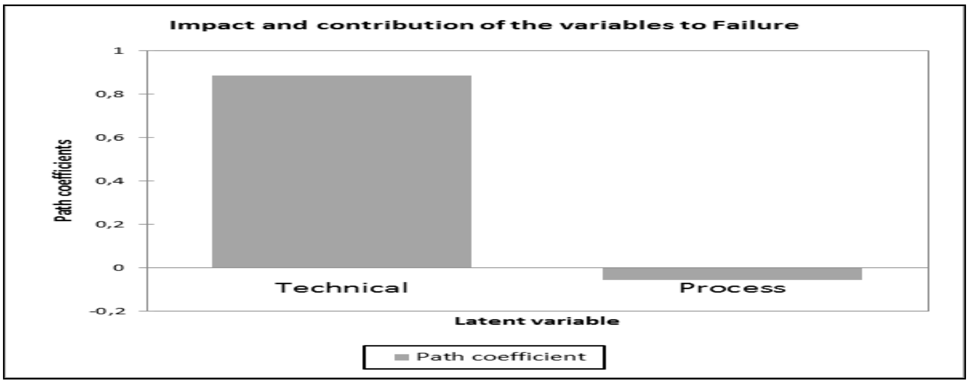
R^2	R^2(Bootstrap)	Standard error	Critical ratio (CR)	Lower bound (95%)	Upper bound (95%)
0,737	0,748	0,066	11,220	0,616	0,874

Based on the results in Table 4.5, the Failure latent variable may be evaluated as indicated in ‘Equation 4.1’.

$$\text{Failure} = -5{,}772*\text{Process}+0{,}884*\text{Technical} \qquad \textbf{(4.1)}$$

Having said that, for the Failure factor (latent variable), the most contribution significantly belonged to Technical factors, other measure’s contribution (Process factor) was considerably low as illustrated in Figure 4.6.

Figure 4.6: Impact of other LVs on the failure LV



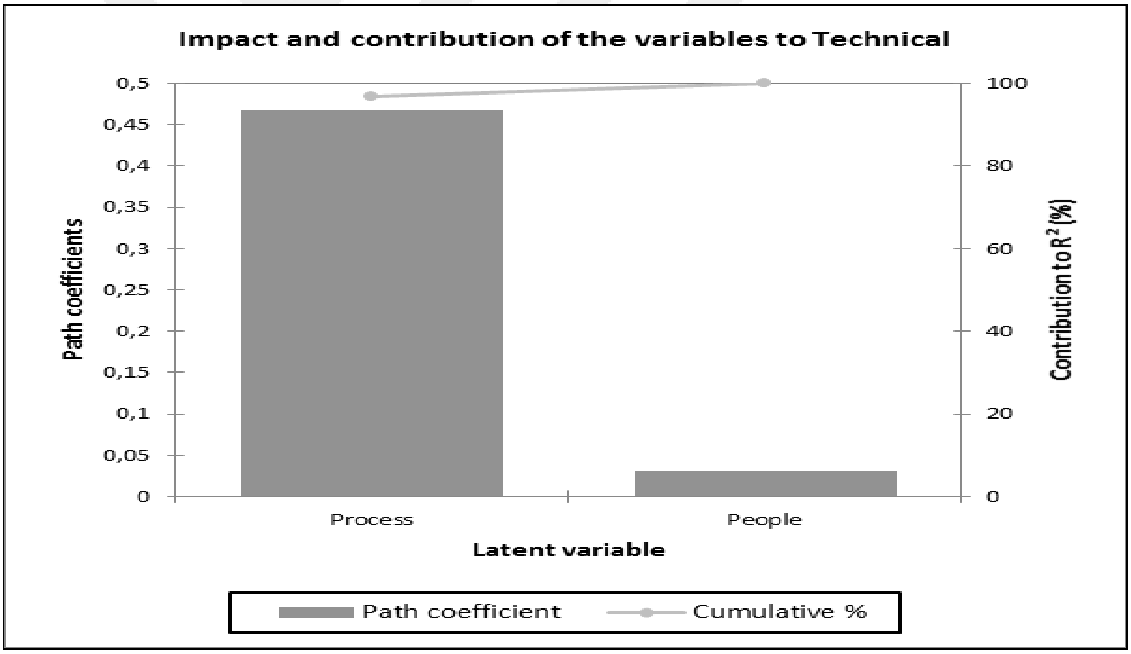
Similarly, if the Technical endogenous latent variable is investigated further, with regards to the path coefficients on Table 4.7, Technical factor mainly depends on Process factors (path coefficient= 0.468) while People factors have low effect (path coefficient= 0,032).

Table 4.7: The technical latent variable

	Process	People
Correlation	0,481	0,229
Path coefficient	0,468	0,032
Correlation * path coefficient	0,225	0,007

In other words, for the Technical factor (latent variable), the most contribution belonged to Process factors (contribution to R^2 is higher than 95 percent), other measure's contribution (People factor) was low (contribution to R^2 is almost 3 percent) as illustrated in Figure 4.7.

Figure 4.7: Impact of other LVs on the technical LV



The Goodness of Fit (GoF) index identifies the overall covariance between the manifest variables, which is evaluated by the default model (Sarstedt et al. 2014). By looking at the results depicted on Table 4.8, absolute GoF index value was calculated as 0.452, which was an acceptable value in a real case model. The relative GoF index value was evaluated as 0.862 which could be considered very high.

Table 4.8: GoF index table

	GoF	GoF (Bootstrap)	Standard error	Lower bound (95%)	Upper bound (95%)
Absolute	**0,452**	0,455	0,032	0,386	0,528
Relative	0,826	0,849	0,059	0,735	0,985
Outer model	0,980	0,971	0,041	0,871	1,000
Inner model	0,843	0,875	0,047	0,795	0,992

4.2.2 Model Analysis By Agile Experience

After evaluating the entire model using all data from the survey, the model was tested against if there were any significant differences based on the agile experience of the respondents'. The data was splitted into two groups, one of them is for the experienced people having 3 years agile experience or more (53 percent of the respondents), and the other group is for the inexperienced people having less than 3 years agile experience (47 percent of the respondents).

Table 4.9 demonstrated the results of the structural model estimates for the model by agile experience.

Table 4.9: The result of structural model assessment by agile experience

		EXPERIENCED		INEXPERIENCED	
Latent variable	Type	R^2	AVE	R^2	AVE
Organizational	Exogenous		0,631		0,611
People	Endogenous	0,083	0,529	0,174	0,517
Process	Exogenous		0,491		0,519
Technical	Endogenous	0,245	0,555	0,255	0,568
Failure	Endogenous	0,667	0,668	0,870	0,497
Mean		0,332	0,566	0,433	0,545

The R^2 values of endogenous (dependent) latent variables were identified to be higher for agile inexperienced respondents. For the Failure latent variable, it was 0,667 for agile experienced people and 0,870 for agile inexperienced people.

In order to explain the critical failure factors and indicators with regards to the agile experience and based on the Failure latent variable, Table 4.10 showed the correlation coefficients connecting the Technical and Process latent variables. For agile experienced people, failure mainly depends on Technical factors (path coefficient= 0.856) while Process factors have negative and low effect (path coefficient= -0,091). For agile inexperienced people, failure mainly depends on Technical factors (path coefficient= 0.935) while Process factors have negative and very low effect (path coefficient= -0,004).

Table 4.10: The failure latent variable by agile experience

	EXPERIENCED		INEXPERIENCED	
	Technical	Process	Technical	Process
Correlation	0,813	0,320	0,933	0,468
Path coefficient	0,856	-0,091	0,935	-0,004
Correlation * path coefficient	0,696	-0,029	0,872	-0,002

Based on the analysis on agile experienced versus agile inexperienced people, Failure factor similarly corresponds to Technical and Process factors, however, the contribution slightly differs. For agile experienced people, the most contribution significantly belonged to Technical factors and other measure’s contribution (Process factor) was considerably low. For agile inexperienced people, similarly Technical factors drastically contributed (higher than the entire model and agile experienced model) and Process factors contributed very low.

By looking at GoF index results depicted on Table 4.11, absolute GoF index value was calculated as 0.433 for agile experienced people (less than the entire model) and 0.486

for agile inexperienced people (higher than the entire model and agile experienced people).

Table 4.11: GoF index table experienced vs inexperienced

	EXPERIENCED			INEXPERIENCED		
	GoF	GoF (Bootstrap)	Standard error	GoF	GoF (Bootstrap)	Standard error
Absolute	0,433	0,446	0,044	0,486	0,493	0,042
Relative	0,896	0,840	0,084	1,103	0,886	0,095
Outer model	0,975	0,953	0,051	0,972	0,960	0,056
Inner model	0,919	0,881	0,080	1,135	0,923	0,083

5. DISCUSSION AND CONCLUSION

As the software projects have high failure rates and the failures of the projects were crucial in business life due to their affect in terms of the loss of money, time and resources, the thesis aimed to figure out the critical failure factors and indicators on agile software projects along with their signifance of impact in order to avoid the failures in return.

The failure factors and indicators were examined in four dimensions (organizational, people, technical, process) and their sub-categories that mainly contributed to the software development methodologies and to the agile specifically.

This research was based on the online survey data to explore the critical failure factors of agile software development projects using quantitative approach. Partial Least Squares Structural Equation Modeling (PLS SEM) was effectively chosen to construct a model and analyze the data to determine the failure factors and indicators and their relative (weighted) impact to the agile projects.

The analysis was performed based on two groups, firstly all survey data was modeled and the impact of the critical failure factors was evaluated, and then secondly, the survey data was filtered by agile experience of the respondents' to measure whether the impact of the critical failure factors changed by the agile experience.

Based on the first model that was developed and analyzed, the Technical factors and indicators (e.g. no or long delivery cycles, lack of developer involvement in prioritization, etc.) was revealed to dominantly lead agile projects to fail. The Process factors and indicators had unexpectedly lower and negative impact on agile project failures, though, if the role of the customer was vague, it was seen as a factor to cause agile project failures. Similarly, resistance from teams or individuals (people factor) and traditional/outdated culture and unsuitable environment (organizational factor) were also determined to lead agile projects to fail considerably.

Technical factors itself were internally affected by Process factors mostly (higher than 95 percent) and by People factors to some extent (almost 3 percent).

Based on the agile experience, the model slightly differed. The respondents were divided into two groups, a) agile experienced people (having 3+ years experience) and b) agile inexperienced people (having less than 3 years experience). Technical factor was still the most influential factor, however, it was identified to be more powerful for agile inexperienced people to lead agile projects to fail compared to agile experienced people. Agile inexperienced organizations or teams should fully concentrate on Technical factors (indicated above) more in order to avoid failures in agile projects when compared to agile experienced organizations or teams. The Process factors and indicators still had unexpectedly lower and negative impact on agile project failures for both agile experienced and inexperienced people.

Based on Chaw and Cao (2008) agile success and failure factor research, incorrect delivery strategy, improper agile software engineering techniques and absence of high-caliber team were found to be critical failure factors leading agile project to fail. This research similarly indicated that the technical factors and no or long delivery cycles were obiously impacting agile projects negatively.

VersionOne (2016) research of the 10th annual state of agile report explained that company culture at odds with agile methodology dominated the agile project failures and secondly lack of knowledge and experience with agile methods impacted and the third from the top was lack of management commitment. The research also revealed that traditional/outdated culture and unsuitable environment which was considered in organizational factors was influental in agile project failures. This study was unable to find sufficient evidence that lack of management commitment (executive or management sponsorship) was noticeably one of the dominant failure factors of agile projects.

Even though there were further studies around critical success factors mostly in agile development, failure factors or indicators were not observed so much to be focused specifically. With the help of this study, critical failure factors and indicators were primarily studied and it should be considered as an example or reference study for further researches.

REFERENCES

Books

Charette, R., 2005. *Why software fails.* IEEE Spectrum. 42(9), pp. 42-49.

Chin, W.W., 1998. *The partial least squares approach to structural equation modeling.* In:Marcoulides GA (ed) Modern methods for business research. Lawrence Erlbaum Associates, Mahwah, NJ, pp. 295–3368.

Cohn, M., 2007. *Agile estimating and planning*. 5th printing. Upper Saddle River, NJ: Prentice Hall PTR.

Efron, B. & Tibshirani, RJ., 1993. *An introduction to the bootstrap*. NY: Chapman and Hall.

Hair, J., Hult, G., Ringle, C. & Sarstedt, M., 2014. *A primer on partial least squares structural equation modeling (PLS-SEM*), California: Sage Publications, Inc.

Mannila, J., 2013. *Key performance indicators in agile software development.*

Martin, R., 2003. *Agile software development: principles, patterns, and practices*, NJ: Prentice Hall PTR.

Sarstedt, M., Ringle, C., Hair, JF & Hult, G., 2014. *A Primer on partial least squares structural equation modeling (PLSSEM).* Journal of Family Business Strategy (5.1), pp. 105-115.

Werts, CE, Linn, RL & Jöreskog, KG, 1974. *Intraclass reliability estimates: testing structural assumptions*. Educ Psychol Meas 34(1), pp. 25–33.

Wold, H., 1985. *Partial least squares*. In S. Kotz, & N. L. Johnson (Eds.), Encyclopedia of statistical sciences. New York: Wiley, pp. 581-591.

Periodicals

Abrahamsson, P., Salo, O., Ronkainen, J. & Juhani, W., 2002. Agile software development: principles, patterns, practices. *Espoo 2002, VTT Publications*. **478**, p. 107.

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

Darwish, N. & Rizk, N., 2015. Multi-Dimensional success factors of agile software development projects. *International Journal of Computer Applications*. **118** (15), pp. 23-30.

Highsmith, J., 2010. Agile project management. **2.** Boston: Pearson Education Inc.

Ibrahim, R., Ayazi, E., Nasrmalek, S. & Nakhat, S., 2013. An investigation of critical failure factors in information technology projects. **10**, p. 88.

Jugdev, K., & Muller, R., 2005. A retrospective look at our evolving understanding of project success. *Project Management Journal*. **36** (4), pp. 19-31.

Karahoca, D., Karahoca, A. & Kurnaz, A., 2015. Analyzing communication dimensions in a ubiquitous learning environment. **22**, pp.138-142.

Koch, A., 2005. Agile Software development: evaluating the methods for your Organization, Norwood: Artech House, Inc.

Misra, S., 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

Nasehi, A., 2013. A quantitative study on critical success factors in agile software development projects.

Sidky, A., Arthur, J., & Bohner, S., 2007. A disciplined approach to adopting agile practices: agile adoption framework. *Innovations in Systems and Software Engineering*. **3** (3), pp. 203-216.

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

Vijayasarathy, L. R., & Turk, D., 2008. Agile software development: a survey of early adopters. *Journal of Information Technology Management*. **21** (2).

Rockart, J.F., 1979. Chief executives define their own data needs. *Harvard Business Review*. **57** (2), p.87.

Rockart, J.F., & Crescenzi, A.D., 1984. Engaging top management in information technology. *Sloan Management Review* **30** (2), p.7.

Other Publications

Ambysoft Scott Ambler Associates, 2013. *IT project success rates survey,* [online]. http://www.ambysoft.com/surveys/success2013.html [accessed 02 May 2016].

Balzano, S. & Trinchera, L., 2010. *Structural equation models and student evaluation of teaching: a PLS path modeling study,* [online]. https://www.academia.edu/390387/Structural_Equation_Models_and_Student_Evaluation_of_Teaching_a_PLS_Path_Modeling_Study [accessed 10 May 2016].

Bloch, M., Blumberg, S., & Laartz, S., 2012. *Delivering large-scale IT projects on time, on budget, and on value,* [online]. http://www.mckinsey.com/business-functions/business-technology/our-insights/delivering-large-scale-it-projects-on-time-on-budget-and-on-value [accessed 29 May 2016].

unningham, L. , 2015. *8 reasons why agile projects fail,* [online]. https://blogs.versionone.com/agile_management/2015/04/09/8-reasons-why-agile-projects-fail/ [accessed 03 May 2016].

Denning S., 2015. *How to make the whole organization agile,* [online]. http://www.forbes.com/sites/stevedenning/2015/07/22/how-to-make-the-whole-organization-agile/#3ff2e110135b [accessed 02 May 2016].

Directions, 2015. *The customer's role in agile project management,* [online]. http://blog.directionstraining.com/event/the-customers-role-in-agile-project-management?doing_wp_cron=1462315355.3164238929748535156250 [accessed 02 May 2016].

Erickson K., 2013. *A comparative look at top agile tools,* [online]. https://www.captechconsulting.com/blogs/a-comparative-look-at-top-agile-tools [accessed 02 May 2016].

Ghahrai A., 2015. *What makes a good agile tester,* [online]. http://www.testingexcellence.com/what-makes-good-agile-tester/ [accessed 02 May 2016]

Jamil, J., 2012. *Partial least squares structural equation modelling with incomplete data.*

Portland Business Journal, 2008. *Why do most IT projects fail? It's not because of technology*. http://www.bizjournals.com/portland/stories/2008/10/20/smallb4.html [accessed 02 May 2016]

Pressman, R., 2009. *Agile development slide set to accompany software engineering a practitioner's approach 7/e.*

Ramaraju, P., 2014. *Organizational culture eats agile transformation, making it fr-agile*.

Roberts, B., 2014. *Agile and the concept of "Multi-skilled" teams,* [online]. https://www.microsoft.com/en-gb/developers/articles/week01oct14/agile-and-the-concept-of-multi-skilled-teams/ [accessed 02 May 2016]

Sanchez, G., 2013. *PLS path modeling with R,* [online]. http://gastonsanchez.com/ [accessed 04 May 2016]

Sauro, J., 2011. *How to interpret survey responses,* [online]. http://www.measuringu.com/blog/interpret-responses.php [accessed 04 May 2016]

Scott Ambler Associates, 2014. *Lean and agile software development is more successful than waterfall,* [online]. http://scottambler.com/backup_muse/lean-and-agile-software-development-is-more-successful-than-waterfall.html [accessed 02 May 2016]

Senapathi, M. & Srinivasan, A., 2013. *Agile Sustainability Questionnaire,* [online]. https://www.surveymonkey.com/r/AgileSustainability_Questionnaire [accessed 01 Feb 2016]

Shapiro, R., 2014. *The U.S. software industry as an engine for economic growth and employment.*

Singh, A., 2013. *Why do companies fail in adopting agile practices,* [online]. https://www.scrumalliance.org/community/articles/2013/december/why-companies-fail-in-adopting-agile-practices [accessed 02 May 2016]

Software Design Consultants, 2011. *Application development agile,* [online]. http://sdc.net.au/services/application-development.aspx [accessed 10 May 2016]

Tabaka, J., 2015. *12 failure modes in agile transformation,* [online]. https://www.rallydev.com/blog/agile/12-failure-modes-agile-transformation [accessed 02 May 2016]

Tenenhaus, M., Vinzi, V., Chatelin, Y. & Lauro, C., 2004. *PLS path modeling*, [online]. www.sciencedirect.com [accessed 04 May 2016]

VersionOne, 2016. *The 10th annual state of agile report*, [online]. http://stateofagile.versionone.com/#results [accessed 29 May 2016]

Waters, K., 2007. *Why most IT projects fail and how agile principles help*, [online]. http://www.allaboutagile.com/why-most-it-projects-fail-and-how-agile-principles-help/ [accessed 02 May 2016]

Wiegers, K., 2013. *Defining project scope managing scope creep*, [online]. http://www.jamasoftware.com/blog/defining-project-scope-managing-scope-creep/ [accessed 02 May 2016]

Worren, M., 2010. *Customer Engagement in Agile Sofware Development*, [online], http://www.diva-portal.org/smash/get/diva2:359163/FULLTEXT01.pdf [accessed 01 Feb 2016]

XLSTAT, 2015. *Create and run a basic PLS path modeling project*, [online]. https://help.xlstat.com/customer/en/portal/articles/2062300 [accessed 04 May 2016]

APPENDICES

APPENDIX-1, TABLE 1 SURVEY QUESTIONS

Personal Information

Full-name*

Age*

Gender*

- ☐ Male
- ☐ Female

Graduate Faculty(e.g Computer Engineering, Electric/Electronic Engineering)*

How long have you been involved in software/systems development? Please enter number of years: *

How long have you been using agile methods or agile practices? Please enter number of years: *

On how many projects have you used Agile methods/practices? Please enter number of projects*

Which best describes your current position?*

- ☐ Agile Coach
- ☐ Scrum Master
- ☐ Business Stakeholder
- ☐ Business Analyst
- ☐ oDesigner
- ☐ Tester
- ☐ Project Manager
- ☐ Other:

*

- ☐ Large (project length of more than one year / 30+ people in team)
- ☐ Medium (project length of 6 months – 1 year / 20 – 30 people in team
- ☐ Small (project length of 3 – 6 months / 10 – 20 people in team)
- ☐ Very small (project length less than 3 months / less than 10 people in team)

Personal Experience

The following questions relate to your beliefs on the teams that you have most worked with or currently working with on agile projects.

10. My team members have a strong sense of identification and commitment to the team*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

11. My team members have the willingness to learn and change*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

12. My team members doesn't have strong interpersonal and communication skills*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

13. My team members are technically competent*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

14. My team members have collaborative attitude*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

Agile Development Methodology Questions

The following questions relate to your beliefs on the agile development methodology to validate the impact of key success or failure factors on the effective and sustained usage of agile methods.

15. Agile methodologies recognize the value of customer engagement and welcomes customer representative in agile team*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

16. Customer involvement in early life cycle of project development motivates customers and makes them feel responsible for the project. *

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

17. Management commitment is not required to support the team to take self-initiatives, decisions and handle the circumstances of the results.*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

18. Committed project sponsor or project manager is required for the investment decisions, project plans and empowers the successful project delivery*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

19. Corporate culture should support the introduction of agile methodologies for being more cooperative instead of hierarchical.*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

20. An organization where agile methodology is followed is more dynamic and fast responsive*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

21. Organizational culture should place high value on face-to-face communication to support agile culture*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

22. Facility with proper agile-style work environment will positively influence team communication and organization culture*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

23. One of the factors that is likely to positively influence the success of an agile software development project is the co-location of the organization of the teams*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

24. Companies involved in distributed international projects will be affected by the cultural and political situations in those regions*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

25. An agile team should be no larger than 9 people*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

26. Customer involvement in early life cycle of project development doesn't help to create much better business engagement and customer satisfaction. *

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

27. Daily sync meeting with the customer should be arranged*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

28. Developers don't need to be experienced with the required skillset*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

29. The customer needs to work locally with the developers*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

30. Daily sync meetings with the customer need to be organized in terms of face-to-face meetings*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

31. The motivation of the individuals (developers) is crucial in the agile development*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

32. Project Manager is not responsible for the motivation of developers.*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

33. Working environment affects the motivation of the developers in the agile development*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

34. Technical challenges affects the motivation of the developers in the agile development*
- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

35. Insufficient agile experience of a project manager doesn't affect agile development*
- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

36. Developers' resistance to agile methodology doesn't cause any failures of agile projects.*
- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

37. Lack of teamwork results in failures of agile projects*
- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

38. Requirements should be determined by the customer*
- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

39. The priorization should be made by the customer.*
- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

40. Clarification of requirements does not have an impact on agile projects.*
- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

41. Progress of the scrum team should be tracked daily using required tools.*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

42. Before starting to implementation process, coding standards should be pre-defined.*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

43. A second engineer should review an engineer's code in terms of code standards*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

44. An engineer should design and implement the code without considering customer's future requirements/enhancements*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

45. As part of scrum meetings, each engineer should tell what and how he/she did in current work-period to others.*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

46. Technical document should be updated clearly during each work-period.*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

47. Technical document should include information only the customer needs to know.*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

48. Code delivery to the customer should be done frequently (along with each/every sprint).*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

49. Most important features of the project should be delivered firstly. *

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

50. The person who implemented the code and the person who will test the code should be the same person.*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

51. Test scenarios should be completed before code implementation.*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

52. Minimum %40 of the project team should consist of expert engineers.*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

53. Training period is not required to consider and plan in project plans*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

54. Integration tests should be run automatically with each delivery.*

- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

55. Agile development doesn't increase the level of software quality.*
- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

56. A requirement cannot be changed by customer in any phase of the project.*
- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

57. Design estimates should be given to the agile team that will work for the project*
- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

58. A project which is comprehensive and difficult to control should consist of small teams.*
- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

59. If there are multiple teams, each team should be aware of their responsibilities and dependencies*
- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

60. Design estimates should be given after a requirement is clear and well understood. *
- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

61. Customer should define the priority of requirements.*
- Strongly Disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly Agree

Additional Feedbacks/Comments

Please let us know your additional feedback or comments about your agile experience.

CURRICULUM VITAE

Name & Surname : Harun ÇALIŞKAN

Permanent Address : Yenişehir St. Cumhuriyet Ave. N:3 Kurtköy/İSTANBUL

Place and Year of Birth : İzmir, 19.07.1981

Foreign Language : English (Fluent in reading, writing, conversation)

Primary Education : Ödemiş Ataturk Primary School

Secondary Education : Ödemiş Super High School (1st ranking)

Undergraduate : Dokuz Eylül Uni. Computer Engineering Eng. (1st ranking), 2003

Working Life : NETAŞ (R&D and Global Product Support Manager, 2010 - ?)

NETAŞ (R&D and Global Product Support Engineer, 2005- 2010)