DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

ASSESSMENT OF AGILE SOFTWARE RESEARCH AND DEVELOPMENT PROJECTS WITH MULTI-CRITERIA DECISION MAKING APPROACHES

by
Elif BİLGİÇ

October, 2021
İZMİR

ASSESSMENT OF AGILE SOFTWARE RESEARCH AND DEVELOPMENT PROJECTS WITH MULTI-CRITERIA DECISION MAKING APPROACHES

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University In Partial Fulfillment of the Requirements for the Degree of Master of Science in Industrial Engineering, Industrial Engineering Program

by
Elif BİLGİÇ

October, 2021
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"ASSESSMENT OF AGILE SOFTWARE RESEARCH AND DEVELOPMENT PROJECTS WITH MULTI-CRITERIA DECISION MAKING APPROACHES**" completed by **ELİF BİLGİÇ** under supervision of **PROF. DR. HASAN SELİM** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

……..

Prof. Dr. Hasan SELİM

Supervisor

……..	……..
Prof. Dr. Mehmet ÇAKMAKÇI	Prof. Dr. Pınar Mızrak ÖZFIRAT
(Jury Member)	(Jury Member)

Prof. Dr. Okan FISTIKOĞLU

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor Prof. Dr. Hasan SELİM for his supporting, knowledge sharing and advice throughout my M.Sc. thesis. His immense knowledge and understanding encouraged me for searching better and complete my thesis successfully.

I would like to offer my special thanks to my family. Especially my mother Gönül BEYGİ and my little nephew Alp BİLGİÇ for their support, positive energy and encouragement.

Elif BİLGİÇ

ASSESSMENT OF AGILE SOFTWARE RESEARCH AND DEVELOPMENT PROJECTS WITH MULTI-CRITERIA DECISION MAKING APPROACHES

ABSTRACT

Today, adapting the agile project management techniques to research and development projects is vital for companies. This research identifies critical success factors for the agile software development projects, as well as the areas where agile businesses should focus their efforts. There exists limited research in the literature examining the impact of critical success factors on the outcomes of research and development projects. In this study, critical success factors in the agile software development processes are investigated, and a group of them is selected for the assessment. The project outputs considered in the study are “scope”, “time”, “cost”, “satisfaction of business partners”, “quality”, and “addressing organizational requirements”.

Each project output is handled as an evaluation criterion and critical success factors are considered as alternatives, and a questionnaire is applied to the experts in the field. By applying SWARA method, weights of the project outputs are determined. By utilizing these weights, the critical success factors are ranked by TOPSIS and MOORA methods. As a result, “correct determination of the scope and constraints of the project” is observed as the most important success factor.

Keywords: Agile software development, multi criteria decision making, SWARA, TOPSIS, MOORA

ÇEVİK ARAŞTIRMA GELİŞTİRME YAZILIM PROJELERİNİN ÇOK KRİTERLİ KARAR VERME YAKLAŞIMLARI İLE DEĞERLENDİRİLMESİ

ÖZ

Günümüzde çevik proje yönetimi tekniklerinin araştırma ve geliştirme projelerine uyarlanması şirketler için hayati önem taşımaktadır. Bu araştırma, çevik yazılım geliştirme projeleri için kritik başarı faktörlerinin yanı sıra çevik işletmelerin odaklanması gereken alanları tanımlamaktadır. Literatürde kritik başarı faktörlerinin araştırma ve geliştirme projelerinin sonuçları üzerindeki etkisini inceleyen sınırlı sayıda araştırma bulunmaktadır. Bu çalışmada, çevik yazılım geliştirme süreçlerindeki kritik başarı faktörleri araştırılmış ve değerlendirme için bunlar arasından bir grup faktör seçilmiştir. Çalışmada dikkate alınan proje çıktıları "kapsam", "zaman", "maliyet", "iş ortaklarının memnuniyeti", "kalite" ve "kurumsal gereksinimlerin karşılanması"dır.

Her proje çıktısı bir değerlendirme kriteri, kritik başarı faktörleri ise alternatifler olarak ele alınmıştır ve alanında uzman kişilere bir anket uygulanmıştır. SWARA yöntemi kullanılarak proje çıktıları yani kriter ağırlıkları belirlenmiştir. Bu ağırlıklar kullanılarak kritik başarı faktörleri TOPSIS ve MOORA yöntemleri ile sıralanmıştır. Sonuç olarak; "projenin kapsamının ve kısıtlarının doğru belirlenmesi" en önemli başarı faktörü olarak ortaya çıkmıştır.

Anahtar kelimeler: Çevik yazılım geliştirme, çok kriterli karar verme, SWARA, TOPSIS, MOORA

CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER ONE
INTRODUCTION

Nowadays, agile software project development is critical. Agile project management helps to keep up with change quickly, and working software can be produced in a short amount of time.

In today's world, where the number of evaluation criteria and alternatives is growing by the day, making quick and correct decisions is essential. As a matter of fact, multi-criteria decision-making methods, a sub-branch of operations research, speed up the decision-making process and allow us to evaluate our choices using scientific methods.

Not only in the agile transformation phase, but also in the agile sustainability phase, critical success factors in agile approaches make a significant contribution to agile businesses. As a matter of fact, understanding the critical success factors in agile software development helps businesses react quickly through the agile transformation process. Businesses that already use an agile approach, on the other hand, have a higher success rate with agile processes.

Critical success factors in the literature were examined in depth in this study. Aside from studies dealing with critical success factors in practice, literature reviews on critical success factors were also investigated during the literature review. In this way, while the concept of critical success factors in the agile software development process was internalized, the contribution of both practice and literature to each other was benefited.This thesis composed of six chapters. Chapter 2 provides information about agile software development process and critical success factors for agile. The critical success factors used in the study, obtained as a result of the literature review, are presented in a table with their references. Then, the critical success factors selected for use in this study were defined. In Chapter 3, the studies on critical success factors, agile methodologies, and evaluation of agile methodologies were examined. In the fourth part of the study, information about multi-criteria decision making is given. Afterwards, SWARA, TOPSIS, MOORA methods are explained in

detail. Chapter 5 includes the application phase in which the criteria were defined. The questionnaire which is applied to the R&D experts in the software development sector is explained in the chapter. In addition, the applications of SWARA, TOPSIS and MOORA methods are included according to data gathered from the practitioners. The results obtained were analyzed in detail in the Results Section. Also, future work directions were suggested at this part of the study. Finally, in Chapter 6, concluding remarks are presented. Also, results of the study and contribution to the literature are explained in the chapter.

CHAPTER TWO
AGILE SOFTWARE DEVELOPMENT

2.1 The Agile Approach

The Agile approach was created with the goal of being able to quickly react to rapidly changing software requirements. The capability to generate and adapt to change is known as agility. It's a strategy for managing with, and eventually excelling in a dynamic and rapidly changing environment (Agile Alliance, n.d.).

Agile project management and software development is an iterative strategy that allows teams to deliver value to clients more rapidly and with fewer challenges.

Agile is a collection of techniques in software development that aims to increase the performance of software development professionals, teams, and organizations. It entails self-organizing and cross-functional teams working together with their customers/end-users to uncover requirements and build solutions. It promotes flexible reactions to changes in needs, resource availability, and knowledge of the issues to be solved, as well as adaptive planning, evolutionary development, early delivery, and continuous improvement (Wikipedia, 2021a).

2.2 Agile Manifesto

The Agile Manifesto was written in 2001 when 17 software developers met for three days to debate their points of view and reach an agreement.

The Agile manifesto is as follows;

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

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

That is, while there is value in the items on the right, we value the items on the left more (Agile Manifesto, 2001).

2.3 Agile Principles

The authors of The Agile Manifesto presented the following twelve principles.

1 Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.	2 Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	3 Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4 Business people and developers must work together daily throughout the project.	5 Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	6 Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
7 Working software is the primary measure of progress.	8 The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	9 Continuous attention to technical excellence and good design enhances agility.
10 Simplicity – the art of maximizing the amount of work not done – is essential.	11 The best architectures, requirements, and designs emerge from self-organizing teams.	12 At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Figure 2.1 Twelve principles of agile manifesto (Agile Manifesto, 2001; Mersino, 2020)

2.4 Agile Methodologies

This section provides an overview of some of the commonly utilized agile techniques.

2.4.1 Scrum

"The history of Scrum can be traced back to 1986 in the Harvard Business Review (HBR) article titled "The New Product Development Game" by Hirotaka Takeuchi & Ikujiro Nonaka. The article describes how companies such as Honda, Canon, and Fuji-Xerox produce new products worldwide using a scalable and team-based approach to product development. This approach emphasizes the importance of empowering self-organized teams." (Digite, n.d.).

Some of the basic statements in Scrum are:

2.4.1.1 Sprint Backlog

The sprint backlog is a list of activities that the Scrum team has identified as needing to be accomplished within a Scrum sprint.

2.4.1.2 Product Backlog

A product backlog, which is a prioritized list of work for the development team, is created using the roadmap and associated requirements. The most critical tasks are displayed at the top of the product backlog, allowing the team to prioritize which tasks should be completed first.

2.4.1.3 Daily Meeting

"The Daily Scrum is a 15-minute event for the Developers of the Scrum Team. To reduce complexity, it is held at the same time and place every working day of the Sprint. If the Product Owner or Scrum Master are actively working on items in the Sprint Backlog, they participate as Developers." (Scrum, n.d.).

2.4.1.4 Product Owner

"The Product Owner (PO) is a member of the Agile Team responsible for defining Stories and prioritizing the Team Backlog to streamline the execution of program priorities while maintaining the conceptual and technical integrity of the Features or components for the team." (Scaled Agile Framework, 2021).

2.4.1.5 Scrum Master

The scrum master is the team member in charge of ensuring the implementation of agile applications.

2.4.2 XP Programming

Extreme Programming (XP) is a software development methodology that prioritizes product quality and adaptability to changing client requirements. Rapid development cycles and frequent releases are encouraged.

XP has four basic principles. These are; "communication", "simplicity", "feedback" and "courage"

Communication is one of the key points to solve the problems that occur or to identify the problems that may occur. Simplicity involves simply doing the work that is really needed simply way rather than doing something that will not be needed at all. Thanks to feedback, customers can spot unnoticed problems or create new opportunities. When it comes to making difficult decisions, "courage" is a virtue to possess.

2.4.3 Feature Driven Development

“An Agile methodology for developing software, Feature-Driven Development (FDD) is customer-centric, iterative, and incremental, with the goal of delivering tangible software results often and efficiently. FDD in Agile encourages status reporting at all levels, which helps to track progress and results.” (Lynn, n.d.).

2.4.3.1 Develop the Overall Model

It is the stage in which the problem of the software development project is defined and a solution proposal is created.

2.4.3.2 Build a Features List

It refers to the procedure for identifying and prioritizing features in a given solution. It's similar to Scrum's product backlog.

2.4.3.3 Plan by Feature

It is the order in which tasks in the features list are prioritized. It is at this point that factors such as potential risks and the distribution of jobs among the workforce are considered.

2.4.3.4 Design by Feature

It is the determination of which features will be completed in the two-week cycle.

2.4.3.5 Build by Feature

This is the stage where feature design is implemented. Necessary tests are made; prototypes are created for customers.

2.5 Critical Success Factors

"Critical Success Factors (CSF) is a management term for an element that is necessary for an organization or project to achieve its mission." (Wikipedia, 2021b).

The critical success factors approach was first put forward by Rockhart (1979) and has evolved to the present day (P. Shakya & Shakya, 2020).

Bullen & Rockhart (1981) describe "critical success factors" as the small number of areas where successful project outcomes assure successful competitive performance for the person, department, or organization (Ramadan & Rizk, 2015).

In this study, critical success factors in the agile software project management in literature were investigated. The selected 31 factors are presented in Table 2.1.

Table 2.1 Critical success factors

Critical Success Factor	*References*
Educating the organization about agile project management	Kim et al., 2016, Cherie et al., 2021
Management's Support and Contribution to Agile Project Management	(Abdullah et al., 201), Aldahmash, 2018, Khoza & Marnewick, 2021
Arranging Social Events	Kim et al., 2016, Naslund &Kale , 2020, Khoza & Marnewick, 2021
Willingness to Take Risk	Naslund & Kale, 2020
Choosing the Appropriate Agile Approach Methodology	Tasneem et al., 2016
Well-defined coding standards	Arcos-Medina, Mauricio, 2020
Code template and code reuse	Nguyen, 2016
Using Simple/Clean Code	Nguyen, 2016, Shahbaz et al., 2018
Rigorous refactoring activities	Arcos-Medina & Mauricio, 2020

Tablo 2.1 Continues

Modular Design	Arcos-Medina & Mauricio, 2020
Portable design	Arcos-Medina & Mauricio, 2020
Extensible design	Arcos-Medina & Mauricio, 2020
Right Amount of Documentation	Silva & Santos, 2015), Chiyangwa & Mnkandla, 2017, Arcos-Medina & Mauricio, 2020)
Integration Testing	Silva & Santos, 2015
Maintenance of Code and Other Documentation	Arcos-Medina & Mauricio, 2020
Backlog Prioritization	Shakya & Shakya, 2020
Daily Stand-up Meetings	Nguyen, 2016, Chiyangwa & Mnkandla, 2017, Arcos-Medina & Mauricio 2020, Shakya and Shakya, 2020
Good Reporting of Project Status	Darwish & Rizk, 2015
Project Schedule	Nasehi , 2013, Darwish & Rizk, 2015, Silva & Santos, 2015, Tasneem et al., 2016, Chiyangwa & Mnkandla, 2017, Syeda, 2017, Bergmann, 2018, Arcos-Medina and Mauricio, 2020, Tsoy & Staples, 2020, AbdulHafeez et al., 2021, Cherie et al., 2021.
Regular Delivery of Software	Tasneem et al., 2016, Arcos-Medina & Mauricio, 2020, AbdulHafeez et al., 2021
Correct determination of the scope and constraints of the project	Syeda, 2017, Bergmann, 2018, Arcos-Medina & Mauricio, 2020)
Training Users	AbdulHafeez et al., 2021, Meenakshi, Singh, Agrawal, 2020
User feedback	Nasehi, 2013, Brown, 2015, Vithana et al., 2015, Silva, Santos, 2015, Kim et al., 2016, Abdullah et al., 2017, Syeda, 2017, Aldahmash, 2018, Bergmann, 2018, Arcos-Medina & Mauricio, 2020, Richa et al., 2020, Tsoy & Staples, 2020, Naslund & Kale, 2020, Shakya & Shakya, 2020, AbdulHafeez et al., 2021
Good Relationships with Business Partners	Kim et al.,2016, Chiyangwa & Mnkandla, 2017, Abdullah et al., 2017, Syeda, 2017, Bergmann, 2018, Arcos-Medina & Mauricio, 2020, AbdulHafeez et al., 2021
Engaging everyone in the organization in the agile approach	Kim et al., 2016, Khoza & Marnewick, 2021
Ensuring Effective Information Flow in the Organization	Bergmann, 2018
Empowering employees to make their own decisions (autonomy)	Naslund & Kale, 2020

Tablo 2.1 Continues

Being focused on providing qualified employees into the business and improving employee quality.	Naslund & Kale, 2020
Recruiting employees who have already worked in an agile environment.	Naslund & Kale, 2020
Small team size	Tasneem et al., 2016, Syeda, 2017, Richa et al., 2020, Cherie et al., 2021
Increased collaboration and openness inside the company, as well as the establishment of trustworthy relationships	Naslund & Kale, 2020

2.6 Definitions of Critical Success Factors

2.6.1 Educating the Organization About Agile Project Management

In many studies, it is argued that managers should receive training on agile. However, in this study, it is argued that not only the managers but also the entire organization should receive training on agile. As it is known, "Individuals and interactions are more important than processes and tools" (Agile Manifesto, 2001). Therefore, when the organization understands and adopts agile culture, progress will be much easier. Organizing training on agile in organizations will affect not only the knowledge level but also the development of the organization's agile capacity.

The debates about training managers on agile in studies do not diminish the importance of organizational training. Similarly, this study's discussion of training the entire organization does not negate the importance of manager training.

2.6.2 Management's Support and Contribution to Agile Project Management

Many studies in the literature have emphasized the importance of top management's support and involvement. However, it is argued in this study that all managers, not just top management, should participate and support the project.

2.6.3 Arranging Social Events

According to the Agile philosophy, people and relationships are more important than processes and tools. Organizing social activities can assist people in communicating with one another more effectively. Corporate morale, motivation, and cooperation all improve as a result of this strategy.

2.6.4 Willingness to Take Risk

Agile organizations are known for their ability to adapt to change and make incremental but consistent progress. Agile teams should be encouraged to take risks as follows, according to an internet resource.

"Agile teams embrace change and often strike out into the unknown for nothing more than the possibility that they may be able to create something greater than for which they could have planned" (Solutions IQ, 2009).

2.6.5 Choosing the Appropriate Agile Approach

Choosing the right agile approach is vital. Many examples of agile approaches can be found in the literature. Although they all have similar characteristics in general, they each have their own set of benefits and drawbacks. As a result, it's critical that the company takes the appropriate approach and integrates it into its own culture.

2.6.6 Well-Defined Coding Standards

While writing the code, coding standards must be followed. Coding standards make it easier to read, understand, and edit codes.

2.6.7 Code Template and Code Reuse

Code template and code resue, refers to the process of creating a new software product using previously used software or software knowledge. As a result, the development time reduces, and we can produce new product at a lower cost.

2.6.8 Using Simple / Clean Code

Clean code can also be defined as code that is simple to read and understand. Because it is simple to understand and designed in a clean structure, such code will be easy to change.

2.6.9 Rigorous Refactoring Activities

"In computer programming and software design, code refactoring is the process of restructuring the existing computer code changing the *factoring* without changing its external behavior. Refactoring is intended to improve the design, structure, and/or implementation of the software (its *non-functional* attributes), while preserving its functionality" (Wikipedia, 2021c).

2.6.10 Modular Design

Modular programming is a programming technique that divides a program into functional components. In fact, in agile management, the product is delivered in usable pieces through an iterative approach, which is a hallmark of the agile methodology.

2.6.11 Portable Design

"Portability in high-level computer programming is the usability of the same software in different environments. The pre requirement for portability is the generalized abstraction between the application logic and system interfaces". (Wikipedia, 2021d).

2.6.12 Extensible Design

Extensible design supports frequent re-prioritization and allows functionality to be implemented in small steps upon request, which are the principles advocated by the agile methodologies and iterative development (Wikipedia, 2021e).

2.6.13 Maintenance of Code and Other Documentation

Software maintenance includes optimization, error correction, the removal of defunct features, and the enhancement of current functions.

2.6.14 Integration Testing

Integration testing is the process of integrating and evaluating individual software modules as a group during the integration testing phase of software testing.

2.6.15 Right Amount of Documentation

Working software is more essential than detailed documentation, according to the Agile Manifesto (2001). The goal of agile methodologies is to deliver software that works all of the time, and getting the right information to the right person is critical to achieving this goal. Focusing on improving documentation, on the other hand, may result in significant time loss. As a result, one of the most crucial issues is the appropriate amount of documentation.

2.6.16 Backlog Prioritization

The backlog is a list of tasks that the team must complete. While establishing the backlog, tasks must be defined flexibly and respond to changing situations.

Backlog prioritization is a strategic decision that must be made. There are a few characteristics to consider into at this point. The factors to take into account during the backlog prioritization stage were broken down into two categories. These fields can be viewed as both general and product-specific criteria that apply to all products. The criteria for each product are as follows:

- Revenues: The expected return on investment is taken into account when evaluating this criterion. Despite the fact that this value is viewed as an expected value, it is critical when making decisions because companies are profit-making structures. As a result, prioritizing tasks with a high rate of return on investment is one of the most important considerations.
- Market fit and market uniqueness: Market fit is all about meeting customer expectations. The term "market uniqueness" refers to a metric that compares your product to that of your competitors. The key thing to remember here is to actualize features that haven't yet surfaced, to meet consumer demands, and to create opportunities by having features that rivals don't have.

- Complexity: Complexity is influenced by the number of features with which the feature interacts, as well as the spawn time. The more functionalities a product can reveal in a short amount of time, the more valuable it is.

The features that should be evaluated depending on the product are as follows (Shecter, n.d.);

- Confidence: It may be defined as the degree of assurance regarding the feature's usefulness. In reality, this item is linked to presenting the important features in general while prioritizing the task list.
- Risk: This criterion may be thought of as inversely proportional to risk; the bigger the risk, the more pushed back it is.
- Cost: While prioritizing, situations that require high implementation costs can be pushed aside. When examining this situation, however, it may be useful to consider the number of features with which it interacts. It may be misleading to make a decision based solely on cost if the feature has a high degree of confidence and affects many features at the same time.

2.6.17 Stand-up Meetings

Teams can focus on critical daily tasks in a short amount of time by holding daily stand-up meetings.

The fact that everyone devotes time to a common project on a daily basis reduces the likelihood of problems. It helps teams work together more effectively. The team's motivation and enthusiasm for achieving a common goal grows.

2.6.18 Good Reporting of Project Status

Working software, rather than documents, should be prioritized, according to the Agile manifesto (2001). However, the precise and clear definition of the project situation, as well as the analysis of the defined situation, have a direct impact on the project's speed and cost. As a result, it's a critical opportunity to accurately reveal the project's status, identify existing issues, and identify potential issues.

2.6.19 Project Schedule

The project schedule includes the project's milestones as well as the creation of the schedule. Depending on the scope of the project, a project schedule should include flexibility to respond to change.

2.6.20 Regular Delivery of Software

Regular software delivery ensures that customer expectations are met and that minor software component modifications can be completed more quickly. Furthermore, delivering software on a regular basis ensures that all details are addressed, as well as improving the ability to keep up with changes and manage client expectations.

2.6.21 Correct Determination of the Scope and Constraints of the Project

The project's scope is directly related to accurately determining the needs and analyzing the current situation.

The three processes involved in project scope management are listed in the following (Clarizen, 2020):

- Planning: This includes capturing and defining the work that needs to be done
- Controlling: This step focuses on scope creep, documenting, tracking and approving/disapproving of changes
- Closing: This includes an audit of the project deliverables and assessing the outcomes of the original plan.

2.6.22 Training Users

It's critical to keep in touch with users about the product and solicit their feedback on a regular basis. As a necessary consequence, training on the produced software not only speeds up the process of making new arrangements based on customer feedback, but it also improves customer satisfaction.

2.6.23 User/Customer Feedback

Gathering feedback on previously developed software is a good way to speed up the project and make changes on time and at a lower cost. Not only in terms of customer satisfaction, but the process of analyzing client feedback and converting it into value also allows the company to differentiate itself from its competitors.

2.6.24 Building Good Relationships with Business Partners

Establishing positive customer relationships is not only a way to ensure customer satisfaction and successful project completion, but it is also a way to secure new successful projects. As a result, one of the most important success factors is learning from customers and communicating effectively and consistently.

2.6.25 Engaging Everyone in the Organization in the Agile Approach

Having everyone involved in agile contributes not only to the success of agile project management but also to the development of team spirit and increased motivation.

2.6.26 Ensuring Effective Information Flow in the Organization

The Agile manifesto emphasizes the importance of individuals and interactions. Constructing an interpersonal, accurate, and effective communication network is one of the most critical situations. In agile approaches, people must be able to communicate clearly and effectively with one another. Individuals must effectively explain the job to each other because agile software development techniques do not collect comprehensive documentation. Maintaining clear language in all interactions with the user as well as the team is critical. It is necessary to transfer key information to the appropriate locations and maintain an information flow in order for this communication to be effective.

2.6.27 Empowering Employees to Make Their Own Decisions

Without a hierarchical structure, autonomy means that the group can make its own decisions. In a hierarchical structure, approval and support from a higher mechanism are always required when making decisions. However, in this structure, the decision

is made by the person who knows the work, which reduces the amount of time spent making a decision.

2.6.28 Recruiting Qualified Employees and Support Them to Increase Their Qualifications

Choosing staff with the right competencies is important not only for agile but also for all approaches. Software development is a job that necessitates knowledge and experience. As a matter of fact, selecting qualified personnel in the software industry becomes increasingly important. In comparison to other industries, the software industry has a higher demand for experienced and knowledgeable workers.

2.6.29 Bringing People with Previous Agile Experience into the Organization

Working with people who have worked with Agile before is helpful in understanding and applying the agile mindset. Because people who have worked in an agile environment have internalized the concept, they can help lead the transition to and maintain agile, as well as contribute to its successful implementation.

2.6.30 Small Team Size

Agile methodologies are known for focusing on individuals and interactions, as well as software that works quickly. Individual interactions within a small number of teams are clearly simpler and more effective when using this logic. Because managing the interaction and communication between individuals will become more difficult as the population grows. Another thing to keep in mind is that working software should take precedence over documentation. Rather than documentation, tasks and requirements must be expressed accurately and clearly for the team in the agile software development process. However, as the number of people on the team and the number of people they interact with grows, time loss and complexity will increase, and in order to manage this complexity.

2.6.31 Increased Cooperation and Transparency in the Organization, Establishing Reliable Relations

On the basis of agile approaches, interpersonal interaction and open communication are critical. As a result, establishing trustworthy relationships in agile

approaches not only boosts employee motivation but also allows for more fluid progress.

CHAPTER THREE
LITERATURE REVIEW

In this section, agile approaches, critical success factors in agile software development approaches and how agile software development methodologies are handled are investigated.

In the literature, there are many methods for selecting software. Some researchers employed a multi-criteria decision-making strategy, while others used a fuzzy inference system approach.

For example, Vavpotič & Vasilecas (2011) have evaluated the different software projects and software development methodologies by decision rules. They have calculated methodology-project values using decision matrices. The Authors have applied their approach to a company. They employed "SCRUM, test-driven development (TDD), extreme programming (XP), RUP, RUP for small projects and Oracle Custom Development Method (CDM)" and the company's custom method". Furthermore, they choose five different projects. The authors examined all projects according to the selected methods. In conclusion, they have found that the most suitable methodology is RUP for the company needs.

Demirtaş, Tuzkaya & Seker (2014) have studied the selection of project management methodology via swot fuzzy AHP. They have examined the criteria as strengths, weaknesses, threats, and opportunities. Then, the proposed approach was applied at an international bank. Alternatives are examined as agile and waterfall methods. At the end of the evaluation, the agile method was selected as the proper method.

Khan, Shameem, Nadeem & Akbar (2021) have identified twenty-three success factors of agile approaches in global software management. They have classified these factors into four main titles as "organizational management", "team", "technology" and "process". Then, a survey has applied to the practitioners from the

Chinese global software industry. The success factors have been prioritized via fuzzy AHP. “Organizational management” has been chosen the most prioritized category.

Öztürk (2013) has used fuzzy logic approach for the selection of software development life-cycle model. Inputs of the fuzzy inference model are defined as “requirements”, “complexity”, “project size”, “development time”, “experience” and “risk”. Also, they have defined the membership functions. For example, Requirements input’s membership functions are; “unstable and unknown”, “known” and “stable and well known”. The outputs of the system have been defined as “Agile”, “Prototype”, “Iterative-Incremental”, “Spiral” and “Waterfall”. Then, the fuzzy rules have been identified.

Khan & Beg (2013) have surveyed about software life-cycle development models. They have searched the effect of wrong software life cycle selection on critical software projects. According to the conclusion, they have found the importance of life cycle model selection. Then, a decision support matrix has been proposed by the authors. In this matrix, software life-cycle models have been examined as traditional and agile models. Thanks to the proposed matrix, life cycle models can be evaluated easily.

Mahajan, Simon & Khatri (2017) have aimed to select agile framework for the software industry using fuzzy AHP approach. They have identified criteria and alternatives via a survey. The criteria considered in the survey are perspective, guideline, cost-efficient, on-time delivery and product quality. The alternatives are Scrum, Scaled Agile Framework, Large Scale Scrum. By using fuzzy AHP, “the perspective” was chosen as the criteria with the highest weight. Also, it was observed that the scaled agile framework was better than other frameworks.

El Beggar (2018) has assessed agile methodologies according to four main and twenty sub-criteria. The main criteria are; organization, technique, modelling and quality assurance. Alternative methodologies are; AMDD, ASD, Crystal Clear, DSDM, FDD, Kanban, Scrum, and XP. Then, the Fuzzy PROMETHEE method was

applied for evaluation. In conclusion, the XP method is chosen as the best alternative. Also, the consistency of the results was supported by sensitivity analysis.

Silva, Schramm & Damasceno (2016) have studied the selection of agile methodology via the SMARTER method. According to the conclusion of the evaluation, XP2 was chosen as the most satisfying alternative.

Rai, Agarwal & Kumar (2018) have proposed a fuzzy-based agile software selection approach. Firstly, they have determined the most important factors which play a role important in the selection of agile software development methodology. Then, the 10 fuzzy rules have been identified. In the study, the authors have used the Mamdani rule base in Matlab. According to the conclusion of the study, Scrum and XP Methods have very high rule strength.

Aljuhani & Alhubaishy (2020) have aimed to prioritize mobile application requirements. Authors have prioritized mobile application needs according to the four criteria namely; Complexity, Value, Risk, and Urgency. According to the determinations of the experts; “value” and “urgency” was chosen as the as the most important and the least important criteria, respectively. In addition, the authors have listed the requirements of the mobile application. They have prioritized the requirements according to criteria, and the “Delete List” requirement was selected as the highest priority. Furthermore, according to AHP method, the “Read / View list with its items” requirement was chosen as the highest priority.

The literature reviews on Agile techniques provide a significant addition to the corpus of knowledge. The following are examples of some of the reviews.

Mohammed & Abushama (2013) have conducted a literature review about agile methodologies and the difficulties of software challenges. The Authors have listed the methods as XP2, Scrum, Crystal and DSDM. They have defined major challenges as the criteria, as three main criteria. These criteria are; Project set-up, project complexity and contract management. Also, they have divided each criterion into sub-criteria. Then, they have used three linguistic measures to compare methods

according to the criteria. The Authors have created a table including all methods evaluated by the criteria. In this way, the study can help the small and medium enterprises for the proper method selection.

One of the important issues is that how critical success factors in Agile software development are addressed in the literature.

Hummel & Epp (2015) have studied success factors in agile information systems development. They use prior literature to establish four success characteristics that serve as the conceptual foundation for their research. The team, the customer, the organization, and communication are the factors. Then they expanded their studies by determining sub-factors for each factor. The authors collected data through interviews with twelve people. Thanks to the study, the positive effects of the four factors determined at the beginning of agile information systems development were examined in detail.

Nasir, Sahibuddin, Ahmad & Fauzi (2015) discussed how PMBOK addresses critical success factors. In their study, they gathered information through the Delphi questionnaire. Then they tabulated their expert opinions. The findings demonstrate that the PMBOK is a very successful strategy to cope with just one important success element. According to the study's experts, PMBOK gives minimal operational assistance in its procedures, and certain customizations are necessary to manage software projects.

Team spirit is very important in agile approaches. Therefore, some studies are only focused on how to increase success for agile teams.

Nguyen (2016) have handled success factors for teams at his/her study. The author determined the characteristics that are essential in creating and managing high-performance agile development teams after conducting a thorough literature analysis. In the study, it was concluded that teams with high performance were also effective on other factors such as project success rate.

In terms of project and performance outcomes, critical success factors in agile organizations are critical. This research is based on critical success factors. For this reason, some important research on the critical success factors are mentioned in the following.

Nasehi (2013) made a literature survey on critical success factors and presented the identified critical success factors. The goal of the study is to determine how organizational, people, process, technical, and project aspects affect the effectiveness of agile software development in terms of quality, scope, time, and cost. As a basis, they conducted a survey and examined the results statistically. Some hypotheses were proven such as the fact that using agile methodology is directly related to product quality rather than total cost reduction, and that using team management techniques can affect timeless factors rather than cost reduction.

Vithana, Fernando & Kapurubandara (2015) evaluated agile critical success factors across the "people" and organizational factors. The researchers surveyed agile software businesses in Sri Lanka and analyzed the implementation results with SPSS. As a result of their research, "Technical competence" has the highest impact on agile project success.

El Hameed, El Latif & Kholief (2016) examined the critical success factors in the dimensions of "People", "Process", "Product", "Project", "Organization" and "Technical". Then a mind map was created for each dimension. The authors also examined the relationships between twelve agile principles and success factors.

Dikert, Paasivaara & Lassenius (2016) examined the success factors and challenges in large-scale agile conversions through a comprehensive literature review. They discovered thirty-five reported difficulties, which were divided into nine categories, and twenty-nine success factors, which were divided into eleven categories.

Aldahmash, Gravell & Howard (2017) examined the critical success factors of agile software development projects. After the literature review, the authors

identified eight critical success factors. They conducted a survey to gather information from agile industry experts. As a result of the questionnaire and analysis, they divided the eight success factors into two components. The two components are "Organizational and People" and "Technical and Project Management". Organizational and People aspects include "organizational culture", "customer involvement", "top management support", "team capability and training", and "communication". Technical and Project Management Aspects includes these; "project management process", "delivery strategy", and "agile software techniques".

Chiyangwa & Mnkandla (2017) examined the contribution of software companies applying an agile methodology to software development project success. The authors gathered the critical success factors under the following 7 main titles. The authors developed hypotheses to examine the impact of critical success factors on project success. Then, they surveyed agile professionals in South Africa and analyzed the results with SPSS. As a result of the study, performance expectancy was found to be the most important factor.

Ghayyur et al. (2018) investigated success barriers as well as agile success factors in their study. The authors, who made a detailed literature review, defined 25 common success factors. Fourteen barriers to hinder agile success were defined. Then they examined these factors in terms of "organization", "people" and" technical".

P. Shakya & Shakya (2020) discussed 23 critical success factors in their study. They divided these factors into 6 basic groups. These basic groups are “organizational", "Environmental", "Technical", "Project", "Team" and "Customer". Additionally, the authors took “Organizational” and “Environmental” factors as independent variables. “Technical”, “project”, “team” and “customer” factors were taken as dependent factors. While evaluating the project output, the authors considered "on-time delivery", "improved quality product", "customer satisfaction", "improved managerial effectiveness", "ability react to change" and "reduced project risk" as outputs. As a result of the study, the dependent and independent variables had an effect on the success of the project.

A detailed literature review was conducted by Naslund & Kale (2020) to determine the critical success factors in the agile software development process. The authors first gathered the critical success factors under 13 titles. These are "Management", "Employees", "Change Leaders", "Choosing and customizing the agile approach", "communication and collaboration", "Culture", "Teams", "Operational Processes", "Role Definition and Deployment", "Incentives and Measures", "Organizational Structure", "Transformation Planning", "Tools". Then, titles were classified into 3 groups; purpose, process, and people. The authors obtained that the process category was more important.

Noteboom, Ofori, Sutrave & El-Gayar (2021) divided the agile critical success factors into three main headings as a result of their detailed literature review. According to the study, the "Dynamic product definition and effort estimation" factor is the most placed one in the literature. The least included factors are "team size" and "employee training".

Muhammad et al. (2021) divided the critical success factors in the agile approach into 6 groups. These main groups are organizational, people, process, project, team and customer. The data obtained from the survey results were examined in the SPSS program by including the 7 main categories mentioned at the beginning. Then, AHP method was applied by including critical success factors with a mean value greater than three and a half. According to the results obtained by using AHP, the main topic of highest importance has been "people". The most important factor was team capability, while the least important factor was culture.

Khoza & Marnewick (2021) evaluated scaled agile approaches from a South African perspective. In the study, expert opinion was taken with a questionnaire and they were asked to evaluate with a five-point Likert scale. The focus of the study is common scaling of the agile approaches, their success factors and the difficulties encountered in the approaches. According to the participants of the study, Scrum is the most adopted agile approach. Another result of the study is that the biggest challenge encountered is the lack of top management support. The participation of the project stakeholders is seen as the most important success factor

CHAPTER FOUR
METHODS

4.1 Multi-Criteria Decision Making

The number of decisions that humans must make is growing by the day, thanks to advances in science and technology. It is critical for the person who wants to enhance his/her life by his choices to make the best selection possible.

The more areas that will be affected by the choice, the more necessary it is to study and review the decision. As a result, we may divide decisions into operational, tactical, and strategic categories. When we look at this classification of decisions in terms of time, we can see that operational decisions are hourly and daily decisions, whereas tactical decisions are weekly or monthly decisions. Longer-term, or yearly decisions, on the other hand, are referred to as strategic decisions.

Although every decision should be regarded seriously, strategic decisions should be given special attention because their impact area and time frame are much larger.

When making easy, daily decisions, the human mind can respond swiftly, but when making complicated decisions, good analysis is required. We must carefully determine the choice variables by doing a thorough study.

In our globalizing world, businesses should be very careful when making decisions. With the widespread use of technology, science and communication tools, every wrong / or inappropriate decision to be taken affects many situations such as brand image, commercial activities, market share, etc. of the enterprise.

Although it is quite difficult to make a decision with the increase of criteria and alternatives that are useful for us in evaluating decisions, thanks to multi-criteria decision-making techniques, we are able to make optimum decisions in a short time in order to make the most appropriate choice and to rank the alternatives.

Multi-criteria decision making is a system of techniques that allows us to evaluate many alternatives simultaneously, according to more than one evaluation criteria. Decision-makers are asked to evaluate each alternative according to criteria, and with the application of the relevant method, the decision-making process results in success by ranking the alternatives or choosing the best alternative.

In addition, the criterion weight obtained by the decision-maker or after a series of evaluations made by the decision-maker, criterion weight can be reflected in the decision making process.

4.2 Classification of Multi-Criteria Decision Making Methods

Multi-criteria decision making problems can be classified as shown in the Table 4.1. In choice problems, the goal is to determine the best alternative. In sorting problems, the aim is to classify alternatives according to certain criteria. In ranking problems, alternatives can be ordered from the best to the worst or vice versa (Yıldırım & Önder, 2015).

Table 4.1 Classification of multi-criteria decision making (Yıldırım & Önder, 2015)

Choice Problems	*Sorting Problems*	*Ranking Problems*
AHP	AHP	AHPSort
ANP	ANP	UTADIS
MAUT/UTA	MAUT/UTA	FlowSort
MACBETH	MACHBETH	ELECTRE-Tri
PROMETHEE	PROMETHEE	
ELECTRE I	ELECTRE III	
TOPSIS	TOPSIS	
Goal Programming		

4.3 MOORA

The MOORA method was proposed in 2006 with the article titled "The MOORA Method and Its Application to Privatization in a Transition Economy". As in all other methods, the decision matrix is created as the first step in the MOORA method. MOORA Ratio Method was employed in this study. Therefore, its steps are explained in detail in the following.

After creating the decision matrix, the second step is to create the normalized matrix. The normalized matrix is calculated with the following formula. The *i* in the formula represents the alternatives, *j* represents the criteria, and x_{ij}^{*} represents the normalized value. Formula(4.1) is presented below (Yıldırım & Önder, 2015).

$$x_{ij}^{*} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^{2}}} \tag{4.1}$$

After the normalization process, the criteria in the prepared table are grouped according to the maximization and minimization direction. Maximization-oriented criteria are collected within themselves, minimization-oriented criteria are collected within themselves. From the maximization-oriented criteria, minimization-oriented criteria are subtracted. By calculating the y_{ij}^{*} value, the sorting process is completed. The y_{ij}^{*} value is calculated with the following formula. (Yıldırım & Önder, 2015).

$$y_{i}^{*} = \sum_{j=1}^{g} x_{ij}^{*} - \sum_{j=g+1}^{n} x_{ij}^{*} \tag{4.2}$$

In some cases, one criterion may be more important than another. In this case, criterion weights appear. Sorting is done by multiplying the normalized matrix by the criterion weight. In this case, we need to use the following formula. (Yıldırım & Önder, 2015).

$$\ddot{y}_{i}^{*} = \sum_{j=1}^{g} s_{j} x_{ij}^{*} - \sum_{j=g+1}^{n} s_{j} x_{ij}^{*} \tag{4.3}$$

4.4 The Step-Wise Weight Assessment Ratio Analysis (SWARA)

The SWARA method is a criterion weighting method. It is a useful method with fewer data and less processing steps.Steps are given below.(Alinezhad & Khalili, 2019).

4.4.1 Steps of SWARA

Firstly; the relative importance of the criteria is ranked by the decision-makers.

Coefficient “K” is then calculated for each decision maker.

$$K_j = \begin{cases} 1 & if\ j = 1 \\ S_j + 1 & if\ j > 1 \end{cases}; j = 1, \dots, n \tag{4.4}$$

The initial weight of each attribute is found as follows.

$$q_j = \begin{cases} 1 & if\ j = 1 \\ \frac{q_j}{K_j} & if\ j > 1 \end{cases}; j = 1, \dots, n \tag{4.5}$$

The *w(j)* value is calculated and the relative weight is found.

$$w_j = \frac{q_j}{\sum_{j=1}^{n} q_j} \tag{4.6}$$

The final weights are ranked and final rankings are found.

4.5 TOPSIS

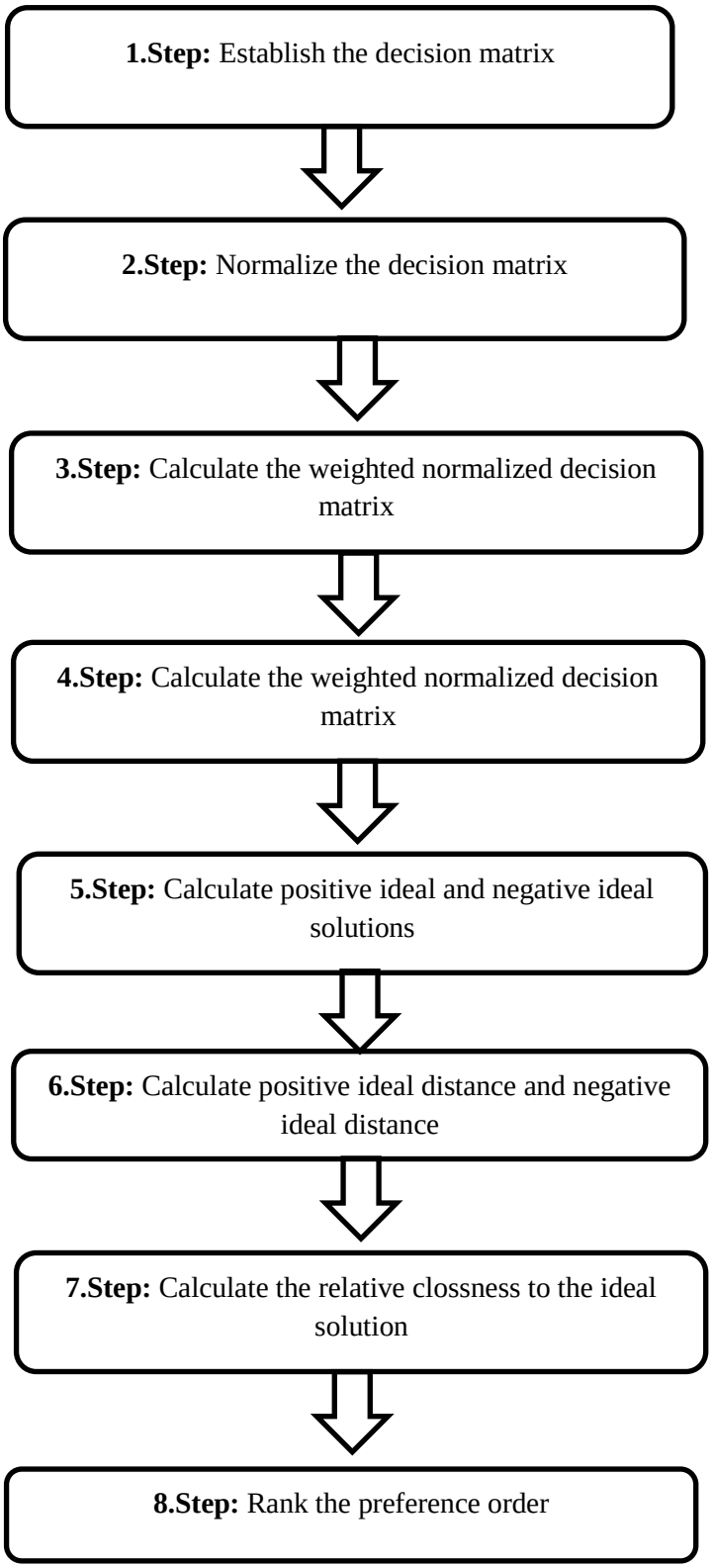

Figure 4.1 TOPSIS steps

In the TOPSIS method, while determining the closeness of an alternative to the ideal solution, it must also be far from the negative ideal solution.

While ranking the alternatives according to the criteria in the TOPSIS method, the first step is to create the decision matrix.

4.5.1 Steps of TOPSIS

The decision matrix is the matrix created by the decision-maker, in which alternatives are evaluated according to the criteria. The matrix *A* in the formula shows the decision matrix. The *m* value in the matrix *A* represents the alternatives and *p* represents the criteria (Yıldırım & Önder, 2015).

$$A_{ij} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mp} \end{bmatrix} \quad (4.7)$$

After the decision matrix is created, the squares of each a_{mp} value are obtained, and the column totals consisting of the sums of these values are obtained. Each a_{mp} is divided by the square root of the column sum to which it belongs. This process is called the normalization process and is calculated by the following formul(Yıldırım & Önder, 2015).

$$n_{ij} = \frac{a_{ij}}{\sqrt{\sum_{i=1}^{m} a_{ij}^2}} \; (i = 1, \ldots, m \; ve \; j = 1, \ldots, p) \quad (4.8)$$

The normalized matrix, *N* can be stated as in the following(Yıldırım & Önder, 2015).

$$N = \begin{bmatrix} n_{11} & n_{12} & \cdots & n_{1p} \\ n_{21} & n_{22} & \cdots & n_{2p} \\ \vdots & & & \vdots \\ n_{m1} & n_{m2} & \cdots & n_{mp} \end{bmatrix} \quad (4.9)$$

The matrix obtained by multiplying each criterion weight with the normalized matrix is called the weighted normalized matrix. One of the most important points to consider when weighting is that the sum of the criteria weights is equal to one. In the formula, *w* represents the criteria weights, and matrix *V* represents the weighted normalized matrix(Yıldırım & Önder, 2015).

$$\sum_{i=1}^{n} w_i = 1 \tag{4.10}$$

$$V = \begin{bmatrix} w_1 n_{11} & w_2 n_{12} & \cdots & w_n n_{1p} \\ w_1 n_{21} & w_2 n_{22} & \cdots & w_n n_{2p} \\ \vdots & & & \vdots \\ w_1 n_{m1} & w_2 n_{m2} & \cdots & w_n n_{mp} \end{bmatrix}_{ij} = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1p} \\ v_{21} & v_{22} & \cdots & v_{2p} \\ \vdots & & & \vdots \\ v_{m1} & v_{m2} & \cdots & v_{mp} \end{bmatrix} \tag{4.11}$$

After the normalization process, ideal and negative ideal solution values should be calculated. Calculations of them are presented in the following(Yıldırım & Önder, 2015).

$$A^* = \left\{ \max_j v_{ij} \,|j = 1, \dots, p; i = 1, \dots, m \right\} \tag{4.12}$$

$$\Rightarrow A^* = \{v_1^*, v_2^*, \dots, v_n^*\} \tag{4.13}$$

$$A^- = \left\{ \min_i v_{ij} \right\} \tag{4.14}$$

$$\Rightarrow A^- = \{v_1^-, v_2^-, \dots, v_n^-\} \tag{4.15}$$

After calculating the ideal and negative ideal solutions, the distance values from the ideal and negative ideal points are calculated with the following formulas.

Herein, d_{ij} is used to determine the Euclidean distance. While s_i^* represents the ideal distance, s_i^- represents the negative ideal distance(Yıldırım & Önder, 2015).

$$d_{ij} = \sqrt{\sum_{k=1}^{p} \left(x_{ik} - x_{jk}\right)^2} \tag{4.16}$$

$$S_i^* = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^*\right)^2} \tag{4.17}$$

$$S_i^- = \sqrt{\sum_{j=1}^{n} \left(v_{ij} - v_j^-\right)^2} \tag{4.18}$$

The following formula is used to calculate the relative closeness to the ideal solution(Yıldırım & Önder, 2015).

$$C_i^* = \frac{S_i^-}{S_i^- + S_i^*} \tag{4.19}$$

CHAPTER FIVE
APPLICATION

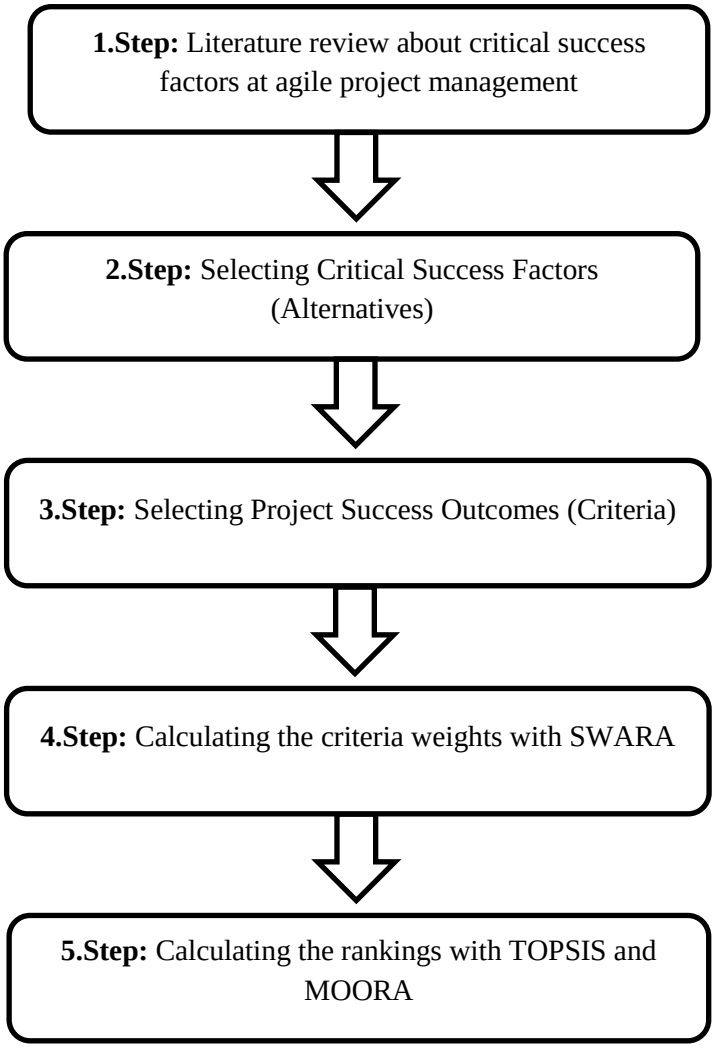

Figure 5.1 Application process flow schema

In this study, firstly, critical success factors in the agile software development process were investigated. Selected critical success factors are explained in detail in the related section. Project success criteria are explained in detail in the following.

5.1 Criteria Definitions

In this study, six criteria were determined in order to evaluate the impact of critical success factors on project success. These are; scope, time, the satisfaction of business partners and quality.

5.1.1 Quality

In general, the quality criterion can be defined as fully meeting the expected needs. Extensibility, simple coding, and the ability to write coding according to certain standards are all features that this definition encompasses.

5.1.2 Scope

The scope is the outline of the needs expected at the start of the project. The project scope determined in agile approaches is managed by dividing it into sub-branches.

5.1.3 Time

While evaluating the time criterion, the positive effect of success factors on the project time is taken as a basis. For example, if the situations that may lead to prolonging the duration of the project are eliminated or reduced, the score of the relevant factor in terms of time criterion will be high.

5.1.4 Satisfaction of Business Partners

Ensuring the satisfaction of business partners is critical to the enterprise's long-term sustainability, sector position, and reputation.

5.1.5 Cost

Commercial enterprises were established to make profit. One of the ways to increase the profit share is to reduce costs. Or, it is to take a proactive approach by determining the situations that will require extra costs in advance.

5.1.6 Addressing the Organizational Needs

It is the addressing of the needs necessary for the survival of the organization. It can be defined as enhancing staff motivation and achievement, conducting marketing studies, and making strategic decisions.

5.2 Survey

In this study, a questionnaire was applied to people in the R&D software industry, who have at least two years and a maximum of ten years of software industry

experience. The positions of the people participating in the survey at the time of the survey are as follows; Three of them are software engineers, and others are business analyst, project manager and software test engineer.

Table 5.1 illustrates the scale which was used to evaluate the alternatives based on the criteria.

Table 5.1 Evaluation scale

Evaluation Scale	Most insignificant	0
	Insignificant	3
	Average	5
	Significant	7
	Very Significant	9
	The Most Significant	10
	Intermediate Values	1,2,4,6,8

5.3 Criterion Weights with SWARA

Through the SWARA Method, each decision-maker was asked to rank the criteria. Then they were asked to answer the question of how many percent more important each criterion is than the others. Then, criterion weights were obtained for each decision-maker. The average criterion weight was found by taking the geometric mean of the weights obtained for each criterion. The evaluations of each decision-maker for each criterion and the relevant calculations are shown in Table 5.2.

Table 5.2 Criteria weights

Criteria/Decision-makers	*DM1*	*DM2*	*DM3*	*DM4*	*DM5*	*DM6*	*Common Criteria Weight*	*Rank*
Quality	0.21	0.17	0.18	0.2	0.2	0.16	0.19	1
Scope	0.20	0.15	0.14	0.13	0.13	0.14	0.15	4
Time	0.18	0.17	0.16	0.15	0.18	0.15	0.16	3
Satisfaction of Business Partners	0.16	0.18	0.21	0.18	0.16	0.22	0.18	2
Cost	0.14	0.19	0.19	0.19	0.19	0.19	0.18	2
Addressing Organizational Needs	0.12	0.14	0.12	0.15	0.14	0.14	0.14	5

5.4 TOPSIS Application

5.4.1 Decision Matrix

The decision matrix was created by using the average of the data collected from the six decision-makers. The matrix is presented in Table 5.3.

Table 5.3 Decision matrix

Question/Criteria	*Quality*	*Scope*	*Time*	*Satisfaction of Business Partners*	*Cost*	*Addressing Organizational Needs*
Educating the organization about agile project management	7.33	7.67	8.17	6.67	7	9
Management's Support and Contribution to Agile Project Management	8.33	7.67	8	7.83	7.5	7.67
Arranging Social Events	6.5	4.83	6.17	6.5	5.5	6.17
Willingness to Take Risk	7	7	8.5	7.83	7.5	7.33
Choosing the Appropriate Agile Approach Method (Scrum, Kanban etc.)	8.67	8.5	9.17	7.67	7.67	8.67
Well-defined coding standards	9.83	6.33	8.33	8	8.33	6.5
Code template and code reuse	8.83	5.83	9.5	6	9	5.67
Using Simple/clean code	9.67	6.17	9.33	5.33	8.33	5.33
Rigorous refactoring activities	7.83	7.17	5.67	7.5	5.33	8.33
Modular Design	7	7.67	6.67	7.17	7.5	8.33
Portable design	7.67	7.67	8	7.5	7.83	8
Extensible design	6.5	7.83	7.5	8.67	6	8.33
Right Amount of Documentation	8.83	8.33	7.5	8.5	6.33	8.83
Integration Testing	8.5	7.5	7.17	8.5	6.67	8.5
Maintenance of Code and Other Documentation	8.83	7	6.67	8.5	6.67	7.33
Backlog Prioritization	6.67	8.17	8	8.83	5.67	7.67
Daily Stand-up Meetings	7.83	6.67	8	7.67	5	7
Good Reporting of Project Status	8.33	8	7.83	8	6.67	7.17
Project Schedule	8.67	9	9.67	9	6.83	8.67
Regular Delivery of Software	9	9	9.5	9.5	6.33	8.67

Table 5.3 Continues

Correct determination of the scope and constraints of the project	9.5	9.5	9.5	9.5	7.83	9.33
Training Users	7	6.67	8.5	9.5	7	9.33
User/customer feedback	8.5	8.83	8.17	9.17	6.5	8.67
Building Good Relationships with Business Partners	7	6.5	7.67	9	6.17	8.33
Engaging everyone in the organization in the agile approach	8.5	7.67	9	7.67	9.17	8.83
Ensuring Effective Information Flow in the Organization	9.33	8.67	8.67	8.67	6.83	8.83
Empowering employees to make their own decisions (autonomy)	8.17	7.33	9.17	7	6	6.83
Being focused on providing qualified employees into the business and improving employee quality	9.83	8.83	8.83	8.17	9.17	9.83
Recruiting employees who have already worked in an agile environment	8,5	8,5	8,5	7,67	8,5	8,17
Small team size (5-11)	7,5	8	8,5	7,17	8,17	8,67
Increased collaboration and openness inside the company, as well as the establishment of trustworthy relationships	7,83	6,83	8	6,17	7,5	6

5.4.2 Normalized Matrix

According to the TOPSIS method, the normalized version of the decision matrix is presented in Table 5.4.

Table 5.4 Normalized matrix

Alternative/Criteria	*Quality*	*Scope*	*Time*	*Satisfaction of Business Partners*	*Cost*	*Meeting Organizational Needs*
Educating the organization about agile project management	0.159953	0.179707	0.177858	0.150290	0.174650	0.201686
Management's Support and Contribution to Agile Project Management	0.181765	0.179707	0.174228	0.176591	0.187125	0.171807

Table 5.4 Continues

Organization of Social Gatherings	0.141777	0.113293	0.134301	0.146533	0.137225	0.138193
Willingness to Take Risk (Acceptable)	0.152683	0.164080	0.185117	0.176591	0.187125	0.164337
Choosing the Appropriate Agile Approach Method (Scrum. Kanban etc.)	0.189036	0.199240	0.199636	0.172833	0.191283	0.194217
Well-defined coding standards	0.214483	0.148453	0.181487	0.180348	0.207917	0.145662
Code template and code reuse	0.192671	0.136733	0.206896	0.135261	0.224550	0.126988
Using Simple/ Clean Code	0.210848	0.144547	0.203266	0.120232	0.207917	0.119518
Rigorous refactoring activities	0.170859	0.167987	0.123411	0.169076	0.133067	0.186747
Modular Design	0.152683	0.179707	0.145190	0.161562	0.187125	0.186747
Portable design	0.167224	0.179707	0.174228	0.169076	0.195442	0.179277
Extensible design	0.141777	0.183613	0.163339	0.195377	0.149700	0.186747
Right Amount of Documentation	0.192671	0.195333	0.163339	0.191620	0.158017	0.197951
Integration Testing	0.185401	0.175800	0.156079	0.191620	0.166333	0.190482
Maintenance of Code and Other Documentation	0.192671	0.164080	0.145190	0.191620	0.166333	0.164337
Backlog Prioritization	0.145412	0.191427	0.174228	0.199134	0.141383	0.171807
Daily Stand-up Meetings	0.170859	0.156267	0.174228	0.172833	0.124750	0.156867
Good Reporting of Project Status	0.181765	0.187520	0.170598	0.180348	0.166333	0.160602
Project Schedule	0.189036	0.210960	0.210525	0.202891	0.170492	0.194217
Regular Delivery of Software	0.196307	0.210960	0.206896	0.214163	0.158017	0.194217
Correct determination of the scope and constraints of the project	0.207212	0.222680	0.206896	0.214163	0.195442	0.209156
Training Users	0.152683	0.156267	0.185117	0.214163	0.174650	0.209156
User/customer feedback	0.185401	0.207053	0.177858	0.206649	0.162175	0.194217
Building Good Relationships with Business Partners	0.152683	0.152360	0.166968	0.202891	0.153858	0.186747
Engaging everyone in the organization in the agile approach	0.185401	0.179707	0.196006	0.172833	0.228709	0.197951
Ensuring Effective Information Flow in the Organization	0.203577	0.203147	0.188747	0.195377	0.170492	0.197951

Table 5.4 Continues

Empowering employees to make their own decisions (autonomy)	0.178130	0.171893	0.199636	0.157804	0.149700	0.153132
Being focused on providing qualified employees into the business and improving employee quality	0.214483	0.207053	0.192377	0.184105	0.228709	0.220361
Recruiting employees who have worked in an agile environment.	0.185401	0.199240	0.185117	0.172833	0.212075	0.183012
Small team size (5-11)	0.163589	0.187520	0.185117	0.161562	0.203758	0.194217
Increased cooperation and transparency in the organization. establishing reliable relations	0.170859	0.160173	0.174228	0.139018	0.187125	0.134458

5.4.3 Weighted Matrix

The weighted normalized matrix was created using the weights acquired using SWARA technique

Table 5.5 Weighted matrix

Criteria Weights	*0.19*	*0.15*	*0.16*	*0.18*	*0.18*	*0.14*
Alternative/Criteria	Quality	Scope	Time	Satisfaction of Business Partners	Cost	Meeting Organizational Needs
Educating the organization about agile project management	0.030391	0.026956	0.028457	0.027052	0.031437	0.028236
Management's Support and Contribution to Agile Project Management	0.034535	0.026956	0.027876	0.031786	0.033683	0.024053
Organization of Social Meetings	0.026938	0.016994	0.021488	0.026376	0.024701	0.019347
Willingness to Take Risk (Acceptable)	0.029010	0.024612	0.029619	0.031786	0.033683	0.023007
Choosing the Appropriate Agile Methodology	0.035917	0.029886	0.031942	0.031110	0.034431	0.027190
Well-defined coding standards	0.040752	0.022268	0.029038	0.032463	0.037425	0.020393
Code template and code reuse	0.036608	0.020510	0.033103	0.024347	0.040419	0.017778
Using Simple/Clean Code	0.040061	0.021682	0.032523	0.021642	0.037425	0.016733
Rigorous refactoring activities	0.032463	0.025198	0.019746	0.030434	0.023952	0.026145

Table 5.5 Continues

Modular Design	0.029010	0.026956	0.023230	0.029081	0.033683	0.026145
Portable design	0.031773	0.026956	0.027876	0.030434	0.035180	0.025099
Extensible design	0.026938	0.027542	0.026134	0.035168	0.026946	0.026145
Right Amount of Documentation	0.036608	0.029300	0.026134	0.034492	0.028443	0.027713
Integration Testing	0.035226	0.026370	0.024973	0.034492	0.029940	0.026667
Maintenance of Code and Other Documentation	0.036608	0.024612	0.023230	0.034492	0.029940	0.023007
Backlog Prioritization	0.027628	0.028714	0.027876	0.035844	0.025449	0.024053
Daily Stand-up Meetings	0.032463	0.023440	0.027876	0.031110	0.022455	0.021961
Good Reporting of Project Status	0.034535	0.028128	0.027296	0.032463	0.029940	0.022484
Project Schedule	0.035917	0.031644	0.033684	0.036520	0.030689	0.027190
Regular Delivery of Software	0.037298	0.031644	0.033103	0.038549	0.028443	0.027190
Correct determination of the scope and constraints of the project	0.039370	0.033402	0.033103	0.038549	0.035180	0.029282
Training Users	0.029010	0.023440	0.029619	0.038549	0.031437	0.029282
User/customer feedback	0.035226	0.031058	0.028457	0.037197	0.029192	0.027190
Building Good Relationships with Business Partners	0.029010	0.022854	0.026715	0.036520	0.027695	0.026145
Engaging everyone in the organization in the agile approach	0.035226	0.026956	0.031361	0.031110	0.041168	0.027713
Ensuring Effective Information Flow in the Organization	0.038680	0.030472	0.030199	0.035168	0.030689	0.027713
Empowering employees to make their own decisions (autonomy)	0.033845	0.025784	0.031942	0.028405	0.026946	0.021439
Being focused on providing qualified employees into the business and improving employee quality	0.040752	0.031058	0.030780	0.033139	0.041168	0.030851
Recruiting employees who have worked in an agile environment.	0.035226	0.029886	0.029619	0.031110	0.038174	0.025622
Small team size (5-11)	0.031082	0.028128	0.029619	0.029081	0.036677	0.027190
Increased cooperation and transparency in the organization. establishing reliable relations	0.032463	0.024026	0.027876	0.025023	0.033683	0.018824

5.4.4 Positive Ideal and Negative Ideal Values

The positive ideal and negative ideal values are reported in Table 5.6.

Table 5.6 Positive ideal and negative ideal values

Alternative/Criteria	*Quality*	*Scope*	*Time*	*Satisfaction of Business Partners*	*Cost*	*Meeting Organizational Needs*
Positive Ideal Solution	0.040752	0.033402	0.033684	0.038549	0.041168	0.030851
Negative Ideal Solution	0.026938	0.016994	0.019746	0.021642	0.022455	0.016733

5.4.5 Ideal Distance, Negative Ideal Distance and Relative Closeness Values

Table 5.7 reports values of the ideal distance, ideal negative distance and relative closeness to the ideal solution. $S_i(+)$ is the ideal distance, $S_i(-)$ is the negative ideal distance, and c_i* is the relative closeness to the ideal solution.

Table 5.7 Positive ideal, negative ideal distances and rankings

Criteria	*Si(+)*	*Si(-)*	*Ci(*)*	Rankings
Correct determination of the scope and constraints of the project	0.009519	0.156101	0.942526	1
Being focused on providing qualified employees into the business and improving employee quality	0.010658	0.127093	0.922627	2
Regular Delivery of Software	0.022177	0.142859	0.865623	3
Ensuring Effective Information Flow in the Organization	0.025485	0.137441	0.843582	4
Project Schedule	0.022761	0.121477	0.842197	5
Recruiting employees who have worked in an agile environment.	0.028769	0.148370	0.837589	6
Engaging everyone in the organization in the agile approach	0.024871	0.113459	0.820202	7
Choosing the Appropriate Agile Approach Method (Scrum, Kanban etc.)	0.027929	0.113179	0.802072	8
User/customer feedback	0.030085	0.120643	0.800401	9
Training Users	0.037069	0.141210	0.792075	10
Well-defined coding standards	0.036067	0.123801	0.774394	11
Small team size(5-11)	0.036629	0.122399	0.769671	12
Right Amount of Documentation	0.035716	0.102548	0.741684	13
Integration Testing	0.040738	0.111344	0.732133	14
Management's Support and Contribution to Agile Project Management	0.039516	0.102406	0.721567	15

Table 5.7 Continues

Portable design	0.041088	0.096408	0.701168	16
Code template and code reuse	0.045640	0.106090	0.699201	17
Empowering employees to make their own decisions (autonomy)	0.050045	0.112267	0.691672	18
Building Good Relationships with Business Partners	0.049467	0.108245	0.686345	19
Maintenance of Code and Other Documentation	0.046517	0.100543	0.683689	20
Extensible design	0.049533	0.097176	0.662372	21
Good Reporting of Project Status	0.043559	0.085139	0.661539	22
Using Simple/clean code	0.048340	0.093817	0.659951	23
Backlog Prioritization	0.048840	0.092440	0.654302	24
Increased cooperation and transparency in the organization, establishing reliable relations	0.056510	0.094659	0.626181	25
Modular Design	0.050301	0.077028	0.604953	26
Daily Stand-up Meetings	0.059099	0.079858	0.574694	27
Rigorous refactoring activities	0.060468	0.078989	0.566404	28
Willingness to Take Risk (Acceptable)	0.046689	0.058546	0.556338	29
Educating the organization about agile project management	0.045876	0.048023	0.511435	30
Social Meetings	0.082562	0.065719	0.443206	31

5.5 MOORA

In the MOORA method, the initial matrix used in the TOPSIS method is used. Table 5.8 reports the weighted normalized matrix.

Table 5.8 MOORA weighted normalized matrix

	0.19	0.15	0.16	0.18	0.18	0.14
Alternative/Criteria	*Quality*	*Scope*	*Time*	*Satisfaction of Business Partners*	*Cost*	*Meeting Organizational Needs*
Educating the organization about agile project management	0.030391	0.026956	0.028457	0.027052	0.031437	0.028236
Management's Support and Contribution to Agile Project Management	0.034535	0.026956	0.027876	0.031786	0.033683	0.024053
Organization of Social Gatherings	0.026938	0.016994	0.021488	0.026376	0.024701	0.019347
Willingness to Take Risk (Acceptable)	0.029010	0.024612	0.029619	0.031786	0.033683	0.023007
Choosing the Appropriate Agile Approach Method (Scrum, Kanban etc.)	0.035917	0.029886	0.031942	0.031110	0.034431	0.027190

Table 5.8 Continues

Well-defined coding standards	0.040752	0.022268	0.029038	0.032463	0.037425	0.020393
Code template and code reuse	0.036608	0.020510	0.033103	0.024347	0.040419	0.017778
Using Simple/Clean Code	0.040061	0.021682	0.032523	0.021642	0.037425	0.016733
Rigorous refactoring activities	0.032463	0.025198	0.019746	0.030434	0.023952	0.026145
Modular Design	0.029010	0.026956	0.023230	0.029081	0.033683	0.026145
Portable design	0.031773	0.026956	0.027876	0.030434	0.035180	0.025099
Extensible design	0.026938	0.027542	0.026134	0.035168	0.026946	0.026145
Right Amount of Documentation	0.036608	0.029300	0.026134	0.034492	0.028443	0.027713
Integration Testing	0.035226	0.026370	0.024973	0.034492	0.029940	0.026667
Maintenance of Code and Other Documentation	0.036608	0.024612	0.023230	0.034492	0.029940	0.023007
Backlog Prioritization	0.027628	0.028714	0.027876	0.035844	0.025449	0.024053
Daily Stand-up Meetings	0.032463	0.023440	0.027876	0.031110	0.022455	0.021961
Good Reporting of Project Status	0.034535	0.028128	0.027296	0.032463	0.029940	0.022484
Project Schedule	0.035917	0.031644	0.033684	0.036520	0.030689	0.027190
Regular Delivery of Software	0.037298	0.031644	0.033103	0.038549	0.028443	0.027190
Correct determination of the scope and constraints of the project	0.039370	0.033402	0.033103	0.038549	0.035180	0.029282
Training Users	0.029010	0.023440	0.029619	0.038549	0.031437	0.029282
User/customer feedback	0.035226	0.031058	0.028457	0.037197	0.029192	0.027190
Building Good Relationships with Business Partners	0.029010	0.022854	0.026715	0.036520	0.027695	0.026145
Engaging everyone in the organization in the agile approach	0.035226	0.026956	0.031361	0.031110	0.041168	0.027713
Ensuring Effective Information Flow in the Organization	0.038680	0.030472	0.030199	0.035168	0.030689	0.027713
Empowering employees to make their own decisions (autonomy)	0.033845	0.025784	0.031942	0.028405	0.026946	0.021439
Being focused on providing qualified employees into the business and improving employee quality	0.040752	0.031058	0.030780	0.033139	0.041168	0.030851
Recruiting employees who have worked in an agile environment.	0.035226	0.029886	0.029619	0.031110	0.038174	0.025622
Small team size (5-11)	0.031082	0.028128	0.029619	0.029081	0.036677	0.027190
Increased cooperation and transparency in the organization, establishing reliable relations	0.032463	0.024026	0.027876	0.025023	0.033683	0.018824

5.5.1 MOORA Rankings

The ranking obtained by MOORA is presented Table 5.9.

Table 5.9 MOORA rankings

Alternative/Criteria	*yi(*)*	*Ranking*
Correct determination of the scope and constraints of the project	0.208886	1
Being focused on providing qualified employees into the business and improving employee quality	0.207747	2
Regular Delivery of Software	0.196228	3
Project Schedule	0.195644	4
Engaging everyone in the organization in the agile approach	0.193534	5
Ensuring Effective Information Flow in the Organization	0.192921	6
Choosing the Appropriate Agile Approach Method (Scrum, Kanban etc.)	0.190476	7
Recruiting employees who have worked in an agile environment.	0.189636	8
User/customer feedback	0.188320	9
Right Amount of Documentation	0.182689	10
Well-defined coding standards	0.182338	11
Small team size (5-11)	0.181777	12
Training Users	0.181337	13
Management's Support and Contribution to Agile Project Management	0.178890	14
Integration Testing	0.177668	15
Portable design	0.177317	16
Good Reporting of Project Status	0.174846	17
Code template and code reuse	0.172765	18
Educating the organization about agile project management	0.172530	19
Maintenance of Code and Other Documentation	0.171889	20
Willingness to Take Risk (Acceptable)	0.171717	21
Using Simple/clean code	0.170065	22
Backlog Prioritization	0.169565	23
Building Good Relationships with Business Partners	0.168938	24
Extensible design	0.168872	25
Empowering employees to make their own decisions (autonomy)	0.168360	26
Modular Design	0.168104	27
Increased cooperation and transparency in the organization, establishing reliable relations	0.161896	28
Daily stand-up meetings	0.159306	29
Rigorous refactoring activities	0.157937	30
Organization of Social Meetings	0.135843	31

5.6. Results

In this study, critical success factors in agile software development processes were evaluated by experts in the R&D software industry. Six decision-makers who participated in the survey evaluated 31 critical success factors according to the six criteria; scope", "time", "cost", "satisfaction of business partners", "quality", and "addressing organizational requirements".

Each decision-maker was asked to prioritize the criteria in the study, and criterion weights were calculated using the SWARA method and the data provided by the decision-makers. The order of importance of the alternatives, namely critical success factors rank, was determined using the criteria weights obtained using the TOPSIS and MOORA methods.

5.6.1 TOPSIS Rankings

Table 5.10 reports the results obtained by TOPSIS method.

Table 5.10 TOPSIS result table

Criteria	*Si(+)*	*Si(-)*	*Ci(*)*	*Rankings*
Correct determination of the scope and constraints of the project	0.009519	0.156101	0.942526	1
Being focused on providing qualified employees into the business and improving employee quality	0.010658	0.127093	0.922627	2
Regular Delivery of Software	0.022177	0.142859	0.865623	3
Ensuring Effective Information Flow in the Organization	0.025485	0.137441	0.843582	4
Project Schedule	0.022761	0.121477	0.842197	5
Recruiting employees who have worked in an agile environment.	0.028769	0.148370	0.837589	6
Engaging everyone in the organization in the agile approach	0.024871	0.113459	0.820202	7
Choosing the Appropriate Agile Approach Method (Scrum, Kanban etc.)	0.027929	0.113179	0.802072	8
User/customer feedback	0.030085	0.120643	0.800401	9
Training Users	0.037069	0.141210	0.792075	10
Well-defined coding standards	0.036067	0.123801	0.774394	11
Small team size (5-11)	0.036629	0.122399	0.769671	12
Right Amount of Documentation	0.035716	0.102548	0.741684	13
Integration Testing	0.040738	0.111344	0.732133	14

Table 5.10 Continues

Management's Support and Contribution to Agile Project Management	0.039516	0.102406	0.721567	15
Portable design	0.041088	0.096408	0.701168	16
Code template and code reuse	0.045640	0.106090	0.699201	17
Empowering employees to make their own decisions (autonomy)	0.050045	0.112267	0.691672	18
Building Good Relationships with Business Partners	0.049467	0.108245	0.686345	19
Maintenance of Code and Other Documentation	0.046517	0.100543	0.683689	20
Extensible design	0.049533	0.097176	0.662372	21
Good Reporting of Project Status	0.043559	0.085139	0.661539	22
Using Simple/Clean Code	0.048340	0.093817	0.659951	23
Backlog Prioritization	0.048840	0.092440	0.654302	24
Increased cooperation and transparency in the organization, establishing reliable relations	0.056510	0.094659	0.626181	25
Modular Design	0.050301	0.077028	0.604953	26
Daily Stand-up Meetings	0.059099	0.079858	0.574694	27
Rigorous refactoring activities	0.060468	0.078989	0.566404	28
Willingness to Take Risk (Acceptable)	0.046689	0.058546	0.556338	29
Educating the organization about agile project management	0.045876	0.048023	0.511435	30
Organization of Social Meetings	0.082562	0.065719	0.443206	31

5.6.2 MOORA Rankings

Table 5.11 presents the results obtained by using MOORA method.

Table 5.11 MOORA result table

Alternative/Criteria	*yi(*)*	*Ranking*
Correct determination of the scope and constraints of the project	0.208886	1
Being focused on providing qualified employees into the business and improving employee quality	0.207747	2
Regular Delivery of Software	0.196228	3
Project Schedule	0.195644	4
Engaging everyone in the organization in the agile approach	0.193534	5
Ensuring Effective Information Flow in the Organization	0.192921	6
Choosing the Appropriate Agile Approach Method (Scrum, Kanban etc.)	0.190476	7
Recruiting employees who have worked in an agile environment.	0.189636	8
User/customer feedback	0.188320	9
Right Amount of Documentation	0.182689	10
Well-defined coding standards	0.182338	11
Small team size (5-11)	0.181777	12
Training Users	0.181337	13

Table 5.11 Continues

Management's Support and Contribution to Agile Project Management	0.178890	14
Integration Testing	0.177668	15
Portable design	0.177317	16
Good Reporting of Project Status	0.174846	17
Code template and code reuse	0.172765	18
Educating the organization about agile project management	0.172530	19
Maintenance of Code and Other Documentation	0.171889	20
Willingness to Take Risk (Acceptable)	0.171717	21
Using Simple/Clean Code	0.170065	22
Backlog Prioritization	0.169565	23
Building Good Relationships with Business Partners	0.168938	24
Extensible design	0.168872	25
Empowering employees to make their own decisions (autonomy)	0.168360	26
Modular Design	0.168104	27
Increased cooperation and transparency in the organization, establishing reliable relations	0.161896	28
Daily Stand-up Meetings	0.159306	29
Rigorous refactoring activities	0.157937	30
Organization of Social Meetings	0.135843	31

5.6.3 Examining of The Alternatives

5.6.3.1 Correct Determination of the Scope and Constraints of the Project

The general lines of the project are determined in agile approaches, and software is produced in small pieces along these general lines. We can easily say that when the project's scope and constraints are not clearly defined, it has a negative impact on all criteria. The lack of a clear definition of the project's scope results in extra effort in the project's progress, which results in additional time and costs.

Quality problems can also be experienced during the integration of the software parts produced with the new feature in agile approaches, in addition to extra-scope requests for software produced in working parts. Employees who have to deal with a similar issue again as a result of a situation that is out of scope may experience low morale and motivation as a result of this situation.

It is clearly seen that this factor directly affects the coverage criteria. If this factor is not handled correctly, the extra costs, loss of time, and deterioration in quality may adversely affect customer satisfaction.

5.6.3.2 Being Focused on Providing Qualified Employees into the Business and Improving Employee Quality

The concept of a servant leader, rather than a manager, is used in agile approaches. In other words, rather than leading the employees, the leader assists them in finding their way. It is critical in Agile approaches for the team to make its own decisions. The team is expected to have specific knowledge and experience in such situations. Quality errors will be reduced, more work will be completed in less time, user expectations will be better understood, and the scope will be determined in a healthy manner if the team is made up of talented and constantly improving individuals.

Customer satisfaction will rise as a result of the presence of an experienced and dependable team, and the team will be motivated by the joy of success. One thing to keep in mind about this factor is that we shouldn't directly assume that including inexperienced people in agile approaches is a bad idea. People are allowed to learn by doing in agile organizations, and their teammates assist them in their learning and self-development.

5.6.3.3 Regular Delivery of Software

Thanks to the “Regular Delivery of Software” factor, the customer who sees the running software experiences, sees that their wishes are embodied and their satisfaction increases. Working software, which is an output of customer requests, an indicator of scope and quality, will be detected in a short time if quality and scope differ from customer expectations, and will cause the cost to change at lower levels, rather than occur in later stages.

5.6.3.4 Project Schedule

One of the most important aspects of agile approaches is project scheduling. The fragmentation and prioritization of the project within the determined scope are the foundations for delivering software in small pieces. During the project's scheduling phase, it's critical to not only prioritize the project's components but also to estimate the time required to complete the work and to assess any potential problems or risks.

Because project management is iterative in Agile approaches, project schedules should be structured in a way that allows for flexibility in response to changes. Time, cost, and customer dissatisfaction will all be affected if the project schedule is disrupted. As a result of the project's disruption, quality issues may arise during the efforts to speed up the project. Employee motivation and work quality can both suffer as a result of the project's disruption.

5.6.3.5 Engaging Everyone in the Organization in the Agile Approach

The fact that everyone in the organization respects the agile approach's rules and standards while working toward similar goals will have a favourable impact on all project outcomes. A team, rather than a person, is basic in Agile methods. The team's decisions are significant. As a result, everyone's participation is required to guarantee the success of the agile mindset in the team as a whole, and when we view the company as a huge team.

5.6.3.6 Ensuring Effective Information Flow in the Organization

Individuals and interactions are more important than processes and tools to the Agile Manifesto (2001) Working software is more important than comprehensive documentation. However, the provision of these two items depends on the establishment of an effective communication network. It is important that the information flows naturally between individuals and that it is clear and understandable. In general, we can say that a software project team consists of a project manager, software engineer/computer engineer, test engineer, agile coaches, and business analysts. In this case, it is critical that the business analyst, who is involved in the analysis of the project, correctly determines the expected requirements and clearly conveys them to the project management and software team. Likewise, the software team needs to clearly articulate to the business analyst whether the project is in expected quality, or not. In this process, the project management team should be in an effective and clear communication flow with both the business analyst and the software team during the use of time and the distribution of cost elements. In other words, the ability to communicate, which is an indispensable part of our lives, is a critically important factor in agile approaches,

which have the principle of producing software that works in a short time and minimum documentation.

5.6.3.7 Recruiting Employees Who Have Already Worked in an Agile Environment

Since people coming from an organization where agile approaches are applied have understood and assimilated the agile mindset, they can facilitate the implementation of agile, especially in businesses that are in transition to agile. Seeing the adoption of the Agile philosophy, it will be easier for other employees to adapt to agile and focus on the positive points brought by agile. The team that adopts the Agile philosophy will make their own decisions faster thanks to autonomy, the cost and time elements will decrease, and the morale and motivation of the team will increase. The morale and motivation of the team will increase and the quality of their work will also increase, and these positive project outputs will also affect customer satisfaction.

5.6.3.8 Appropriate Agile Approach Methodology

Each agile methodology has its own characteristics. There are many studies in the literature regarding the selection of these methodologies. Therefore, there are many factors that affect the selection of the appropriate methodology, and making this choice requires rigor. The method chosen should be suitable for the structure of the organization. Each of the methods has advantages and disadvantages over the other. A general conclusion can be made that the method that is most easily adapted to the organization and where you get the best results is the best. Therefore, this factor is very important as the proper implementation of agile practices will also directly affect the project outputs.

5.6.3.9 Well-Defined Coding Standards

We can say that coding standards actually facilitate communication between software developers. Since in Agile approaches, working software is given priority over documentation, when the software developer writes the code according to a certain standard, it will be easier to understand when he looks back. In this case, it will help the codes to be easy to understand, reduce quality problems, and reduce

time and cost-related problems as the complexity of the code is overcome. When there are changes in the scope of the project, it will be easier to return to the code and make changes. In addition, another programmer will easily understand the code written by a software developer, a kind of code communication will be established within the team.

5.6.3.10 User Feedback

Agile approaches are known for being dynamic, their capacity to adapt to changes. However, the basis of these changes is the user. The experience of the user of the software is more important than the producer. For this reason, being in constant interaction with the user will serve the perfection of the software. Thanks to the user feedbacks, the arrangements to be made within the scope of the project will be determined more clearly. Distinguishing important and unimportant features for the user will also affect the time, quality, and cost factors. The satisfaction of the user, who sees that their ideas and opinions are cared for and transformed into working software, will also increase. Acting with the correct interpretation and understanding of the feedback from the customer, the team will distinguish the works that do not add value and will be effective in meeting the organizational needs.

5.6.3.11 Code Template and Code Reuse

Designing codes in a reusable structure directly affects quality, cost, and time. The factor that positively affects these outputs will naturally also affect customer satisfaction. Here is an example of this factor. Let's say a software company produces similar projects. Since this company does similar work, standardizing its work and using the code structure from one project in another project will enable the work to be completed in a shorter time, and the completion of the work in a shorter time will naturally affect the cost positively. The enterprise, which has a certain code standard, will naturally have reached a certain quality standard. Although this factor seems not to affect the scope, it is also very important for scope. By determining the code templates in the working scope, the codes that are included in the scope but not ready will be written as templates, thus creating a basis for future projects and a standard for the current project.

5.6.3.12 Using Simple/Clean Code

Since the code is simple and easy to understand, it will prevent time and cost losses that will occur during the software phase. Also, the software developer will have information about the scope of the code thanks to its easy understandability. In cases where the user requests change, it will be easier to change the code and adapt it to the requests, and the customer's satisfaction will naturally increase as their requests are met quickly.

5.6.3.13 Right Amount of Documentation

Saying that working software is more important than documentation in Agile philosophy does not make documentation unimportant. Understanding customer expectations correctly, providing the right amount of documentation, and conveying them to the project team clearly and understandably are key points. In cases where customer requirements are not documented correctly, customer/user requests will be discussed again, which will lead to loss of time and customer dissatisfaction. Doing the same job again will cause extra costs. The morale and motivation of the team will be adversely affected if similar work is incomplete due to simple information confusion.

5.6.3.14 Small Team Size

Agile approaches are considered important for the implementation of Agile methods. Agile approaches require verbal communication within the team, the interaction between individuals, and the team's self-decision. For this reason, as the number of people increases, the implementation of the basic principles of agile will become more difficult. This will affect the success of the project.

5.6.3.15 Management's Support and Contribution to Agile Project Management

However, since managers make strategic decisions in businesses, the contribution of managers and the provision of the necessary agile environment and its continuation during the implementation of agile approaches are extremely important for the implementation of agile approaches.

5.6.3.16 Training Users

Educating users about the software not only makes the software easier to understand but also allows customers to offer ideas that will contribute to the software development process. This contributes positively to project outputs, as well as customer feedback.

5.6.3.17 Good Reporting of Project Status

A proper understanding of the project status ensures the correct use of project time. Efficient use of project time, on the other hand, reduces costs. By checking the suitability of the project status to the scope of the project, it is ensured that the time management is done correctly, as well as the determination of the changes to be made and their inclusion in the project plan.

5.6.3.18 Portable Design

Ensuring the usability of the same software in different environments is not only a feature of agile approaches but also among the requirements of today. Therefore, writing the same software to be adapted to different environments will prevent cost, time, and quality losses.

5.6.3.19 Integration Testing

Integration testing is a critical success criterion in agile approaches where piecemeal software is produced. Therefore, giving the necessary importance to the integration test will prevent time, cost, and quality problems.

5.6.3.20 Maintenance of Code and Other Documentation

The last update and maintenance according to the requirements of the day prevents errors that may arise in the software. In this case, timely corrections in the software provide a proactive approach to time, quality, and cost problems.

5.6.3.21 Willingness to Take Risk

Agile's key features include keeping up with change. Taking risks at an acceptable level can bring success in agile.

5.6.3.22 Empowering Employees to Make Their Own Decisions

As is known, agile has no hierarchical structure. The decision-making power rests with the team. This, in turn, will have a positive impact on many factors such as quality and cost, resulting in a reduction in costs and an increase in quality, due to time losses and the fact that people who know the job take an active role in the decision-making mechanism.

5.6.3.23 Backlog Prioritization

In short, this factor, which means the prioritization of tasks, is actually of critical importance during the management of the project. It is a factor that takes many factors into consideration besides which job is the priority, the duration of the job and the scope of the cost. If mistakes are made during backlog prioritization, problems may arise in project cost and time management.

5.6.3.24 Educating the Organization About Agile Project Management

Informing the organization about agile project management and its importance will ensure the assimilation and conscious implementation of the agile mindset. In this case, it will naturally affect the project outputs positively.

5.6.3.25 Good Relationships with Business Partners

The point to be considered in this factor is not only customers but also all contacts in the business relationship that are meant by business partners. It is emphasized at every opportunity how important communication and interactions are in agile. Effective communication is very important for situations such as understanding the scope correctly and quickly obtaining commercial data to be obtained from business partners with whom information is shared. Although it seems that effective communication has no direct effect on cost or quality, the reality is that it is very important. It is among the possible possibilities in business life that you can obtain important data in the software development process only because you have established good relations with the competitor before. Or, in case of any quality problems or customer needs, turning the crisis into an opportunity will not only help you get ahead of your competitors but will improve your quality and increase customer satisfaction.

5.6.3.26 Extensible Design

Producing software parts in short time intervals is among the basic principles of agility. This factor is therefore critical to success in agile. Although Scope should be understood correctly at the beginning of the project, no project is simple enough to be predicted down to the smallest detail. Variations in scope and customer requests are very likely as features and diversity increase, especially in today's advanced software world. For this reason, the software being "Extensible" reduces time losses, affects quality, and prevents cost increases.

5.6.3.27 Increased Collaboration and Openness Inside the Company

According to Agile, communication and interaction between people are very important. One of the most important points in agile businesses where comprehensive documentation is not kept and team spirit is essential is to establish reliable relationships. Documentation is the source of information, but if we consider that the knowledge and experience are in the people and this information can be transferred through effective communication, establishing reliable relationships and providing collaboration are the main factors for the success of the agile project.

5.6.3.28 Modular Design

Modular design is the principle of designing software that works in independent parts. We can easily say that this factor is the basis of the production of software that works in a short time in Agility.

5.6.3.29 Daily Stand-up Meetings

In project management, it provides benefits in many areas such as daily monitoring of projects, situation analysis and information sharing, and the emergence of problems and opportunities that may arise from the determination of the project status.

5.6.3.30 Rigorous Refactoring Activities

Rigorous refactoring activities are s the cases of making changes to the internal structure of the code without changing the external structure. It can have a positive effect on the quality of the code.

5.6.3.31 Arranging Social Events

Organizing social meetings within the organization has a positive effect on the interaction and communication within the team, increasing the morale and motivation of the team.

5.7 Discussion

In this study, critical success factors for agile software project management were evaluated by using SWARA, TOPSIS, MOORA methods. The literature is based on the determination of critical success factors. It is thought that it would be beneficial to be informed about the main success factors from the sector and to be included in the studies.

A questionnaire was applied to six experts in the research and development sector. More expert opinions can be obtained in future studies. A sector-based agile success criterion can be determined by obtaining data from experts in software projects in different sectors. In addition, these studies may lead to sector-specific success factors. In addition, by applying fuzzy multi-criteria decision-making approaches, verbal evaluations can be adapted to the study more effectively.

CHAPTER SIX
CONCLUSION

Critical success factors in agile project management were investigated using multi-criteria decision making approaches in this study. Experts in the software R&D industry assessed critical success criteria derived from the literature. The priorities that firms in the R&D software sector should pay attention to while using agile project management have been identified as a result of the ranking.

It is thought that this study will contribute to the adaptation of R&D software companies to agile project management. There are few studies in the literature that examine project success in detail. In this study, the success of the project was not evaluated in general, but divided into six sub-headings and examined in more detail. A detailed examination of the success of the project has been provided and a contribution to the literature has been made.

REFERENCES

Agile Alliance. (n.d.). *What is agile?* Retrieved July 8, 2021, from https://www.agilealliance.org/agile101/

Agile Manifesto. (2001). *Manifesto for agile software development.* Retrieved July 15, 2021, from https://agilemanifesto.org/iso/en/manifesto.html.

Aldahmash, A. M. (2018). *A review on the critical success factors of agile software development: An empirical study*. Phd Thesis, University of Southampton.

Aldahmash, A., Gravell, A., & Howard, Y. (2017). Using factor analysis to study the critical success factors of agile software development. *Journal of Software, 12*(12), 960-961. doi:10.17706/jsw.12.12.957-963

Alinezhad, A., & Khalili, J. (2019). New methods and applications in multiple attribute decision making (MADM). In: C. C. Price, (Ed.). *International series in operations research & management science* (Vol. 277). Cham: Springer.

Aljuhani, A., & Alhubaishy, A. (2020). Incorporating a decision support approach within the agile mobile application development process. *3rd International Conference on Computer Applications & Information Security (ICCAIS)*, 1-6. doi:10.1109/ICCAIS48893.2020.9096751.

Arcos-Medina, G., & Mauricio, D. (2020). Identifying factors influencing on agile practices for software development. *Journal of Information and Organizational Sciences,, 44*(1). doi:10.31341/jios.44.1.1.

Bergmann, T. (2018) *The Relationship Between Agile Project Management and Project Success Outcomes*. Phd Thesis, University of Central Florida.

Brown G, A. (2015). *An examination of critical success factors of an agile project.* ProQuest Number: 3684950 Phd Thesis, School of Business and Technology. Western Michigan University.

Chiyangwa, T. B., & Mnkandla, E. (2017). Modelling the critical success factors of agile software development projects in South Africa. *South African Journal of Information Management, 19*(1), a838. doi:10.4102/sajim.v19i1.838.

Clarizen. (2020). *Why project scope is so important*. Retrieved July 8, 2021, from https://www.clarizen.com/project-scope-important/?cn-reloaded=1

Darwish, N, R., & Rizk, N, M. (2015). Multi-dimensional success factors of agile software development projects. *International Journal of Computer Applications, 118*(15), 23-26

Demirtaş, N., Tuzkaya, U. M. ve Seker. S. (2014). Project management methodology selection using SWOT-Fuzzy AHP. *Proceedings of the World Congress on Engineering Vol II*. ISSN: 2078-0966. http://www.iaeng.org/publication/WCE2014/WCE2014_pp1121-1126.pdf

Digite. (n.d.). *What is scrum?* Retrieved July 8, 2021, from https://www.digite.com/agile/scrum-methodology/

Dikert, K., Paasivaara, M., & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *The Journal of Systems and Software, 119*(2016), 87-108. doi:10.1016/j.jss.2016.06.013

El Beggar, O. (2018). Multicriteria decision aid for agile methods evaluation using fuzzy PROMETHEE. *Journal of Software: Evolution and Process,* 30. doi:10.1002/smr.2108.

Ghayyur, S. A. K., Ahmed, S., Ali, M., Naseem, A., Razzaq, A., & Ahmed, N. (2018). A systematic literature review of success factors and barriers of agile

software development. *International Journal of Advanced Computer Science and Applications, 9*(3), 287-289.

Hameed, T. A. E., Latif, M. A. E., & Kholief, S. (2016). Identify and classify critical success factor of agile software development methodology using mind map. *International Journal of Advanced Computer Science and Applications, 7*(5), 83-91.

Hummel, M., & Epp, A. (2015). Success factors of agile information systems development: A qualitative study. *48th Hawaii International Conference on System Sciences.* doi:10.1109/HICSS.2015.598.

Ishizaka, A., & Nemery, P. (2013). *Multi-criteria decision analysis methods and software.* United Kingdom: John Wiley & Sons Ltd.

Khan, A. A., Shameem, M., Nadeem, M., & Akbar, M. A. (2021). Agile trends in Chinese global software development industry: Fuzzy AHP based conceptual mapping. *Applied Soft Computing,* 102. doi:10.1016/j.asoc.2021.107090.

Khan. P. M., & Beg, M. M. S. (2013). Extended decision support matrix for selection of SDLC-models on traditional and agile software development projects. *Third International Conference on Advanced Computing and Communication Technologies (ACCT),* pp. 8-15. doi:10.1109/ACCT.2013.12.

Khoza, L., & Marnewick, C. (2021). Challenges and success factors of scaled agile adoption–A South African perspective. *The African Journal of Information Systems, 13*(2), 169-170.

Lynn, R. (n.d.). *What is in fdd in agile?.* Retrieved August 15, 2021, from https://www.planview.com/resources/articles/fdd-agile/

Mahajan, G., Simon, R., & Khatri, S. K. (2017). Selection of framework for implementing agile in software industry by fuzzy AHP approach. *International*

Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS), pp. 713-718. doi:10.1109/ICTUS.2017.8286100.

Meenakshi, E., Singh, A., & Agrawal, D. (2020). A review on success factors of agile software development. *International Journal of Applied Science and Engineering, 8*(1), 33-36. doi:10.30954/2322-0465.1.2020.4

Mohammed, A. M., & Abushama, H. M. (2013). Popular agile approaches in software development: Review and analysis. *Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference.* doi:10.1109/ICCEEE.2013.6633925.

Muhammad, A., Siddique, A., Naveed Q., Saleem, U., Hasan, M., & Shahzad, B. (2021). Investigating crucial factors of agile software development through composite approach. *Intelligent Automation & Soft Computing*, 27(1). doi:10.32604/iasc.2021.014427

Nasehi, A. (2013). *A Quantitative Study on Critical Success Factors in Agile Software Development Projects; Case Study IT Company.* University of Boras.

Nasir, M. H. N. M., Sahibuddin, S., Ahmad, R., & Fauzi, S. S. M. (2015). How the PMBOK addresses critical success factors for software projects: A multi-round Delphi study. *Journal of Software,* 1283-1300. doi:10.17706/jsw.10.11.1283-1300.

Naslund, D., & Kale, R. (2020). Is agile the latest management fad? A review of success factors of agile transformations. *International Journal of Quality and Service Sciences*, 12(4), 489-450. doi:10.1108/IJQSS-12-2019-0142

Nguyen, D. S. (2016). Success factors for building and managing high performance agile software development teams. *International Journal of Computer, 20*(1), 51-82.

Nguyen, D. S. (2016). Success factors that influence agile software development project success. *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, *17*(1), 171-222.

Noteboom, C., Ofori, M., Sutrave, K., & El-Gayar, O. (2021). Agile project management: A systematic literature review of adoption drivers and critical success factors. *Proceedings of the 54th Hawaii International Conference on System Sciences*, pp. 6777-6778.

Öztürk, V. (2013). Selection of appropriate software development life cycle using fuzzy logic. *Journal of Intelligent & Fuzzy Systems, 25*(2013), 797–810. doi:10.3233/IFS-120686.

Rai, A. K., Agarwal, S., & Kumar, A. (2018). A novel approach for agile software development methodology selection using fuzzy inference system. *2018 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 518-526. doi:10.1109/ICSSIT.2018.8748767.

Scaled Agile Framework. (2021). *Product owner.* Retriveed July 15, 2021, from https://www.scaledagileframework.com/product-owner/

Scrum. (n.d.). *What is a daily Scrum?* Retrieved July 8, 2021, from https://www.scrum.org/resources/what-is-a-daily-scrum

Shakya, P., & Shakya, S. (2020). Critical success factor of agile methodology in software industry of Nepal. *Journal of Information Technology and Digital World*, *2*(3), 135-143. doi:10.36548/jitdw.2020.3.001

Shecter, O. (n.d.). *The product backlog: A step-by-step guide*. Retrieved August 30, 2021, from https://www.toptal.com/product-managers/agile/product-backlog-step-by-step-guide

Silva, K. M. B., & Santos, S. (2015). Critical factors in agile software projects according to people, process, and technology perspective. *56th Brazilian Workshop on Agile Methods,* pp. 48-54. doi:10.1109/WBMA.2015.19

Silva, V. B. S., Schramm, F., & Damasceno, A. C. (2016). A multicriteria approach for selection of agile methodologies in software development projects. *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 002056-002060. doi:10.1109/SMC.2016.7844542.

Sinha, R., Shameem, M., & Kumar, C. (2020, February 27–29). SWOT: strength, weaknesses, opportunities, and threats for scaling agile methods in global software development. *ISEC 2020*, Jabalpur, India. doi:10.1145/3385032.3385037

Solutions, I. Q. (2009). *Why agile teams need to embrace risk?* Retrieved 12 August, 2021, from https://www.solutionsiq.com/resource/blog-post/agile-teams-need-embrace-risk/

Syeda, S. H. (2017). *An Exploratory Study to Identify Critical Success Factors of Agile Systems Engineering.* Proquest Number:10743062.

Tsoy, M., & Staples, D. (2020). Exploring critical success factors in agile analytics projects. *Proceedings of the 53rd Hawaii International Conference on System Sciences,* pp. 984-986.

Vavpotič, D., & Vasilecas, O. (2011). An approach for assessment of software development methodologies suitability. *Electronics and Electrical Engineering*, 8(114), 107-110. doi:10.5755/j01.eee.114.8.707.

Vithana, V. N., Fernando, S. G. S., & Kapurubandara, M. (2015). Success factors for agile software development – a case study from Sri Lanka. *International Journal of Computer Applications*, *113*(17), 12-14.

Wikipedia. (2021a). *Agile software development*. Retrieved July 8, 2021, from https://en.wikipedia.org/wiki/Agile_software_development

Wikipedia. (2021b). *Critical success factor*. Retrieved July 8, 2021, from https://en.wikipedia.org/wiki/Critical_success_factor

Wikipedia. (2021c). *Code refactoring*. Retrieved August 10, 2021, from https://en.wikipedia.org/wiki/Code_refactoring

Wikipedia. (2021d). *Software portability*. Retrieved August 15, 2021, from https://en.wikipedia.org/wiki/Software_portability

Wikipedia. (2021e). *Extensibility*. Retrived August 8, 2021, from https://en.wikipedia.org/wiki/Extensibility

Yıldırım, B. F. ve Önder, E. (Ed.) (2015). *Operasyonel, yönetsel ve stratejik problemlerin çözümünde çok kriterli karar verme yöntemleri* (2. Baskı). Bursa: Dora Yayınevi.