

Investigating Crucial Factors of Agile Software Development through Composite Approach

AbdulHafeez Muhammad¹, Ansar Siddique^{2,*}, Quadri Noorulhasan Naveed³, Usman Saleem¹, Mohd Abul Hasan⁴ and Basit Shahzad⁵

¹Department of Computer Science, Bahria Uuniversity Lahore Campus, Punjab, 54600, Pakistan

²Department of Software Engineering, University of Gujrat, Punjab, 50700, Pakistan

³College of Computer Science, King Khalid University, Abha, 62529, Saudi Arabia

⁴Department of Civil Engineering, King Khalid University, Abha, 62529, Saudi Arabia

⁵Department of Software Engineering, National University of Modern Languages (NUML), Islamabad, 44000, Pakistan

*Corresponding Author: Ansar Siddique. Email: dr.ansarsiddique@uog.edu.pk

Received: 20 September 2020; Accepted: 12 October 2020

Abstract: The major emphasis of Software Engineering (SE) discipline is to produce successful software systems. The success of software projects is estimated through quadruple measures including budget, cost, scope, and quality. To meet this aim of SE, several software development processes are presented in the literature. Such processes are categorized into two different methodologies which are known as traditional and agile software development methodologies. The issue with traditional software development methodologies is that they had not shown any remarkable progress towards the fundamental goal of SE. Consequently, software development organizations have started to adopt agile methodologies in the pursuit of successful software development. However, agile adoption does not come without challenges that vary from one context to another. Therefore, it is necessary to figure out the key factors of agile software development for successful project outcomes. In the wake of such need, this study investigated the Critical Success Factors (CSFs), categorized and prioritized them through a mixed-method approach. Such an approach was based on the detailed literature review and Delphi method accompanied with Multi-Criteria Decision Analysis (MCDA) technique. Twelve CSFs were revealed and categorized into people, organization and technical dimensions. Among these factors, 'team capability' was found the most significant factor where 'culture' was revealed as the least significant factor. The findings of the study would be promising for agile software development that is carried on in the local software industry.

Keywords: Critical success factors; agile methodology; software development; analytical hierarchical process

1 Introduction

Software development is carried on using two kinds of methodologies namely traditional vs. agile software development methodologies [1]. Former methodologies encompass heavyweight characteristics



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

where the latter consists of lightweight characteristics. The term 'agile' means a rapid and flexible response to changes in software development. Agile alliance considers agile methodologies as a better way to approach software development because it focuses on interactions over processes, working software over documentation, collaboration over a contract and responding to change over following a plan [1,2]. At present, agile has become a prominent approach to software development as nearly 67% of software development organizations are employing agile methodologies to produce software [3]. According to the findings of the Standish group, the success rate of agile software projects is far greater than the projects being developed using traditional methodologies (i.e., 39% vs. 11%) [4]. It has been shown that one of the reasons for software failure is that software engineering practices at different stages of the software development life cycle (SDLC) are not being properly followed. It happens mainly owing to the inflexibility and heaviness of traditional software development methodologies [5].

The adoption of agile methodologies is motivated by its strength to deal with the inherent complexities of the software development. Such methodologies increase the likelihood of success by rapid and iterative responses to stakeholder needs [6]. The lightweight methodologies could help software organizations in developing successful software solutions. Initially, agile methodologies were applied in small teams, projects, and companies but currently, its usage has been bolster for the development of larger systems [7]. However, the adoption of agile software development methodologies always poses some challenges as the developer community has accustomed to traditional methodologies. So the experience with newer methodology may give rise to certain issues that are contextual in nature and vary from one industrial context to another [8]. Therefore, for an effective realization of agile software development methodologies in the context of Pakistan software industry, there is a need to examine critical success factors (CSFs) of agile software development in that particular context. The identification of such factors would obviously provide insight to the software development community and help them in the smooth application of agile methodologies. Presently agile methodologies have become the leading methodologies for all kind of software development [9]. It is being applied equally in both local and global software development settings [5]. Recently, agile methodologies are also being considering beyond pure software development projects such as projects in the domain of mechatronics [10]. As agile is massively utilized for software development so defining the factors that may have an impact on the success of software development projects, drawn the attention of researchers since the last many years [9]. The researchers have investigated different contexts to determine critical success factors. For example, a study has been conducted with software developers who worked with IT companies of Yugoslavia countries. Similarly, some other studies have been conducted in different countries including South Africa, Europe, USA, UK & Brazil [9,11,12,13]. These are the survey studies which have been conducted on the CSF of agile software development projects through quantitative research techniques. Some other studies had identified the CSF on agile software development projects using systematic literature review (SLR) [14,15]. However, the existing research lacks in identifying and prioritizing CSFs of agile software development considering their relative importance towards success of software projects. There is no structured model exists to specify the key factors that could play part in bringing success in agile software development. The formulation of such models considering diverse contexts of software industry are necessary to guide developers about the successful implementation of agile software projects.

This study aimed to identify, classify and prioritize (*based on their relative importance*) key success factors of agile software development. A mixed-method approach has utilized to look at the factors first from the literature review and later involving experts from the local software industry as well as academia. The outcome of this research could be taken into account while implementing software projects using agile software development methodologies. This research has conducted to explore the answer to these three research questions (i) What are the critical factors which influence the success of agile software development specifically in the context of the local software industry of Pakistan? (ii)

What are the major dimensions which represent different identified factors? (iii) Which factors are more important relative to other identified factors for successful agile software development?

The rest of the paper organized as follows: Section 2 describes the background and related work to the study. Section 3 defines the methods and material used in this study. Section 4 presents the results and discussion part of the paper. Section 5 is about the conclusion and future research.

2 Background and Related Work

The term agile was first used in software development during 2001, which had already been used in flexible manufacturing practices for a decade. A group of 17 software engineering consultants independently created change tolerant approaches which later came to know as agile software development methodologies. Agile methodologies emerged as an attempt to aid clients by embracing a higher rate of change in their expectations and software requirements where traditional development methodologies appeared unsuccessful [16]. The software engineering consultants and practitioners set up an Agile Alliance in 2001 and wrote an Agile Manifesto which revolves around to uncover better ways to develop software solutions. Furthermore, they had come to four values and twelve agile principles. The methodologies initially embraced by the agile alliance include Extreme Programming (XP), Feature Driven Development (FDD), Adaptive Software Development (ASD), Dynamic Systems Development Methods (DSDM), and Scrum [16,17]. During the last 25 years, many approaches to software development had introduced which categorized into two major classes (*i.e.*, *traditional vs. agile*) [1,17]. The proclamation of agile manifesto has brought exceptional changes to software engineering practices. The characteristics of agile methodologies showed that it has the potential to deal with the intricacies of software development and can produce better results in comparison to traditional methodologies [1,5,18]. It has been revealed that the software projects implemented using agile methodologies have a better success rate as compared to projects implemented through traditional software development methodologies. Agile methodologies outperformed during the implementation of projects of all sizes including small, medium and large relative to traditional methodologies, the failure rate was also lower correspondingly. Similarly, the challenging projects of the agile methodologies were also less in number than the projects implemented through traditional methodologies [18].

The research literature highlighted many reasons of challenging and failed software projects including lack of user involvement, incomplete requirements, changing requirements and specifications, unrealistic expectations, unclear objectives, lack of executive support, lack of planning, ineffective project management, technology incompetence, non-compliance to standards and quality issues [19,20]. These are the major issues that always hinder the progress of software projects. Among them, the most detrimental are problems related to software requirements which can seriously affect the project outcomes if not addressed properly. Agile methodologies emerged with the ability to rapidly respond to the dynamic nature of software development [16]. In essence, agility means reacting to changes quickly and efficiently. Likely changes required in software projects include budget, schedule, resources, software requirements, teams, and technology. These are the possible areas of changes on which agile methodologies concentrate by delivering the first increment in a couple of weeks and the whole product in the next few months [21]. In line with the agile principles articulated in the agile manifesto allow software developers—relying on technical excellence and simple design, causing business value by continuously delivering operational software at short intervals. Agile principles brought about a range of practices that are accounted for to deliver substantial value to clients. The core of such practices is the notion of self-organizing teams which are co-located and work at a pace that sustains their creativity and productivity. The principles encourage practices to serve requirement change at any stage of the software development life cycle. Moreover, users remain involved throughout the development process, provide

timely feedback and reflect their expectations that lead to more satisfying outcomes [6]. However, agile alliance members don't view methodologies as a silver bullet theory and stresses to uncover better ways by practicing and exploring these methodologies so that successful outcomes could be ensured [22]. A project considered successful if it is completed within budget, on time and meeting objectives, scope, performance requirement and obtaining user acceptance [23]. Critical success factors are those couple of elements, attributes or variables when appropriately addressed, have a greater impact on project work. So, if the CSFs are known in advance, the agile team may kick-start, plan and manage the project more effectively that would certainly support to close a project successfully [24]. A great deal of research has been done to find out the CSFs of agile software development. Boehm has confirmed that the key factors related to the success of agile software projects are people factors [25]. Livermore investigated the factors of agile software development using a survey study conducted with software development practitioners. The response rate of the survey was very low (i.e., 5.76%), however, results revealed few factors related to project management [26]. Chow & Cao conducted a survey study to find the critical success factors of agile software development involving software practitioners from different countries of America, Europe, Asia and Africa. This study collated success factors from agile literature and consolidated them through reliability and factor analysis into possible twelve factors and five categories including Organizational, People, Process, Technical and Project. The survey results showed that among twelve factors, only six factors were considered CSFs. Similarly, among five dimensions, technical and people were found as dimensions that greatly impact the success of agile software projects. Process and organizational dimensions also impact project success to some extent but the project dimension has shown no effect. The major limitation of this study is the small sample size so results are deficient in terms of statistical significance [27]. Hairul et al. conducted an extensive literature review of CSFs that seriously affect the success of software projects. They analyzed the research published during the year (1990–2010) using content analysis and frequency analysis to develop a list of CSFs. As a result, 26 CSFs were found relevant to the success of agile software projects [24]. Stankovic et al. presented the results of an empirical study conducted to determine CSFs that influence the agile software projects. The sample of the study consists of 23 software developers and project managers from IT companies located in Yugoslavia countries. This study challenged the factors identified by chow & cao in their research [9]. Another survey-based study has been conducted to investigate the factors affecting the agile methodologies to produce software in the scenario of the Indian industry [15]. IBM has also reported certain CSFs to adopt the agile software development methodologies in true spirit so that agile software could be produced successfully [28]. Silva & Santos conducted a systematic literature review to find the CSFs of agile software development. They identified fourteen key factors and categorized them into three categories including people, process and technology [29]. Kelle et al. examined social factors that could influence the success of software development projects particularly agile software projects [30]. Darwish & Rizk extensively reviewed agile literature to give the multi-dimensional view of success factors in agile software development projects. To ease the understanding, the success factors grouped into multiple categories including organizational, people, process, project and technical [31]. A qualitative research study was carried out to determine the CSFs of agile projects, examining earlier research. The CSFs were further categorized into dimensions and mapped to agile principles using the mind map technique [32].

Nguyen conducted an online survey with ten project managers of US-based companies to identify the CSFs of agile software projects. The results stressed the need for customer involvement and management support for a successful outcome in agile environment [33]. Pourhanifeh & Mazdeh conducted a case study considering an Iranian software company to find and prioritize CSFs for agile software projects. They identify and prioritize five factors among other factors using Analytical Hierarchical Process (AHP) [34]. Aldahmash et al. explored the research published during the last ten years (2006–2016) in order to identify the CSFs of agile software development methodologies [14]. Chiyangwa & Mnkandla identified

and have given insight of CSFs that may impact the success of software development projects using agile methodologies in South Africa. The data was collected through a web-based survey using a structured questionnaire. Results revealed that the organizational dimension significantly impacts the success of agile software projects. This study discovered a comprehensive model to provide guidelines for agile professionals [35]. Islam conducted a systematic literature review (SLR) to identify CSFs their related attributes in agile software projects. The findings were further analyzed using frequency analysis to see the contribution of each factor towards success. Moreover, the study revealed the technical dimension as the most significant in terms of successful agile software development [36]. Liu identify the CSFs at the agile planning phase along with the challenges associated with each factor. Two methods including systematic mapping and survey used to carry out the research objectives [37]. Alqudah & Razali performed a systematic literature review to identify the key factors that should be considered to successfully adopt the agile software development methodologies [38]. Jovanovic et al. reviewed the literature to identify the success factors of agile software projects [39]. Kalenda et al. highlighted success factors through a focused literature review which further guided action research in a software company with the aim of scalability of agile methodologies [40].

Tab. 1 present various CSFs of agile software projects related to different dimensions, identified during various research studies conducted before.

Table 1: Different factors of agile software development

Dimensions	Factors	Literature references
Organizational	Management support, Team distribution, Facility with agile-style work environment, Leadership, Risk management, Organization maturity, Co-located teams, Culture	[24,26,27,31,33–35,37–40]
People	Capable and motivated team members, Managers having knowledge of Agile process, Agile experience, Good client relationship, Customer involvement, Training and Education, staffing culture, values, communication, amicability	[25,26,27,31,37,40]
Process	Agile-oriented requirement management, Strong communication, Regular working schedule, Strong customer commitment, Project scope is well-defined, Good customer relationship, Regular delivery of software	[9,11,24,26,27,31,33,37]
Project	Budget, schedule, Projects with small team, Project complexity, Project nature	[9,11,27,32,37,38,39]
Team	Project team skills, Project team commitment, Internal project communication, Participation of project team, Project team commitment	[3,26,38]
Customer	Knowledgeable user, User involvement, User support education, Client experience, Customer trainings	[9,25,31,38,39]

Although the CSFs of agile software projects have drawn a lot of attention in the literature, still existing research has some deficiencies. For example, majority of the studies conducted through survey method comprised of small sample size as well as the chosen participants were well versed only in few agile methodologies. Therefore, based on identified factors, the connection with project success could not be developed. Some other studies systematically reviewed the literature to identify a set of CSFs. The

sample of the survey should involve diverse subjects (i.e., both from academia and industry) having experience and knowledge of agile methodologies to identify the factors with real value. Though many factors were highlighted by earlier research, still there is a need to determine the significance of each factor in relation to other identified factors so that the factors could be prioritized in terms of their influence towards the success of agile software development. Furthermore, it is widely recognized that project success is a multidimensional idea with interrelated technical, economic, behavioral, psychological and political aspects [41]. Therefore, we think that the Multi-Criteria Decision Analysis (MCDA) techniques [42] to be able to show more relevant results. One of the MCDA techniques is Analytic Hierarchy Process (AHP) which is a robust technique to identify and prioritize CSFs of software projects by making pairwise comparison [43]. As per authors' knowledge, the relative significance of CSFs towards agile software development has not been investigated so far considering both qualitative and quantitative data. This research aimed at the identification and prioritization of the most critical factors of the agile software development through a rigorous methodology that encompasses different layers, methods and techniques involving a variety of agile experts at each stage.

3 Methods and Materials

This study has used a mixed-method approach to answer research questions posed in this study. The research initiated by extensively reviewing the recent literature to find the key factors which influence the agile software development. The identified factors were further validated through agile professionals using three rounds of the Delphi method. Delphi method allows collecting opinions about an issue from a wider population. Delphi method has been used as a way to structure the process for group communication to enable a group of experts to deal with complex problems [44]. In this study, Delphi method was used because of these reasons. (i) This study is about the investigation of the factors that could make agile software projects successful. The complex issue needs knowledge from experts who understand the issue technically, socially, managerially and organizationally. Delphi method helps to collect required knowledge. (ii) A group of experts may appropriately respond to the research questions instead of individual responses. Delphi is a group method. (iii) The design of the Delphi method is flexible, follow-up interviews may have conducted to deeply understand the issue. The steps followed to collect and analyze data during the Delphi process includes [45] (i) *Select a Facilitator* (ii) *Identify the Experts* (iii) *Portray the problem* (iv) *Round One Questions* (v) *Round Two Questions* (vi) *Round Three Questions* (vii) *Results & Findings*.

Analytical Hierarchy Process (AHP) technique has combined with the Delphi method to prioritize the CSFs of agile software development. AHP is one of the techniques of Multi Criteria Decision Analysis (MCDA) which based on the hierarchical structure. AHP is a robust and useful technique for decision making which support decision-makers to reach an appropriate decision that best suits to their goals and understanding of the problem [46]. A graphical illustration of the research methodology is shown in Fig. 1.

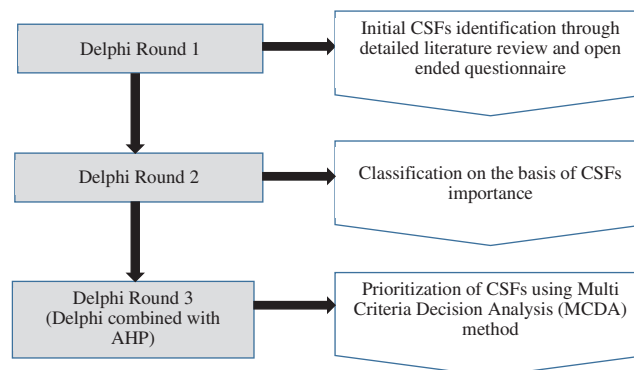


Figure 1: Research methodology

3.1 Round 1: Identifying Critical Success Factors

To identify the critical factors influencing the success of agile projects, the Delphi method used along with the support of the literature review. During round 1, first, a detailed review of literature has been taken through various sources such as Google Scholar (<https://scholar.google.com.pk/>), Science Direct (<https://www.sciencedirect.com/>) and IEEE Xplore (<https://ieeexplore.ieee.org>). After reviewing the literature and extracting different factors, a survey conducted to validate the findings of the literature. The survey participants consisting of 144 qualified agile experts who deeply understand the issue (*i.e., agile software development*) and have an important and valuable knowledge in this regard. The survey participants include academicians (*i.e., university professors having expertise in agile software development methodologies*), postgraduate students (*i.e., engaged in research activities in agile software development methodologies*) and agile software engineers linked to the local software industry. Of the 144 respondents, 24.31% (35) of the respondents were professors, 40.27% (58) of the respondents were software engineers and 35.42% (51) of the respondents were research scholars.

This research applies a snowball sampling method which means the participants selected through personal contacts. The authors identified academicians with expertise in agile software development methodologies through faculty portals and using their personal contacts. Afterward, they were personally visited to identify further pertinent faculty in their contact as well as students working under their supervision on research projects related to agile software development methodologies. Some local software development organizations also visited and focal persons were contacted to identify the software engineers involved in agile software development. The identified agile practitioners mentioned the further developers who have been implementing software projects using agile methodologies. During visits to academic institutes and local industry, a paper-based survey was completed and a link of a google form was shared with rest of the subjects to collect data efficiently.

The questionnaire was consisting of open-ended questions formulated based on the factors extracted from literature. Such questions prompt participants to highlight the factors which lead to agile software development towards successful projects.

3.2 Round 2: Likert Scale Survey

Based on the qualitative data generated as a result of the round 1 questionnaire, a Likert scale questionnaire was developed during round 2 to determine the important factors and to classify them. Before being sent out to the respondents, the content validity of the questionnaire was analyzed and a pilot test with five experts was also done. Afterward, the instrument was distributed among those participants of the original sample who have expertise in agile software development methodologies (*e.g., including software engineers with minimum three years' development experience and university professors and postgraduate researchers with at least three years R&D experience*). They were asked to highlight the contribution of each factor toward successful agile software development through a five-point Likert scale.

3.3 Round 3: Prioritization of CSFs Using AHP

In this round, the AHP technique was combined with the Delphi method to overcome the likely biases in expert responses. The final-round Delphi survey aimed to set the priorities and to give weights to the factors identified during the previous round. To prioritize the factors influencing the success of agile software development, the results of the second round were arranged in the format of AHP pair-wise comparisons. The pairwise comparison method was adopted to estimate the weights of CSFs. In general, the pairwise comparison is a process of comparing items in pairs to estimate which of each entity has a greater importance, or whether or not the two items are similar. The pairwise comparison approach was implemented to assign relative weights to CSFs as explained below [47].

Step 1: The pairwise comparison matrix was completed. The importance of two factors were evaluated using index value from 1 to 9. For example, if factor F_i was equally important as factor F_j , this pair received an index value of 1. If F_i was enormously important than F_j the index was 9, possible gradations are shown in [Tab. 3](#). For a “less important” relationship, the fractions 1/1 to 1/9 were available: if F_i was extremely less important than F_j , the rating was 1/9. The index values were calculated on the basis of opinion of agile experts, to achieve this, a questionnaire was designed and distributed among 40 participants (i.e., senior experts having more than five years’ experience either in agile software development or research experience with agile methodologies). The experts were asked to evaluates the importance of each key factor in relation to other key factors and record their judgements. If the judgements of various experts were varying, a consensus approach was used to reduce the variation. The final approved judgements were recorded row by row into a cross-matrix C ($n \times n$). First, the diagonal of C was completed by values of 1, in accordance to “[Eq. \(1\)](#)”. Second, the right upper half of C was completed until each factor had been compared to every other one. If F_i to F_j was rated with the relative importance of m (i.e., $C_{ij} = m$), F_j to F_i had to be rated with $1/m$ (i.e., $C_{ji} = 1/m$). Finally, the lower left half of C was completed with the consistent fraction, in accordance to “[Eq. \(2\)](#)”. (Note that i and j are positive integers $\leq n$, C_{ij} is the element of C located in row i and column j)

$$C_{ij} = 1, i = j \quad (1)$$

$$C_{ij} = \frac{1}{C_{ji}}, i \neq j \quad (2)$$

Step 2: Calculate the normalized comparison matrix: A normalized comparison matrix C' was generated by dividing each element in matrix C by the sum of the elements in its column. This is depicted in “[Eq. \(3\)](#)”.

$$C'_{ij} = C_{ij} / \sum_{i=1}^n C_{ij} \quad (3)$$

Step3: Calculate of the relative weights of the factors: To obtain the weight w_i of each singular factor F_i , the mean of each row in C' was calculated as illustrated by “[Eq. \(4\)](#)”.

$$w_i = \frac{1}{n} \sum_{j=1}^n C'_{ij} \quad (4)$$

Such weights were already normalized; their sum was 1, as shown in “[Eqs. \(5\) and \(6\)](#)”.

$$0 \leq w_i \leq 1 \quad (5)$$

$$\sum_{i=1}^n w_i = 1 \quad (6)$$

Step 4: Check consistency of the results of the pairwise comparisons: According to Saaty [48], the consistency ratio is acceptable if it is less than 10% otherwise pairwise comparisons need to be revised [49]. The ratio of consistency (CR) is shown by “[Eq. \(7\)](#)”.

$$CR = \frac{CI}{RI} \quad (7)$$

where CI is consistency index that is depicted by “[Eq. \(8\)](#)”.

$$CI = \frac{\lambda_{\max} - n}{n - 1} \quad (8)$$

λ_{\max} indicates the maximum eigenvalue and n shown the rank of pairwise comparison matrix [49].

RI states the random index of consistency, which has various values centered on different criteria, as shown in [Tab. 2](#).

Table 2: Relationship between (RI and n)

N	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

Table 3: AHP scale (Index values in pairwise comparison matrix)

Definition	Index	Definition	Index
Equally important	1	Equally important	1/1
Moderately more important	3	Moderately less important	1/3
Much more important	5	Much less important	1/5
Far more important	7	Far less important	1/7
Extremely more important	9	Extremely less important	1/9

4 Results and Discussion

To achieve the objective of the research, three rounds were conducted involving agile experts from academia and the local software industry of Pakistan. This section presents the findings for each round. The findings mainly consist of CSFs of agile software development investigated through published observations of various researchers and deep knowledge and experience of agile experts.

4.1 Round 1: Extracted Critical Success Factors

In the first round, relevant literature was reviewed and a survey was conducted to identify the key factors. The questionnaires both in the paper-pencil form and google form distributed among 144 subjects of the survey. Out of 144, there were only 52 participants who furnished with appropriate responses. The rest of the questionnaires were either incomplete or participants found irrelevant to the study. The responses of the survey were examined and organized keeping in view likeness among them. As a result, 17 important factors extracted from the research literature and data gathered by the agile community. The description of the extracted critical success factors is as follows:

a. Communication: Communication is often perceived as an important dimension of interaction between the user and the software development team [41]. It has been demonstrated that communication is one of the main factors that influence the success of agile methodologies [38].

b. Management Support: Management support plays a crucial role in successful software development. The term management refers to the group of executives and decision-makers with responsibility for the overall strategic direction of the organization [41].

c. Customer Involvement: Customer involvement is defined as active involvement and collaborative interaction of customers with developers on regular basis [50].

d. Agile Software Techniques: Agile software techniques refer to a group of agile techniques that assist agile teams to deal with the complexities and unpredictability of software development [27].

e. Training & Education: Training & Education refers to an act of inculcating theoretical knowledge of agile methodologies and specific skills in the agile team so that they could effectively perform on agile projects [38].

f. Risk Management: Risk management is a process to evaluate the probability of loss or harm to a project and then take measures to deal with potential risks [41].

g. Leadership: Leadership is the position to direct an agile team, or the ability to lead a team [41].

h. Project Management Process: Project Management Process refer to processes and strategies which support to achieve the project objectives and deliver the project product on time [27].

i. Quality: According to ISO 9000 quality is defined as “the totality of characteristics of an entity that bear on its ability to satisfy stated or implied needs” [51].

j. Team Capability: Team capability refers to a set of roles performed by the agile team to deliver technical or business outcomes of agile software development [38].

k. Culture: Culture refers to shared assumptions, values and beliefs which define behavior towards software development, deal with customer and work as a team [41].

l. Productivity: It measures how efficiently software is being produced [1].

m. Delivery Strategy: Delivery strategy indicates how to deliver appropriate deliverables to the customer on regular basis [12].

n. Requirement Specification: Software requirement specification describes the software system to be developed from its functional and non-functional aspects [1].

o. Infrastructure: Refers to the technical infrastructure which is required to execute the project [41].

p. Performance: Performance indicates how well agile software methodologies meet project timeliness requirements [1].

q. Partnership: Partnership indicates the good working relationship between the customer and the agile team [50].

4.2 Round 2: Importance Level of Factors and Dimensions

During second round, questionnaires were distributed among 52 participants but only 29 responses were found appropriate. During this round, experts were asked to provide their viewpoint regarding the importance of each factor towards the success of agile software development. The responses were recorded and statistically analysed through the Statistical Package for Social Sciences (SPSS) [52]. Tab. 4 present the factors identified during this round along with their frequency and corresponding percentage. Frequency analysis is a descriptive statistical method that indicates how many times a particular response has chosen by the respondents of the survey [53]. It is clearly shown in Tab. 4, the factors which have chosen at most by respondents. The more frequently chosen factors include communication and management support whereas less frequently chosen were performance and partnership. Out of the above discussed 17 factors, 16 factors considered in further analysis, ‘*project management process*’ was discarded because more relevant factors related to project management were already included in the list. The mean and standard deviation of each factor was calculated as shown in Tab. 5. Tab. 5 shows the identified factors, categorized into three different categories related to software development including organization, people and technical. In “*organization*” dimension the ‘*risk management*’ found as the most significant parameter with SD .693 and ‘*partnership*’ found as the least significant parameter with 0.508 SD. In “*people*” dimension ‘*customer involvement*’, with .669 SD, and in “*technical*” dimension ‘*agile software techniques*’, with 1.05 SD, found the most important parameters. On the other hand, ‘*requirement*

specification’, with .687 SD, and *performance*’, with 0.574 SD, found the least important parameters of the “*people*” dimension and “*technical*” dimension.

Table 4: Factors of agile software development

S#	Factors	Frequency	Percentage
1	Communication	29	55.7
2	Management support	28	53.8
3	Customer involvement	22	42.3
4	Agile software techniques	22	42.3
5	Training and education	21	40.3
6	Risk management	20	38.4
7	Leadership	19	36.5
8	Project management process	19	36.5
9	Quality	17	32.6
10	Team capability	16	30.7
11	Culture	16	30.7
12	Productivity	14	29.6
13	Delivery strategy	13	25
14	Requirement specification	8	15.3
15	Infrastructure	5	9.61
16	Performance	4	7.69
17	Partnership	3	5.76

Table 5: Analysis of data

Dimensions	Factors	Importance level					Mean	St. Deviation
		Very important	Important	Neutral	Less important	Not important		
Organization	Risk Management	N 0	0	5	15	9	4.13	0.693
		% 0	0	17.2	51.7	31.03		
	Management Support	N 0	0	2	8	18	4.51	0.687
		% 0	0	6.89	27.5	62.09		
	Communication	N 0	0	0	15	14	4.48	0.508
		% 0	0	0	51.7	48.2		
	Culture	N 0	1	6	7	15	4.24	0.912
		% 0	3.44	20.6	24.1	51.7		
	Infrastructure	N 2	19	8	0	0	2.20	0.559
		% 6.89	65.5	27.5	0	0		
	Partnership	N 15	14	0	0	0	1.48	0.508
		% 51.7	48.2	0	0	0		

(Continued)

Table 5 (continued).									
Dimensions	Factors	Importance level					Mean	St. Deviation	
		Very important	Important	Neutral	Less important	Not important			
People	Customer involvement	N	0	0	3	13	13	4.34	0.669
		%	0	0	10.3	44.8	44.8		
	Leadership	N	0	0	0	14	15	4.51	0.508
		%	0	0	0	48.2	51.7		
	Team capability	N	0	1	4	11	13	4.24	0.803
		%	0	3.44	13.7	37.9	44.8		
Training & education	N	0	0	1	12	16	4.51	0.574	
	%	0	0	3.44	41.3	55.1			
Requirement specification	N	0	2	12	14	1	3.48	0.687	
	%	0	6.89	41.3	48.2	3.44			
Technical	Agile software technique	N	0	3	5	6	15	4.13	1.05
		%	0	10.3	17.2	20.6	51.7		
	Delivery strategy	N	0	0	6	12	11	4.17	0.75
		%	0	0	20.6	41.3	37.9		
	Productivity	N	0	0	1	14	14	4.44	0.572
		%	0	0	3.44	48.2	48.2		
	Quality	N	0	0	0	19	10	3.51	0.949
		%	0	0	0	65.5	34.4		
	Performance	N	0	16	12	1	0	2.48	0.574
		%	0	55.1	41.3	3.44	0		

4.3 Round 3: AHP Result Analysis

The basic objective of the third round was to prioritize the factors based on their contribution to successful agile software development. In this round, only those factors were considered which have a mean value greater than or equal to 3.50 (*i.e.*, *threshold value /cutoff that is selected on the basis of experts'/researchers' choice*). Therefore, four factors were eliminated and twelve factors were selected for further analysis. The AHP is being used by researchers in different fields of study in order to make decisions. To perform the decision-making process, AHP takes several steps that are incorporated by Expert Choice software [54]. This section describes the implementation of AHP through Expert Choice software and produced the model for successful agile software development.

The AHP steps along with the related results are discussed below.

4.3.1 Define Goal

The goal of this study was the identification of CSFs of agile software development.

4.3.2 Identifying Criteria and Sub-Criteria

The subsequent step is taken to identify the major dimensions (*i.e.*, people, technical, organization) and factors related to each dimension. Such dimensions and factors were elicited from agile experts and defined into expert choice software.

4.3.3 Hierarchal Model for Analysis

A model representing hierarchy of three dimensions and twelve factors of agile software development was designed during the implementation of AHP. The nodes of hierarchical model (dimensions and factors) were pairwise compared against each other to make comparisons in order to set priorities using formula $n(n-1)/2$ where ‘n’ represents the number of dimensions or factors [48].

4.3.4 Data Gathering

In this step, data was collected using AHP questionnaire, designed based on previously identified factors. The questionnaires distributed among 40 participants belongs to original sample but include only those who have at least five years’ experience either in agile software development or in R&D activities. Amongst the 40 responses, only 20 responses meet the criteria which considered to do further analysis.

4.3.5 Pair Wise Comparison

At each node of the hierarchical model, a matrix represents the comparisons of different dimensions and factors made by participants. The responses of participants were measured using a geometric mean approach and relative importance of dimensions.

4.3.6 Consistency Ratio

The consistency test performed using Expert Choice software to meet Saaty criteria [48]. According to his criteria, only the responses with less than 10% consistency ratio are acceptable [49]. Therefore, out of forty, twenty responses with greater than 10% consistency ratio were excluded from the study.

4.3.7 Global and Local Weights

The global and local weights of the dimensions and related factors were calculated using Expert choice software. Global weights are values of major dimensions whereas local weights present values related to factors fall within each dimension. Using expert choice, local priorities of the factors were calculated and subsequently global weights were generated to produce the final outcome of this study. The local weights of factors related to people dimension are shown in [Tab. 6](#).

Table 6: Local weights and ranking of people factors

Factors	Weights (Local)	Rank (Local)
Team capability	0.364	1
Training and Education	0.266	2
Leadership	0.256	3
Customer involvement	0.113	4

The local weights of factors related to the technical dimension shown in [Tab. 7](#).

The local weights of factors related to the organization dimension shown in [Tab. 8](#).

[Tab. 9](#) shows the weights of globally prioritized dimensions of agile software development.

The global weights and global ranking of main dimensions are shown in [Tab. 9](#). The CSFs of the “people” dimension ranked at top owing to its higher weight (i.e., 0.415) where the “organization” dimension placed at lower position due to its lesser weight (i.e., 0.284).

Table 7: Local weights and ranking of technical factors

Factors	Weights (Local)	Rank (Local)
Quality	0.390	1
Productivity	0.264	2
Agile software technique	0.178	3
Delivery strategy	0.167	4

Table 8: Local weights and ranking of organization factors

Factors	Weights (Local)	Rank (Local)
Communication	0.482	1
Risk management	0.266	2
Management support	0.168	3
Culture	0.084	4

Table 9: Global weights and ranking of dimensions

Factors	Weights (Local)	Rank (Local)
People	0.415	1
Technical	0.301	2
Organization	0.284	3
People	0.415	1

h. *Final Ranking and Overall Weight:* **Tab. 10** illustrates the weight and contribution level of the main dimensions and related factors. The “people” dimension was found as the most impactful dimension with 41.5 percent contribution value towards the success of a project. The major contributing factor of people dimension was ‘team capability’ with 36.4 percent contribution value. The results show that the agile software development team should highly competent, technically sound and greatly motivated to get the job done. The managers should knowledgeable of agile methodologies and possess leadership qualities to effectively coordinate with the entire development team as well as develop a good working relationship with the customer.

These findings are similar to the results of previous research [9,27,38,41]. However, in addition to team capability and customer involvement, this study emphasizes training and leadership needs to keep the agile team knowledgeable and motivated which resultantly complement the people dimension. Moreover, the primary value of agile manifesto underlines that individuals are much important than processes and tools [2].

The “technical” dimension was found as the second most important dimension with a 30.1 percent contribution value. It indicates that agile software development should put into execution using the most effective agile methodologies. The agile methodology may be selected as a single, pair of two or a combination of more than one agile methodologies to achieve maximum software quality and productivity. Similarly, the most effective agile techniques related to requirement development, coding and testing should take into account [55]. Furthermore, an effective strategy should be devised to deliver

software at regular intervals so that customer satisfaction could be achieved. The most contributing factor of technical dimension was ‘*quality*’ with 39 percent contribution value. The parameter ‘*quality*’ has paramount importance within the literature of software engineering. It has crucial consideration in the implementation of any software project as well as considered to determine the success and failure of software projects [56]. The result implies that during agile software development, the agile team should select those practices of agile methodologies that focus on delivering high-quality software products. For example, a sprint review meeting is the main practice of SCRUM which leads to quality outcomes. Similarly, pair programming, refactoring and simple design are the practices of XP which also focus on producing good quality products [38].

Table 10: Overall weight and ranking

Dimensions	Global weight with contribution%	Global ranking of dimensions	Factors	Local weight with contribution%	Local ranking of factors global ranking of factors
People	0.415(41.5%)	1	Team capability	0.364 (36.4%)	1 1
			Training and education	0.266 (26.6%)	2 2
			Leadership	0.256 (25.6%)	3 4
			Customer involvement	0.113 (11.3%)	4 9
Technical	0.301(30.1%)	2	Quality	0.390 (39%)	1 3
			Productivity	0.264 (26.4%)	2 6
			Agile software technique	0.178 (17.8%)	3 8
			Delivery strategy	0.167 (16.7%)	4 10
Organization	0.284 (28.4%)	3	Communication	0.482 (48.2%)	1 5
			Risk management	0.266 (26.6%)	2 7
			Management support	0.168 (16.8%)	3 11
			Culture	0.084 (8.4%)	4 12

The third important dimension was “*Organization*” with 28.4 percent contribution value. The findings indicate that a sound communication mechanism is indispensable for the effective functioning of an agile team and it plays a pivotal role in the success of software projects. There are certain factors of agile software development which make it challenging such as agile methodologies accelerate the software development, allow changes at any stage and depends a lot on the credibility of people involved in the projects. Therefore, a proactive risk management approach should be an integral part of each iteration of the agile software development cycle [57]. Furthermore, the allocation of necessary resources to projects requires management support which also helps to control the organizational obstacles [14,41]. Culture presents the values, beliefs and patterns of behavior within an organization also needs the support of management [36]. ‘*Communication*’ has shown as the most important factor of the “*organization*” dimension with 48.2 percent contribution value. It is a key dimension of interaction between the user and the software development team. It has widely established that effective communication channels can overcome misunderstandings, disagreements and conflicts that occur between different project

stakeholders [14,41]. The result is consistent with Project Management Body of Knowledge (PMBok) which stressed the need for communication for the successful execution of a project. Additionally, software development is a social process that extensively requires communication to understand the organizational context, problems, stakeholder requirements and team management [58].

As an outcome of this study, a hierarchical model shown in Fig. 2 has proposed to guide agile developers particularly those who are working in the local software industry of Pakistan. The proposed model represents the CSFs, ranked according to their significance towards the success of agile software development. The model can help practitioners with respect to the criteria which should take into consideration while implementing agile software projects.

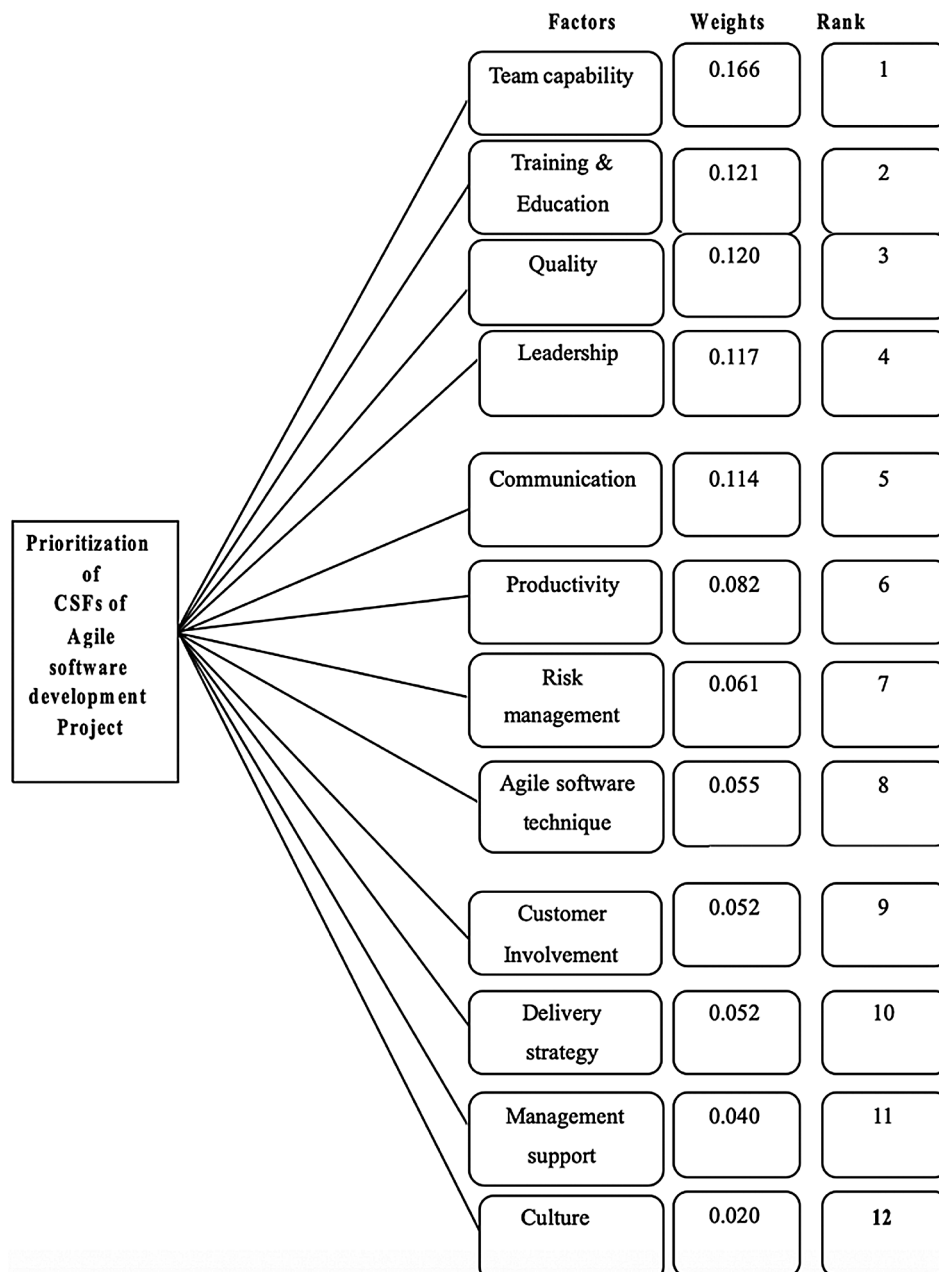


Figure 2: Proposed model for agile software development

We believe the proposed model would be useful for the implementation of agile projects because this model based on such factors which substantially supported by research literature and agile experts involved either in agile software development or research activities related to agile methodologies.

5 Conclusion and Future Work

The last many years has produced a very great extent of research concentrated on identifying the factors that impact agile software development. However, the research carried out previously has accomplished either through systematic literature review or quantitative methods. This study is conducted combining traditional and judgment methods in order to create a better model. The criteria to identify agile CSFs was the review of corresponding research literature which was further validated by experts through different rounds. Finally, this process created a model that represents three dimensions and twelve key factors prioritized with respect to their influence on the success of agile software development. Three factors of each dimension including '*team capability*', '*communication*', and '*customer involvement*' are found as important factors in relation to other factors of the corresponding domain. Moreover, among the twelve success factors, the first and foremost factor is '*team capability*' whereas the relatively least significant factor is the culture of development organization. Though '*culture*' is relatively ranked at low in the list of identified CSFs but it is not the least factor. It has its own unique importance because to be an agile company is a cultural thing. If the culture is not supporting, then the organization cannot be the right candidate for agile software development. It is therefore suggested to take it also into consideration while operating as an agile driven software development organization. The results of the study imply the structuring of an agile environment keeping in view identified CSFs. Furthermore, the CSFs have structured into the model in such a way that it may easily be adapted to the project situation. For example, the CSFs classification into different dimensions, each factor importance with respect to corresponding dimension as well as the overall importance of CSFs toward the successful outcome also allows agile practitioners to tailor model according to the size, nature and budgetary constraints of a project. We think the findings revealed in this research would be useful in implementing agile software because the subjects involved in discovering the CSFs were well versed in agile methodologies and experienced in agile software development being carried out in the local software industry. Future research will concentrate on applying the proposed model to multiple agile software projects of different sizes (i.e., small, medium and large) to show further evidence. Moreover, the research will also focus on the investigation of CSFs of agile software development in the global software development setting as well as explore the key factors of agile beyond the pure software development such as embedded applications and mechatronics.

Acknowledgement: Authors thankfully acknowledge all those who contributed this work scientifically, administratively, financially and academically during the whole research process.

Funding Statement: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through research Grant No. (RG- P2/85/41).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding this study.

References

- [1] M. A. Awad, "*A comparison between agile and traditional software development methodologies*," (Hons. Thesis). School of Computer Science and Software Engineering, The University of Western Australia (UWA), Australia, 2005.
- [2] D. Zima, "Modern methods of software development," *TASK Quarterly*, vol. 19, no. 4, pp. 481–493, 2015.
- [3] P. Serrador and J. K. Pinto, "Does Agile work ? — A quantitative analysis of agile project success," *International Journal of Project Management*, vol. 33, no. 5, pp. 1–12, 2015.

- [4] Standish Group, Chaos Report. Standish Group 2015 Chaos Report–Q&A with Jennifer Lynch. [Online]. Available: <https://www.infoq.com/articles/standish-chaos-2015/>.
- [5] G. Papadopoulos, “Moving from traditional to agile software development methodologies also on large, distributed projects,” *Procedia–Social Behavioral Sciences*, vol. 175, no. 2, pp. 455–463, 2015.
- [6] T. Dingsoyr, S. Nerur, V. Balijepally and N. B. Moe, “A decade of agile methodologies: Towards explaining agile software development,” *Journal of Systems and Software*, vol. 85, no. 6, pp. 1213–1221, 2012.
- [7] J. Nuottila, K. Aaltonen and J. Kujala, “Challenges of adopting agile methods in a public organization,” *International Journal of Information Systems and Project Management*, vol. 4, no. 3, pp. 65–85, 2016.
- [8] H. Hajjdiab and A. S. Taleb, “Adopting agile software development: Issues and challenges,” *International Journal of Managing Value and Supply Chains*, vol. 2, no. 3, pp. 1–10, 2011.
- [9] D. Stankovic, V. Nikolic, M. Djordjevic and D. B. Cao, “A survey study of critical success factors in agile software projects in former Yugoslavia IT companies,” *Journal of Systems and Software*, vol. 86, no. 6, pp. 1663–1678, 2013.
- [10] U. Eklund and C. Berger, “Scaling agile development in mechatronic organizations-a comparative case study,” in *Int. Conf. on Software Engineering–Software Engineering Practice Track, ICSE–SEIP*, Buenos Aires, Argentina, pp. 173–182, 2017.
- [11] B. Tawanda and E. Mnkandla, “Modelling the critical success factors of agile software development projects in South Africa,” *South African Journal of Information Management*, vol. 19, no. 1, pp. 8, 2017.
- [12] S. C. Misra, V. Kumar and U. Kumar, “Identifying some important success factors in adopting agile software development practices,” *Journal of Systems and Software*, vol. 82, no. 11, pp. 1869–1890, 2009.
- [13] D. Sato, D. Bassi, M. Bravo, A. Goldman and F. Kon, “Experiences tracking agile projects: An empirical study,” *Journal of the Brazilian Computer Society*, vol. 12, no. 3, pp. 45–64, 2006.
- [14] A. Aldahmash, A. M. Gravell and Y. Howard, “A review on the critical success factors of agile software development,” in *European Conf. on Software Process Improvement, EuroSPI*, Ostrava, Czech Republic, pp. 504–512, 2017.
- [15] D. Shahane, P. Jamsandekar and D. Shahane, “Factors influencing the agile methods in practice-literature survey & review,” in *Int. Conf. on Computing for Sustainable Global Development, INDIACom*, USA, pp. 556–560, 2014.
- [16] L. Williams, “Agile software development methodologies and practices,” In: M. V. Zelkowitz (ed.), *Advances in Computers*, Amsterdam: Elsevier, vol. 80, pp. 1–44, 2010.
- [17] P. Abrahamsson, O. Salo, J. Ronkainen and J. Warsta, “Agile software development methods: Review and analysis,” arXiv Prepr. arXiv1709.08439, 2017.
- [18] A. P. Veiga, “Project success in Agile development projects,” arXiv Prepr. arXiv1711.06851, 2017.
- [19] J. Verner, J. Sampson and N. Cerpa, “What factors lead to software project failure?,” in *2nd Int. Conf. on Research Challenges in Information Science RCIS*, Marrakech, Morocco, pp. 71–80, 2008.
- [20] Y. K. Dwivedi, K. Ravichandran, M. D. William, S. Miller, B. Lal *et al.*, “IS/IT project failures: A review of the extant literature for deriving a taxonomy of failure factors,” in *Int. Working Conf. on Transfer and Diffusion of IT*, Bangalore, India, pp. 73–88, 2013.
- [21] M. R. J. Qureshi, “Agile software development methodology for medium and large projects,” *IET Software*, vol. 6, no. 4, pp. 358–363, 2012.
- [22] K. Beck, M. Beedle, A. V. Bennekum, A. Cockburn, W. Cunningham *et al.*, “Manifesto for agile software development,” 2001. [Online]. Available: <http://AgileManifesto.org>.
- [23] R. Berntsson-Svensson and A. Aurum, “Successful software project and products: An empirical investigation,” in *International Symposium on Empirical Software Engineering ISESE*, Rio de Janeiro, Brazil, pp. 144–153, 2006.
- [24] M. Hairul, N. Nasir and S. Sahibuddin, “Critical success factors for software projects: A comparative study,” *Scientific Research and Essays*, vol. 6, no. 10, pp. 2174–2186, 2011.
- [25] B. Boehm and R. Turner, “People factors in software management: Lessons from comparing agile and plan-driven methods,” *Crosstalk–The Journal of Defense Software Engineering*, vol. 10, no. 12, pp. 4–8, 2003.

- [26] J. A. Livermore, "Factors that significantly impact the implementation of an agile software development methodology," *Journal of Software*, vol. 3, no. 4, pp. 31–36, 2008.
- [27] T. Chow and D. B. Cao, "A survey study of critical success factors in agile software projects," *Journal of Systems and Software*, vol. 81, no. 6, pp. 961–971, 2008.
- [28] IBM Centre for the Business of Government, A Guide to Critical Success Factors in Agile Delivery. 2014. [Online]. Available: <http://www.businessofgovernment.org/sites/default/files/A%20Guide%20to%20Critical%20Success%20Factors%20in%20Agile%20Delivery.pdf>.
- [29] B. G. Tavares, C. E. S. da Silva and A. D. de Souza, "Analysis of scrum practices for risk treatment," *Product: Management and Development*, vol. 14, no. 1, pp. 38–46, 2016.
- [30] E. van Kelle, J. Visser, A. Plaat and P. van der Wijst, "An empirical study into social success factors for agile software development," in *Int. Workshop on Cooperative and Human Aspects of Software Engineering, CHASE*. New Washington DC, USA, 77–80, 2015.
- [31] N. R. Darwish and N. M. Rizk, "Multi-dimensional success factors of agile software development projects," *International Journal of Computer Application*, vol. 118, no. 15, pp. 23–30, 2015.
- [32] T. A. El Hameed, M. A. E. L. Latif and S. Kholief, "Identify and classify critical success factor of agile software development methodology using mind map," *International Journal of Advance Computer Science Applications*, vol. 7, no. 5, pp. 83–92, 2016.
- [33] D. S. Nguyen, "Success factors that influence agile software development project success," *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, vol. 17, no. 1, pp. 171–222, 2016.
- [34] G. H. Pourhanifeh and M. M. Mazdeh, "Identifying the critical success factors of organization with analytic hierarchy process approach (case study–Iran Argham company)," *Problems and Perspectives in Management*, vol. 14, no. 4, pp. 54–60, 2016.
- [35] T. B. Chiyangwa and E. Mnkandla, "Modelling the critical success factors of agile software development projects in South Africa," *South African Journal of Information Management*, vol. 19, no. 1, pp. 1–8, 2017.
- [36] M. Islam, "A systematic literature review on the critical factors that contribute to success of agile development projects," (M. S. Thesis). Department of Computer Science and Engineering, Chalmers University of Technology, Sweden, 2016.
- [37] D. Liu and Z. Zhai, "An empirical study of Agile planning critical success factors," (M.S. Thesis). Department of Software Engineering, Bleking Institute of Technology, Sweden, 2017.
- [38] M. K. Alqudah and R. Razali, "Key factors for selecting an Agile method: A systematic literature review," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 7, no. 2, pp. 526–537, 2017.
- [39] M. Jovanović, A. L. Mesquida, A. Mas and B. Lalić, "Towards the development of a sequential framework for agile adoption," in *Int. Conf. on Software Process Improvement and Capability Determination, SPICE*, Palma de Mallorca, Spain, pp. 30–42, 2017.
- [40] M. Kalenda, P. Hyna and B. Rossi, "Scaling agile in large organizations: Practices, challenges, and success factors," *Journal of Software, Evolution and Process*, vol. 30, no. 10, pp. 1–24, 2018.
- [41] L. McLeod and S. G. MacDonell, "Factors that affect software systems development project outcomes: A survey of research," *ACM Computing Survey*, vol. 43, no. 4, pp. 1–56, 2011.
- [42] K. Kabassi, C. Karydis and A. Bottonis, "AHP, fuzzy SAW and fuzzy WPM for the evaluation of cultural websites," *Multimodal Technology Interaction*, vol. 4, no. 1, pp. 5, 2020.
- [43] L. Rodriguez-Repiso, R. Setchi and J. L. Salmeron, "Modelling IT projects success: Emerging methodologies reviewed," *Technovation*, vol. 27, no. 10, pp. 582–594, 2007.
- [44] C. Okoli and S. D. Pawlowski, "The delphi method as a research tool: An example, design considerations and applications," *Information & Management*, vol. 42, no. 1, pp. 15–29, 2004.
- [45] A. Muhammad, A. Shaikh, Q. N. Naveed and M. R. N. Qureshi, "Factors affecting academic integrity in e-learning of Saudi Arabian Universities. an investigation using delphi and AHP," *IEEE Access*, vol. 8, pp. 16259–16268, 2020.

- [46] B. L. Golden, E. A. Wasil and D. E. Levy, "Applications of the analytic hierarchy process a categorized annotated bibliography, " In: B. L. Golden, E. A. Wasil and P. T. Harker (eds.), *The Analytic Hierarchy Process: Applications and Studies*. New York: Springer Verlag, pp. 37–58, 1989.
- [47] A. Muhammad, A. Siddique, A. E. Youssef, K. Saleem, B. Shahzad *et al.*, "A hierarchical model to evaluate the quality of web-based e-learning systems," *Sustainability*, vol. 12, no. 10, pp. 4071, 2020.
- [48] T. L. Saaty, "How to make a decision: The analytic hierarchy process," *European Journal of Operational Research*, vol. 48, no. 1, pp. 9–26, 1990.
- [49] E. F. Lane and W. A. Verdini, "A consistency test for AHP decision makers," *Decision Sciences*, vol. 20, no. 3, pp. 575–590, 1989.
- [50] K. Kautz, "Customer and user involvement in agile software development," in *Int. Conf. on Agile Processes and Extreme Programming in Software Engineering*, Sardinia, Italy, pp. 168–173, 2009.
- [51] E. Mnkandla and B. Dwolatzky, "Defining agile software quality assurance," in *Int. Conf. on Software Engineering Advances, ICSEA*, Papeete, Tahiti: French Polynesia, pp. 36, 2006.
- [52] I. B. M. Corportation, *IBM SPSS Statistics for Windows (Version 25.0 Armonk)*. NY: IBM Corporation, 2017.
- [53] G. Marshall and L. Jonker, "An introduction to descriptive statistics: A review and practical guide," *Radiography*, vol. 16, no. 4, pp. e1–e7, 2010.
- [54] A. Ishizaka and A. Labib, "Analytic hierarchy process and expert choice: Benefits and limitations," *OR Insight*, vol. 22, no. 4, pp. 201–220, 2009.
- [55] D. Parsons, H. Ryu and R. Lal, "The impact of methods and techniques on outcomes from agile software development projects," in *Int. Working Conf. on Organizational Dynamics of Technology-Based Innovation*, Manchester, UK, pp. 235–249, 2007.
- [56] P. Bourque, R. Dupuis, A. Abran, J. W. Moore and L. Tripp, "The guide to the software engineering body of knowledge," *IEEE Software*, vol. 16, no. 6, pp. 35–44, 1999.
- [57] D. G. E. Odzaly, "Lightweight risk management in agile projects," in *Int. Conf. of Software Engineering & Knowledge Engineering*, Vancour, Candad, pp. 576–581, 2014.
- [58] K. H. Rose, "A guide to the project management body of knowledge," *Project Management Journal*, vol. 3, no. 44, pp. e1, 2013.