IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 10, 2017, accepted September 29, 2017, date of publication October 23, 2017, date of current version March 9, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2765351

Calculating Completeness of Agile Scope in Scaled Agile Development

SUNNIA AMJAD[1], NAVEED AHMAD[1], TANZILA SABA[2], ADEEL ANJUM[1], UMAR MANZOOR[3], MUHAMMAD A. BALUBAID[4], AND SAIF UR REHMAN MALIK[1]

[1]Department of Computer Science, COMSATS Institute of Information Technology, Islamabad 44000, Pakistan
[2]College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia
[3]Computer Science Department, Tulane University, New Orleans, LA 70118, USA
[4]Industrial Engineering Department, King Abdulaziz University, Jeddah 21589, Saudi Arabia

Corresponding author: Naveed Ahmad (naveedahmad@comsats.edu.pk)

ABSTRACT Flexible nature of scope definition in agile makes it difficult or impossible to measure its completeness and quality. The aim of this paper is to highlight the important ingredients of scope definition for agile projects and to present a method for agile projects in order to measure the quality and completeness of their scope definitions. The proposed method considers the elements that are retrieved as a result of the systematic literature review. An industrial survey is conducted to validate and prioritize these elements. Elements are then assigned weights according to their importance in scope definition to build a scorecard for calculating the score of user stories present in the product backlog. The proposed method is able to identify the clear and complete user stories that can better be implemented in the coming iteration. Formal experiments are performed for the evaluation of the proposed method, and it suggests that the method is useful for experts in order to quantify the completeness and quality of scope definition of an agile software project.

INDEX TERMS Agile software development, project management, project planning, project scope management, scope definition.

I. INTRODUCTION

Scope is considered significant as it has a direct impact on timing and budget of the overall project; yet scope management seems to be one of the most neglected domain in both the traditional and agile project management. Defining the scope of the project means that coming to a common understanding of what the project will deliver and what it will not deliver [2]. Project scope definition process is foundation of project scope management as it defines all the work and functions that need to be performed in order to deliver a successful product [1]. Scope refers to all the processes that are involved in performing all the work that is required to deliver a valued product to the customer [3]. According to Project Management Institute (PMI), scope definition is one of the most important and difficult tasks of the project management. Success and failure of any project are strongly linked to its scope definition [4]–[7]. Contribution of scope definition in success of project can be analyzed by a study conducted by NASA which revealed the following interesting facts: companies that spent less than 5% of the total project cost on requirement elicitation experienced cost overruns between 80% and 200% while the companies that spent 8% to 14% of total cost on scope elicitation had cost overruns of less than 60% [1].

Scope definition in agile is a difficult task due to flexible nature of project scope [8], [9]. It occurs iteratively in agile projects. Five levels of planning are considered throughout for defining project scope, namely: *product vision planning, product roadmap, release planning, iteration planning, daily meeting* [10]. *Product vision planning* is the highest/most abstract level of scope definition describing the big picture. *Product roadmap* shows the evolution of the product for next few releases and finally ends up giving a product backlog for the next phase of planning. The third level of planning is the *release planning*, it prioritizes the features in the backlog and divides them into iterations that would be sufficient for release. *Iteration planning* phase is where features are elaborated, tasks are identified and estimated. Iteration plan continues to update until a realistic scope commitment is made. Finally, *daily meeting* measures the progress of the work according to the scope defined in the iteration planning phase.

As mentioned earlier, initially scope definition in agile is at a high level which is revisited in each iteration. Customers are involved in this process of scope definition, constantly validating and reprioritizing the features to be developed [11]. Agile scope definition is a result of constant planning, prioritization and changing requirements within time frame of an iteration. Features with the highest market value are developed first [12]. Agile scope definition in the product backlog is in the form of user stories and is continuously reviewed, prioritized for each iteration during iteration planning [13]. Planning and scoping the project features for delivering valued product to customer on short-term is the essence of agile software development. It satisfies the changed requirements of customers at end of process, instead of the values defined at early stages of the project [14].

Final agile project scope definition is built as a result of scope definitions for each iteration and can possibly be changed up to 30% during each iteration. This iterative approach of planning for scope definition is helpful in faster project execution and delivering large number of benefits on early basis [15]. Agile methodologies believe that change can occur anytime during the life of the project and customer is responsible for giving direction to the project by making requirement, priority and functionality changes in the scope definition of the project [16].

Release plan updates throughout the project because of continuous iteration planning at the start of each iteration. Team make decisions and adjustments needed to the scope after acquiring additional information from iterations [17], [18]. Continuous integration, team communication, customer involvement and documenting requirement are important practices for scope definition as they reduce the rate of over scoping [19]. Release plan, scope elements and product backlog collectively are used for defining scope of the project under the scope management strategies from the start of the project till end [20]. So, here it can be concluded easily that a well-defined scope is very important for making a successful project.

Different perspectives and situations are considered while defining a project scope. Project scope usually contains information about deliverables, technical structure, functionality and data [21]. Scope statement is an essential element of a scope definition process which contains information about the product scope description, business needs, products, services, deliverables, acceptance criteria, exclusion, constraints and assumptions, requested party and project team should agree to all the terms and conditions mentioned in the scope statement before the project starts [22], [23].

Problems introduced by the traditional approaches tend to decrease due to the arrival of modern software development methodology (agile). Agile is considered a remedy for shortcomings of traditional projects because of its emphasis on individuals and interactions, change control, working software and customer collaboration, expanded test coverage, reduced time and cost, collaborative environment, customer satisfaction, improved self-esteem, maintainable code etc [24], [25]. But still there exists some scope related challenges like scope creep, gold plating, scope inflation etc. [25], [26]. Recognizing the certainty of some of the scope related problem in agile highlights that there are very less number of tools and methods for scope definition which leads to the project failure.

Project scope definition has its significance in other industries as well and a lot of work has been done to quantify scope in these industries. CII developed a Project Definition Rating Index (PDRI) tool for industrial, building and infrastructure projects to measure the completeness of their scope definitions and is really helpful in increasing their success rate [4], [5]. Afterwards concept of PDRI was also used in both nuclear and non-nuclear traditional construction projects [27]. The US DOE, NASA, ESKOM, and LLC have developed similar tools [28]. The EM-PDRI is also a tool that is also developed using the concept of PDRI and provides a numerical assessment of how well a project is planned [29]. Inspired from PDRI we then developed a method for software industry named Software Project Scope Rating Index (SPSRI), which was able to gauge the quality and completeness of scope definition of traditional software projects [30]. There is no such tool in software industry for agile based projects to check the completeness and quality of scope. Focus of this research is on checking the completeness and quality of scope and to provide direction to the team members to clarify the scope of project during project planning phase by identifying the critical elements from literature. Literature suggests that projects will be a great failure in terms of cost, time and quality if they don't have a proper scope definition and are planned poorly. This research objects to propose a method that will be able to address the issue of poor planning, vague and unstable scope definition of agile projects by investigating following research questions.

- **RQ-1:** How can we quantify the quality and completeness of scope of iterations?
  - RQ-1.1: What is the appropriate information that can be used to quantify the quality and completeness of scope of iterations?
- **RQ-2:** How can the model be constructed which can be useful in defining scope and measuring its completeness?
  - RQ-2.1: How the quality and completeness of scope can be measured effectively by using such model?

To answer the RQ 1 an extensive literature review was conducted to find out the tools and techniques for measurement of scope in software and other industries. Literature review findings are discussed in section 2. Section 3 builds upon the findings of the literature review to develop a method for checking the completeness and quality of scope of agile projects. In order to answer the RQ 2, the proposed method is demonstrated through an example. Proposed approach is evaluated through series of experiments, which is discussed in section 4. Section 5 discussed the results of evaluation and the limitations of the proposed method. Finally, the conclusion of this research work is presented in section 6.

II. LITERATURE REVIEW

Properly organizing the software development process is of great interest for delivering the systems with better quality. Due to increased rate of failure using traditional approaches, best suggestion for improvement of software development is to adopt agile software development methodologies [31]. Agile methodology was a new concept in late 1990's but it evolved with time as a risk mitigation and technology evolution approach throughout the software development process [32]. Agile is considered a remedy for shortcomings of traditional projects as it greatly emphasis on individuals and interactions, change control, working software and customer collaboration [24]. Agile has various factors that are advantageous for the project success like expanded test coverage, reduced time and cost, collaborative environment, customer satisfaction, improved self-esteem, maintainable code etc. Based on these success factors it was concluded that 75% of the people use agile in almost all of their projects and only 8% of the people use agile rarely in their projects [25].

Project management in very important for every enterprise to make its projects successful. According to Standish group's 2011 Chaos report agile projects are three times more likely to succeed than traditional projects [8]. According to Standish Chaos report of 2013 almost all of the agile projects are completed on time, within budget and with no defects [33]. Research benefits of agile over traditional projects can be summarized as increase in success rate by a huge improvement of 29% in cost, 71 % in schedule, 122% improvement in performance, 75% improvement in quality and 70 % improvement in customer satisfaction [34], but still there exists some challenges and issues in agile that lead to project failure [24]. Agile approach was proposed because of high criticism on non-incremental, inflexible and change resistant nature of traditional approaches but transition from traditional to agile was not an easy process as it still involves a lot of factors that lead to project failure [35], [36]. To address these problems there is a strong need to understand the problems and the reasons behind these problems that are continually involved in causing project failures.

A. CHALLENGES OF AGILE SOFTWARE PROJECTS

Due to software project failures, discipline of software engineering was born in 1968 [37]. Before considering checking measures, it is important to analyze the causes of failures to get a clear view of why the failure occurs [38]. In case of agile, project failure reasons lay under four dimensions; organizational, people, process and technical. Under these four dimensions the factors that cause project failure are:

- Lack of: executive sponsorship, management commitment, agile logistical arrangements, necessary skill-set, project management competence, team work, progress tracking mechanism, customer presence, complete set of correct agile practices.
- Ill-defined: project scope, project requirements, project planning, and customer role.
- Unsuitable tools and technology.
- Organizational size is too large with a traditional and political culture.
- Resistance from groups or individuals.
- Bad customer relationship [39], [40].

Lack of planning, structure and documentation were highlighted as the failure factors or drawbacks in a research conducted by Australian IT consulting firm in 2003 [25]. Factors that lead to failure of agile project includes ignorance in adopting agile approaches, lack of facilities for pair programming, resistance by individuals, lack of training and peer support, complete rely on economic evaluation criteria and organizational resistance to change [25].

Agile development has some challenges or factors that are critical to success of project; like slow participant buy-in, inappropriate mechanism for rewarding individuals, lack of detailed cost evaluation, lack of focus on infrastructure and maintainability, lack of customer involvement and management support, extensive monitoring and scope creep which is the main focus of this research [25].

Some of the challenges for agile projects are due to the scope of the project which are highlighted by some of the authors in their articles: scope in agile may cause problems at the end for the fixed price of the project [25]. Scope creep is one of the basic cause of failure in agile projects, so there should be a change control mechanism for reducing the chances of failure [20]. Slowly increasing and changing scope can easily affect the success of project because of its nonlinear nature [41]. Scope creep problem from business user's side can lead to a technology creep problem for developers. Unavailability of a newer technology to handle a change request can cause problems in achieving the scope of project within time and budget. With the need of a new technology all the plans, estimates, schedule need to be reevaluated and a new model or prototype need to be designed in order to achieve the changed scope of an iteration [41].

It is clearly highlighted by the authors that most of the scope related challenges are due to the poor project planning: poor project planning and scope definition is the major reason behind the failure of software projects [42], [43]. Poor planning in agile development projects also leads to unwanted involvement of people, wastage of money on unwanted and undesired features that are out of scope of the project [44]. Agile projects usually fail due to improper iteration and daily planning meetings. Sometime agile projects fail due to over scoping by the addition of more backlog items in a single iteration [45]. Scope management in agile is done through iterative and incremental process or planning. The success of agile project relies on the delivery of value product to customer and the valued product is achieved through definition and redefinition of project scope throughout agile planning [10]. This research focuses on the importance of clear scope definition and its impact on other areas like time management, cost management, change management etc.

Where a poorly defined scope and poor planning activities are the reasons behind project failure, a well-defined scope and proper-continuous planning are the special ingredients for the success of agile projects [1], [24], [32], [33], [40], [43].

Importance of scope can be judged easily by considering its effect on other areas, so by seeing its importance some of the tools and methods for checking the completeness of scope in agile software and other industries are discussed in the next section.

B. SCOPE TOOLS AND METHODS IN SOFTWARE INDUSTRY

Several studies have put a great emphasis on the importance of project scope definition [10], [20], [21], [46], [47]. According to authors project scope definition is the step towards successful execution of projects [48], [49]. But still there is very limited literature on the scope definition tools for both traditional and agile projects.

According to PMBOK Guide scope planning, scope definition, scope verification and scope control are all the processes that are involved in resolving the scope creep problem [10]. There exist some scope tools for agile which are helpful in managing scope of the project but there exists no tool which is able to quantify the completeness and quality of scope of an iteration.

Feature breakdown structure is a scope tool which shows the proper breakdown of work and is helpful in defining and verifying scope [10]. Story mapping is a scope control tool which provides help in determining what's in scope for that release. It organizes all the stories of large backlog to verify the scope easily and provides a good sense of the whole product [26], [50], [51]. Function Points (FP) is a well-known technique for scope estimate, it provides a metric that includes: Total number of function points or user stories delivered per iteration, productivity of the team per iteration in terms of effort spent by team in days or hours, percentage of change in velocity and defect density, total function points or user stories delivered at the end of project after all the changes that are made during the project [52]–[54].

Agile EVM is a scope control tool which provides help in integrating scope, schedule and resources. It counts and controls the changes in scope of project which ultimately affect the areas of time and cost [55], [56]. Stacked Area Chart is helpful in visualizing big picture of the project and visualizing the status and progress of the project towards achieving its scope in terms of time [26]. The Iceberg list contains total budgeted story points that are allowed for each iteration. It has two parts where "Above water" part shows the story points that will be delivered in the current iteration and the "below water" part shows the lists of story points that will be delivered in next coming iteration [56]. Parking Lot Diagram big picture is a visualization technique for checking that either the team is on schedule while achieving the project scope or not and estimating the project scope (in user stories) [26]. Tree Map is another visualization technique for a larger product backlog where block size tells the size of the product backlog/scope and shades represent the progress of the project towards achieving its objectives [26].

Burndown Chart is a tool for measurement of planning and monitoring progress. It measures team progress in terms of speed and time and represents the amount of remaining work. It suggests and control quick changes in scope to set everything back on the track and shows team progress during the iteration. It is also helpful in getting a valuable information about the amount and frequency of scope change [26], [51], [56]–[58]. Burnup Chart is an alternative of burndown chart. It is able to rate and trend the project track. It tracks incremental progress and tells about amount of work done [56], [57]. Iteration Status Chart provides stakeholders a detailed picture of all the story points involved in each iteration [56].

Agilefant is a simple tool which provides support for release and iteration management in terms of time and effort. Monitor's the progress of project iterations by generating charts like burndown and burnup charts [59]. Cumulative flow diagram is useful in representing the quantity of work done in each iteration. It removes bottleneck and improves throughput by monitoring and controlling progress [57]. TinyPM, ScrumDesk, Agile for Trac and VersionOne are four similar type of planning tools which are helpful in defining and controlling; they provide support for decomposition of releases into iterations, selection of iteration length, prioritization of user stories, decomposition of iterations into tasks and progress tracking [60]. Kanban board is a tool that allows mapping the flow of how team works, setting limits for work in progress. It also manages the work flow in order to know about delivering speed, feedback opportunities and key metrics in order to tune the process [61]. Task board is helpful in finding out the items of interest easily, quickly seeing the completion rate of projects, showing the progress of individuals or groups and quickly updating task status and remaining work [61].

Analogy technique is another estimation technique that considers the past projects that are like the current one and then make estimates on the basis of information gained and the experience with past projects [53]. Price-to-win is a method which only estimates the project considering the customer's ability to pay but it results in cost and time overruns and forced developers to give overtime in order to complete the project [53]. Wideband Delphi method is an appropriate method to estimate the projects that is familiar to all the members and with clear requirements. Members made the estimates for tasks based on their experiences and justify their estimates until all the members end up with a final estimate for the tasks [53], [54]. Source Lines of Code (SLOC) is a cost estimation method that makes estimates by demonstrating the total number of program statement. Previous projects of the same size are considered in order to estimate the effort, scope and cost of the current project [53].

Object point is calculated for a software with well-defined requirements, by dividing the size and complexity (simple, medium and difficult) of all its objects by the developer's productivity (low to highest) [53]. COCOMO methods are widely used to estimate the cost and effort of software projects with clear and detailed requirements [53], [54], [62]. AgileMOW makes use of both the expert judgment and an algorithm to estimate the cost and effort of developing web applications and it only considers people factors [53], [63]. Constructive agile estimation algorithm first makes an initial overall estimate of the project and then it makes iterative

TABLE 1. Software scope definition methods and tools.

Tools/Techniques	Agile / Traditional / Both	Scope Definition				
		Scope	Scope control	Identify features	Quantifying scope	Gauge completeness
Work Breakdown Structure [65]	Both	✓	✓	✓		
Feature breakdown structure [10]	Agile	✓	✓	✓		
Story Mapping [26] [50] [51]	Agile	✓	✓	✓		
Scope Change Control System [66] [67]	Both	✓	✓			
Function Points (FP) [52] [53] [54]	Both		✓	✓	✓	
Functional Size Measurement [68]	Both		✓	✓	✓	
Expert Judgement [53] [66] [67]	Both			✓		
Templates, Forms and Standards [66] [67]	Both			✓		
Identification of Alternatives [66] [67]	Both			✓		
Stakeholder Analysis [66] [67]	Both			✓		
Benefit Analysis [66]	Both			✓		
TinyPM, ScrumDesk, Agile for Trac, VersionOne [60]	Agile		✓	✓		
Performance Analysis [66] [67]	Both		✓			
Requirement Matrix [69]	Agile		✓			
Cumulative Flow [57]	Agile		✓			
Agile EVM [55] [56]	Agile		✓			
Stacked Area Chart [51]	Agile		✓			
Burndown Chart [26] [51] [56] [57]	Agile		✓			
Burnup Chart [56] [57]	Agile		✓			
Iteration Status Chart [56]	Agile		✓			
Agilefant [59]	Agile		✓			
Iceberg List [56]	Agile		✓			
Kanban Board [61] [70]	Agile		✓			
Tree Map [26]	Agile		✓			
Parking Lot Diagram[26]	Agile		✓			
Task Board [61]	Agile	✓	✓			
Planning Poker [53] [71] [72]	Agile	✓	✓	✓	✓	
Software Project Scope Rating Index (SPSRI) [30]	Traditional					✓
Analogy Technique [53]	Both	✓			✓	
Price-to-Win [53]	Both	✓	✓		✓	
Wideband Delphi [53][54] [73]	Agile	✓		✓	✓	
Source Lines of Codes (SLOC) [53]	Both			✓	✓	
Object Point [53]	Both			✓	✓	
Constructive Cost Model (COCOMO) [53] [54] [62]	Agile	✓		✓	✓	
AgileMOW [53] [63]	Agile	✓			✓	
Constructive Agile Estimation Algorithm [53]	Agile				✓	
PERT [54]	Both				✓	
Activity Based (Top-down) [53] [64]	Both			✓	✓	
Task Based (Bottom-up) [53] [64]	Both			✓	✓	

estimates at the start of each iteration to get the final estimate [53]. Bottom-up or activity based estimation method is beneficial when requirements are clear at early stage, able to separately estimate each individual component and then give a final estimate after summing up all [53], [64]. Task based or top-down method provides an overall estimate of the system then provides help in individually estimating the individual components or functions based on initial total estimate [53], [64]. Table 1 shows the list of scope definition tool and techniques for both traditional and agile software projects.

Limitations of Existing Scope Definition Tools and Methods: Most of the existing tools and methods for agile software projects provide support for visualizing, verifying, monitoring and controlling changes to project scope definition. Some of them are helpful in determining the amount of work that need to be done in each coming iteration. Existing scope tools for both traditional and agile software projects are mentioned in Table 1. Among them there exists no tool which is able to quantify the completeness and quality of scope definition for agile projects. So, there is a strong need of one such method which can help experts in order to measure the completeness and quality of scope definitions.

C. SCOPE TOOLS AND METHODS IN OTHER INDUSTRIES

Scope definition has its importance in other industries too. In 1992 Hackney was the first man who made a scope development tool that was used to define and quantify the important elements of scope definition, He identified the critical elements for scope definition and then assigned them weights to measure the impact rate of that element on the success of project [5]. Inspired from the work of Hackney, CII in 1994

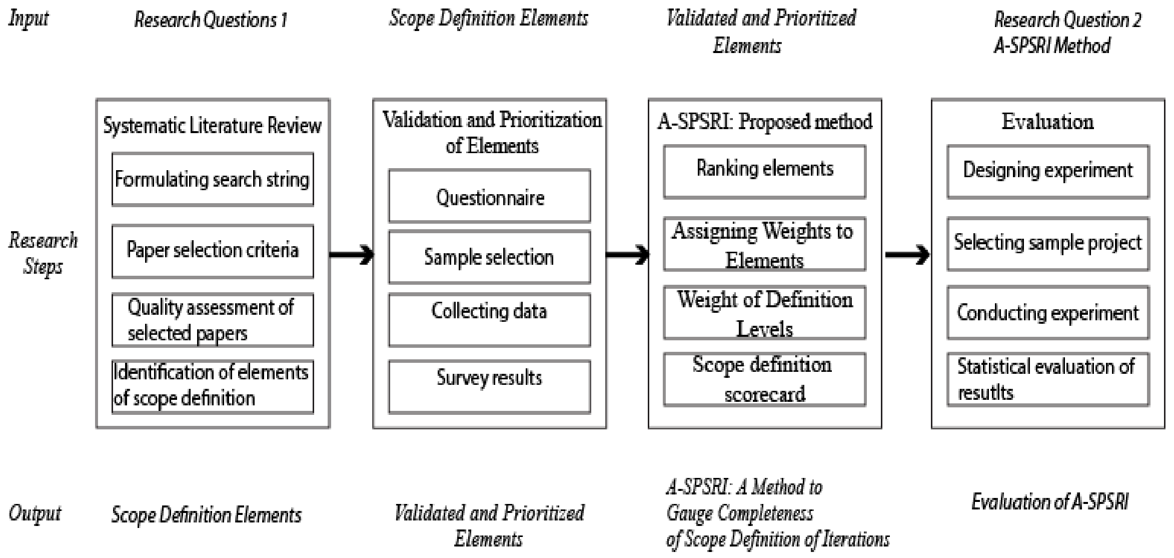

Input
Research Questions 1
Scope Definition Elements
Validated and Prioritized Elements
Research Question 2 A-SPSRI Method
Research Steps
Systematic Literature Review
Formulating search string
Paper selection criteria
Quality assessment of selected papers
Identification of elements of scope definition
Validation and Prioritization of Elements
Questionnaire
Sample selection
Collecting data
Survey results
A-SPSRI: Proposed method
Ranking elements
Assigning Weights to Elements
Weight of Definition Levels
Scope definition scorecard
Evaluation
Designing experiment
Selecting sample project
Conducting experiment
Statistical evaluation of resutlts
Output
Scope Definition Elements
Validated and Prioritized Elements
A-SPSRI: A Method to Gauge Completeness of Scope Definition of Iterations
Evaluation of A-SPSRI

FIGURE 1. A-SPSRI methodology.

sponsored a series of studies focused on the development of tools that could help project teams in attaining a greater level of scope definition and for measuring the completeness of the industrial project. 70 critical elements were identified and were given weights to show their importance within a scope definition. Tool provides a checklist for determining the definition of a project at the time of analysis and is helpful in predicting the success or failure of project. Team members can use this tool to focus on the areas where project is lacking behind the scope. It is helpful for the planning team as it provides them with the ability to quantify, rate, and assess the level of scope definition prior to approval for detailed design or construction. Until now CII has developed three tools for industrial in 1996, buildings in 1999 and infrastructure projects in 2010 [4], [5].

After successful implementation of PDRI in construction industry, it was also used in both nuclear and non-nuclear traditional construction projects [27]. The US DOE, NASA, ESKOM, Chang *et al.*, Project Auditors and LLC have developed similar tools based on this methodology [28]. The EM-PDRI is also a tool that is also developed using the concept of PDRI, it provides a numerical assessment of how well a project is planned [29].

Another notable study that can help understand project scope definition is done by Texas Department of Transportation (TxDOT), they introduced a risk management tool named Advanced Planning Risk Analysis (APRA) that was designed to focus on a project's scopes of infrastructure projects and was also helpful in improving the clarity, comprehensiveness and entirety of those scopes and was helpful in verifying the scope [4].

III. METHOD: AGILE SOFTWARE PROJECT SCOPE RATING INDEX (A-SPSRI)

This research emphasis to propose a method for agile software projects. The proposed approach is termed as Agile Software Project Scope Rating Index (A-SPSRI), it provides help in quantifying the completeness and quality of agile project scope definition and it also works as a guide for experts to find out the weaknesses in scope definitions of agile projects. Research questions presented in Section 1 directs the whole research. The research method is divided into four key steps: 1) Systematic Literature Review, 2) Validation and Prioritization of Elements, 3) Proposed Method, Agile Software Project Scope Rating Index (A-SPSRI) to measure the completeness and quality of scope definitions, and 4) Evaluation. Research method is shown in Fig. 1.

TABLE 2. Notations and their meanings.

Notation	Meaning
PL_i	i^{th} planning level among the set of planning levels PL
E	The set of elements
e_i	i^{th} element, $e_i \epsilon$ E
M	Total number of planning levels in set PL
N	Total number of elements in set E
p_i	Priority percentage of elements
r_i	Rank of element e_i
dl_i	Definition level of an element e_i
X	Rank of definition level, where x = 1,2,3 or 4
Z	Total number of definition levels
W_{e_i}	Weight of i^{th} element
MAX	1000 (Maximum score of A-SPSRI)
W_{dl_i}	Weight of i^{th} definition level
Scr_{e_i}	Score of the i^{th} element
Scr_{PL_i}	Score of i^{th} planning level
T_{scr}	Total scope score

Basic Notations: Table 2 shows the notations that are involved throughout the process for scope score calculation.

Planning Levels and Elements in A-SPSRI

$$PL = \{PL_1, PL_2, \ldots, PL_m\}$$
$$E = \{e_1, e_2, \ldots, e_n\} \qquad (1)$$

Weights of an element e_i

Rank Centroid Method [74], [75] is used to assign weights to all elements. As elements are not sorted in the order of

TABLE 3. Search strings.

Database	Search String
Google Scholar	Software AND (project OR projects OR product OR products) AND (Scope OR “Scope management” OR “Project scoping” OR “scope determination” OR “Scope definition”) AND (agile, agility, scrum, Extreme programming) AND (Planning OR “planning levels” OR “planning benefits”) AND (Challenges" OR problems OR success factors OR “failure reasons” OR factors OR root causes) AND (Scope-creep OR over-scoping OR “scope failure” OR “inflexible scope”) AND (Techniques OR tool OR tools OR methods OR methodology OR methodologies).
Science Direct	Software AND (project OR product) AND (Scope OR “Scope management”) AND (agile "Extreme programming") AND (“planning levels”) AND (Challenges OR problems OR "success factors") AND ("Scope-creep" OR "over-scoping") AND (Techniques OR tool OR methods)
Springer Link	Software AND (project OR product) AND (Scope OR “Scope management”) AND (agile "Extreme programming") AND (“planning levels”) AND (Challenges OR problems OR "success factors") AND ("Scope-creep" OR "over-scoping") AND (Techniques OR tool OR methods)
IEEE	Software AND (project OR product) AND (Scope OR Scope management) AND (agile OR Extreme programming) AND (planning levels) AND (Challenges OR problems) AND (Scope-creep OR over-scoping) AND (Techniques OR tool OR methods)
ACM	Software AND (project OR product) AND (Scope OR Scope management) AND (agile OR Extreme programming) AND (planning levels) AND (Challenges OR problems) AND (Scope-creep OR over-scoping) AND (Techniques OR tool OR methods)

ranks so the equation below has been changed.

$$W_{e_i} = \left(\frac{1}{N}\right)\sum_{j=r_i}^{N}\left(\frac{1}{j}\right) * MAX; \qquad \sum_{i=1}^{22} W_{dl_i} = 1000 \quad (2)$$

Weights of definition levels

$$\begin{aligned} \mathrm{W}_{\mathrm{dl_i}} &= \left(\frac{1}{\mathrm{Z}}\right)\sum_{\mathrm{x=i}}^{\mathrm{Z}}\left(\frac{1}{\mathrm{x}}\right)\mathrm{W}_{\mathrm{dl_i}} \\ &= \left\{ 0 < \mathrm{W}_{\mathrm{dl_i}} \right. \\ &< \begin{matrix} 1, & \mathrm{i} = 1 \\ 1, & \mathrm{i} = 2, 3, 4 \\ 0, & \mathrm{i} = 0 \end{matrix} \end{aligned} \quad (3)$$

Score of an element e_i

$$\mathrm{Scr}_{\mathrm{e_i}} = \left(\mathrm{W}_{\mathrm{e_i}} * \mathrm{W}_{\mathrm{dl_i}}\right) \quad (4)$$

Score of planning level PL_i

$$\mathrm{Scr}_{\mathrm{PL_i}} = \sum\nolimits_{\mathrm{i=1}}^{\mathrm{N}} \mathrm{Scr}_{\mathrm{e_i}} \quad (5)$$

Total scope score

$$\mathrm{T}_{\mathrm{scr}} = \sum_{\mathrm{i=1}}^{\mathrm{M}} \mathrm{Scr}_{\mathrm{PL_i}} \quad (6)$$

A. SYSTEMATIC LITERATURE REVIEW: PROTOCOL, SCOPE ELEMENTS

An extensive literature review was performed on scope management and scope definition, existing scope tools and methods in software and other industries, in order to answer the RQ 1. Review of literature on other industries resulted in the observation that they identified scope definition elements for measuring the completeness of scope definitions. Therefore, in our previous work we identified the important scope definition elements by conducting a systematic literature review on scope definition of traditional software projects in order to measure their completeness. As this research focuses on agile software development, so we conducted a systematic literature review on scope management in agile to identify the important scope definition aspects or elements of scope definition.

Search String used: Criteria for search strings used and paper selection from five databases are described below in Table 3.

1) IDENTIFICATION OF RELEVANT STUDIES

It was crucial to discover agile project scope definition elements as there is very little work done on agile scope management. Five well-known journal databases IEEE Xplore, Science Direct, ACM, Springer Link and Google Scholar were extensively searched with different search strings mentioned above. Google books on agile were also considered while collecting material related to agile scope definition elements, as we have mentioned earlier that there exists very limited literature on agile scope management in the form of research articles.

Research articles from year 2000 to 2017 were searched in order to discover scope definition elements of agile projects, selection of paper consists of two steps: 1) Related 534 papers retrieved from the above-mentioned databases were filtered and 244 were selected out of them based on the title. Selected 244 papers were again filtered based abstract and keywords relevance and thus 88 papers were selected. 2) Most likely studies were identified; then full text filtration was performed by applying inclusion/exclusion criteria; text was also evaluated in order to find out the agile scope definition elements 50 articles were selected finally. References of all the selected articles were also analyzed in order to find out the articles that we missed in the initial search. Thus, 4 related articles were identified and then included. They were focusing on

TABLE 4. Occurrence of elements in literature.

Elements	Occurrence in Literature
e_1. Product Future Statement	[44] [78]
e_2. Market Strategy VS Project Strategy	[79]
e_3. Release Definition and Selection	[44] [80] [81]
e_4. Release Plan	[44] [82] [83] [84] [85]
e_5. Iteration Mission	[86] [58] [8] [12] [87]
e_6. Resource Estimation	[58] [10] [86] [88] [89] [90]
e_7. Iteration Schedule	[86] [91]
e_8. Task Identification	[58] [14] [80] [85] [86] [88]
e_9. Managing Dependencies	[10] [88]
e_{10}. Key Deliverables	[21] [22] [58] [92]
e_{11}. Identifying Constraints	[86]
e_{12}. Managing Risks/Uncertainties	[58] [86] [93] [94] [41] [95] [12] [96] [97] [98]
e_{13}. Adjust Priorities	[58] [88] [11] [12] [99]
e_{14}. Review/ Update Release Plan	[17] [18] [20] [58]
e_{15}. User/Client Involvement	[19] [24] [25] [58] [86] [12] [100] [95] [101] [100] [102] [103] [104] [105] [106][107] [108] [109]
e_{16}. Building Trust in Team	[86] [58] [110] [111] [112] [113]
e_{17}. Manager's Role	[86] [114] [115]
e_{18}. Contractual Terms and Conditions	[58] [116] [117] [118] [119]
e_{19}. Use of Technology	[41] [12] [120] [121]
e_{20}. Client Acceptance and Change Control	[58] [16] [24] [25] [101] [12] [87] [88] [11] [102] [105] [106] [107] [122] [113] [123] [124] [92]
e_{21}.Collaborative Development Environment	[86] [24] [25] [102] [103] [105] [107] [122] [124]
e_{22}. Daily Meetings	[44] [125] [11] [104] [111] [113] [90]

the scope definition in agile, failure and success reasons of agile, tools/methods for estimating agile software projects plus they were also highlighting the important aspects or elements of scope definition that were helpful in measuring its completeness and quality. From the selected 50 research articles 9 were excluded later as they were not providing any valuable information.

As there is very limited literature on Agile scope management, so we also included books from year 2000 to 2017 in the SLR. Selected books were the ones whose focus was on agile scope management, agile scope definition, agile scope tools or methods and provided valuable information about agile scope definition elements.

2) IDENTIFICATION OF ELEMENTS

Content analysis method was used for discovering elements from literature. Content analysis is a widely used qualitative research technique, it consists of three different approaches; conventional, directed, or summative. These are used to interpret meaning from the text data [76]. Approach was first used in social sciences for the study of human communication and many of the authors used this approach in engineering studies [77]. Agile scope elements were identified after analyzing the selected books and research articles. Table 4 shows the occurrence of each element in literature.

Extensive literature review provided different elements that are important in scope definition of agile projects. Most of the elements are related to the people, organizational, process and technical dimension and these have a great impact on the scope of the project [39], [40]. By adopting the approach devised by CII for the division of elements into sections and categories [4], [5], we first collected the elements in a spreadsheet and then divided into five planning levels of agile to develop a final list. Table 5 shows the final list of 22 elements which are divided into five levels of agile planning.

B. VALIDATION AND PRIORITIZATION OF ELEMENTS

Once elements were identified and divided into planning levels, each element of agile software project was defined in detail for clear understanding. For this purpose, an extensive literature review on agile way of development was performed to get views of different authors about the collected elements. Information collected from literature was then formulated in form of proper definitions for clearly understanding those elements. A complete list of elements along with their description is available in Appendix A.

The next step in A-SPSRI is to validate the elements and their descriptions and to prioritize them based on their importance. The elements were then validated and prioritized through a survey. To conduct a survey different companies were considered. Finally, questionnaire was sent to almost 20 companies where the main targets were CEO's, project managers, senior agile developers, testers and team leads. The survey was conducted locally in Pakistan and the population involved those companies which were highly involved in doing agile based projects. Questionnaire was sent via email and within three-weeks the response rate was 60%, where most of the respondents were project managers and the ones who had 4 to 5 years' experience of working in agile. Questionnaire was divided into two sections, in first section

TABLE 5. Elements, their respective ranks and weights.

Planning Levels/Elements	Priority Percentage p_i	Rank r_i	Weight w_{ei}
PL_1. Product Vision Planning			
e_1. Product Future Statement	3.17	22	2.06
e_2. Market Strategy VS Project Strategy	3.41	19	8.89
PL_2. Product Roadmap			
e_3. Release Definition and Selection	3.61	13	26.70
PL_3. Product Release Planning			
e_4. Release Plan	3.71	7	56.40
PL_4. Iteration Planning			
e_5. Iteration Mission	3.64	12	30.49
e_6. Resource Estimation	3.84	6	63.97
e_7 Iteration Schedule	3.71	8	49.90
e_8. Task Identification	3.97	2	122.30
e_9. Managing Dependencies	3.58	14	23.21
e_{10}. Key Deliverables	3.94	3	99.58
e_{11}. Identifying Constraints	3.51	8	11.42
e_{12}. Managing Risks/Uncertainties	3.69	11	34.62
e_{13}. Adjust Priorities	3.89	4	84.43
e_{14}. Review/ Update Release Plan	3.58	15	19.96
e_{15}. User/Client Involvement	3.84	5	73.06
e_{16}. Building Trust in Team	3.97	1	167.76
e_{17}. Manager's Role	3.56	16	16.93
e_{18}. Contractual Terms and Conditions	3.30	21	4.23
e_{19}. Use of Technology	3.53	17	14.09
e_{20}. Client Acceptance and Change Control	3.71	9	44.22
$e_{21.}$Collaborative Development Environment	3.33	20	6.50
PL_5. Daily Commitment Planning			
e_{22}. Daily Meetings	3.71	10	39.17

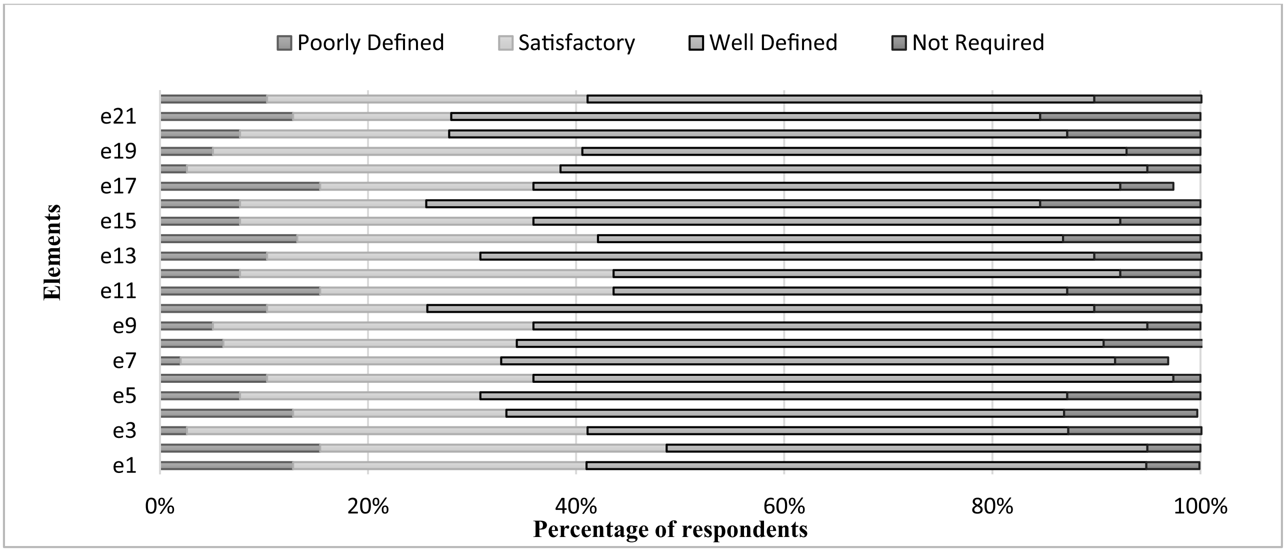


FIGURE 2. Validation of elements from practitioners.

respondents were asked to read the elements descriptions and then rate their agreement to validate these elements on a four-point scale (not required, poorly defined, satisfactory, well define,). Fig 2 shows the results. Second part of the questionnaire was designed in a way that respondents were asked to give their views on a five-point Likert scale (very limited, limited, average, extensive, and very extensive) in order to find out the importance of elements depending on their use in the scope definition. Then to summarize the results of the survey weighted mean approach [139], [140] is used for the collected responses against each element. Formula used for calculating the weighted mean of responses against each element is discussed below:

$$\bar{x} = \frac{\sum_{i=1}^{n} (x_i * w_i)}{\sum_{i=1}^{n} x_i} \tag{7}$$

Where, w_i = the weight assigned to the options against questions and x_i = number of respondents

As in the questionnaire, options provided to the respondents for prioritizing elements were: 1-Very Limited, 2-Limited, 3-Average, 4- Extensive, 5-Very extensive. Then to calculated mean of responses the weights are assigned to these options: 1,2,3,4 and 5 respectively. The weighted mean values calculated using equation (7) for each element are shown in Table 5, these values show the priority of each element in scope definition of the project.

C. A–SPSRI

1) RANKING ELEMENTS

Considering equal importance of all the 22 found elements is a wrong approach. So, it is very important to rank these elements based on their importance in the scope definition. Weighting approach for A-SPSRI is divided into two main steps: ranking of elements and assigning weights to them. According to Newing, ranking provides order of preference and then weights are assigned based on the ranks to get a relative magnitude [126]. In order to rank the validated elements, results of the survey were considered which gave the priority percentage of each element based on its importance in scope definition. Now the ranks of the elements are calculated using these values, higher the value greater the rank. Ranks of the elements from 1(most important) to 22 (least important) are also shown in Table 5.

2) ASSIGNING WEIGHTS TO ELEMENTS

Once ranks are identified, Rank Order Centroid (ROC) method is used to calculate the weights of each element which are ranked based on their importance. According to Barron and Barret (1996) ROC is more useful, accurate and practical method of weight assignment, it takes ranks of elements as input which makes it easy to calculate exact weights and overcomes biasness [129]. In A-SPSRI, just like SPSRI [30] ranks are used as an input to obtain weight of each element, which are then multiplied with 1000; this process is repeated for all the element to get total 1000 points for A-SPSRI. Priority percentages, ranks and weights of all elements are shown in Table 3. Sample weight calculation using equation (2) is shown below for e_9 (user/client involvement).

$$\begin{aligned}\mathrm{W}_{\mathrm{e}_9} &= \left(\frac{1}{\mathrm{N}}\right)\sum\nolimits_{\mathrm{j}=\mathrm{r}_9}^{\mathrm{N}}\left(\frac{1}{\mathrm{j}}\right) * \mathrm{MAX} = \left(\frac{1}{22}\right)\sum\nolimits_{\mathrm{j}=5}^{22}\left(\frac{1}{\mathrm{j}}\right) * 1000\\ &= \left(\frac{1}{22}\right)\left(\frac{1}{5}+\frac{1}{6}+\ldots+\frac{1}{22}\right) * 1000 = 73.06\end{aligned}$$

Contribution and importance of e_9 in the total score of A-SPSRI can be shown by its value of 73.06. Weights of all the other elements are calculated the same way and are shown in Table 5. After these calculations, it is easy to see the contributions of critical elements, among which top five elements in descending order of their weights are: 1) Building trust in team (e_{10}), task identification (e_{17}), key deliverables (e_{20}), adjust priorities (e_7) and user/client involvement (e_9).

3) WEIGHT OF DEFINITION LEVELS AND A-SPSRI SCORE CARD CALCULATION

Individual weights of all the elements are used in calculating the total scope score (T_{scr}). Quality of an element is one of the important aspect when evaluating scope definitions of the projects. Here quality is stated as the definition level (dl). For example, element (e_{14}) says that use of technology should be shown in making a successful scope statement of a project and it can be shown in various ways depending on the type and need of technology, hence score of the element will be adjusted.

A scale is defined for determining the definition level (dl) of an element and the levels in the scale are defined as: 0-Not Applicable, 1-Poor Definition, 2-Minor Deficiencies, 3-Major Deficiencies, 4-Complete Definition. It is important to find out the weights of definition levels as according to Barron and Barret it is not sufficient to have definition levels alone [129]. Equation (3) under its constraints is considered for calculating the score of definition levels, the same way we did it for SPSRI [30]. Definition level ‘0’ is added only to satisfy that element which is at this level is not relevant to the project and it does not affect the results of the A-SPSRI. Hence, weight will always be 0 for definition level ‘0’. Shown below is the calculation for the definition level ‘3’.

$$W_{dl_i} = \left(\frac{1}{Z}\right)\sum\nolimits_{x=i}^{Z}\left(\frac{1}{x}\right) = \left(\frac{1}{4}\right)\left(\frac{1}{3}+\frac{1}{4}\right) = 0.15$$

Weights for definition levels 0, 1, 2, 3 and 4 are 0, 0.52, 0.27, 0.15 and 0.06 respectively. According to the equation (3), these weights do not satisfy the constraints. So, weights need to be rescaled between 0 and 1 which is performed using equation (8).

$$\begin{aligned}\mathrm{W}_{\mathrm{dl_i}} = &\left(\frac{\mathrm{W}_{\mathrm{dl_i}} - \text{lowest value of existing scale}}{\text{Width of existing scale}}\right)\\ &\times (\text{Width of new scale})\\ &+ \text{Lowest value of new scale} \qquad (8)\end{aligned}$$

Calculation for definition level ‘3’ after rescaling is performed using equation (8), which shows following results.

$$\mathrm{W}_{\mathrm{dl}_3} = (0.15 - 0/0.52) \times 1 + 0 = 0.28$$

Weights of the definition levels 0,1,2,3,4 changed to 0,1, 0.51, 0.28, and 0.11 respectively after rescaling. These weights of definition levels are very helpful in quantifying the scope of the project along with the individual weight of each element. Fig 3 shows the scope scorecard, where each element is given a definition level, for example, element ‘e_4’ has given a definition level of ‘dl_4’ then the scope score of ‘e_4’ ($\mathrm{Scr}_{\mathrm{e}_4}$) is calculated using equation (4) as:

$$\mathrm{Scr}_{\mathrm{e_i}} = \left(\mathrm{W}_{\mathrm{e_i}} * \mathrm{W}_{\mathrm{dl_i}}\right) = 56.40 \times 0.11 = 6.204$$

To get a total scope score of a project’s scope statement this process is repeated for all the elements.

Just like scoring scheme used for SPSRI [30], a lower value of T_{scr} means a better or well-defined scope statements for A-SPSRI too.

a

t	Definition Levels (*dl*)				
	'0'	'4'	'3'	'2'	'1'
e₁			☒		
e₂				☒	
. . .					
e₂₂		☒			
Total Score (T_{scr})					

b

Elements	**Definition Levels (*dl*)**					**Scr_{e_i}**
	'0' -(0.00)	'4' -(0.11)	'3' -0.28()	'2' -(0.51)	'1' -(1.00)	
e₁(2.06)			☒			
e₂ (8.89)				☒		
. .						
e₂₂ (39.17)		☒				
Total Score (T_{scr})						**$\sum Scr_{e_i}$**

FIGURE 3. (A) Sample un-weighted scorecard; (B) Sample weighted scorecard.

4) SCOPE SCORE CALCULATION

In A-SPSRI final step is to calculate the total scope score (T_{scr})of a project's scope statement using scope scorecard. But before calculating total scope score of iteration, scores of each planning levels are calculated. Each planning level element is scored using scope scorecard depending on its definition level; this is done for all the elements in all the planning levels same way. Score of each planning level can be calculated using equation (5).

$$\mathrm{Scr}_{\mathrm{PL_i}} = \sum\nolimits_{i=1}^{N} Scr_{e_i}$$

Score calculations of planning levels for the scope scorecard shown in figure 4 are as follow:

For Planning Level 1:

$$\mathrm{Scr}_{\mathrm{PL_1}} = \sum\nolimits_{i=1}^{2} Scr_{e_1} + Scr_{e_2}$$

For Planning Level 2:

$$\mathrm{Scr}_{\mathrm{PL_2}} = Scr_{e_3}$$

For Planning Level 3:

$$\mathrm{Scr}_{\mathrm{PL_3}} = Scr_{e_4}$$

For Planning Level 4:

$$\mathrm{Scr}_{\mathrm{PL_4}} = \sum\nolimits_{i=5}^{21} Scr_{e_5} + Scr_{e_6} + \ldots + Scr_{e_{21}}$$

For Planning Level 5:

$$\mathrm{Scr}_{\mathrm{PL_5}} = Scr_{e_{22}}$$

Final score of scope definition of each iteration can be calculated using equation (6). Total scope score calculations for the scope scorecard shown in Fig 4 are as follow:

$$T_{scr} = \sum\nolimits_{i=1}^{M} Scr_{PL_i}$$
$$= 6.59 + 13.61 + 28.76 + 561.16 + 39.17 = 649.2$$

Fig 4 shows the scope scorecard calculation of all the planning levels using weighted scorecard. Scope statement of *iteration 4* of "Malisic Marketing Application" is used, which is shown in Appendix B.

In the calculations above all the five planning levels of A-SPSRI are evaluated for iteration 4. Definition levels are assigned to all the elements after reading the descriptions. To find out the scores of each planning levels, scores of elements are written in the "Scr_{e_i}" column. These scores are then added for all the levels to obtain total score of each planning level. Then to calculate the total score of user story, scores of all the planning levels are added. In order to calculate scope scores of other iterations, repeat the above-mentioned steps for all of them.

D. EVALUATION

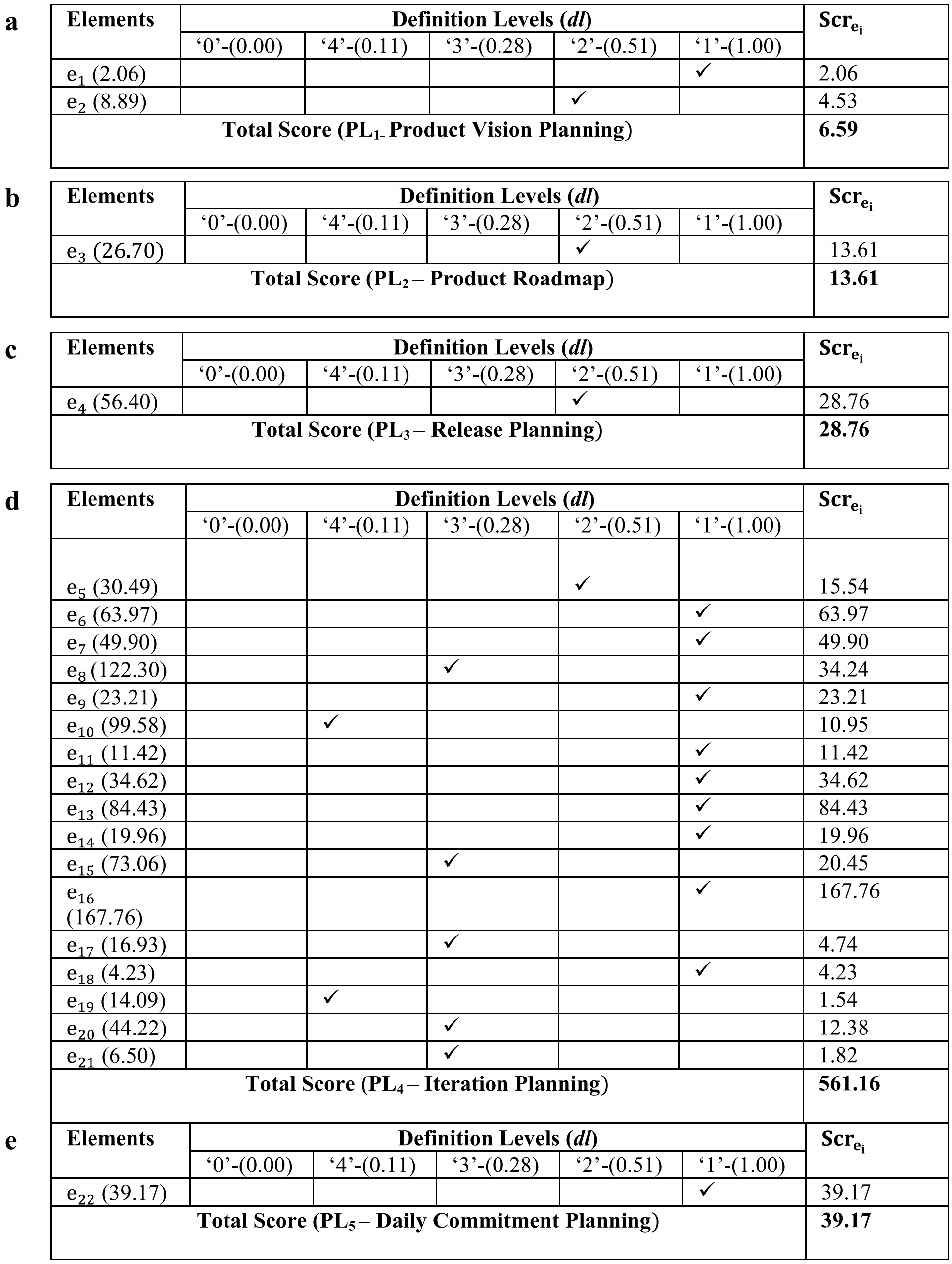
In order to find out the usability and utility of A-SPSRI a formal experiment is performed. Results of experiment are then analyzed statistically.

E. APPLICATION OF A-SPSRI

Application of our proposed method can be seen during iteration planning. Scope scores calculated for iterations using A-SPSRI can be helpful in defining the scope of iterations and selecting the complete/well defined iteration to implement first. Fig 5 shows the application of A-SPSRI in iteration planning.

IV. USABILITY AND UTILITY EVALUATION OF A-SPSRI

Establishing criteria for measuring the usability and usefulness of A-SPSRI is essential in order to evaluate it. A-SPSRI is considered usable if it will allow users to evaluate the quality and completeness of scope definitions in considerably less time and with minimum involvement of the authors. A-SPSRI is considered useful if it will allow users to identify maximum number of problems from a scope statement of any user story. Two hypothesis are defined below, in order to evaluate usability and usefulness of A-SPSRI. Criteria and hypothesis for eval-



a

Elements	Definition Levels (*dl*)					Scr_{e_i}
	'0'-(0.00)	'4'-(0.11)	'3'-(0.28)	'2'-(0.51)	'1'-(1.00)	
e_1 (2.06)					✓	2.06
e_2 (8.89)				✓		4.53
Total Score (PL_1- Product Vision Planning)						**6.59**

b

Elements	Definition Levels (*dl*)					Scr_{e_i}
	'0'-(0.00)	'4'-(0.11)	'3'-(0.28)	'2'-(0.51)	'1'-(1.00)	
e_3 (26.70)				✓		13.61
Total Score (PL_2 – Product Roadmap)						**13.61**

c

Elements	Definition Levels (*dl*)					Scr_{e_i}
	'0'-(0.00)	'4'-(0.11)	'3'-(0.28)	'2'-(0.51)	'1'-(1.00)	
e_4 (56.40)				✓		28.76
Total Score (PL_3 – Release Planning)						**28.76**

d

Elements	Definition Levels (*dl*)					Scr_{e_i}
	'0'-(0.00)	'4'-(0.11)	'3'-(0.28)	'2'-(0.51)	'1'-(1.00)	
e_5 (30.49)				✓		15.54
e_6 (63.97)					✓	63.97
e_7 (49.90)					✓	49.90
e_8 (122.30)			✓			34.24
e_9 (23.21)					✓	23.21
e_{10} (99.58)		✓				10.95
e_{11} (11.42)					✓	11.42
e_{12} (34.62)					✓	34.62
e_{13} (84.43)					✓	84.43
e_{14} (19.96)					✓	19.96
e_{15} (73.06)			✓			20.45
e_{16} (167.76)					✓	167.76
e_{17} (16.93)			✓			4.74
e_{18} (4.23)					✓	4.23
e_{19} (14.09)		✓				1.54
e_{20} (44.22)			✓			12.38
e_{21} (6.50)			✓			1.82
Total Score (PL_4 – Iteration Planning)						**561.16**

e

Elements	Definition Levels (*dl*)					Scr_{e_i}
	'0'-(0.00)	'4'-(0.11)	'3'-(0.28)	'2'-(0.51)	'1'-(1.00)	
e_{22} (39.17)					✓	39.17
Total Score (PL_5 – Daily Commitment Planning)						**39.17**

FIGURE 4. Example scope evaluation of user stories.

uating usability and usefulness of A-SPSRI are described below:

Usability Evaluation of A-SPSRI: A-SPSRI allows user to evaluate the scope definition in less time than without A-SPSRI.

Hypothesis 1: Using A-SPSRI user evaluates the scope definition in consistently less time than without A-SPSRI.

$$\mathbf{H_01}: \mu_{\mathbf{A-SPSRI}} \geq \mu_{\mathbf{without\ A-SPSRI}}$$

$$\mathbf{H_11}: \mu_{\mathbf{A-SPSRI}} < \mu_{\mathbf{without\ A-SPSRI}}$$

μ is mean time taken by people using A-SPSRI and not using A-SPSRI.

Utility Evaluation of A-SPSRI: A-SPSRI allows user to become familiar with the project to identify missing items from the scope definition as compared to the user without A-SPSRI.

Hypothesis 2: Using A-SPSRI user familiar with the project identifies more problems in comparison to the user without A-SPSRI.

$$\mathbf{H_02}: \mu_{\mathbf{A-SPSRI}} \leq \mu_{\mathbf{without\ A-SPSRI}}$$

$$\mathbf{H_12}: \mu_{\mathbf{A-SPSRI}} > \mu_{\mathbf{without\ A-SPSRI}}$$

μ is mean problems identified by people using A-SPSRI and not using A-SPSRI.

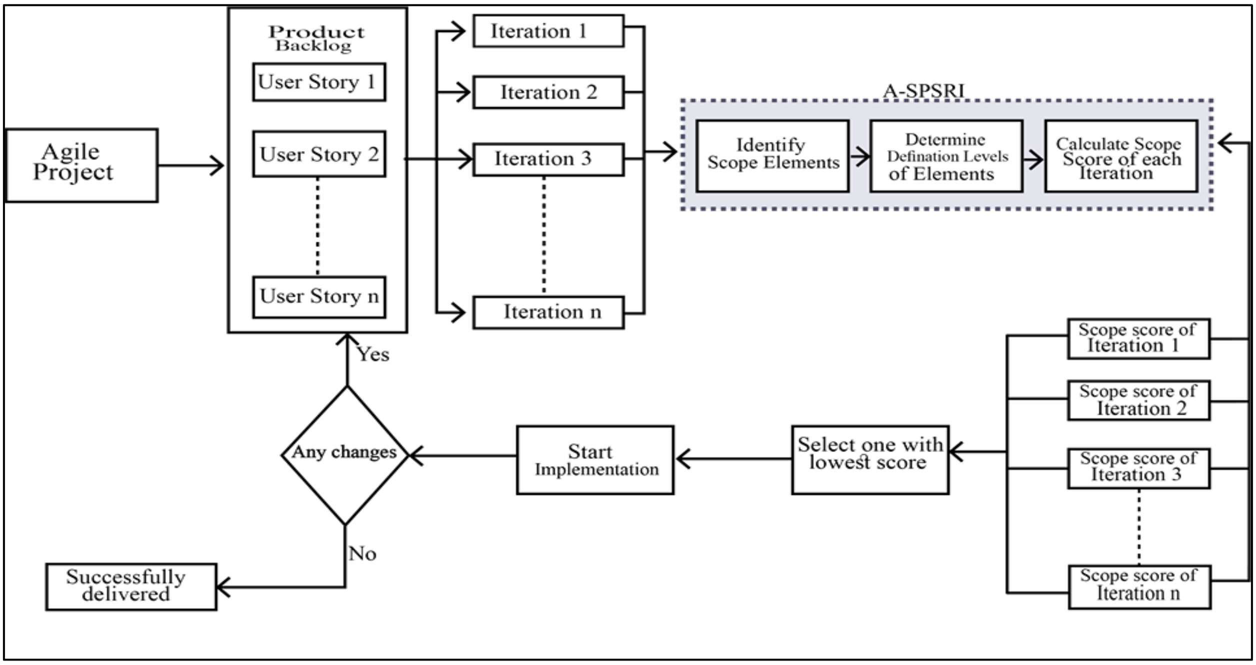

Agile Project
Product Backlog
User Story 1
User Story 2
User Story n
Iteration 1
Iteration 2
Iteration 3
Iteration n
A-SPSRI
Identify Scope Elements
Determine Defination Levels of Elements
Calculate Scope Score of each Iteration
Scope score of Iteration 1
Scope score of Iteration 2
Scope score of Iteration 3
Scope score of Iteration n
Select one with lowest score
Start Implementation
Any changes
Yes
No
Successfully delivered

FIGURE 5. Application of A-Spsri in iteration planning.

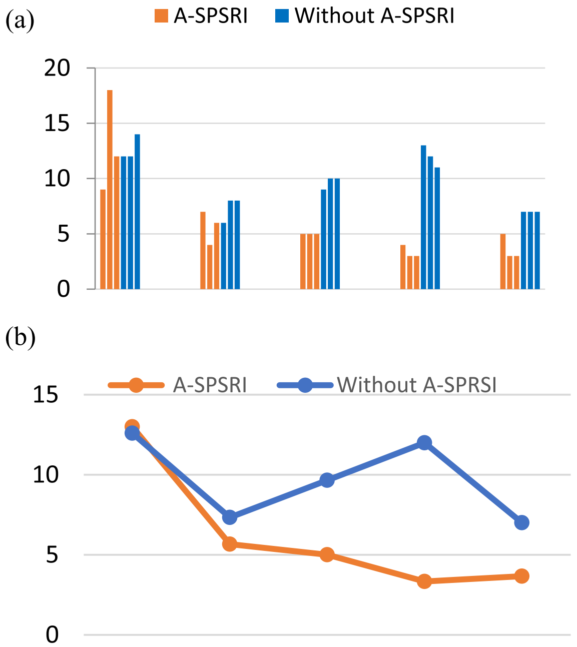


FIGURE 6. (a) Usability Evaluation of A-SPSRI (Each Participant). (b) Usability Evaluation of A-SPSRI (Average time for each iteration).

This research also claims that in terms of time and the number of problems identified, A-SPSRI is able to evaluate scope definitions with lesser variations. Variation in scope evaluation using A-SPSRI and without A-SPSRI can be tested as well by making another hypothesis.

Hypothesis 3: Group evaluating scope statement using A-SPSRI is significantly different than the group not using A-SPSRI.

$$\mathbf{H_0 3} : \sigma_{\mathbf{A-SPSRI}} = \sigma_{\mathbf{without\ A-SPSRI}}$$

$$\mathbf{H_1 3} : \sigma_{\mathbf{A-SPSRI}} \neq \sigma_{\mathbf{without\ A-SPSRI}}$$

σ is standard deviation of the total time taken and the number of problems identified by the people using A-SPSRI and without A-SPSRI during scope evaluation.

The above-mentioned criteria were then evaluated through a formal experiment.

A. FORMAL EXPERIMENT TO ASSESS A-SPSRI

Once the criteria for evaluation were decided, there was a strong need to choose a method for evaluation of scope statements using A-SPSRI. So, formal experiments were chosen and performed on an agile software project. For this experiment, two groups were made in order to evaluate the scope of user stories of an agile project. Group 1 used A-SPSRI for evaluating the scope statements of user stories; however, Group 2 evaluated the scope statements without A-SPSRI. Experiment arrangement details are given below:

- *Selection of Project:* An agile software was taken from a local software house located in Islamabad, Pakistan to evaluate A-SPSRI's usability and utility. Brief description of the project is given below:
  Project - Malisic Marketing Application: This project was a web based "Malisic Marketing Application" for Windows. In scope document, detailed features of the application were stated in the form of user stories including all the hardware, software and technology

TABLE 6. Usability evaluation (unit of time = minutes).

Observations	Group 1- with A-SPSRI	Group 2 – without A-SPSRI
Participant 1-Iteration 1	9	12
Participant 2-Iteration 1	18	12
Participant 3-Iteration 1	12	14
Participant 1-Iteration 2	7	6
Participant 2-Iteration 2	4	8
Participant 3-Iteration 2	6	8
Participant 1-Iteration 3	5	9
Participant 2-Iteration 3	5	10
Participant 3-Iteration 3	5	10
Participant 1-Iteration 4	4	13
Participant 2-Iteration 4	3	12
Participant 3-Iteration 4	3	11
Participant 1-Iteration 5	5	7
Participant 2-Iteration 5	3	7
Participant 3-Iteration 5	3	7
Mean time taken (μ)	*6.1333*	*9.7333*
Standard Deviation (σ)	*4.12*	*2.52*

details. This application was in implementation phase, few of the features were already built

- *Selection of Participants:* Two groups were formed for performing the experiment and each group consisted of three researchers from agile software domain. Group 1 evaluated the scope statements using A-SPSRI while Group 2 evaluated the statements using Planning Poker (PP) approach. The only controlled variable was "Type of Participants".
  Group 1 - with A-SPSRI
  We briefed Group 1 on A-SPSRI and the main objective was to guide participants about the scope elements of A-SPSRI, way to find out elements in the scope document and how to assign definition levels to all the elements. Scope document of agile project, A-SPSRI description, an un-weighted scorecard (to avoid biasness) and definitions of scope elements were provided to the group members in order to calculate scope score using A-SPSRI.
  Group 2- without A-SPSRI
  Scope document of agile project was given and briefed to Group 2. They had to evaluate the quality of scope statement using PP [53], [71]. They had to provide their evaluation in form of comments on the provided sheets.
- *Time Limit:* Both the groups were given an hour to complete the evaluation of scope statements of all the iterations and all the participants were observed individually to confirm that time limit is observed.

1) ASSESSMENT OF A-SPSRI USABILITY

Usability evaluation criterion established for A-SPSRI was that a user who uses A-SPSRI will evaluate the scope definitions of iterations in lesser time than a user who does not use it. Time taken by each participant of both groups was considered to evaluate this criterion. Time taken by each participant was recorded for each iteration of an experiment (Fig. 6 (a)) Average time taken by each participant of both groups are then calculated for this experiment using these individual values (Fig. 6 (b)). Both groups were highly able to perform scope evaluation of each iteration within the stated time limit. It also highlights that Group-1 performed evaluation in considerably less time taken by Group-2.

It is clearly highlighted in Table 6 that mean time taken by Group-1 with A-SPSRI is considerably less than the time taken by Group-2 with PP. Similarly, standard deviation of Group-1 is higher which indicates that it is dependent of the type of scope statement being evaluated and there is very greater chance of variation in evaluation time, whereas lower is the standard deviation of Group-2 which indicates that type of dependency does not exist and there is a less chance of inconsistency in terms of time taken. Mean time and standard deviation of both groups are shown in Table 6. Statistical significance of this consistency is tested in Section 4.2.

2) ASSESSMENT OF A-SPSRI UTILITY

Utility criterion was evaluated by recording the results of experiment for all the participants. Table 6, Table 7 and

TABLE 7. A-SPSRI scores for iterations.

Planning Levels/Elements	Experiment 1									
	User Story 1		User Story 2		User Story 3		User Story 4		User Story 5	
	dl	Scr_{e_i}	*dl*	Scr_{e_i}	*dl*	Scr_{e_i}	*dl*	Scr_{e_i}	*dl*	Scr_{e_i}
PL_1. Product Vision Planning										
e_1. Product Future Statement	2	1.05	1	2.06	1	2.06	1	2.06	1	2.06
e_2. Market Strategy VS Project Strategy	2	4.53	2	4.53	1	8.89	2	4.53	3	2.48
Scr_{PL_1}		***5.58***		***6.59***		***10.95***		***6.59***		***4.54***
PL_2. Product Roadmap										
e_3. Release Definition and Selection	1	26.70	3	7.47	2	13.61	2	13.61	3	7.47
Scr_{PL_2}		***26.70***		***7.47***		***13.61***		***13.61***		***7.47***
PL_3. Product Release Planning										
e_4. Release Plan	2	28.76	2	28.76	2	28.76	2	28.76	3	15.79
Scr_{PL_3}		***28.76***		***28.76***		***28.76***		***28.76***		***15.79***
PL_4. Iteration Planning										
e_5. Iteration Mission	3	8.53	3	8.53	3	8.53	2	15.54	3	8.53
e_6. Resource Estimation	3	17.91	2	32.62	1	63.97	1	63.97	1	63.97
e_7. Iteration Schedule	4	5.48	3	13.97	1	49.90	1	49.90	1	49.90
e_8. Task Identification	4	13.45	4	13.45	4	13.45	3	34.24	2	62.37
e_9. Managing Dependencies	2	11.83	3	6.49	1	23.21	1	23.21	1	23.21
e_{10}. Key Deliverables	3	27.88	4	10.95	4	10.95	4	10.95	4	10.95
e_{11}. Identifying Constraints	3	3.19	2	5.82	1	11.42	1	11.42	3	3.19
e_{12}. Managing Risks/Uncertainties	1	34.62	1	34.62	1	34.62	1	34.62	1	34.62
e_{13}. Adjust Priorities	1	84.43	1	84.43	2	43.05	1	84.43	1	84.43
e_{14}. Review/ Update Release Plan	3	5.58	2	10.17	1	19.96	1	19.96	3	10.17
e_{15}. User/Client Involvement	3	20.45	3	20.45	3	20.45	3	20.45	3	20.45
e_{16}. Building Trust in Team	1	167.76	1	167.76	1	167.76	1	167.76	1	167.76
e_{17}. Manager's Role	3	4.74	2	8.63	2	8.63	3	4.74	1	16.93
e_{18}. Contractual Terms and Conditions	1	4.23	1	4.23	1	4.23	1	4.23	1	4.23
e_{19}. Use of Technology	4	1.54	4	1.54	4	1.54	4	1.54	4	1.54
e_{20}. Client Acceptance and Change Control	1	44.22	1	44.22	1	44.22	3	12.38	2	22.55
e_{21}.Collaborative Development Environment	3	1.82	3	1.82	3	1.82	3	1.82	2	3.31
Scr_{PL_4}		***457.6***		***469.7***		***527.7***		***561.1***		***588.1***
PL_5. Daily Commitment Planning										
e_{22}. Daily Meetings	1	39.17	1	39.17	1	39.17	1	39.17	1	39.17
Scr_{PL_5}		***39.17***		***39.17***		***39.17***		***39.17***		***39.17***
Total Scope Score (T_{scr})		**557.8**		**551.6**		**620.1**		**649.2**		**655.0**

Table 8 shows the evaluation of Group 1 using A-SPSRI and Group-2 without A-SPSRI.

Scope Score (T_{scr}) of all the iterations for the experiment are shown in Table 7. Lower scope score of iteration means better and well defined scope statement. Scope scores of iterations 1,2,3,4 and 5 are 557.8, 551.6, 620.1, 649.2 and 655.0 respectively. So, it can be said that iteration 2 has well defined scope statement so better will be to implement it first. Scope scorecard of A-SPSRI also indicates the problems or reasons behind the desirable and undesirable scores of iterations; this also provides help to practitioners in identifying the areas that need improvement.

Table 8 shows the comments made by Group-2 who evaluated the scope statements without using A-SPSRI. It can be seen from these comments that method only highlights that there are some ambiguities in the features and it provides no information that which iteration is defined better and which will be best to implement first. However, A-SPSRI provides information about the missing items plus the things that need improvement, this method is also able to highlight that which iteration is well defined and better to implement first.

To assess the usability criterion, number of problems identified by both groups is considered using Table 7 and Table 8 which are shown in Table 9 below. It shows the number of problems identified by each participant for each iteration. In case of Group-1 (with A-SPSRI), elements identified as "poorly defined" and with "major deficiencies" are collected and summed up to find out the total number of problems within iteration. However, for Group-2 (without A-SPSRI) number of possible problems is based on the comments given by the participants, few of the comments are shown in Table 8. Group-1 easily highlighted more problems using the definition levels of elements. Here, standard deviation of Group 2 is lower than Group-1 which does not help in the case of problems identified. Higher standard deviation of Group-1 means that participants highlighted more problems in scope statements of each iteration using A-SPSRI. It also indicated that in case of A-SPSRI number of problems identified varies more depending on the quality of user stories than without ASPSRI. Number of problems identified by each participant was recorded for each iteration of an experiment (Figure 8(a)). Mean number of problems by each participant

TABLE 8. Utility evaluation group 2- without A-SPSRI.

	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
Participant 1	*"No clear information provided regarding permissions, plans, email reminders, records of customers."*	*"Not enough information about stock details, buyer profile information."* *"Features are not defined separately."*	*"Ambiguities in details of profile, invoice format, tasks and their period, how these features will help vendors to perform better?"*	*"Planning is a tough job and there are not enough details to support planning activities."* *"Building Online Behavior Tracking system is a complex task and there are no details provided regarding budget and schedule."*	*"Ambiguities in features that need to be developed."* *"No details of company policies, on which system needs to generate reports on regular basis."*
Participant 2	*"Ambiguity in definition of vendor and admin."* *"Flow of information is not fine."* *"Giving permission access to multiple stakeholders at same time will create security breach and data loss."*	*"Status details needs refinement."* *"Payment procedure details are missing."*	*"Tasks are not demonstrating the importance of user story."* *"Activity period is not correctly and properly defined."* *"Stated irrelevant information."* *"Not proper invoices details."*	*"Ambiguous user story with respect to marketing generic case."*	*"Ambiguous in the way that there is no link defined between reporting mechanism and maintenance system, targeted audience is not defined, no details about ads."*
Participant 3	*"It will be difficult for admin to write reports for every task."* *"Access control feature will be difficult to implement completely during this iteration."*	*"Explicitly define what will be maintained in user's profile."* *"Transfer methods should be implemented separately."*	*"Define what information of vendors should be visible to all types of users."* *"Ambiguity in send invoices."*	*"Marketing should not be only visible to authorized users."*	*"Not clearly defined that what is meant by reports about everything?"* *"Marketing reports are required to implement some features in this iteration."*

of both groups are then calculated for this experiment using these individual values (Figure 8(b)).

B. STATISTICAL ANALYSIS AND HYPOTHESIS TESTING USING T-TEST AND MANOVA

For statistical significance of utility and usability criteria for Group-1 (with A-SPSRI) and Group-2 (without A-SPSRI), t-test and MANOVA test has been conducted. Results of these hypothesis tests are discussed in this section.

1) T-TEST ANALYSIS

T-test is performed at 5% level of significance (α) for Hypothesis 1 and Hypothesis 2, results of which are shown in Table 10 (T-Test Table). Hypothesis 1 is tested and it produced a p-value of 0.0013 which is less then α ($p<\alpha$). It means that result is significant and we can reject null hypothesis. Therefore, H_01 is rejected at 5% level of significance. So, it can be said that A-SPSRI allows user to evaluate the scope definition in less time than without A-SPSRI.

Similarly, T-Test is performed for Hypothesis 2 which is related to number of problems identified. It gives a p-value of 0.000080 which are shown in Table 10 below.

2) MANOVA WITH ASSUMPTION TESTING

Hypothesis 3, to test difference in type of evaluation used by Group-1 and Group-2, is tested using MANOVA. For both types of evaluation with A-SPSRI and without A_SPSRI two dependent variables are recorded, time and problems identified. MANOVA creates a linear combination of these two variables and it tests to see if there are any differences between two types of evaluation. It is also used to further identify which dependent variable can explain these differences better. Table 11 shows detailed results of MANOVA, along with its assumptions. Four different assumptions were tested, including: 1) Outliers; 2) Linearity; 3) Normality; 4) Collinearity.

The assumptions are tested using different methods. The data is checked for outliers using regression; the results indicate that there is one outlier in data. This is participant 2 iteration 1 where this participant took 18 minutes to identify 18 problems in evaluation of a user story. Linearity is tested using scatter matrix (see 7). Fig. 7 shows that a linear relationship exists between both the dependent variables for two groups of independent variable. Shapiro Wilk is used to test normality and results indicate that dependent variables have a normal distribution; Collinearity – it is tested using correla-

TABLE 9. Number of problems identified in each iteration.

Observations	Group 1- with A-SPSRI	Group 2 – without A-SPSRI
Participant 1-Iteration 1	11	8
Participant 2-Iteration 1	18	4
Participant 3-Iteration 1	12	4
Participant 1-Iteration 2	13	7
Participant 2-Iteration 2	18	4
Participant 3-Iteration 2	13	3
Participant 1-Iteration 3	16	7
Participant 2-Iteration 3	18	6
Participant 3-Iteration 3	16	4
Participant 1-Iteration 4	15	7
Participant 2-Iteration 4	13	5
Participant 3-Iteration 4	14	3
Participant 1-Iteration 5	13	6
Participant 2-Iteration 5	15	9
Participant 3-Iteration 5	14	4
Mean time taken (μ)	14.6	5.4
Standard Deviation (***σ***)	2.22	1.88

TABLE 10. T-test table.

	Hypothesis 1	Hypothesis 2
Description	A-SPSRI allows user to evaluate the scope definition in less time than without A-SPSRI.	A-SPSRI allows user to become familiar with the project to identify missing items from the scope definition as compared to the user without A-SPSRI.
H_0	$\mu_{A\text{-}SPSRI} \geq \mu_{without\ A\text{-}SPSRI}$	$\mu_{A\text{-}SPSRI} \leq \mu_{without\ A\text{-}SPSRI}$
H_1	$\mu_{A\text{-}SPSRI} < \mu_{without\ A\text{-}SPSRI}$	$\mu_{A\text{-}SPSRI} > \mu_{without\ A\text{-}SPSRI}$
Critical Value		
α	0.05	0.05
df	14	14
$\mu_{A\text{-}SPSRI}$	6.1333	14.6000
$\mu_{without\ A\text{-}SPSRI}$	9.7333	5.4000
t-value	-3.6561	11.5000
P value	0.0013	0.000080
Result	**Significant –** Reject Null Hypothesis	**Significant -** Reject Null Hypothesis

tion, and results indicate that dependent variables do not have collinearity. Among these four assumptions, assumption 1 (i.e. Outliers) failed; so it changed our choice of multivariate test while conducting MANOVA. In multivariate test of MANOVA, Pillai's trace is used instead of Wilk's Lamda. It produced significant results with a p-value of 0.0000, which is much smaller than significance level ($p \ll \alpha$). So, on account of multivariate test of MANOVA, null hypothesis was rejected. It can be said that there is significant difference between Group-1 (A-SPSRI) and Group-2 (without A-SPSRI).

Moreover, to further understand that where this variation lies in the groups Test of Between-Subject Effects is conducted. It shows that evaluation method (A-SPSRI or without A-SPSRI) has significant effect on both dependent variables. Another interpretation is that there is significant different across levels of independent variables for both the dependent variables. This can be further explored to identify the dependent variable which can explain this difference better. Eta Squared value in univariate test results indicate that problems identified can explain 84.2% differences across different groups of independent variable, whereas time taken can only explain 22.9% of this variation.

V. DISCUSSION

A-SPSRI addresses the research questions that we discussed in Section 1. Just to recap:

- RQ-1: How can we quantify the quality and completeness of scope of user stories?

TABLE 11. MANOVA table.

	Hypothesis 3
Description	*There is significant difference in the group using A-SPSRI in comparison to the group without A-SPSRI for the two dependent variables time and problems identified.*
H_o	$\mu_{A\text{-}SPSRI} = \mu_{withoutA\text{-}SPSRI}$
H_1	$\mu_{A\text{-}SPSRI} \neq \mu_{withoutA\text{-}SPSRI}$
α	0.05
	Data assumptions
Independent variable	Type of Evaluation
Groups of independent variable	A-SPSRI, Without A-SPSRI
Dependent variables	Time taken, Problems identified
Sample size (N)	15
	Assumption 1: Test for outliers Regression
Test used	Regression
Mahalanobis Distance	Maximum: 16.271
Critical value	Critical value (Two dependent variables) = 13.86
Result	There are outliers in the data. There is only one observation whose Mahalanobis distance is greater than critical value, so there is one outlier in the data.
	Assumption 2: Test for linearity
Test used	Scatter plot (see Fig. 7)
Result	Linear relationship exits
	Assumption 3: Test for normality
Test used	Shapiro Wilk
Sig. Time taken	0.091
Sig. Problems identified	0.012
Result	Since for Problems Identified Sig. < 0.05 (α), so this dependent variables does not have normal dist.
	Assumption 4: Test for multicollinearity
Test used	Correlation
Pearson (r)	-0.444
Result	0.2 < r < 0.8, we don't have multicollinearity but sufficient correlation between dependent variables
	MANOVA
Levene's test	Sig. (Time_Taken = 0.382), Sig. (Problems_Identified) = 0.569. Meets the assumption
Multivariate test	Pillai's Trace is used because of not meeting the outlier and normality assumption
P value	0.0000
Result	**Significant** - Reject null hypothesis
Test of Between-Subjects Effects	TimeTaken (**Sig. = 0.007**) ProblemsIdentified (**Sig. = 0.000**)
Univariate test	Eta Squared (Time_Taken) = 0.229 Eta Squared (Problems_Identified) = 0.842

In this study, we did a detailed literature review on scope management in agile, existing scope tools and methods that are being used. After this extensive literature review we find out that it is not possible to create a clear and detailed scope statement at start of agile projects, as agile welcomes changes throughout the lifecycle of the project. There exists no tool or method that can measure the quality and completeness of scope definitions of agile projects. We did this type of work before for traditional software projects [30], after taking inspiration from other industries, especially CII.

- RQ-1.1: What is the appropriate information that can be used to quantify the quality and completeness of scope of user stories?

A list of elements is identified as a result of extensive literature review on scope definition in agile. This list of elements is then validated and prioritized through an industrial survey to know their importance in scope definition. As A-SPSRI is designed to measure completeness of scope definition on the basis of absence or presence of identified elements, weights are assigned to elements to know their level of importance. Scoring of planning levels is done to find out the final score of the scope statement of each user story.

- RQ-2: How can the model be constructed which can be useful in defining scope and measuring its completeness?

Construction of A-SPSRI is shown in Section 3. Proposed Method take identified elements as input, then rank these elements based on their priority percentage, assign weights to elements using ROC, define their level of definition in the given scope statement and finally it generates a final scope score card based on the weights and definition levels of the elements of all the agile planning levels.

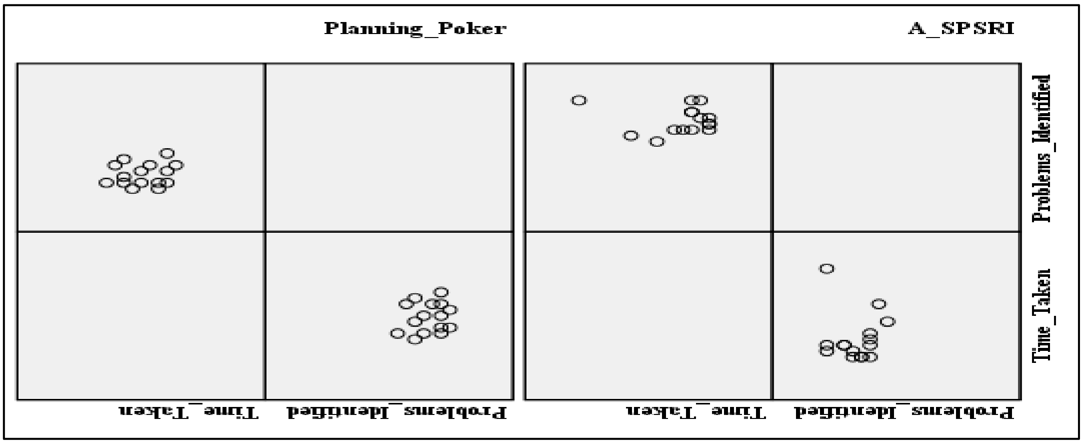


FIGURE 7. Test of linearity – scatter matrix of dependent variables for both group.

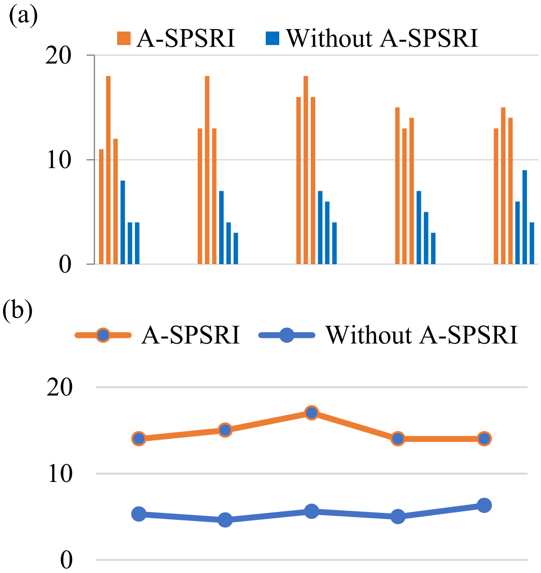


FIGURE 8. Utility Evaluation of A-SPSRI (Problems identified by each Participant) (b) Utility Evaluation of A-SPSRI (Average problems identified in each iteration).

- RQ-2.1: How the quality and completeness of scope can be measured effectively by using such model?

Formal experiment was conducted to evaluate the usability and utility of A-SPSRI. After an experiment, it was observed that there was consistency in results of all the participants who were involved in evaluation. Division of elements among planning levels of agile helped participants in evaluating the user stories on similar characteristics easily. Finally, statistical evaluation was performed and the results showed significant influence of A-SPSRI on evaluation of scope definition.

A. LIMITATIONS

Validation of A-SPSRI is performed using only one software project which is not sufficient to build confidence on the use of the method. It will be beneficial to consider multiple detailed projects for further enhancements. Another limitation of evaluation is that we did not consider the experience of participants that affects the quality of obtained results. So, it can be said that various perceptions can be achieved if we consider a greater number of projects and participants for evaluation.

Subjective view of authors while defining the elements is another limitation of this research. Industrial experts then validated and prioritized the elements considering the views of authors that are subjective in identifying the occurrence of elements in literature.

VI. CONCLUSION

Project success has a relationship with proper planning and completely well-defined scope definition. Most of the failure reasons of agile software projects are due to the poorly or incompletely defined scope definitions. Despite the importance of scope definition, there exists limited literature on agile scope management and the tools that exists only help in controlling, monitoring, estimating, verifying and defining scope. There exists no tool for measuring the quality and completeness of user stories that shows the quality and completeness of scope definition of an iteration. Inflexible nature of agile needs a method that can help developers, managers in order to control the project effectively.

A-SPSRI is presented in this paper; it is a method which is able to measure the quality and completeness of scope definition of all the user stories. Method first identified the critical scope definition elements from literature. These identified elements are then divided into the five levels of agile planning and are then validated and prioritized through a survey. Weights to the elements and their definition levels are then assigned using ROC, then a scorecard is designed which is used in order to evaluate the scope definition of user stories. Finally, a formal experiment is performed on one of the agile project in order to measure the usability and utility of A-SPSRI.

There can be many areas of future work using this concept of A-SPSRI. Considering the structure of A-SPSRI, where elements are divided into agile planning levels will be helpful in defining a scope definition language for agile software projects. This could be possible by defining a schema where all planning level elements can be defined separately.

Evaluation of A-SPSRI highly needs an attention to get more confidence and it can be done by applying it on multiple projects. The results of these evaluations can then be used as a benchmark for a satisfactory scope score of iterations for scaled agile development projects. This effort may also lead to developing more methods for scope management in agile which needs attention.

APPENDIX A
A-SPSRI ELEMENTS' DESCRIPTIONS
A. PRODUCT VISION PLANNING

e_1. *Product Future Statement:* Product future statement describes the idea of a product owner; it explains the future of the product being developed. Product future statement also includes product goal, product features, required effort, resource plus schedule estimates, target audience, competitors, and changes required from previous or competitive products. In addition, it also includes a statement for the expected features of final product, and project team can use this as a guide.

e_2. *Market Strategy VS Project Strategy:* Making a market strategy is responsibility of the product marketers; they express about the profits as they understand the market conditions and product being developed. On the other hand, project strategy is the responsibility of a project manager; he maps the project strategy according to the market strategy.

B. PRODUCT ROADMAP

$e_{3.}$*Release Definition and Selection***:** Release definition and selection is truly a responsibility of the product owner. In brief, it tells when releases are needed, what functionality is required, and details about each release. It further elaborates on the business values obtained from the each release. Product backlog with all the desired features along with their priorities is designed before starting planning for releases.

C. RELEASE PLANNING

e_4. *Release Plan:* Release plan includes details of all the iterations within a release. These details are related to constraints, schedule, budget, technical aspects, business impact and priorities of stakeholders. Product backlog is used as an input for planning the events of a release. This planning has a great impact on time to launch the product in market, gratification of the customer and on the constancy of development process.

Multiple stakeholders are involved in making the release plan, as they prioritize the features based on their needs. Release plan contains details of all type of constraints like risks, budget, technology etc. It states the effort and time required for implementation by determining the list of features for iterations. Release plan is the high level plan and it includes the details about project mission including: high level goals, priorities, budgets, focus areas, assigned team and their mission, product backlog with high level use cases, initial rough estimates, major milestones, number of iterations, and objective of each iteration.

D. ITERATION PLANNING

$e_{5.}$*Iteration Mission:* The Project team identifies the goal of iteration and prioritizes the required features accordingly. Iteration goal describes the milestones needed to accomplish the iteration target. Mission of the iteration is prioritized list of objectives which is understood and agreed by all the involved parties. Mission of iteration is to successfully implement the requirements of customers, which result in producing the new deliverables.

$e_{6.}$*Resource Estimation:* Resource estimation is estimation of tasks/activities in ideal hours. It includes accurate estimates for equipment/tool, man power, funding, facilities etc. Estimates are revised at start of all iterations against the progress made by the project and the lessons learned. Estimates are more reliable for iterations that occur at mature phase of a project, mainly because the project manager has gained experience from previous iterations. Resource estimation at the iteration level involves commitment of the team members working on the features with all the other available resources. All the efforts of the team will contribute to achieving the scope or objectives of iteration. Furthermore, it accounts for all the holidays, training day, and leave of absence etc. when team will not be working.

$e_{7.}$*Iteration Schedule:* Along with the objectives for iteration, the management must agree to its schedule and understand how it impacts the entire project plan. Schedule can be simplified by setting the iteration's start and end dates, and understanding how the iteration fits into the overall project plan.

$e_{8.}$*Task Identification:* Task identification involves appropriate set of user stories to be selected; each user story is decomposed into the set of tasks necessary to deliver the new functionality.

In agile success or failure of a project means success or failure of the entire team. Hence, team members work together on all identified tasks and commits to do everything to achieve objectives of iteration.

Team members on their own take responsibility of certain roles and make a promise towards its achievement. They also try their best to complete the new tasks that are discovered during iteration. Team members share responsibilities of their tasks and collective commitment by all the team members at the start of iteration helps completing their tasks. Team members help each other in order to guarantee that team commitments are encountered and share the collective responsibility in case of failure. Iteration plans covers details of task assignment and responsible parties.

$e_{9.}$*Managing Dependencies:* Dependencies can be of the overall project, which is the responsibility of project manager. In order to determine task dependencies, team members will first determine that what they are going to work on and with whom, secondly they sequence the activities during iteration planning in a way that they are able to find out the dependencies, like any technical dependency in the product backlog. Finally, the product backlog is updated accordingly after informing the customer.

e_{10}.*Key Deliverables:* Deliverables include delivery date of tangible and intangible deliverables. In detail it also includes reports, documents, service, product, work package etc. that are going to be delivered at the end of iteration.

e_{11}. *Identifying Constraints:* Constraints include unexpected absence of the team members. Unexpected absence may result from illness, medical leave, and/or paid time off. It also includes constraints related to fur concerns of square. i.e.: Scope of project, delivery time, quality of project, cost constraints, and availability of stakeholders.

e_{12}. *Managing Risks and Uncertainties:* Iteration plan considers all types of possible risks.Uncertainty comes in multiple forms. There is often uncertainty about exactly what the customer or users need, what the velocity of the team will be, the length of iteration and about technical aspects of the project. Uncertainties or risks can be related to length of iteration due to changing requirements. Uncertainties because of inadequate information about task dependencies, risk chances occur when team members stop working due to unrelated emergency situations or miscalculated required work effort. Risks come from multiple sources and it is the responsibility of the project manager to identify risks, prioritize the risks, to

APPENDIX B: MALISIC MARKETING APPLICATION - SCOPE STATEMENT

Project Title: Malisic Marketing Application

Company Details:

Malisic is divided in two sub-companies to divide expenses:

Malisic MP

Mo-Grading.

A company has a shopping center and has its own 10 retail shops (Dry Cleaner's, Pharmacy, Perfumery, Boutique, Electronics, Kitchenware, Furniture store, restaurant, decoration items and stationary) out of total 30 shops in the center.

Three owners of this company that need to be able to see all information about expenses.

Five store managers of their own retail shops.

Application Objective:
To handle the shopping center in an effective and easy manner.
Platform:
Windows XP, 7, 8.1, 10 and Vista
Tools and Technologies:
PHP, NetBeans IDE or Eclipse
About Application:
Following three login features are already developed for the system, so they are not the part of product backlog.

Company Users (Vendors) should have the option to sign up in order to maintain their sales and purchases easily. This information must contain their full names, email addresses, password, shop number and shop name.

User (Customers) must login to do online shopping through credit cards, in order to save their time.

Admin must have a login in order to maintain the system.

Product Backlog (User Stories):

As a customer, I want following admin features so that I can manage things effectively.

Admin must have profile

Controls everything in the application (Standard admin features)

Gives permissions to vendors, users and other shops in the center to different parts of the application

Insert plans of center and shops (for example: ADD EVENT or ADD ACTIVITY)

Set reminders

Set auto mailing for vendors and users

Add date of marketing activity beginning

Add due date for everything to be prepared before the beginning of marketing activity

Exports data (by vendors, by time, by retail shop, etc.)

Add records of vendors and assign tasks to vendors that immediately get notified that they are responsible for that and they would receive email remainders for period that I set if the task is not marked finished and admin can either approve or reject the suggestions made by vendors or find and assign a different vendor.

Add events of shopping center

Add planned activities for each retail shop

Each activity should have option to insert what is required to prepare that activity, vendors who are assigned for which task.

Write report after task is finished (how it went)

Write good and bad things for each project so that this information can be used in planning similar activities

Add records of customers (Users with valid logins can only purchase things online from the company's shops and can only view plans and advertisements of other shops in the center)

Add audio radio commercial

Add video commercial (YouTube maybe)

Add description for tasks

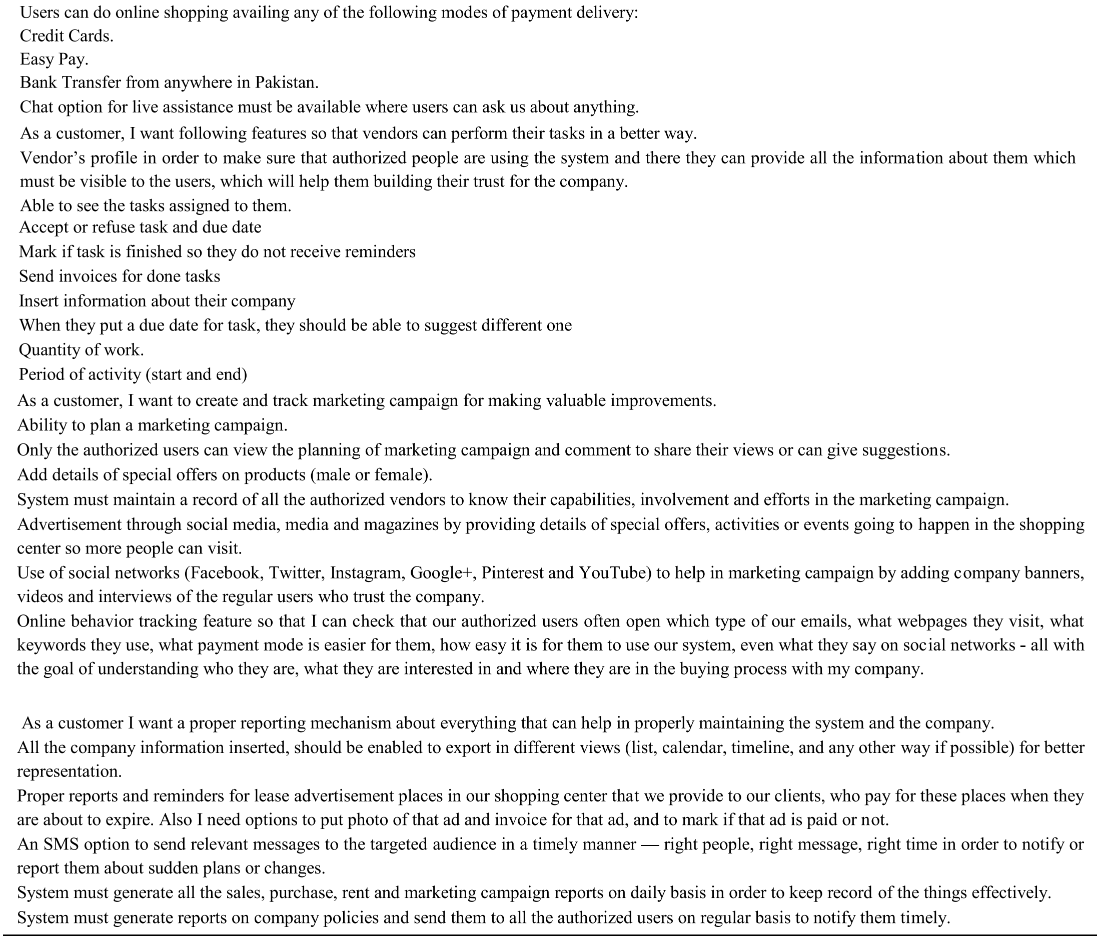
As a customer, I want a shop search option so that users can easily search the shop and view all of its details.

User profile (buyer profile) option, so that authorized users can view all the stock details, can make purchases and maintain their history.

Different shop categories to easily find out the required one.

Ability to view all the stock available in all the categorized shops.

Category wise stock information along with their status (available or not available), exchange or return policy and price details, in order to make purchase easier for users.



Users can do online shopping availing any of the following modes of payment delivery:
Credit Cards.
Easy Pay.
Bank Transfer from anywhere in Pakistan.
Chat option for live assistance must be available where users can ask us about anything.
As a customer, I want following features so that vendors can perform their tasks in a better way.
Vendor's profile in order to make sure that authorized people are using the system and there they can provide all the information about them which must be visible to the users, which will help them building their trust for the company.
Able to see the tasks assigned to them.
Accept or refuse task and due date
Mark if task is finished so they do not receive reminders
Send invoices for done tasks
Insert information about their company
When they put a due date for task, they should be able to suggest different one
Quantity of work.
Period of activity (start and end)
As a customer, I want to create and track marketing campaign for making valuable improvements.
Ability to plan a marketing campaign.
Only the authorized users can view the planning of marketing campaign and comment to share their views or can give suggestions.
Add details of special offers on products (male or female).
System must maintain a record of all the authorized vendors to know their capabilities, involvement and efforts in the marketing campaign.
Advertisement through social media, media and magazines by providing details of special offers, activities or events going to happen in the shopping center so more people can visit.
Use of social networks (Facebook, Twitter, Instagram, Google+, Pinterest and YouTube) to help in marketing campaign by adding company banners, videos and interviews of the regular users who trust the company.
Online behavior tracking feature so that I can check that our authorized users often open which type of our emails, what webpages they visit, what keywords they use, what payment mode is easier for them, how easy it is for them to use our system, even what they say on social networks - all with the goal of understanding who they are, what they are interested in and where they are in the buying process with my company.

As a customer I want a proper reporting mechanism about everything that can help in properly maintaining the system and the company.
All the company information inserted, should be enabled to export in different views (list, calendar, timeline, and any other way if possible) for better representation.
Proper reports and reminders for lease advertisement places in our shopping center that we provide to our clients, who pay for these places when they are about to expire. Also I need options to put photo of that ad and invoice for that ad, and to mark if that ad is paid or not.
An SMS option to send relevant messages to the targeted audience in a timely manner — right people, right message, right time in order to notify or report them about sudden plans or changes.
System must generate all the sales, purchase, rent and marketing campaign reports on daily basis in order to keep record of the things effectively.
System must generate reports on company policies and send them to all the authorized users on regular basis to notify them timely.

do mitigation and contingency planning and finally triggering the events. In agile the common risks include: business risks, technical risks, schedule risks, project management methodology risks, supplier risks, and people risks etc.

e_{14}. *Adjust Priorities:* Iterations are selected from a priority list based on their priority level. Priorities of iterations change with the changing project conditions and business's plan. New functionalities that are added during the iteration based on their priorities should be demonstrated to stakeholders and all the other interested parties to clearly identify the project's goal. Also, the risk associated tasks are considered the top priority.

e_{15}. *Review/Update Release Plan:* Release plan updates throughout the project because of continuous iteration planning at the start of each iteration. Team make decisions and adjustments needed to the scope after acquiring additional information from iterations.

e_{16}. *User/Client Involvement:* After delivering iteration, the client provides feedback in iteration review meeting. Client's feedback is helpful in tracking project's status. Customer representatives are provided with an opportunity to comment on work in progress items and impact the future direction of the development may cause. They also have the power to accept or reject the iteration deliverables. Feedback on regular basis is helpful than setting expectations. It reminds the development team of its responsibility to ensure the new technology is providing benefits to the business and its operations. Client is the content expert and no project can be successful without the involvement of the client. Client is closely involved and providing us quick feedback on every feature which results in increased flexibility of making required changes.

e_{17}.*Building trust in Team*: To build trust among team members it is important to provide collaborative environment, rights, and equal participation during the project. Demonstration of working at end of iteration, provoking the issues that are problematic for the abilities of team members, collaboration and an environment of sharing experiences are the activities that build trust among team member's its users and sponsors. It is the responsibility of a project manager to make sure all the team members that they are allowed to do whatever they think is beneficial for the success of the project.

e_{18}.*Managers Role:* Managers in agile are able to perform different roles. He/she is responsible for all the development activities, management of the product, ensuring that the project is being developed under proper monitoring and control. Agile divides managers into two main groups: Project managers, who are responsible for scheduling, budgeting, priorities etc. and line managers, who are responsible for setting the company's strategy, businesses processes, and marketing goals etc. Results of the iteration are assessed and tracked against the iteration schedule to review the success or failure of iteration.

e_{18}. *Contractual Terms and Conditions*: Contractual team identifies items that need procurement. Vendors are involved throughout the iteration and provide all required functionalities; they also try to remove weaknesses using these functionalities. Contracts and negotiation are critical to business relationships. So agile sets forth the idea that a buyer and a seller should work together to create products within certain conditions. Relationship between them is also important for agile to provide valuable system to client.

e_{19}. *Use of Technology:* Scope creep problem from business user's side can lead to a technology creep problem for developers. Unavailability of a newer technology to handle a change request can cause problems in achieving the scope of project within time and budget. With the need of a new technology all the plans, estimates, schedule need to be reevaluated and a new model or prototype need to be designed in order to achieve the changed scope of an iteration.

e_{20}. *Client Acceptance and Change Control*: After each iteration, client acceptance is required which works as the input to define and update the scope of next iteration. Change request from the client is recorded along with its impact, and then incorporated in the reprioritization and planning process which ultimately results in re-planning the scope at start of each iteration. In Agile, changes can be recognized earlier and included with earlier iterations to get development risks out of the way.

e_{21}. *Collaborative Development Environment:* The checks and balance and the collaboration inherent in the iterative approach provide the mechanisms for achieving concurrence among the project's stakeholders, which increases the chances of success. It helps members in learning from experience gained during iteration and enables them to react appropriately to any changes.

E. DAILY COMMITMENT PLANNING

e_{22}.*Daily Meetings:* A very short period session in which all the team members share their plans, achievements, hurdles and give suggestions for making the project better. In the meeting, all types of issues are elevated and then resolved. Meetings are helpful in defining the success rate of delivering the features. Daily meetings are a 15-minute activity that is performed to fully inspect the status of the project. It is helpful in making the project flow more flexible and faster. Team members can take advantage of this in the form of early feedback, as all of them share their work and cooperatively knows what to do more, what not to do and what to improve.

REFERENCES

[1] J. Moustafaev, *Project Scope Management: A Practical Guide to Requirements for Engineering, Product, Construction, IT and Enterprise Projects*. Boca Raton, FL, USA: CRC Press, 2014.

[2] J. Hallows, *Information Systems Project Management: How to Deliver Function and Value in Information Technology Projects*. New York, NY, USA: American Management Association, 2015.

[3] K. Schwalbe, *Information Technology Project Management*. Boston, MA, USA: Cengage Learn., 2015.

[4] E. Bingham, "Development of the project definition rating index (PDRI) for infrastructure projects," Ph.D. dissertation, Arizona State Univ., Phoenix, AZ, USA, 2010.

[5] P. R. Dumont, G. E. Gibson, Jr., and J. R. Fish, "Scope management using project definition rating index," *J. Manag. Eng.*, vol. 13, no. 5, pp. 54–60, 1997.

[6] J. M. McQuighan and R. J. Hammell, II, "Scope as a leading indicator for managing software development," in *Proc. 9th Int. Conf. Softw. Eng. Res., Manage. Appl. (SERA)*, 2011, pp. 235–241.

[7] M. N. Mirza, Z. Pourzolfaghar, and M. Shahnazari, "Significance of scope in project success," *Proc. Technol.*, vol. 9, pp. 722–729, Jan. 2013.

[8] M. C. Layton, *Agile Project Management for Dummies*. Hoboken, NJ, USA: Wiley, 2012.

[9] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review," *J. Syst. Softw.*, vol. 119, pp. 87–108, Sep. 2016.

[10] M. Sliger and S. Broderick, *The Software Project Manager's Bridge to Agility*. Reading, MA, USA: Addison-Wesley, 2008.

[11] I. Inayat, S. S. Salwah, S. Marczak, M. Daneva, and S. Shamshirband, "A systematic literature review on agile requirements engineering practices and challenges," *Comput. Human Behav.*, vol. 51, pp. 915–929, Oct. 2015.

[12] E. Bjarnason, K. Wnuk, and B. Regnell, "Are you biting off more than you can chew? A case study on causes and effects of overscoping in large-scale software engineering," *Inf. Softw. Technol.*, vol. 54, no. 10, pp. 1107–1124, 2012.

[13] M. Tomanek and J. Juricek, "Addressing main problems in project management (PRINCE2) by adopting agile principles (Scrum)," in *Proc. IADIS*, vol. 5. 2012, pp. 385–389.

[14] A. J. Sohi, M. Hertogh, M. Bosch-Rekveldt, and R. Blom, "Does lean & agile project management help coping with project complexity?" *Proc.-Soc. Behav. Sci.*, vol. 226, pp. 252–259, Jul. 2016.

[15] M. Špundak, "Mixed agile/traditional project management methodology—Reality or illusion?" *Proc.-Soc. Behav. Sci.*, vol. 119, pp. 939–948, Mar. 2014.

[16] A. L'ecuyer and S. A. Ahmed, "Controlling change on agile software development projects," *Univ. J. Manag.*, vol. 4, no. 1, pp. 42–49, 2016.

[17] M. Drury, K. Conboy, and K. Power, "Obstacles to decision making in agile software development teams," *J. Syst. Softw.*, vol. 85, no. 6, pp. 1239–1254, 2012.

[18] T. Suomalainen, R. Kuusela, and M. Tihinen, "Continuous planning: An important aspect of agile and lean development," *Int. J. Agile Syst. Manage.*, vol. 8, no. 2, pp. 132–162, 2015.

[19] M. Talluri and H. M. Haddad, "Best managerial practices in agile development," *Proc. ACM Southeast Reg. Conf.*, 2014, pp. 3:1–3:5.

[20] H. Oktaba, *Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies: Techniques and Case Studies*. Hershey, PA, USA: IGI Global, 2008.

[21] N. Turbit, "Defining the scope of a project," White Paper, 2005.

[22] S. E. Portny, *Project Management for Dummies*, 4th ed. Hoboken, NJ, USA: Wiley, 2013.

[23] *Creating a Project Scope Statement*. Accessed: Feb. 2, 2016. [Online]. Available: http://www.free-management-ebooks.com/faqpm/scope-05.htm

[24] J. Cho, "Issues and Challenges of agile software development with SCRUM," *Issues Inf. Syst.*, vol. 9, no. 2, pp. 188–195, 2008.

[25] L. Vijayasarathy and D. Turk, "Agile software development: A survey of early adopters," *J. Inf. Technol. Manag.*, vol. 19, no. 2, pp. 1–8, 2008.

[26] K. Wongsaroj. (2012). *Visualizing the Big Picture of Your Agile Project*. Accessed: Feb. 25, 2016. [Online]. Available: https://www.infoq.com/articles/visualize-big-picture-agile

[27] (2010). *U.S. Department of Energy Project Definition Rating Index Guide for Traditional Nuclear and Non-Nuclear Construction Projects—DOE Directives, Delegations, and Requirements*. Accessed: Mar. 10, 2016. [Online]. Available: https://www.directives.doe.gov/directives-documents/400-series/0413.3-EGuide-12-admchg1

[28] S. J. Lotz, "Generalized application of the project definition rating index," Tech. Rep., 2015.

[29] *Em. Pdri*, E. M. P. M. Anual, 2001.

[30] I. Hassan, N. Ahmad, and B. Zuhaira, "Calculating completeness of software project scope definition," *Inf. Softw. Technol.*, vol. 94, pp. 208–233, Apr. 2017, doi: https://doi.org/10.1016/j.infsof.2017.10.010

[31] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Inf. Softw. Technol.*, vol. 50, no. 9, pp. 833–859, 2008.

[32] R. Green, T. Mazzuchi, and S. Sarkani, "Communication and quality in distributed agile development: An empirical case study," *Proc. World Acad. Sci. Eng. Technol.*, vol. 61, pp. 322–328, Jan. 2010.

[33] A. Cline, *Agile Development in the Real World*. Springer, 2015.

[34] D. F. Rico, H. H. Sayani, and S. Sone, *The Business Value of Agile Software Methods: Maximizing ROI With Just-in-Time Processes and Documentation*. Washington, DC, USA: J. Ross Publishing, 2009.

[35] S. Nerur, R. Mahapatra, and G. Mangalaraj, "Challenges of migrating to agile methodologies," *Commun. ACM*, vol. 48, no. 5, pp. 72–78, 2005.

[36] B. Boehm and R. Turner, "Management challenges to implementing agile processes in traditional development organizations," *IEEE Softw.*, vol. 22, no. 5, pp. 30–39, Sep. 2005.

[37] P. Naur and B. Randell, "Software engineering: Report of a conference sponsored by the NATO science committee," in *Proc. NATO*, Garmisch, Germany, Oct. 1968, pp. 7–11.

[38] J. J. Rooney and L. N. Vanden Heuvel, "Root cause analysis for beginners," *Qual. Prog.*, vol. 37, no. 7, pp. 45–56, 2004.

[39] 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," *J. Syst. Softw.*, vol. 86, no. 6, pp. 1663–1678, 2013.

[40] T. Chow and D.-B. Cao, "A survey study of critical success factors in agile software projects," *J. Syst. Softw.*, vol. 81, no. 6, pp. 961–971, 2008.

[41] B. Unhelkar, *The Art of Agile Practice: A Composite Approach for Projects and Organizations*. Boca Raton, FL, USA: CRC Press, 2016.

[42] G. P. Prabakhar, "What is project success: A literature review," *Int. J. Bus. Manag.*, vol. 3, pp. 3–10, Mar. 2008.

[43] R. Kaur and J. Sengupta. (2013). "Software process models and analysis on failure of software development projects." [Online]. Available: https://arxiv.org/abs/1306.1068

[44] H. Smits, "The impact of scaling on planning activities in an agile software development center," in *Proc. 40th Annu. Hawaii Int. Conf. Syst. Sci. (HICSS)*, 2007, pp. 274c–274c.

[45] J. Schiel, *The Scrum Master Study Guide*. Boca Raton, FL, USA: CRC Press, 2011.

[46] S. W. Ambler and M. Lines, *Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise*. Indianapolis, IN, USA: IBM Press, 2012.

[47] S. Ambler, *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process*. Hoboken, NJ, USA; Wiley, 2002.

[48] B. Councill and G. T. Heineman, "Definition of a software component and its elements," in *Component-Based Software Engineering: Putting the Pieces Together*. Boston, MA, USA: Addison-Wesley, Jun. 2001, pp. 5–19.

[49] C. B. Seaman, "Qualitative methods in empirical studies of software engineering," *IEEE Trans. Softw. Eng.*, vol. 25, no. 4, pp. 557–572, Jul. 1999.

[50] J. Patton and P. Economy, *User Story Mapping: Discover the Whole Story, Build the Right Product*. Sebastopol, CA, USA: O'Reilly Media, 2014.

[51] J. Gregory and L. Crispin, *More Agile Testing: Learning Journeys for the Whole Team*. Reading, MA, USA: Addison-Wesley, 2014.

[52] B.-A. K. Keote, "Function points and agile—Hand in hand," Tech. Rep., 2010.

[53] S. W. Munialo and G. M. Muketha, "A Review of agile software effort estimation methods," *Int. J. Comput. Appl. Technol. Res.*, vol. 5, no. 9, pp. 612–618, 2016.

[54] A. Ren and C. Yun, "Research of software size estimation method," in *Proc. Int. Conf. Cloud Serv. Comput. (CSC)*, Nov. 2013, pp. 154–155.

[55] T. Sulaiman, B. Barton, and T. Blackburn, "AgileEVM—Earned value management in scrum projects," in *Proc. AGILE*, 2006, p. 10.

[56] S.-C. Wu, "Traditional and agile earned value management processes," to be published.

[57] T. Javdani, H. Zulzalil, A. A. A. Ghani, A. B. M. Sultan, and R. M. Parizi. (Jan. 2013). "On the current measurement practices in agile software development." [Online]. Available: https://arxiv.org/abs/1301.5964

[58] M. Cohn, *Agile Estimating and Planning*. London, U.K.: Pearson Education, 2005.

[59] M. A. Boschetti, M. Golfarelli, S. Rizzi, and E. Turricchia, "A Lagrangian heuristic for sprint planning in agile software development," *Comput. Oper. Res.*, vol. 43, pp. 116–128, Mar. 2014.

[60] S. Dimitrijević, J. Jovanović, and V. Devedžić, "A comparative study of software tools for user story management," *Inf. Softw. Technol.*, vol. 57, no. 1, pp. 352–368, 2015.

[61] (2015). *Agile Process Work Item Types and Workflow*. Accessed: Mar. 15, 2016. [Online]. Available: https://www.visualstudio.com/en-us/docs/work/guidance/agile-process-workflow

[62] R. Dillibabu and K. Krishnaiah, "Cost estimation of a software product using COCOMO II. 2000 model—A case study," *Int. J. Project Manage.*, vol. 23, no. 4, pp. 297–307, 2005.

[63] R. Litoriya and A. Kothari, "An efficient approach for agile web based project estimation: AgileMOW," *J. Softw. Eng. Appl.*, vol. 6, pp. 297–303, Jun. 2013.

[64] R. Goswami, "Impact of different estimation approaches in traditional and agile development," in *Proc. 2nd Int. Conf. Comput. Intell. Commun. Technol. (CICT)* 2016, pp. 679–682.

[65] R. T. Hans, "Work breakdown structure: A tool for software project scope verification," *Int. J. Softw. Eng. Appl.*, vol. 4, no. 4, pp. 19–25, 2013.

[66] P. Sanghera, *Fundamentals of Effective Program Management: A Process Approach Based on the Global Standard*. John Ross, 2008.

[67] A. B. Badiru, *STEP Project Management: Guide for Science, Technology, and Engineering Projects*. Boca Raton, FL, USA: CRC Press, 2009.

[68] M. Bundschuh and C. Dekkers, "Functional size measurement methods (FSMMs)," in *The IT Measurement Compendium: Estimating and Benchmarking Success With Functional Size Measurement*. Springer, 2008, pp. 365–395.

[69] T. Gautam, "Software measurement metrics in project scope management," M.S. thesis, Univ. Tampere, Tampere, Finland, 2014.

[70] M. Ikonen, "Lean thinking in software development: Impacts of kanban on projects," Univ. Helsinki, Helsinki, Finland, Tech. Rep., 2011.

[71] F. Calefato and F. Lanubile, "A planning poker tool for supporting collaborative estimation in distributed agile development," in *Proc. 6th Int. Conf. Softw. Eng. Adv. (ICSEA)*, 2011, pp. 14–19.

[72] N. C. Haugen, "An empirical study of using planning poker for user story estimation," *Proc. Agile Conf.*, Jul. 2006, pp. 9–34.

[73] N. Dalkey, *The Delphi Method: An Experimental Study of Group Opinion*, vol. 3. Santa Monica, CA, USA: Rand Corporation, 1969.

[74] W. Edwards and F. H. Barron, "SMARTS and SMARTER: Improved simple methods for multiattribute utility measurement," *Organ. Behav. Hum. Decision Process.*, vol. 60, no. 3, pp. 306–325, 1994.

[75] "Procedures for determining the weights of selection factors in the weighted-matrix delivery decision approach (tier 2)," Tech. Rep.

[76] Content and Discourse Analysis. (2009). *Content and Discourse Analysis*, Accessed: Mar. 7, 2016. [Online]. Available: http://admn502awiki.pbworks.com/w/page/10041942/

[77] O. R. Holsti, *Content Analysis for the Social Sciences and Humanities*. Reading, MA, USA: Addison-Wesley, 1969.

[78] J. L. G. Benassi, F. Junior, L. Dias, and D. C. Amaral, "Evaluating methods for product vision with customers involvement to support agile project management," in *Proc. 18th Int. Conf. Eng. Design (ICED)*, vol. 10. 2011, pp. 1–10.

[79] R. Pichler, *Agile Product Management With Scrum: Creating Products That Customers Love (Adobe Reader)*. Reading, MA, USA: Addison-Wesley, 2010.

[80] G. Meszaros, "Using storyotypes to split bloated XP stories," in *Proc. Conf. Extreme Programm. Agile Methods*, 2004, pp. 73–80.

[81] M. Block, "Evolving to Agile a story of agile adoption at a small SaaS company," in *Proc. Agile Conf.*, 2011, pp. 234–239.

[82] D. Greer and G. Ruhe, "Software release planning: An evolutionary and iterative approach," *Inf. Softw. Technol.*, vol. 46, no. 4, pp. 243–253, 2004.

[83] J. Momoh and G. Ruhe, "Release planning process improvement—An industrial case study," *Softw. Process, Improvement Practice*, vol. 11, no. 3, pp. 295–307, 2006.

[84] G. Ruhe, "Software release planning," in *Handbook of Software Engineering and Knowledge Engineering*, vol. 3. Singapore: World Scientific, 2005, pp. 365–394.

[85] T. Stober and U. Hansmann, *Best Practices for Large Software Development Projects*. Berlin, Germany: Springer, 2010.

[86] K. Bittner and I. Spence, *Managing Iterative Software Development Projects*. Reading, MA, USA: Addison-Wesley, 2006.

[87] J. Chen, J. Xiao, Q. Wang, L. J. Osterweil, and M. Li, "Refactoring planning and practice in agile software development: an empirical study," in *Proc. Int. Conf. Softw. Syst. Process (ICSSP)*, 2014, pp. 55–64.

[88] O. Lewitz, "LNCS 3092—Story management," in *Proc. Int. Conf. Extreme Programm. Agile Process. Softw. Eng.*, 2004, pp. 278–281.

[89] O. Ktata, "Designing and implementing a measurement program for Scrum teams: What do agile developers really need and want?" in *Proc. 3rd Conf. Comput. Sci. Softw. Eng.*, 2010, pp. 101–107.

[90] S. Kang, "Model-based dynamic cost estimation and tracking method for agile software development," in *Proc. 9th IEEE/ACIS Int. Conf. Comput. Inf. Sci.*, Aug. 2010, pp. 743–748.

[91] M. M. Hohman, "Estimating in actual time [extreme programming]," in *Proc. Agile Develop. Conf. (ADC)*, Jul. 2005, pp. 132–138.

[92] T. Franklin, "Adventures in agile contracting: Evolving from time and materials to fixed price, fixed scope contracts," in *Proc. Agile Conf.*, 2008, pp. 269–273.

[93] A. Moran, *Managing Agile: Strategy, Implementation, Organisation, People*. Springer, Mar. 2015.

[94] G. Chin, *Agile Project Management: How to Succeed in the Face of Changing Project Requirements*. AMACOM Div American Mgmt Assn, Jan. 2004.

[95] H. Baumeister *et al.*, "Agile processes in software engineering and extreme programming," in *Proc. 14th Int. Conf. XP*, 2013.

[96] M. Tomanek and J. Juricek, "Project risk management model based on prince2 and scrum frameworks," *Int. J. Softw. Eng. Appl.*, vol. 6, no. 1, Jan. 2015.

[97] R. Morien, "Agile management and the Toyota way for software project management," in *Proc. 3rd IEEE Int. Conf. Ind. Informat. (INDIN)*, Aug. 2005, pp. 516–522.

[98] A. Albadarneh, I. Albadarneh, and A. Qusef, "Risk management in Agile software development: A comparative study," in *Proc. IEEE Jordan Conf. Appl. Elect. Eng. Comput. Technol. (AEECT)*, 2015, pp. 1–6.

[99] M. Golfarelli, S. Rizzi, and E. Turricchia, "Sprint planning optimization in agile data warehouse design," in *Proc. 14th Int. Conf. Data Warehousing Knowl. Discovery*, 2012, pp. 30–41.

[100] R. K. Wysocki, *Adaptive Project Framework: Managing Complexity in the Face of Uncertainty*. Reading, MA, USA: Addison-Wesley, 2010.

[101] K. Johansen and A. Perkins, "Establishing an agile testing team: Our four favorite 'mistakes,'" in *Proc. Conf. Extreme Programm. Agile Methods*, 2002, pp. 52–59.

[102] R. Mantovani, I. Mantovani, P. Andrea, S. Reinehr, and A. Malucelli, "Processes versus people: How should agile software development maturity be defined?" *J. Syst. Softw.*, vol. 97, pp. 140–155, Nov. 2014.

[103] W. Ketter, R. Guikers, B. V. Jibes, A. Kayser, and B. V. Jibes, "Introducing an agile method for enterprise mash-up component development," in *Proc. Commerce Enterprise Comput.*, 2009, pp. 293–300.

[104] V. K. Vankayala, "Developing realtime business intelligence systems the agile way," in *Proc. Syst. Conf.*, 2007, pp. 1–7.

[105] N. Jastroch, V. Kirova, C. S. Ku, T. J. Marlowe, and M. Mohtashami, "Adapting business and technical processes for collaborative software development," in *Proc. 17th Concurrent Enterprising (ICE)*, 2011, pp. 1–8.

[106] J. Mcavoy, D. Sammon, I. Owens, J. Mcavoy, D. Sammon, and I. Owens, "A simple tool to assist in agile methodology adoption decisions a simple tool to assist in agile methodology adoption decisions," *J. Decision Syst.*, vol. 16, no. 4, pp. 451–468, 2017.

[107] M. Coram and S. Bohner, "The impact of agile methods on software project management a brief look at agile methods," in *Proc. 12th IEEE Int. Conf. Workshops Eng. Comput.-Based Syst. (ECBS)*, Apr. 2005, pp. 363–370.

[108] A. Moore and W. T. Flannery, "Use of extreme programming methodologies in IT application design processes: An empirical analysis," in *Proc. PICMET*, 2007, pp. 5–9.

[109] T. J. Lehman, "Software development as a service: Agile experiences," in *Proc. Annu. SRII Global Conf.*, 2011, pp. 749–758.

[110] J. A. Crowder and S. Friess, *Agile Project Management: Managing for Success*. Cham, Switzerland: Springer, 2015.

[111] J. F. Tripp, C. Riemenschneider, and J. B. Thatcher, "Job satisfaction in agile development teams: Agile development as work redesign," *Assoc. Inf. Syst.*, vol. 17, no. 4, pp. 267–307, 2016.

[112] C. T. Schmidt, S. A. P. Ag, J. Tripp, T. Kude, A. Heinzl, and K. Spohrer, "Team adaptability in agile information system development," in *Proc. 34th Int. Conf. Inf. Syst.*, 2013, pp. 1–11.

[113] D. Turk, R. France, and B. Rumpe. (2014). "Limitations of agile software processes." [Online]. Available: https://arxiv.org/abs/1409.6600

[114] D. J. Anderson, *Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2003.

[115] Á. Medinilla, *Agile Management: Leadership in an Agile Environment*. Berlin, Germany: Springer-Verlag, 2012.

[116] A. Martin, R. Biddle, and J. Noble, "When XP met outsourcing," in *Proc. Int. Conf. Extreme Programm. Agile Processes Softw. Eng.*, Berlin, Germany, 2004, pp. 51–59.

[117] K. Laakkonen, *Contracts in Agile Software Development*. 2014.

[118] E. Pilios, *Contracting Practices in Traditional and Agile Software Development*. 2015.

[119] D. Jamieson, K. Vinsen, and G. Callender, "Agile procurement and dynamic value for money to facilitate agile software projects," in *Proc. 32nd EUROMICRO Conf. Softw. Eng. Adv. Appl. (EUROMICRO-SEAA)*, 2006, pp. 248–255.

[120] A. Q. Gill, B. Henderson-sellers, and M. Niazi, "Scaling for agility: A reference model for hybrid traditional-agile software development methodologies," in *Information Systems Frontiers*. 2016.

[121] J. Wan and R. Wang, "Empirical research on critical success factors of agile software process improvement," *Softw. Eng. Appl.*, vol. 12, no. 3, pp. 1131–1140, 2010.

[122] M. Seymour, S. Coyle, and S. Coyle, "Towards a research agenda for adopting agile project management in creative industries towards a research agenda for adopting agile project," in *Proc. 11th Int. Res. Workshop Inf. Technol. Project Manag. (IRWITPM)*, 2016, pp. 166–175.

[123] S. D. Based, "Requirements management process model for software development based on legacy system functionalities," in *Proc. 9th Int. Conf. Telecommun.*, 2007, pp. 115–122.

[124] A. Fruhling, P. Mcdonald, and C. Dunbar, "A case study: Introducing eXtreme programming in a US government system development project," in *Proc. 41st Hawaii Int. Conf. Syst. Sci.*, 2008, p. 464.

[125] K. S. Rubin, *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Reading, MA, USA: Addison-Wesley, 2012.

[126] H. Newing, *Conducting Research in Conservation: Social Science Methods and Practice*. Evanston, IL, USA: Routledge, 2010.

[127] G. Grela, "Does weighted average really work." in *Proc. Zadar, Croatia Int. Conf.*, 2013, pp. 1219–1227.

[128] J. Noppen, "Fuzzy weighted average: Alternative approach," in *Proc. Annu. Meet. North Amer. Fuzzy Inf. Process. Soc. (NAFIPS)*, 2007, pp. 1–5.

[129] F. H. Barron and B. E. Barrett, "Decision quality using ranked attribute weights," *Manage. Sci.*, vol. 42, no. 11, pp. 1515–1523, 1996.

SUNNIA AMJAD received the M.S. degree in computer science from the COMSATS Institute of Information Technology, Pakistan, in 2017. Her research interests include agile software development, in particular, managing scope in agile projects.

NAVEED AHMAD received the doctoral degree in engineering design from the University of Cambridge, Cambridge, U.K., and joined the COMSATS Institute of Information Technology in 2011, where he now serves as an Assistant Professor in the Computer Science Department. He is a member of research groups on software engineering and mobile application development at CIIT. His research interests include change management, process management, enterprise information systems in particular enterprise resource planning software, and mobile application development.

TANZILA SABA received the Ph.D. degree in document information security and information management from the Faculty of Computing, Universiti Teknologi Malaysia (UTM), Malaysia in 2012. Her research interests include intelligent data mining, forensic documents analysis and information security. She won the 2012 Best Student Award in the Faculty. Currently, she serves as an Assistant Professor in the College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia. Due to her excellent research achievement, she is included in Marquis Who's Who (S & T) 2012.

ADEEL ANJUM received the Ph.D. degree with distinction in 2013. He is an Assistant Professor in the Department of Computer Sciences, CIIT Islamabad Campus. His area of research is data privacy using artificial intelligence techniques. He has several publications in international conferences. He also authored a book on data privacy. He serves in the technical program committees of various international conferences.

UMAR MANZOOR is currently working as a PostDoc Researcher at the Computer Science Department, Tulane Univeristy, USA. He received the B.S. and M.S. degrees in computer science from the National University of Computer and Emerging Sciences, Pakistan, and the Ph.D. degree in multi-agent systems from the University of Salford, Manchester, U.K., in 2003, 2005, and 2011, respectively. He has a strong interest in machine learning, natural language processing, multi-agent systems, artificial intelligence, and has published extensively in these areas.

MUHAMMAD A. BALUBAID received the B.S. and M.S. degrees in manufacturing engineering from the University of Warwick, U.K. and the Ph.D. degree in operations management and logistics from the University of Manchester, Manchester, U.K.. He is currently working as an Associate Professor at the Faculty of Engineering (Industrial Engineering Department), King Abdulaziz University, Jeddah, Saudi Arabia.

SAIF UR REHMAN MALIK received the Ph.D. degree from North Dakota State University, USA. He is currently working as an Assistant Professor in the COMSATS Institute of Information Technology, Islamabad, Pakistan. His research interests include formal verification analysis and modeling, cyber physical systems, and large scale computing systems.