

Research article

A software-based cost estimation technique in scrum using a developer's expertise

Shariq Aziz Butt^{a,*}, Ayesha Khalid^a, Tuncay Ercan^b, Paola Patricia Ariza-Colpas^c,
Acosta-Coll Melisa^c, Gabriel Piñeres-Espitia^c, Emiro De-La-Hoz-Franco^c,
Marlon Alberto Pineres Melo^d, Roberto Morales Ortega^c

^a Department of Computer Science and Information Technology, The University of Lahore, Pakistan

^b Department of Management Information Systems, Yasar University, Turkey

^c Universidad de la Costa, Colombia

^d Universidad del Norte, Colombia

ARTICLE INFO

Keywords:

Scrum projects
Cost estimation techniques
Issues in the scrum

ABSTRACT

Background: Scrum methodology is the most useful and adopted part of the agile methodology. Many organizations are adopting Scrum for the last decade to meet their software requirements. It is facilitating the software industries to develop software applications according to the user's requirements.

Objective: Scrum helps to fulfill the customer's requirements by interacting with the client and developer. Instead of this, scrum still has some challenges that influence the cost of a scrum. These challenging factors include coordination, team size, change request, complexity, and issues in daily meeting sessions. Due to these factors mostly the cost and time exceed scrum-based projects. To overcome these issues from the accurate and precise estimation in the scrum project we have developed a web-based software system to estimate the effort of a user story as well as the budget.

Methodology: There are some sophisticated cost estimation techniques for scrum-based projects. But all estimation techniques have some limitations that make them less useful for estimation and overcoming the effects of these factors in the scrum. We have surveyed the multiple software industries to get the static findings of the survey. We have designed the 30 hypotheses that main aim to reveal how much the proposed estimation technique is valuable for the accuracy of estimation. For the inclusion of participants, we have set the age criteria to get the opinions of the most experienced programmers.

Results: The main aim of this paper is to provide knowledge about key challenging factors in Scrum that cause exceed in cost and time, and also briefly explain why existing scrum cost estimation techniques are not pertinent for projects. In the last, we propose a framework as a solution to manage these factors and control cost and time from exceeding. We also adopted this estimation technique in different case studies in software industries for findings and results. The technique calculates the efforts autonomously. We design a software-based estimation technique that collects inputs from team/s and estimates the cost and time of a project.

Conclusion: In outcomes, we found that our cost estimation approach reduced the issues from scrum software project development and make it more proficient for estimation of projects. Our proposed estimation technique introduced the new concept of estimation that facilitate the client, software industry, and developers as well to meet the requirements of the client, fulfill the changes during the sprints/s development and deliver the project within the time and cost.

1. Introduction

Software Project Management is a combination of different processes like planning, organizing, controlling, and interaction between

members. Agile software methodology is widely used in software industries due to its flexible and dynamic nature. In agile methodology, the project is developed in different iterations and there is a quick response to changes. The main methods of agile are XP, Scrum, DSDM,

* Corresponding author.

E-mail address: Shariq2315@gmail.com (S.A. Butt).

<https://doi.org/10.1016/j.advengsoft.2022.103159>

Received 10 April 2022; Received in revised form 23 May 2022; Accepted 26 May 2022

Available online 17 June 2022

0965-9978/© 2022 Elsevier Ltd. All rights reserved.

FDD, and Crystal. But the Scrum is the most important and trendy component of agile methodology in industry and research. Scrum software development process is well known for its lightweight process with fixed series of iterations. Scrum characteristics are team self-determination, self-organized team, quick response to changes, flexibility in nature, promote interaction between team members, work in sprints and daily meeting sessions. Scrum has been found very effective and efficient with small-scale and co-located teams. Scrum plays a vital role by helping the executives to get business results as they want.

Nowadays Scrum become more significant due to its flexibility and rich interaction. Adaptation of Scrum is difficult to maintain as it will be complex to implement between the teams. It uses Scrum Of Scrums (SOS) meetings for collaboration and coordination. Human communication and knowledge sharing are major concerns. There is no effective technique or approach that formulates to coordinate work with co-located teams and there is also no proper technique or method to organize and manage the internal dependencies, these dependencies become a cause of project failure. The software industries are facing problems and issues like coordination, meetings issues, team size, complexity, and change request. There are only a few empirical studies that focus on cost estimation techniques by which productivity and accurate estimation of cost and time can be achieved with keeping in mind these factors. This study examines the challenges faced by organizations while using Scrum. Also, discuss these factors that how affect the scrum project success of completion. This study aims to provide knowledge about key challenges of Scrum cost estimation and also briefly explain why current scrum cost estimation techniques are not pertinent for project effort estimation. Lastly, propose a framework to estimate the cost of the project as per the expertise and knowledge of the developer, team, and project manager. Further, we have developed a web-based system to estimate the efforts of the project by using the suggested framework. This web-based cost estimation mechanism enables the estimation of efforts autonomously. Developed a software-based estimation system that collects inputs from team/s and developer/s to estimate the cost and time of a project.

2. Literature work

The effort estimation method means the process that is used to come up with the effort estimates. Input in the effort and cost estimation process is the objective of the estimation process and the data that is used during estimation. This data can be categorized as qualitative and quantitative measures. This data can be collected from previously completed projects and other developmental activities. The quantity of data is not the only factor that is necessary for the accuracy of the estimation process.

2.1. Agile scrum methodology

Agile methodology is well known as the alternative way to traditional software development processes. The agile model works with small teams and use for the small size project development. The agile model develops any software in iterations and allows change requests at any stage of a project and any iteration. It always gives priority to customer satisfaction and involves the customer in software development [1,2]. There are various agile methodologies but the most in use are the scrum methodology [3,4]. The scrum process starts by creating a product backlog that is a prioritized list of features and other requirements needed to develop a successful product. The product owner (client) is responsible to define the product backlog (which contains the user stories). The work itself is performed by sprints that it's called "iterations" [5,6]. At the start of each sprint, the team plans which feature of the product backlog is created. Scrum masters and team members can plan the new iteration after the end of the previous iteration for improving and adapting the changes [7]. A sprint backlog is a list of prioritized features and contains all functionality of the product

identified by team members to be completed during the sprint [8,9]. The team chooses the highest priority items from the product backlog during the Sprint planning meeting that is usually in the form of user stories. Sprint backlog refers to the estimation that how many hours each task will take to be completed. The product backlog is measured by relative story points whereas the sprint backlog is measured by hours [10,11]. For the priority of these user stories, a sprint meeting was conducted to finalize the stories. In meeting the development team arranges a 15–20 min time-boxed meeting every day to make a plan for the next 24 h and to synchronize tasks. The major purpose of daily scrum meetings conducted by the scrum master is to analyze the progress towards the goals and also examine that progress is trending towards executing the final task as shown in Fig. 1 [12,13].

3. Factors that influence the cost of scrum

We have done an extensive literature study to find the factors that directly influence on exceed of cost and time in the scrum projects. After the detailed analysis, we have narrowed down some factors that frequently impact on project's cost and time. We also don't find any study that worked on these factors collectively and proposed any estimation technique that addresses all of these factors [14,15]. All these factors are listed in Fig. 2.

3.1. Coordination issue

Co-ordination is the combination of integration of tasks to reach a specified goal. The coordination represents the dependencies of teams on each other related to functionality and code dependency. Due to a lack of ineffective communication coordination issue arises. Co-ordination among co-located teams may be an effect due to misunderstandings and lack of knowledge of requirements. Because the Developers and other team members in cross-functional teams cannot put their best without an efficient understanding of tasks and user requirements [16,17]. Scrum was designed for small-scale and co-located teams due to quick feedback to change the request of customers and complete the user stories. But due to inefficient coordination between team members, the projects become fail and this cause exceeds in cost and time [18–20].

3.2. Team size

Team size refers to the number of teams and communication between different co-located teams to a shared understanding regarding the project. The projects have multiple scrum teams for completion with the project type and nature. Many team models are: Isolated Scrum team, Distributed Scrum of Scrum, Fully Integrated Scrum team, etc. It is analyzed that frequent meetings are required among Scrum master, product owner team, and developer to ensure better and more effective communication and collaboration. This category needs a lot of attention and effective communication at an early stage of the project [21,22]. The issue related to the team size in the scrum methodology is that when the team size becomes increases coordination and communication become difficult between developers that causing cost increases in the scrum model [23,24].

3.3. Complexity

Complexity in Scrum can be categorized in different terms like task complexity and story point complexity. All the estimation techniques ignore the developer's skills and expertise to estimate the user stories' complexities. In the scrum methodology, the teams estimate the user story complexity with the help of story points. Every developer gives different story points to the user story then at the end the total hours are estimated for the user stories for the current iteration/sprint. In the scrum methodology, the issue related to estimating the user stories is

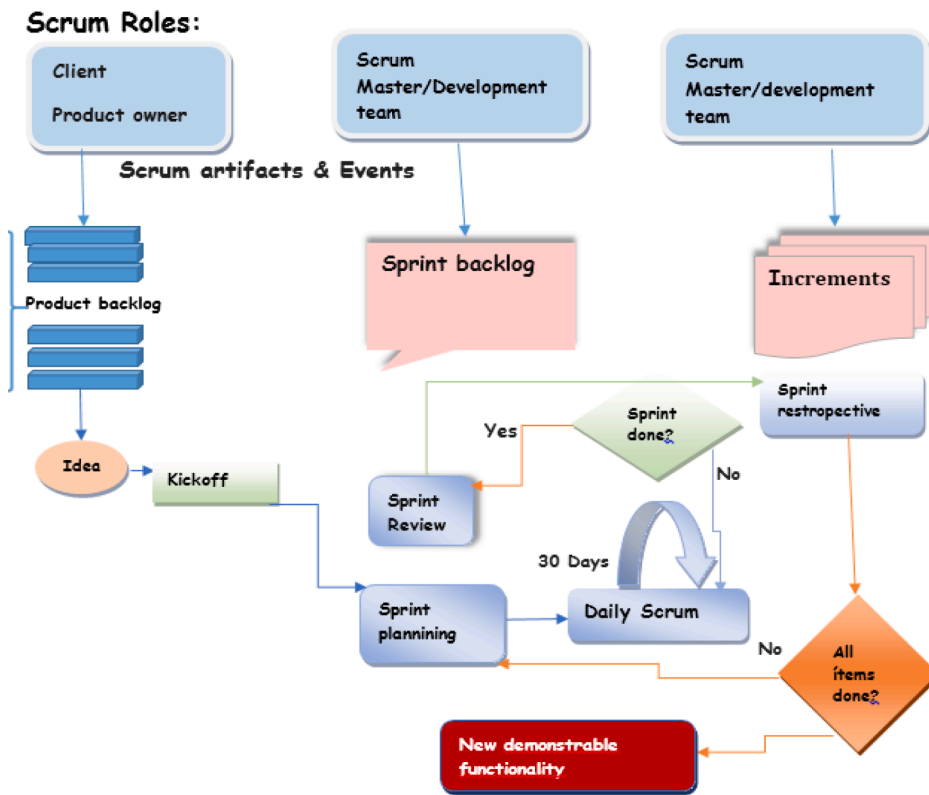


Fig. 1. Scrum Role and sprint planning.

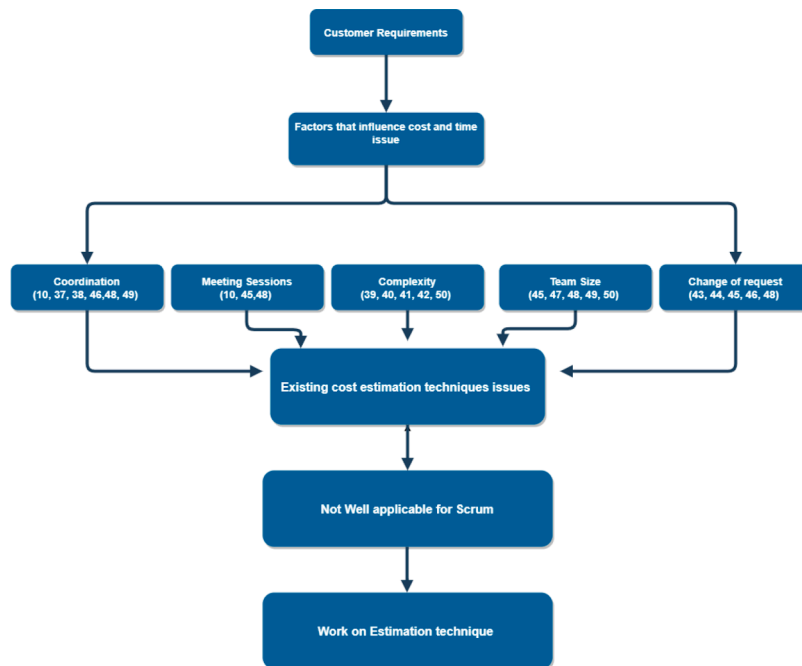


Fig. 2. Factors that influence the Cost Estimation Process.

that teams estimate the user stories at run time. After the 2–3 sprints delivery the team performance can be measured able. In the scrum methodology due to the self-organized, teams select and prioritize easy user stories first to do that ultimately left the complex user stories at the end and causes not project completion at right time [25,26].

3.4. Meeting sessions

Scrum enhances communication through Sprint planning meetings, Daily sprint meetings, and Retrospective meetings. The scrum master manages the information flow between the onsite and offsite teams. But some challenges suffer the meeting schedule. It is difficult to manage a meeting time where each member from a different team can attend.

Meetings like Sprint Review sometimes overlap with other meetings. There are many reasons behind this like developers feeling hesitant, lack of requirements knowledge, etc. [27–29].

3.5. Change of request

In Scrum, the development team members define the process for managing and approving the changes throughout the project. The changes at the small level are directly approved by the Product owner. Scrum projects warmly welcome the change of request at a small level but the change of request at a large level during the development process influence the cost estimation process and cause late delivery of the project [30–32].

4. Related work

Problems with the existing approaches

To maintain the schedule and budget of a project, many estimation techniques and models have been presented. Many existing techniques/methods are divided into 2 categories:

- Algorithm based methods
- Non-algorithmic based method

Non-Algorithmic based methods

The non-algorithmic techniques are techniques that do not work with any numerical equation. These techniques only estimate the effort with theoretical concepts.

4.1. Planning poker

According to this technique, team members discuss cost and effort estimation. Each member has different requirements regarding estimation, and all team members participate in the discussion and exchange their requirements regarding the estimation process. After a brief discussion session, team members finalized the estimation process and requirements by comparing each member's requirements. This technique is useful to ensure team member interaction. This technique has less empirical evidence regarding accuracy and is less applicable in the software industry. This technique is not a predictive base [33–36].

Issues:

There are several challenges in planning poker:

- Planning poker is less accurate when there is a lack of previous experience with the same projects.
- The anchoring effect is another issue of planning poker which means building own opinion by using an initial piece of information and skills as a base. It's like an idea that comes up from the initial discussion and has a deep impact and influence on the remaining part of the discussion.
- Domination is also another issue in planning poker. In a meeting session, when a most experienced member selects a card with a random number of user stories, test objects, specifications, requirements, test cases or bugs, etc. then the remaining members with less experience follow the same pattern of picking the card. This influence the results.
- Like expert estimation and planning poker also required the involvement of highly extensive vs. skilled and experienced persons who represents a variety of viewpoints.
- In planning poker like expert judgment, the judgment may be biased by irrelevant and misleading information and planning poker does not provide such a method or procedure that can be reused.

4.2. Wideband Delphi

This technique puts focus on the interaction between team members.

The whole task is structured into a breakdown framework where each member is assigned a task. The team members perform [37,38] their relevant tasks for estimation purposes. It includes experts' opinions. This technique is useful for agile-based projects where interaction and communication between team members is an essential principles for development.

Issues:

- Wideband Delphi also required extensive skills and experience People and opinions can be biased by providing irrelevant information.
- Most experts fail to provide objective and quantitative information regarding project effort dependencies like what kind of project characteristics may affect the scope and which characteristics affect the extent.
- False confidence is developed due to biased judgments. The judgments containing irrelevant data may develop the false confidence of estimators like to complete the story point in an estimated period.
- It is difficult to repeat the technique again and again with different groups. The wideband Delphi technique does not provide such a specific model for estimation that can be used again and again for different projects.

4.3. Expert judgment method

This approach involves discussion with experts or a group of experts to use their experience and their domain knowledge about the project to reach the estimation. Delphi is introduced to satisfy broad communication bandwidth [39–41]. The Delphi technique has been successfully used in estimation techniques. The steps of the estimation procedure are given as follows:

Issues:

- Highly and extensively dependent on experts' opinions and experience. So needed the most experienced and skilled person for judgment and this enhances the cost.
- This technique is sometimes dependent on measurement data, this data support experts to estimate the project effort. But this messy data and information mislead the experts due to data deficits like incompleteness and inconsistency and lead the project to the wrong estimation. This data must be given after several analyzed techniques.
- This technique is not a predictive base, because most human judgments are not repeatable and reusable. Here the repeatable and reusable mean that previous judgments for a similar project cannot be considered for a current project. Because these estimations are human-based judgments and can be biased by misleading and irrelevant information, usually requirements remain incomplete.

Algorithm-based methods

It includes the techniques that used numerical work or equation modeling to estimate the efforts. It no resides only in the theoretical way of estimation of time and cost of any project.

4.4. Story point

The story point is used to estimate the size. The story point is the unit of software size and intuitive combination of complexity and inherent risk development effort. It is assigned as a Fibonacci number sequence. The user stories are measured as story points. A story point can be divided into smaller, easier, and estimated blocks.

Issues:

- Mostly estimators face difficulties to choose story points in case of multiple user stories. The selection of a particular user story impacts the estimation process. There is no specific tool or procedure for the selection/ priority of user stories. The selection method of story

points is based on an estimator. If the estimator has sufficient knowledge and skills then the project will not suffer but in case of lack of project understanding the project will suffer.

- Desire to achieve more and more, sometimes team members inflate the story points. The team under pressure to achieve more story points mostly falls into the illusion of higher velocity. This illusion creates low quality of work.
- Team may not be comfortable in an early phase of story point estimation due to different agile methodology in the team [42–44] and if there is a lack of coordination then the project will suffer.

4.5. Use case point

This method is best suited for object-oriented software development. In this method use cases and the number of actors is used to calculate the Unadjusted Use Case Point [45,46]. Like function points, it also used environmental adjustment and technical complexity to find adjusted UCP. As in the case of Agile, it counts transactions as per use case.

Issues:

- Use case points do not capture the non-functional story points like portability, maintainability, performance, etc. properly.
- UCP can only be used when the user stories are written in the form of use-cases, uniformly structured, and with the goal. These user stories must contain detailed information about functionality. If there is no detailed information then the resulting UCP may not be accurate.
- UCP is well useful for initial estimation for the whole project but it is not useful when there is iteration to iteration work. Like if the team completed the first 2 sprints and working on the next sprint and then a change of request is requested by the user then it impacts on overall project results.

4.6. Velocity

The working capability of team members in each iteration is called Velocity. Velocity is a unit to calculate the team progress rate. Story points are used as metrics of velocity. Effort estimation consists of story size, velocity, and complexity [47,48]. Velocity refers to helping the team to polish their estimates of a project. It enables overtime calculation and releases planning.

$V = \text{Story point completed in one iteration} / \text{Story point in one user story}$.

There are two types of Velocity measurement (1) Actual Velocity and (2) Expected Velocity.

4.6.1. Actual velocity

The actual velocity is the real out of the team estimated after delivery of some sprints. The team output is determined when some user stories are done by developers [49,50]. The total actual velocity can be measured: $\text{Actual Velocity} = (V_a)$. $(V_a) = \Sigma \text{Achieved Feature Story Points} / \text{Sprint}$

4.6.2. Expected velocity

The Expected Velocity is the velocity that is expected from the team as an output before the delivery of any user story or sprint [62–64].

4.7. SLOC

Estimate/ size is one of the most important elements of software development. It is the key marker to telling about the cost, effort, and time of the venture. The size of the undertaking is likewise the base unit to determine different measurements for the project. The source line of code is not the sole supporter to estimate the cost, effort, and time of the project [51,52]. It just serves as the input to estimate the cost, effort, and time of the project. SLOC contribution to the estimation of the project is very important because the SLOC tells about the size of the project and

the effort required for the project.

Issues:

- SLOC does not effectively correlate with the software functionalities like a program with less size code may contain more functionalities as compared to large size because some developers developed the same functionalities of large size code with small size code.
- It was a tradition of one-to-one correspondence between instruction and physical lines that was broken due to new emergence in technologies like high-level programming. Functionality must be the same on two different platforms. This thing badly affects the measuring metric SLOC.

Now we describe the features and issues of existing estimation methods in Table 1 forms. Table 1 shows a comparison of techniques for estimation.

5. Proposed cost estimation technique

It is analyzed through literature review and survey that Scrum has some barriers in implementation. There is a strong need to propose an Estimation technique after examining the Literature Review and survey results that will be a predictive base. This technique would help in improving the planning of development of the team at the initial level. The proposed estimation technique will assist in effort and cost estimation. Our estimation framework starts with the review session meeting that will be held at the initial of every project. The project manager/ Scrum master briefly explains the project description to the team members (Team leaders, Developers, and project manager) to find out the project cost, time, and effort. In this review meeting, a form is distributed to all members to collect their feedback. This feedback consists of some values that will give development experience and categorization of project type (easy/ complex). In this review session, [53–55] team members like project managers, developers, and team leaders will communicate with each other and update which modules have been completed and which ones are new for the development team. Fig. 3 shows the solution scenario:

Categories of Modules:

- Easy

The first category of the module is Easy which means there are maximum modules/sprints that have been developed by the members in other projects/ applications that are similar to the current project. There is a need for minor modifications to meet the requirements. If the number of modules is maximum which is already developed by developers previously then the module will be categorized as "Easy" depending on the developer's experience [56,58,59,70].

- Complex

The second category is the complex module which means the whole application is new for developers. They have never developed the same module in previous applications. The developers have no experience to meet the modifications of the current project/sprint. The developers have to put great effort, and time to accomplish the task and they need more time and cost to handle the change of request [57,70,71,73].

These both categorize we have adopted after extensive study of previous cost estimation techniques used in the scrum projects. The significance of the study has been argued with the comprehensive literature relevant to effort estimating approaches, and some of such techniques assist and encourage the proposed new technique. Scrum Methodology, Function Point, Expert Analysis, and [45,46] Delphi Technique are the methodologies listed in the Table 2. Many of such approaches have shortcomings in terms of accurately measuring the cost and time for agile development with user-proposed changes.

Table 1
Features and issues of existing methods.

Method	Type	Features	Issues	Refs.
<i>Poker planning</i>	Non-algorithmic	<ul style="list-style-type: none"> ü Useful for enhancing team members' interaction ü Not required historical data 	<ul style="list-style-type: none"> ü Less empirical evidence regarding the accuracy ü Less applicable ü No reusability ü Extensive skilled people required ü Biased by providing misleading data 	[17,21, 23,24, 31,32]
<i>Wideband Delphi</i>	Non-algorithmic	<ul style="list-style-type: none"> ü No required historical data/particular for its input ü Intuitive and easy to apply 	<ul style="list-style-type: none"> ü Need repetition at each estimation ü False confidence developed ü Experts can be biased ü Fail to provide objective and quantitative analysis 	[23,24]
<i>Expert judgment</i>	Non-algorithmic	<ul style="list-style-type: none"> ü Rapid prediction 	<ul style="list-style-type: none"> ü Highly dependent on expert's opinions and experience ü Requirements remain incomplete mostly ü Biased by providing irrelevant data ü Extensive skilled and experienced people required ü Cost increased 	[17,23, 28,34]
<i>Use-case Points</i>	Algorithmic	<ul style="list-style-type: none"> ü Best suited for object-oriented development 	<ul style="list-style-type: none"> ü Need detailed data ü Not useful for iteration work 	[18,23, 32,35]
<i>SLOC</i>	Algorithmic	<ul style="list-style-type: none"> ü Serves the input to estimate the cost, time 	<ul style="list-style-type: none"> ü Difficult to estimate the exact line of code ü Not effectively correlated with functionalities ü New emergence in technologies like high-level programming 	[20–23, 25,26]
<i>Story Point</i>	Algorithmic	<ul style="list-style-type: none"> ü Foster collaboration ü Easy, intuitive and take less time 	<ul style="list-style-type: none"> ü The illusion of higher velocity ü Quality is sacrificed ü Teams may inflate story points 	[18–20, 23,24, 27,30, 32]
<i>Velocity</i>	Algorithmic	<ul style="list-style-type: none"> ü User Story Estimation ü Team Output 	<ul style="list-style-type: none"> ü Wrong prediction of output ü Team delay project 	[61,62, 63]

This categorization assists the manager to conduct information about team members' experience and their review of the current module estimation process. This categorization will be done by a Form that is given by the manager to all the members in the review session at the initial stage. The form is shown in Fig. 4:

The proposed algorithm is shown in Fig. 5. The purpose of this algorithm is to design, a system that automatically selects the team members and reduces the factor of cost and time [57]. The system will work in the following steps:

- Firstly the manager will send the form shown in Fig. to every developer randomly.
- Every developer receives the form on their system.

- The developers will fill out the form according to their particulars (working experience, project, and language).
- This form mainly put focuses on the developer's opinion like Easy/Complex.
- Then the developers will submit the form to the system database individually.
- After the submission of forms to the system database, the system will sort the input into the output sorting list. This output is sorted on behalf of the easy/ complex category.
- If the system will select first easy then the next category will be complex, if the system will select the first complex then the next will be easy.
- After that, the system will be able to show which developer is best suited for a project.

6. Implementation

We have automated the proposed solution and tested it on different projects. We have collected surveys from several software industries about the system's working, features and parameters. The developed system works as follows: (i) first get the developer's basic information, (ii) get the project's type as per the expertise of the developer, and (iii) estimate the effort needed to complete the project.

In this form, the team members individually fill the required data. This form consists of 2 sections: (i) the First section contains the developer's name, designation, experience, and salary. (ii) Second section requires project detail like project name, project type, language, and duration. After the developer's information system gets the project types from developers that are either Easy or Tough. The developer chooses a type as per his experience and skills. This form is filled by every team member individually and nobody knows about any other developer's feedback. After filling out the requirements the form is submitted to the main database. After that, the information is only viewed by the project manager because he is responsible to respond each query of the client. The main purpose of this form in Fig. 6 is to reduce the coordination and communication gap which removes experts' biasness as an issue in planning poker.

Fig. 7 explains the filled form with the project details mentioned in Fig. 6. In the case of multiple projects, if any particular project's information is required then only with one button press the manager can get the desired project's details. Moreover, new projects can be added with names and descriptions by clicking the send button. This thing reduces the pile of documentation and extra other burden of project management.

Fig. 8 contains the results of every team member or developer. The manager can analyze the results of every project. This section gives brief information about the project's team members' experience and their assessment including easy/complex, time, and cost regarding every project.

7. Analysis of existing technique with the proposed solution

Planning poker analysis:

According to this technique, team members discuss cost and effort estimation. Each member has different requirements regarding estimation, and all team members participate in the discussion and exchange their requirements regarding the estimation process. After a brief discussion session, team members finalized the estimation process and requirements by comparing each member's requirements. This technique is useful to ensure team member interaction [58,60,62].

- Planning poker has an anchoring effect
- Due to the increase in number the size and cost of the project increased
- Decision can be biased and difficult to control
- Dominating factor involve estimation.

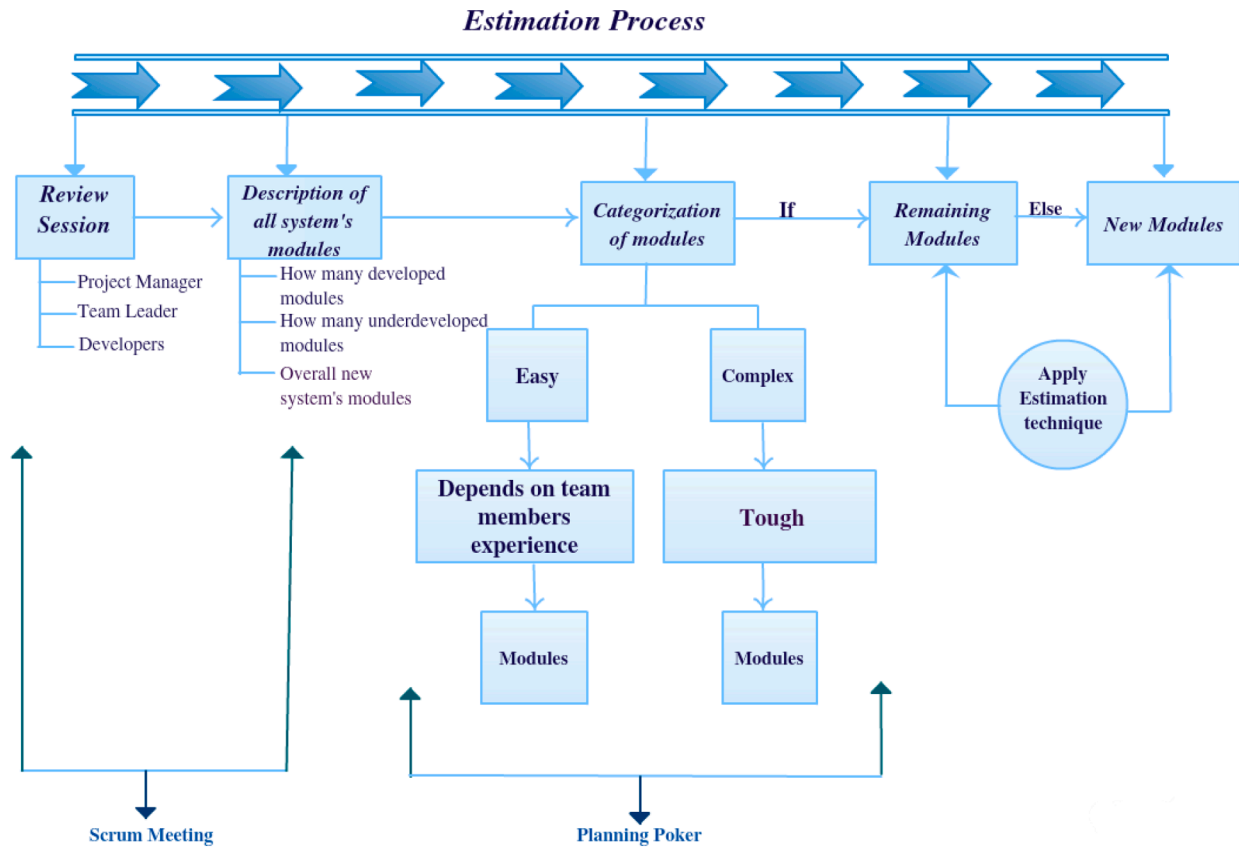


Fig. 3. Estimation Process.

Table 2
Motivational techniques.

Techniques	Motivational Work
Function Point	Low, Average, and High Categorizations
Expert Opinion	Professional Judgment About Project
Scrum Methodology	Daily Meeting Session
Delphi Technique	Prediction about Software

Proposes solution will remove the expert’s biasness through automation selection of team members. Team members will be selected randomly instead of a self-based approach. After the selection method, there are team members that will be most suitable to fulfill the project requirements.

Wideband Delphi analysis

This technique puts focus on the interaction between team members. The whole task is structured into a breakdown framework where each member is assigned a task. The team members perform their relevant tasks for estimation purposes. It includes experts’ opinions. The quality aspect is also considered in this technique [59,70].

- False confidence is developed due to biased judgments. The judgments containing irrelevant data may develop the false confidence of estimators like to complete the story point in an estimated period.

The proposed solution will randomly select the team members that reduce the expert’s biasness and false confidence by selecting the most appropriate and best suitable team members for the project.

Expert judgment analysis

This approach involves discussion with experts or a group of experts to use their experience and their domain knowledge about the project to reach the estimation. This approach is widely used by companies. Delphi

Name: _____

Designation: _____

Experience: _____

Salary: _____

Working Experience

Current project:

Project Type: _____

Project duration: _____

Language: _____

Previous Project/ Module:

Team Size: _____

Project duration: _____

Language: _____

No of Modules: _____

Categorization of project/ Modules

- **Easy:**
- **Tough:**

Fig. 4. Categorization Form.

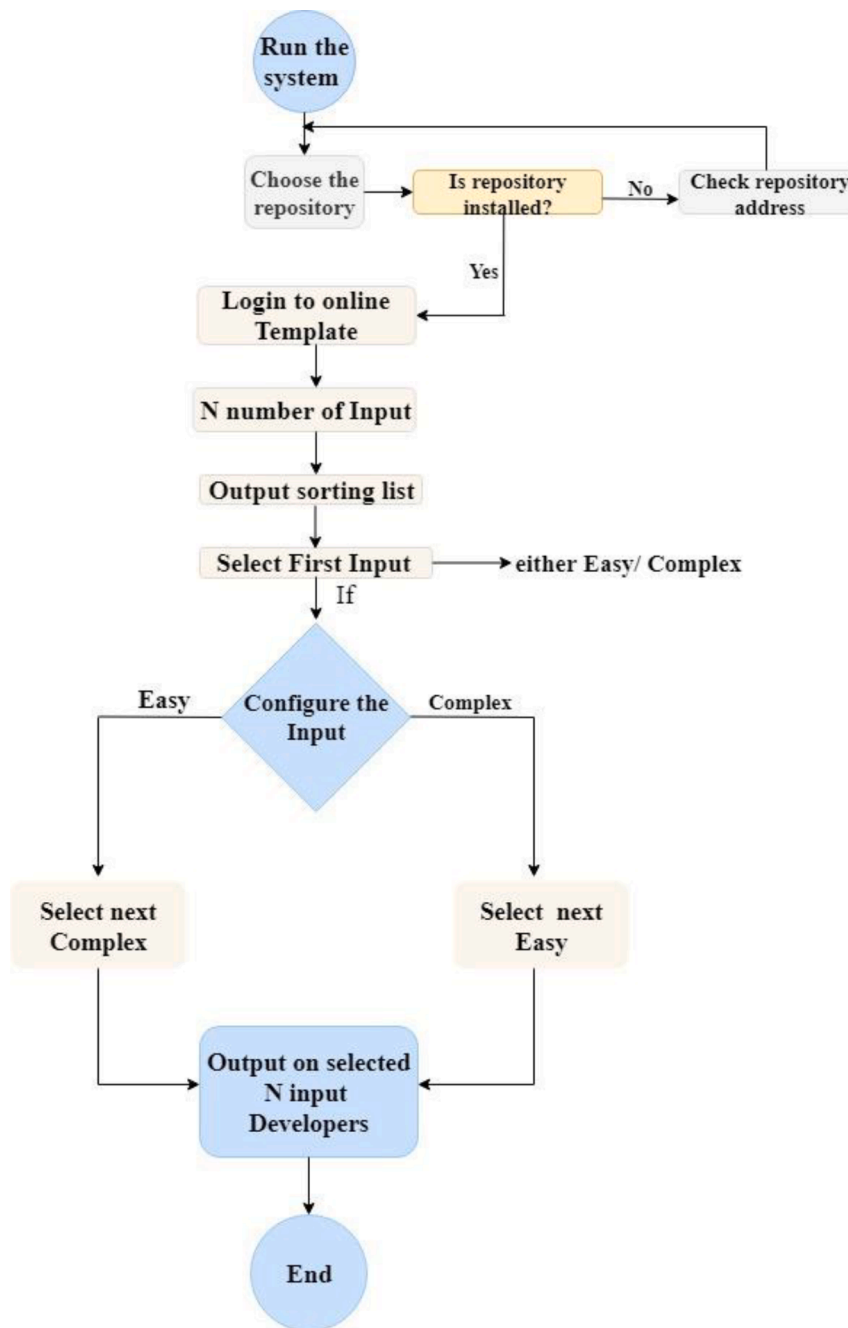


Fig. 5. Working of Algorithm for Cost estimation.

is introduced to satisfy broad communication bandwidth [60,81].

- Highly and extensively dependent on experts' opinions and experience. So needed the most experienced and skilled person for judgment and to control cost.

Through the proposed solution form filling feature every member will be randomly selected so the dependency on the expert's opinions will be decreased because there is the most competent and skilled person regarding every project due to their development experience. Every member will submit a form with the required necessary information. For every project, the team members will be selected according to the strong relationship between the developer's skills and project scope. This feature of the proposed solution will enhance the team communication and relationship for every project due to most of the similarities of their

skills [61,74,80].

Story points analysis

The story point is used to estimate the size. The story point is the unit of software size and intuitive combination of complexity and inherent risk development effort. It is assigned as a Fibonacci number sequence. The user stories are measured as story points. A story point can be divided into smaller, easier, and estimated blocks.

- Mostly estimators face difficulties to choose story points in case of multiple user stories. The selection of a particular user story impacts the estimation process. The selection method of story points is based on an estimator. If the estimator has sufficient knowledge and skills then the project will not suffer but in case of lack of project understanding the project will suffer.

Welcome

Name

Designation

Experience

Salary

Current Project:

Project: Project Type:

Duration: Language:

SEND

Fig. 6. Project and Developer Data for a project.

Welcome
☰

Add New Project

Project Name

Project Description

SEND

Projects

Show entries Search:

Name	Description	Created at	Actions
project 4	real estate	2019-09-19	Delete
project 3	property website	2019-09-19	Delete
project 1	pension management	2019-09-19	Delete
project 2	education	2019-09-19	Delete

Showing 1 to 4 of 4 entries Previous Next

Delete All

Fig. 7. Deception of Projects.

The proposed solution in case of selection of story points will provide a smooth channel that prioritizes the user stories. All story points are briefly explained at the initial level of a project [62,68].

Use case point analysis

This method is best suited for object-oriented software development. In this method use cases and the number of actors is used to calculate the Unadjusted Use Case Point. Like function points, it also used environmental adjustment and technical complexity to find adjusted UCP.

- UCP is well useful for initial estimation for the whole project but it is not useful when there is iteration to iteration work. Like if the team completed the first 2 sprints and working on the next sprint and then a change of request is requested by the user then it impacts on overall project results. It is also not feasible for the change request to come on sprint. It is not useful for the estimation of the change request impact on the project time and cost.

The proposed solution focuses on whole project estimation and also changes of request on each iteration. In the Review Session, all team members collaborate, and in this session, they also discuss how many sprints are completed and how many are remaining, also a change of request must take into account [63,79,80].

SLOC analysis

Estimate/ size is one of the most important elements of software development. It is the key marker to telling about the cost, effort, and time of the venture. The size of the undertaking is likewise the base unit to determine different measurements for the project. According to the Boehm perspective about the cost estimation of the project, the size of the project is a fundamental part of the estimation models.

- It was a tradition of one-to-one correspondence between instruction and physical lines that was broken due to new emergence in technologies like high-level programming. Functionality must be the

Welcome ☰

Survey Results

Show entries Search:

Name	Designation	Experience	Salary	Project Name	Project Type	Duration	Language	Acti
a	developer	12	999	project 1	Easy	11	php	Dele
a	UI	18	10,0000	project 1	Tough	27	php	Dele
a	developer	19	100	project 1	Tough	10	.NET	Dele
a	UI	8	778879	project 1	Tough	22	php	Dele
a	developer	19	6566565	project 1	Tough	27	.NET	Dele
a	UI	18	11000000	project 4	Easy	27	php	Dele
a	UI	7	10001	project 4	Tough	11	php	Dele
a	developer	12	11000000	project 4	Tough	10	.NET	Dele
a	UI	12	10,0000	project 4	Easy	11	php	Dele
a	developer	12	11000000	project 4	Tough	27	.NET	Dele
Search Name	Search Designation	Search Experience	Search Salary	Search Project Name	Search Project Type	Search Duration	Search Language	

Showing 1 to 10 of 65 entries Previous 2 3 4 5 6 7 Next

Fig. 8. Each Project's Final Suggestions by the developers.

same on two different platforms. This thing badly affects the measuring metric SLOC.

The proposed solution is mostly selecting the developers who have the same language expertise to develop the project. Whenever the change request comes from the client the developer can easily complete the change request. It will enhance the project development due to the same programming language of the developers additionally will improve the coding style [64,68,79].

8. Analysis with existing techniques

The proposed solution removes issues from factors:

8.1. Coordination

The coordination represents the dependencies of teams on each other related to functionality and code dependency. The coordination issue arises due to lack of communication, lack of project requirements understanding, and functional dependencies. Our selection approach selects the most relevant and qualified developers for the project. Our approach discourages the self-organized approach that causes a lack of communication and understanding of the project requirements. Our approach selects the 50% highly experienced developers and 50% less experienced developers. This combination removes the coordination barrier because there is a combination of highly and less experienced persons and they communicate on the edge of requirements and other aspects related to the project. Most developers that are selected have enough skills and experience relevant to the project. Through the proposed selection approach there is a better understanding of requirements that promotes healthy communication between developers and manage the coordination of the project [65].

8.2. Team size

Team size refers to several teams and communication between

different co-located teams to a shared understanding regarding the project. In Scrum, a self-made team approach is adopted for every project development. In the scrum to meet the requirements/user stories within the sprint's deadline the team mostly increase the team size to involve one or more developers to complete the user story. The exceed in team size makes it difficult to complete the project successfully. The self-organized approach also causes an increase in team size and causes failures in projects. Due to the self-organized team project becomes fail. With the proposed solution approach team selection is autonomous and members are selected randomly. Some developers belong to different ranges of experience and skills. There is a combination of highly and less experienced developers that promotes better understanding and coordination regarding project requirements/user stories.

8.3. Complexity

Complexity in Scrum can be categorized in different terms like task complexity and story point complexity. In the scrum methodology due to the self-organized, teams select and prioritize easy user stories first to do that ultimately leaving the complex user stories at the end and cause of not completing the project at right time. The proposed solution approach measures the user stories at the initial level during sprint planning rather than measuring the stories at the end. The proposed solution removes the complexity by measuring the user stories at the initial stages of a project. The user stories at the initial level will be divided into an aspect of easy/complex [66].

8.4. Meeting sessions

The scrum master manages the information flow between the onsite and offsite teams through meeting sessions like Sprint planning meetings, Daily sprint meetings, and Retrospective meetings. It is difficult to manage a meeting time where each member from a different team can attend. Meetings like Sprint Review sometimes overlap with other meetings. The proposed solution schedule meeting twice a week and ask for a daily update on the project. This thing will reduce the collapse of

meeting session's time slots [67].

8.5. Change of request

Change of request is a part of project development. In Scrum, the development team members define the process for managing and approving the changes throughout the project. Changes in requests during the development process influence the cost estimation process and cause late delivery of a project. We cannot deny the factor of change of request. It can be controlled through an accurate equation. The proposed solution provides an equation that will manage the change of request during the development process [68].

9. Research methodology and analysis

The main objective of the paper is to propose an estimation technique to control the influence of factors on cost and time increase. We have collected the survey to validate the factors also from the software industries. We also have to find the reasons behind these factors [69]. We adopted two strategies for research methodology one is a literature review and the second one is the survey that we conducted from the software industries that use scrum methodology for software development. The adopted methodology is shown in Fig. 9.

- Data Collection

The selected data collection method was the semi-structured survey. We conducted a survey and filled the questionnaire in those industries where Scrum is implemented in most projects. The questionnaires were filled by experts, developers, and representatives of 7 different software development industries. That enables the developers, and experts to touch upon the different issues as needed as come up with new aspects and themes. To get in-depth information, and clarify our understanding

and respondents' full cooperation, we surveyed through interviews and email conversations. We created a quantitative and qualitative analysis with the help of completed questionnaires that were used to answer the research question [70]. When the questionnaire was made, the research process was going through some steps:

- (1) Schedule the time for the interview
- (2) Conduct the survey
- (3) Rewrite the interview by analyzing the grouping statements In interviews
- (4) Firstly we explain the motive behind this research to the interviewers (candidates),
- (5) We gave them the questionnaire and then started discussing the answers. The main objective of this survey is to find out the actual key challenges in the industry faced by the developers and managers.

Analysis Procedure

We analyzed the results of the interviews purely quantitatively. The data to be analyzed was going to be quantitative textual data. Because there was a restraint on time, we decided that a minimum of one person for each role must be interviewed. In other words, all roles must be covered in different teams where multiple persons are working for a single role. We categorized the answers into a table like "team size", "complexity", "coordination" etc. the text and figures were placed in the relevant categories. The segments that are not related to the research question are dismissed. We further extracted the results for each research question and refine the answers in the form of a graph [71].

- Validity Procedure

Our main purpose was to build and use the interview guideline to support the validity of the results. Firstly we select the study areas so

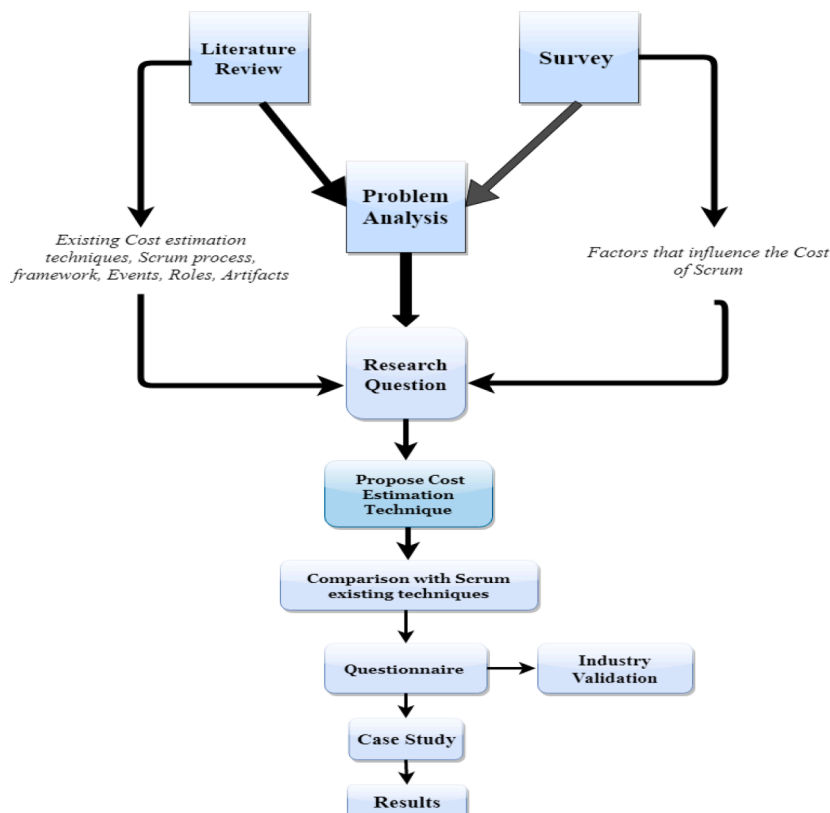


Fig. 9. Research Methodology.

make sure to avoid any interference between them. At initial, we stated the purpose to the interviewees to give fair and honest answers. The questionnaire and factors that influence the cost in scrum are mentioned below. The Reference Column is highlighting that the factors and survey questions exist in these papers.

The main objective of this research is to highlight the issues of Scrum that leads to projects being over-budgeted and complex. The research goal is to analyze the Scrum framework to explore the issues and variations regarding the estimation process. we designed a questionnaire that covers some major issues belonging to the estimation process. We ended this section by exploring the issues through a survey and Literate Review.

The scope of this research is to develop a new estimation technique. To get in-depth information, and clarify our understanding and respondents' full cooperation, we surveyed through interviews and email conversations. There are 20 respondents from these software development organizations. We examined through survey results that there are some challenges like issues in daily meeting session, communication gap, and time zone that exists in industries. The study concluded that the more issue while using the scrum methodology in the software industries is the change of request with 80% that comes from the client-side after it the scrum implementation in the industry is the biggest challenge due to the non-cooperative behavior of the software developers or teams. The teams in the software industries are still focusing on and following the traditional way of development. The other major challenges in the software industries while adopting scrum methodology are cost and time increase from the project's actual values. These existence rates in the software industries are 70–75%. The remaining challenges are self-organized team, coordination, and daily meeting session of 60%, 63%, and 55%. The developers become agitated during the daily meeting session. Our study concludes that most of the developers do not participate in the scrum daily meetings. Then teams have no accurate update of the project and in the end, the whole project becomes failing. The self-organized team mechanism also promotes the political policies among the teams as done in the planning poker. In the self-organized team mechanism, the developers dominate their one team member and follow its inputs instead he/she is giving accurate inputs or not. This issue also affects the team's coordination, when the project is distributed and all teams are dependent on their inputs.

After analyzing the survey results and literature review the research question was found. The survey and literature review aim to look for problems in software industries and research papers that are visible through results [72,73].

• Survey Question

- SQ1: Do team members generally communicate without confusion?
- SQ2: Is there a face-to-face communication gap between persons at different sites?
- SQ3: Do you face issues/challenges while conducting daily meeting sessions? Please explain/highlight
- SQ4: Does the team member understand the project vision and goals and do they truly believe that they can solve any problem to achieve any goals as a team?
- SQ5: Is the scrum team engaged in the decision-making process (rather than making or succumbing to decisions from others)?
- SQ6: Quality control and frequent deliveries are made?
- SQ7: Does the quality of the product being developed can be assessed?
- SQ8: Current project time and size in which using scrum methodology?
- SQ9: Is your team self-organizing, rather than functioning in command and control (top-down organization)?
- SQ10: While using Agile Scrum methodology for software development frequent change requests come from the client-side?

- SQ11: Does the cost of the projected increase after the change request come from the client-side?
- SQ12: Does the time of the projected increase after the change request come from the client-side?

• Methodology

This section explains the research method that was selected to produce the results for this study. This chapter includes the sampling design, research instrument, and data analysis procedures that were carried out in this study. The survey questionnaire method would also produce a more consistent range of responses from its participant thus making it easier to accumulate and analyze the data [74].

• Population

Six different Software houses in Pakistan were selected for data collection in this study. The different software houses targeted in size and population, who develops different software's houses. Sample sizes were selected through G power and 30 sample size were selected at the medium level.

• Sample

The sample was selected from the different software houses for this study, 30 questionnaires were distributed through online ULR. Only 30 questionnaires were returned, resulting in a 100 percent response rate. The sample for research was 15 males and 15 females were selected. Six software houses and 20 to 65 age range participants were included in this study and

Below 20 and above 65 years age ranges also were not selected and other software houses were excluded.

• Measures

The SPSS tool has consisted of survey analysis and results. The Likert-type items were used to measure the Factors Affecting Cost Estimation for Scrum Projects (12 items). The questionnaire measures some independent and dependent factors. To measure the dependent variable and independent variables regarding cost time size and manager role etc. Questions were designed to ask at the very start of the questionnaire as mentioned in Table 3.

Cronbach's Alpha Reliability

According to Varma (2006), a Cronbach's alpha value ranges from 0 to 1.00 and a value of 0.7 to 0.8 and above indicate high internal consistency whereas values lower than 0.7 indicate an unreliable scale. The result of the reliability test for this study is shown in Table 4.

• Validity of Measurement scale

Validity of activity is said to be the degree to that the tool produces the correct results. A validity check is applied to visualize the accuracy of the form. Eigenvalues, content validity, and item loading ranges are

Table 3
Factors study under the software houses.

Dependent variables	Independent variables
COST	Communication Manager Role
Time	Change of Request Module / sprint formation Team-based issue
Requirement understanding	language experience
User stories Handling	Developers participation Self-made team

Table 4

Alpha showed the reliability of the Factors affecting the cost estimation questionnaire.

Scale Name	Item Number	Alpha
Factors affecting cost estimation questionnaire	12	.77

taken from the previous analysis. The same construct validity techniques are applied to our collected information. Positive Eigenvalues are thought of pretty much as good. All freelance variables fall under positive criteria. Variety of item loading varies ≥ 0.4 and ≤ 0.6 are thought of to be sensible. Eigenvalues and things loading vary are given within the following Table 5.

Criterion validity is applied to ascertain the link of the variable quantity with all dependent variables. In our paper, the structure performance may be a variable quantity and it shows positive relations with all the opposite freelance variables. Talus plot of elbow shapes is thought to be acceptable. The below grid shows the talus plots of all variables mentioned in Fig. 10.

- Data Collection Procedure

The Questionnaire instrument completed by participants was made available through an easy-to-remember URL (Online survey.com), which pointed to the survey hosted by Google form, an online survey service. For this study, nonrandom sampling techniques were selected for respondents a Google form is shared via different social media websites who were running the software houses or working as a developer in some private and government software houses. A total of 12 questions were asked from respondents for this study. The questionnaire is consisting of different sub-variables. In the very first part respondent were asked general demographic questions about their organization. Twelve questions consist of the factors affecting cost estimation [75].

- Data Analysis

All of the data have been entered into and analyzed by the Statistical Package for the Social Science (SPSS) for Windows, version 20.0. Before statistical analyses, data cleaning and handling of missing values were performed. Frequency distributions of all the variables were checked for outliers, missing data, and typing errors. Normal distributions of the dependent and independent variables were assessed. Summary statistics, including the computation of means, standard deviations, frequency counts, and percentages of all data, were performed. The Chronbach’s alpha coefficients for internal consistency reliability and validity of the questionnaire short form were evaluated.

Mean scores and standard deviations were computed for all factors. Pearson r correlation has been used to answer research question three in this study. Pearson r correlation has been used to examine significant relationships between the continuous variables of all factors. The data

Table 5
Eigenvalues of variables.

Independent Variables	Eigen Value	Depended Variables	Eigen Value
Communication	.615	Cost	.356
Manager Role	.874	Time	.727
Change of Request	.473	Requirement understanding	.945
Module / sprint formation	.450	Users stories handling	.953
Team-based issue	.299		
Language	.945		
experience	.945		
Developers participation	.953		
Self-made team	.781		

have been evaluated to identify if statistical assumptions are met. An alpha level of confidence was at 0.001 for statistic tests. All data had been entered into SPSS 20.0 for Windows software for analyses. Descriptive statistics and measures of central tendency for all variables, and internal consistency reliability coefficients and means and standard deviations for the two study instruments were computed where appropriate. Alpha was set at a 0.001 level of confidence. To investigate the hypotheses and research question, regression analysis and correlation were used to estimate the unique relationship between each of the questionnaire and outcome variables in this study. Furthermore, to assess where there were significant differences amongst the thorough *t*-test [76–78].

10. Testing of hypotheses and presentation of results

The comprises two sections which are a demographic profile of respondents and inferential statistics. The Respondent’s characteristics such as gender, age, designation, software house, and experience are described in the demographic profile. For Inferential Statistic, Independent *T*-Test, Pearson Correlation Analysis, and Regression were applied to test the hypotheses of the study as stated in Table 6.

Descriptive statistics Table (1) outlines the mean scores and standard deviations of each of all variables under this study. The mean and SD of the demographic variables is the gender ($M = 1.500, SD=5085$), age ($M = 1.9667, SD=0.8899$), experience ($M = 16.33, SD=9643$), designation ($M = 1.9333, SD=1.0148$). The mean and SD scores of the independent variables are time ($M = 10.93, SD=2.545$), cost ($M = 15.60, SD=3.348$), requirement understanding ($M = 2.133, SD=0.9371$), and users stories handling ($M = 2.633, SD=0.7648$).

The mean and SD score of the independent variables are communication ($M = 1.733, SD=0.8276$), manger role ($M = 2.4000, SD=0.8550$), change of request ($M = 6.366, SD=1.129$), module/sprint formation ($M = 2.100, SD=1.028$), time base issues ($M = 10.56, SD=2.500$), language ($M = 2.133, SD=0.9371$), experience ($M = 2.133, SD=0.9371$), developer participation ($M = 2.633, SD= 0.7648$),self made team ($M = 6.200, SD=2.074$) and factors effecting cost estimation ($M = 2.0667, SD=0.2537$).

Based on the numbers of participants 50% female and 50% male responded to the questionnaire. The age factor of 70% lead to (0–5) years of experience and the 30% age factor lead to (5–15) years of experience. Overall six different software houses were included in the research and software houses have different percentages shown above the pie chart. In this study, 43% of developers, 30% of programmers, 16.67% of team leaders, and 10% of other designation post respondents record their responses.

This chapter commences with a detailed analysis of data that provides an insight into the findings of the study, accompanied by numerical and graphical representations of the data and interpretation of results. The presentation of detailed analysis and findings, which were extracted from questionnaires used to collect data discussed in this section. The hypotheses for this study were evaluated using inferential statistics in the form of the Pearson correlation analysis. Multiple regression analysis was used to ascertain the strongest relationship between variables that were measured.

- Findings of the study

A co-relational study between transformational leadership styles, transactional leadership styles, and employee satisfaction in the IT service department. The findings for each hypothesis are explained as follows:

H01: There would be a significant relationship between all variables cost, communication, and manager role.

H02: There would be a significant relationship between all variables time, change of request, module/sprint formation, and team-based issues.

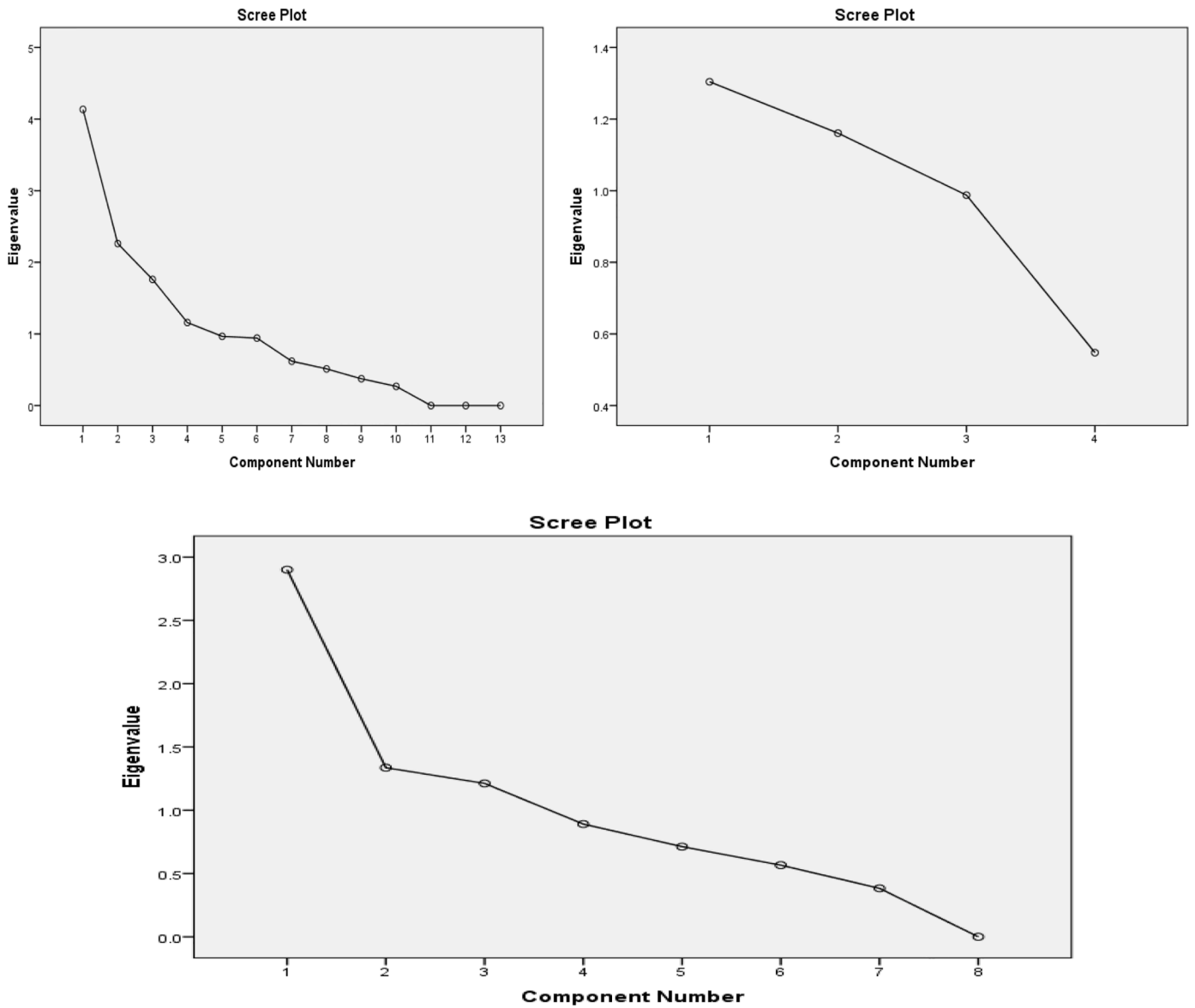


Fig. 10. Showed the plot screen validity of the questionnaire.

Table 6 showed the mean, SD.

Descriptive Statistics about the Variables	N	Minimum	Maximum	Mean	Std. Deviation	Skewness
Gender	30	1.00	2.00	1.5000	.50855	.000
Age	30	1.00	4.00	1.9667	.88992	.697
Experience	30	1.00	4.00	1.6333	.96431	1.324
Designation	30	1.00	4.00	1.9333	1.01483	.777
COST	30	5.00	20.00	15.6000	3.34870	-1.710
Time	30	5.00	15.00	10.9333	2.54522	.120
Requirement understanding	30	1.00	4.00	2.1333	.93710	.258
User stories Handling	30	2.00	5.00	2.6333	.76489	1.250
Communication	30	1.00	3.00	1.7333	.82768	.551
Manager role	30	1.00	4.00	2.4000	.85501	.156
Change of request	30	4.00	8.00	6.3667	1.12903	-0.337
Module/Sprint Formation	30	1.00	5.00	2.1000	1.02889	.806
Team base issues	30	7.00	15.00	10.5667	2.50080	.419
Languages	30	1.00	4.00	2.1333	.93710	.258
Experience	30	1.00	4.00	2.1333	.93710	.258
Developers participation	30	2.00	5.00	2.6333	.76489	1.250
Self-made team	30	2.00	10.00	6.2000	2.07448	.111
Factors Affecting Cost Estimation	30	2.00	3.00	2.0667	.25371	3.660

Note: A 5-point Likert scale ranged from 1= strongly agree, to 5 = strongly disagree.

H03: There would be a significant relationship between all variables requirement understanding, language, and experience.

H04: There would be a significant relationship between all variables user story handling, developer’s participation, and self-made team.

In Table 7 the Spearman rho correlation of nonparametric tests was also used to support these findings. There will be a significant and negative relationship between all the dependent and independent variables. The Pearson r correlation was computed to examine significant relationships between the dependent variables and all the independent variables of the instrument. The correlation coefficients were negatively significant in some variables shown in the above table. Most of the variables have a weak correlation with the computed variables. So H01, H02, and H03 were accepted because there would be a founded correlation between all the variables [77].

T-test compares means used to assess the gender difference in this study so this t-test applied to the gender difference. Table 8 mean average scores showed that there is no gender difference found in the tables. So hence proves that H05 is accepted.

In Table 9 linear regression analysis result indicated that cost predicted the 16% experience and developers’ participation. So the H06 is accepted.

In Table 10 linear regression analysis result indicated that time predicted the 92% change in request and module/sprint formation. So the H07 is accepted.

In Table 11 linear regression analysis result indicated that requirement understanding predicted the 1.000% language, manager role, and self-made team. So the H08 is accepted.

The Table 12 linear regression analysis result indicated that user’s story handling predicted the 3% teams’ base issues and communication. So the H09 is accepted.

We have found these factors highly impact cost and time exceed during the scrum project. The proposed estimation technique controls these factors in the estimation process and reduces exceed during development. Our technique reduced the cost and time and facilitates the estimation autonomously. We also found that the automated estimation approach is more viable as compared to other techniques because all the other techniques support the biases in estimation. During the survey limitations, we faced that some of the participants didn’t respond completely to all survey questions. Thus, we have excluded all such responses from the results analysis stage. We have sent many software developers to get survey outcomes but few of them did not fill the survey questions.

11. Conclusion

Scrum methodology is the most useful and adopted part of the agile methodology. Many organizations are adopting Scrum for the last

Table 7
factors of the questionnaire.

i. Using correlation Variables	Pearson correlation	Interpretation
Cost	.310	Significant correlation
Time	−0.254	Negative correlation
Requirement understanding	.127	Significant correlation
User stories Handling	.008	Weak correlation
Communication	−0.214	Negative correlation
Manager role	.022	Signification correlation
Change of request	.104	Weak correlation
Module/Sprint Formation	.102	Weak correlation
Team base issues	.016	Weak correlation
Languages	.127	Weak correlation
Experience	.127	Weak correlation
Developers participation	.008	Weak correlation
Self-made team	−0.236	Negative correlation

**Correlation is significant at the 0.01 level (1-tailed). * Correlation is significant at the 0.05 level (1-tailed).

Table 8
Showed the t-test comparing mean.

ii. Using of simple mean t-test
H05: There would be no gender difference among Factors Affecting Cost Estimation for Scrum Projects.

Factors Affecting Cost Estimation			
Gender	Mean	N	Std. Deviation
Female	2.0667	15	.25820
Male	2.0667	15	.25820
Total	2.0667	30	.25371

Table. 9
Showed linear regression.

iii. Using of linear regression
H06: Cost would be predicted by the developer’s participation and experience.

Model	R	R Square	Adjusted R Square	Std. An error in the Estimate
1	.128 ^a	.016	−0.057	3.44211

- a. Dependent Variable: COST.
- b. Predictors: (Constant), Experience, Developer’s participation.

Table. 10
showed linear regression.

H07: Time would be predicted the change of Request and Module/sprint formation.

Model	R	R Square	Adjusted R Square	Std. An error in the Estimate
1	.303 ^a	.092	.024	2.51413

- a. Dependent Variable: Time.
- b. Predictors: (Constant), Change of request, Module/Sprint Formation.

Table. 11
Showed linear regression.

H08: Requirement understanding would be highly predicted the manager role, self-made team, and language.

Model	R	R Square	Adjusted R Square	Std. An error in the Estimate
1	1.000 ^a	1.000	1.000	.00000

- a. Dependent Variable: Requirement understanding.
- b. Predictors: (Constant), languages, Manager role, Self-made team.

Table. 12
Showed linear regression.

H09: Users’ story handling would be predicted the communication and team-based issues.

Model	R	R Square	Adjusted R Square	Std. An error in the Estimate
1	.051 ^a	.003	−0.071	.79168

- a. Dependent Variable: User stories Handling.
- b. Predictors: (Constant), Team base issues, Communication.

decade to meet their business requirements. It is facilitating the software industries to develop software applications according to the user’s requirements. Scrum helps to fulfill the customer’s requirements through client interaction and development. Instead of this, scrum still has some challenges that affect the cost. These challenging factors include coordination, team size, change request, complexity, and issues in daily meeting sessions. Due to these factors mostly the cost and time exceed during scrum development. There are some sophisticated cost estimation techniques for scrum-based projects. But all estimation techniques have some limitations that make them less useful for estimation and overcoming the effects of these factors in a scrum. The main aim of the paper is to provide the knowledge about key challenging factors in Scrum that cause the cost and time to exceed, and also briefly explain why existing scrum cost estimation techniques are not pertinent for projects. In the last, we will propose a framework as the solution to

manage these factors and control cost and time from exceeding. We also adopted this estimation technique in different case studies in software industries for findings and results. The technique calculates the efforts autonomously. We design a software-based estimation technique that collects inputs from team/s and estimates the cost and time of a project.

Author contributions

All authors contributed equally.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] Balaji S, Sundararajan Murugaiyan M. Waterfall vs V-model vs AGILE: life Cycle. *Int J Inf Technol Bus Manag* 2012;29: June.
- [2] Ruparelia NB. Software development lifecycle models. *ACM SIGSOFT Softw Eng* 2010. May.
- [3] Sutherland J. Scrum. a revolutionary approach to building teams, beating deadlines and boosting productivity. Random House Business Books; 2014.
- [4] Nuevo ED, Piattini M, Pino FJ. Scrum-based methodology for distributed software development. *IEEE Computer Society*; 2011.
- [5] Suliman SM, Kadoda G. Factors influence that effect the software project cost and schedule estimation. In: *Proceedings of the Sudan conference on computer science and information technology (SCCSIT)*. IEEE; 2017.
- [6] Hossain E, Bannerman P, Jeffery R. Towards an understanding of tailoring scrum in global software development: a multi-case study. In: *Proceedings of the international conference on software and systems process*; 2011.
- [7] Moe NB, Dingsoyr T, Dyba T. A teamwork model for understanding an agile team: a case study of a Scrum project. *Inf Softw Technol* 2010. ELSEVIER.
- [8] Anwer F, Aftab S, Shah S, Shah M, Waheed U. Comparative analysis of two popular agile models: extreme programming and scrum. *Int J Comput Sci Telecommun* 2017;8(2). volumeissueMarch.
- [9] Lindsjorn Y, Sjöberg DIK, Dingsoyr T, Bergersen GunnarR, Dyba Tore. Teamwork quality and project success in software development: a survey of agile development teams. *J Syst Softw* 2016. ELSEVIER.
- [10] Stray V, Sjöberg DIK, Dyba T. The daily stand-up meeting: a grounded theory study. *J Syst Softw* 2016. ELSEVIER.
- [11] Kaisti M, Mujunen T, Tuomas M, Rantala V, Lehtonen T. Agile principles in the embedded system development. In: *Proceedings of the international conference on agile software development*. Springer Italy; 2014.
- [12] Miller. The Characteristics of agile software process" Granville together soft. In: *Proceedings of the 39th International conference and exhibition on Technology of object-Oriented language and Systems*. IEEE; 2001.
- [13] Singh Matharu G, Mishra A. Harmeet singh and Priyanka Upadhyay "empirical study of agile software development methodologies: a comparative analysis. In: *Proceedings of the ACM SIGSOFT software engineering Notes* page 1; 2015. January.
- [14] Streule T, Miserini N, Bartlome O, Klippel M, Soto BGD. Implementation of scrum in construction industry. In: *Proceedings of the creative construction conference*. Elsevier; 2016.
- [15] Schwaber K, Beedle M. Agile software development with scrum. Prentice-Hall: Upper Saddle River; 2002.
- [16] Sutherland J., Schwaber K., and Sutherl C.J., "The scrum papers: nuts, bolts, and origins of an agile process", 2007.
- [17] Choudhari J., Suman U., "Story Points based effort estimation model for software maintenance", ELSEVIER, 2011.
- [18] Rosa W, Madachy R, Clark B, Boehm B. Early phase cost models for agile software processes in US DoD. In: *Proceedings of the ACM/IEEE international symposium on empirical software engineering and measurement (ESEM)*; 2017.
- [19] Ahmed AR, Tayyab M, Bhatti SNr, Alzahrani AJ, Baber MI. Impact of story point estimation on product using metrics in Scrum Development process. *IJACSA Int J Adv Comput Sci Appl* 2017.
- [20] Coelho E, Basu A. Effort estimation in agile software development using story points. *Int J Appl Inf Syst IJ AIS* 2012.
- [21] Williams L, Nagappan N. Scrum + engineering practices: experience of three microsoft teams. *Software Engineering*, IEEE; 2011.
- [22] Caballero E., Calvo-Manzano J.A. and Felii T.S., "Introducing Scrum in a very small enterprise: a productivity and quality analysis", Springer, EuroSPI 2011, CCIS 172, pp. 215–224-2011.
- [23] Trendowicz A, Jeffery R. Software Project effort estimation. Springer, International Publishing Switzerland; 2014.
- [24] Zhang Z. The benefits and challenges of planning poker in software development: comparison between theory and practice. Auckland University of Technology; 2017. retrieved from, <http://hdl.handle.net/10292/10557>.
- [25] Lavazza L, Morasca S, Taibi D, Tosi D. Applying SCRUM in an OSS development process. In: *Proceedings of the Springer, international conference on agile*; 2010.
- [26] Desharnais JM, Buglione L, Kocaturk B. Using the COSMIC method to estimate agile user. In: *Proceedings of the 12th international conference on product focused software development and process improvement*; 2011. Pages 68-73.
- [27] Hayata T, Han J. A hybrid model for IT project with scrum. In: *Proceedings of IEEE international conference on service operations, logistics and Informatics*; 2011.
- [28] Popli R, Chauhan N. Cost and Effort estimation in agile software development. In: *Proceedings of international conference on reliability optimization and information technology (ICROIT)*. IEEE; 2014.
- [29] Tayh Raslan A, Darwish NR, Ahmed Hefny H. Towards a Fuzzy based framework for effort estimation in agile software development. *Int J Comput Sci Inf Secur (IJCSIS)* 2015.
- [30] Torrecilla-Salinas CJ, Sedeno J, Escalona MJ, Mejias M. Estimating, planning and managing agile web development projects under a value-based perspective. *J Syst Softw* 2015. ELSEVIER.
- [31] Moharreri K, Vinayak Sapre A, Ramanathan J, Ramnath R. Cost-Effective supervised Learning models for software effort estimation in agile environment. In: *Proceedings of the 40th annual computer software and applications conference (COMPSAC)*. IEEE; 2016.
- [32] Prakash B, Viswanathan V. A Survey on software estimation techniques in traditional and agile development models. *Indones J Electr Eng Comput Sci* 2017.
- [33] Bilgaiyan S, Sagnika S. Samaresh mishra and madhabananda, "a systematic review on software cost estimation in agile software development". *J Eng Sci Technol Rev* 2017.
- [34] Vyas M, Bohra A, Lamba CS, Vyas A. A review on software cost and effort estimation techniques for agile development process. *Int J Recent Res Aspects* 2018;5(1):2349-7688. ISSNVol., Issue.
- [35] Yuliansyah H, Qudsiyah SN, Zahrotun L, Arfiani I. Implementation of use case point as software effort estimation in Scrum framework. In: *Proceedings of the 1st international conference on engineering and applied technology (ICEAT)*; 2017.
- [36] Lenarduzzi V, Lunesu I, Matta M, Taibi D. Functional size measures and effort estimation in agile development: a replicated study. Springer; 2015.
- [37] Strode DE, Huff SL, Hope B. Sebastian Link "Coordination in co-located agile software development projects. *J Syst Softw* 2012. ELSEVIER.
- [38] Brede Moe N, Dingsoyr T. Tore Dybå "a teamwork model for understanding an agile team: a case study of a Scrum project. *Inf Softw Technol* 2010. ELSEVIER.
- [39] Nguyen V., Deed-Rubin S., Tan T. and Bohem B., "A SLOC counting standard", August 2007.
- [40] Bhatti K, Tarey V, Patel P. Analysis of source line of code (SLOC) metrics. *Int J Emerg Technol Adv Technol* 2012.
- [41] Olszewska M, Walden M. DevOps meets formal modelling in high-criticality complex systems. In: *Proceedings of the 1st international workshop*. ACM; 2015.
- [42] Ozkan N. Risks, challenges and issues in a possible scrum and COBIT marriage. In: *Proceedings of the Asia-pacific software engineering conference*. IEEE; 2015.
- [43] Samarawickrama SS, Perera I. A continuous Scrum: a framework to enhance Scrum with DevOps. In: *Proceedings of the international conference on advances in ICT for emerging regions (ICTer)*; 2017.
- [44] Pohl C., Hof H.J., "Secure Scrum: development of secure software with scrum", arXiv: 1507.02992v1, 2015.
- [45] Vithana VN. Scrum requirements engineering practices and challenges in offshore software development. *Int J Comput Appl* 2015.
- [46] Dingsoyr T, Nerur S, Balijepally V, Moe NB. A decade of agile methodologies: towards explaining agile software development. *J Syst Softw* 2012. ELSEVIER.
- [47] Oomen S, De Waal B, Albertin A, Ravesteyn P. How can Scrum be successful?. In: *Proceedings of the competences of the scrum product owner", twenty-five European conference on information systems (ECIS)*; 2017.
- [48] Drury M, Conboy K, Power K. Obstacles to decision making in agile software development teams. *J Syst Softw* 2012. ELSEVIER.
- [49] Hoda R, Murugesan L,K. Multi-level agile project management challenges: a self-organizing team perspective. *J Syst Softw* 2016. ELSEVIER.
- [50] Singhto W, Denwattana N. An experience in blending the traditional and agile methodologies to assist in a small software development project. In: *Proceedings of the international joint conference on computer science and software engineering*; 2016. IEEE.
- [51] Vlietland J, Vliet H. Towards a governance framework for chains of Scrum teams. *Inf Softw Technol* 2015;57:52–65.
- [52] Moharreri K, et al. Cost-effective supervised learning models for software effort estimation in agile environments. In: *Proceedings of the computer software and applications conference (COMPSAC)*, IEEE 40th Annual. 2. IEEE; 2016. Vol.
- [53] Stray V, Sjöberg DIK, Dybå T. The daily stand-up meeting: a grounded theory study. *J Syst Softw* 2016;114.
- [54] Ahmed AR, et al. Impact of story point estimation on product using metrics in scrum development process. *Int J Adv Comput Sci Appl* 2017;8(4):385–91.
- [55] Salmanoglu M, Hacıoglu T, Demirors O. Effort estimation for agile software development: comparative case studies using COSMIC functional size measurement and story points. In: *Proceedings of the 27th international workshop on software measurement and 12th international conference on software process and product measurement*. ACM; 2017.
- [56] Samarawickrama SS, Perera I. Continuous scrum: a framework to enhance scrum with DevOps. In: *Proceedings of the advances in ICT for emerging regions (ICTer)*, seventeenth international conference on. IEEE; 2017.
- [57] Lindsjorn Y, et al. Teamwork quality and project success in software development: a survey of agile development teams. *J Syst Softw* 2016;122:274–86.
- [58] Vlietland J, Van Solingen R, van Vliet H. Aligning codependent Scrum teams to enable fast business value delivery: a governance framework and set of intervention actions. *J Syst Softw* 2016;113:418–29.

- [59] Cooper RG, Sommer AF. Agile-stage-gate for manufacturers: changing the way new products are developed integrating agile project management methods into a Stage-Gate system offers both opportunities and challenges. *Res Technol Manag* 2018;61(2):17–26.
- [60] Drury M, Conboy K, Power K. Obstacles to decision making in Agile software development teams. *J Syst Softw* 2012;85(6):1239–54.
- [61] Przybylek A, Zakrzewski M. Adopting Collaborative games into agile requirements engineering. In: *Proceedings of the 13th international conference on evaluation of novel approaches to software engineering (ENASE'18)*; 2018. <https://doi.org/10.5220/0006681900540064>.
- [62] Bumbary KM. Using velocity, acceleration, and jerk to manage agile schedule risk. In: *Proceedings of the international conference on information systems engineering (ICISE)*. IEEE; 2016 (pp. 73-80).
- [63] Mahnic V. A capstone course on agile software development using Scrum. *IEEE Trans Educ* 2012;55(1):99–106.
- [64] Dong, W. (2015). Exploring and theorizing velocity flux in agile development.
- [65] Hassanein EE, Hassani SA. Cost efficient scrum process methodology to improve agile software development. *Int J Comput Sci Inf Secur (IJCSIS)* 2020;18(4).
- [66] Krishna VV. Agile methodology (SCRUM) approach for web application testing process to reduce time, cost and improve the quality. *Turk J Comput Math Educ (TURCOMAT)* 2021;12(11):5467–80.
- [67] Kom EAPS, Suryani E. Designing cost measurement system in a small scrum based software company using activity based costing model (Case Study: ABC company). In: *Proceedings of the international conference on information and communications technology (ICOIACT)*. IEEE; 2019. pp. 943-947.
- [68] Przybylek A, Albecka M, Springer O, Kowalski W. Game-based Sprint retrospectives: multiple action research. *Empir Softw Eng* 2022;27. <https://doi.org/10.1007/s10664-021-10043>. vol.
- [69] Butt SA. Study of agile methodology with the cloud. *Pac Sci Rev B Humanit Soc Sci* 2016;2(1):22–8.
- [70] Butt SA, Abbas SA, Ahsan M. Software development life cycle & software quality measuring types. *Asian J Math Comput Res* 2016;112–22.
- [71] Khalil MA, Kotaiah B. Implementation of agile methodology based on SCRUM tool. In: *Proceedings of the international conference on energy, communication, data analytics and soft computing (ICECDS)*. IEEE; 2017 (pp. 2351-2357).
- [72] Kom EAPS, Suryani E. Designing cost measurement system in a small scrum based software company using activity based costing model (case study: ABC company). In: *Proceedings of the international conference on information and communications technology (ICOIACT)*. IEEE; 2019. pp. 943-947.
- [73] Butt SA. Analysis of unfair means cases in computer-based examination systems. *Pac Sci Rev B Humanit Soc Sci* 2016;2(2):75–9.
- [74] Periyasamy K, Chianelli J. A project tracking tool for scrum projects with machine learning support for cost estimation. *EPIC Ser Comput* 2021;76:86–94.
- [75] Alostad JM, Abdullah LR, Aali LS. A Fuzzy based model for effort estimation in scrum projects. *Int J Adv Comput Sci Appl* 2017;8(9):270–7.
- [76] Gandomani TJ, Faraji H, Radnejad M. Planning poker in cost estimation in agile methods: averaging vs. consensus. In: *Proceedings of the 5th conference on knowledge based engineering and innovation (KBEL)*. IEEE; 2019. p. 066–71.
- [77] Arora M, Verma S, Chopra S. A systematic literature review of machine learning estimation approaches in scrum projects. *Cogn Inf Soft Comput* 2020:573–86.
- [78] Anwer F, Aftab S, Shah SM, Waheed U. Comparative analysis of two popular agile process models: extreme programming and scrum. *Int J Comput Sci Telecommun* 2017;8(2):1–7.
- [79] Chavarria RC, Madriz FL. ROI estimation in a scrum project: a case study. In: *Proceedings of the 28th international conference*. 64; 2019. p. 78–87 (Vol., pp.).
- [80] Aziz Butt S, Piñeres-Espitia G, Ariza-Colpas P, Tariq MI. Project management issues while using agile methodology. In: *Proceedings of the international conference on lean and agile software development*. Cham: Springer; 2022. pp. 201-214.
- [81] Meyer B. Making sense of agile methods. *IEEE Softw* 2018;35(2):91–4.