IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 3, 2020, accepted July 13, 2020, date of publication July 21, 2020, date of current version August 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3010968

An Intelligent Recommender and Decision Support System (IRDSS) for Effective Management of Software Projects

MUHAMMAD HAMID[1,2], FURKH ZESHAN[3], ADNAN AHMAD[3,4], FAROOQ AHMAD[3], MUHAMMAD ALI HAMZA[5], ZUHAIB ASHFAQ KHAN[3], SAIMA MUNAWAR[2], AND HANAN ALJUAID[6]

[1]Department of Statistics and Computer Science, University of Veterinary and Animal Sciences (UVAS), Lahore 54000, Pakistan
[2]School of Computer Science, National College of Business Administration and Economics (NCBA&E), Lahore 54000, Pakistan
[3]Department of Computer Science, COMSATS University Islamabad, Lahore Campus, Lahore 54000, Pakistan
[4]Department of Mathematics and Computer Science, Saint Louis University, 28003 Madrid, Spain
[5]Department of Economics and Business Management, University of Veterinary and Animal Sciences (UVAS), Lahore 54000, Pakistan
[6]Computer Sciences Department, College of Computer and Information Sciences, Princess Nourah bint Abdul Rahman University (PNU), Riyadh 84428, Saudi Arabia

Corresponding authors: Furkh Zeshan (drzfurkh@cuilahore.edu.pk) and Adnan Ahmad (adnanahmad@cuilahore.edu.pk)

ABSTRACT The management and estimation of agile projects are challenging tasks for software companies due to their high failure rates. This paper emphasizes how to improve management and estimation challenges in the context of scrum, which is an agile process widely used for the development of small to medium size software projects. The scrum emphasis on code results in spending inadequate time on the estimation process. Mostly, the scrum master, along with the scrum team, estimates the upcoming software projects based on experience or historical data. Many issues can arise in a case where expert judgment is not available or historical data are not properly organized. In this paper, an Intelligent Recommender and Decision Support System (IRDSS) is proposed that can help the scrum master to better estimate an upcoming software project in terms of cost, time, and recommendations of human resources. Formal specification of IRDSS is also performed using the formalism known as Z language. Furthermore, an experiment on fifteen web projects was performed to validate the proposed approach and compared it with Delphi and Planning Poker estimation methods. The overall results indicate that the proposed system can produce better estimation than Planning Poker and Delphi methods by applying MMRE and PRED evaluation. This research opens new directions for the scrum community for the development of software projects within the allocated time and cost.

INDEX TERMS Cost estimation, decision support system (DSS), human resource selection, intelligent recommender system (IRS), software project management, time estimation.

I. INTRODUCTION

Estimation of time and cost plays an important role in the planning and management of software projects [1], [2]. The precision of estimation has a direct effect on the output of a software project; underestimation may lead to schedule or budget overruns while overestimation may have a negative influence on organizational competitiveness [3], [4]. Research efforts on time and cost estimation are spanned over 30 years and are commonly divided into expert-based and model-based methods [5]–[7]. Expert-based methods rely on the expertise of human resources to predict new projects. Model-based approaches select the data from similar type of already developed software projects for the judgment of upcoming software projects. Most of the research [7]–[14] in this domain has focused on traditional models such as waterfall. These traditional models estimate time and cost for complete software project at the beginning of the software life cycle.

In the modern era, the scrum-agile process is mostly used for the development of software projects. In the scrum process, the software project is developed through small iterative cycles, allowing for adaptation of changing requirements at any point during a project's lifetime. An agile project may have several iterations (e.g. sprints [15], [16]) where each iteration is usually 2 - 4 weeks long, in which the development team designs, implements, tests, and delivers a distinct product increment to the client. Each iteration requires the

The associate editor coordinating the review of this manuscript and approving it for publication was Resul Dash.

completion of several user stories, which is a common way for scrum teams to address user requirements [17]. This is a shift from the traditional model where usually all functionalities are delivered together (in a single delivery), which makes estimation and decision process much complex.

In agile project management, the focus of scrum master remains on the estimation of time and cost of a single user story at a time making the estimation more reliable. It is a common practice for scrum teams to go through each user story to estimate its ''size''. Story points are commonly used as a unit of measurement for specifying the overall size of a user story [18]. Based on these story points, the scrum master estimates the time and cost required to complete a user story or sprint. In this regard, experienced scrum masters play an essential role as the time and cost estimation done by the scrum masters are inaccurate by at least 50% in majority of cases [19].

These issues of software development highlight two problems in the scrum development processes [20].

1) How better estimation can be suggested based on the best practices of successful scrum masters or historical data?
2) How scrum master can take more good decisions through gaining experience and tacit knowledge through the lessons learned?

These problems arise because in most cases all the tacit knowledge of software companies is usually lost when the scrum master leaves the company. To handle such kind of problems, this paper proposes an Intelligent Recommender and Decision Support System that will help scrum masters during estimation (of time and cost) and decision-making. The following question is formulated by keeping in mind the above-mentioned problems.

Can we develop a system to assist the scrum master in project estimation and decision making?

To answer this research question, below is our research objective.

To develop an Intelligent Recommender and Decision Support System (IRDSS) to assist the scrum master during project estimation and decision making based on the best practices of successful managers or historical data.

The rest of the paper is organized as: In Section II, a review of the literature is discussed. The detailed methodology is described in Section III. In Section IV, the prototype design and development is presented. In Section V, the demonstration and evaluation is performed while Section VI concludes the research and provides directions for future work.

II. LITERATURE REVIEW

Jamieson *et al.* [21] highlighted the scrum master aspect of estimation that leads a software toward failure. The scrum master often allocates budget without an adequate understanding of the requirements. In this case, the allocated budget does not reflect the actual scope of the project and ultimately causes delay and cost overruns. Gupta *et al.* [22] conducted a case study in the context of scrum projects. The results showed that key challenges in the transformation of scrum development process were wrong estimation of time and cost. In the same vein, Adnan and Afzal [23] designed a tool to predict the effort required for scrum development projects and uses the knowledge acquired during the development for upcoming software projects. However, their tool does not recommend human resources or cost estimation based on historical data or best practices. Furthermore, Colomo-Palacios *et al.* [24] introduced a hybrid recommender system through the semantic technologies [48] based on rough set and fuzzy set approaches. Their system maintains the tasks distribution among the team members according to their capacity. The results were found efficient regarding expert opinion between team recommendations. However, the system does not predict the time and cost based on best practices or historical data. Moreover, Lin *et al.* [25] presented two approaches (Multi-Agent System and K-CRIO ontology) for improvement of the scrum process. They used ASPECS methodology and expressed the system design through use case modeling. Their developed tool assists the scrum master in predication of efforts during development but did not support in the predication of time and cost or recommendation of human resources according to project's specifications. Besides, Yuliansyah *et al.* [26] performed a literature review related to Use Case Point (UCP) of scrum process and proposed an approach for the estimation of agile projects. Their estimation approach is a combination of use case model and scrum process. They also developed a simulation application for evaluation. The authors further suggested that there is a need to explore other factors related to estimation and development of a system that can predict effort as well as cost. In addition, Nepomuceno and Fontana [27] proposed a decision support system using scrum for effective handling of software projects. Their system facilitates the scrum master for time estimation of upcoming software projects or sprint. The scrum team enters their information including their seniority level. Then the scrum master assigns the user stories according to the seniority level of the team members. The system predicts time but does not consider project domain, relevant project type, and priority level. Further, Bhalerao and Ingle [28] critically reviewed the literature related to agile software development and identified some critical factors that play important role in the estimation of agile-scrum process. Based on these identified factors (project domain, configuration, security, complex processing, operation ease, etc.), they also developed a constructive agile estimation algorithm (CAEA). The scrum master gives input to the algorithm in terms of these factors and it estimates time for a software project. However, the algorithm does not use knowledge base or best practices for prediction. Additionally, Alhazmi [29] thoroughly reviewed the literature related to scrum project management and proposed a Sprint Planning decision Support System (SPESS) for prediction of upcoming scrum projects. Their proposed system uses Planning Poker method by considering various human factors (competency level, dependency among tasks, developers seniority) as input from scrum master and predicts the effort required to complete a particular sprint or project. They further suggested that there is a need to develop a system that can predict the time and cost of scrum projects based on historical data and developers' seniority. Likewise, Ramos *et al.* [30] enhanced the scrum process using the k-Nearest Neighbors (k-NN) algorithm for identification of the Non-Functional Requirements (NFRs) in the early stage of development. In their proposed approach, the k-NN algorithm, from the available data set, predicts the NFRs that can be implemented more effectively by the scrum development team. Their study overall indicated that for good prediction of upcoming software projects the scrum process needs further exploration in terms of estimation and human resources recommendation.

In the agile-scrum process context, the story point values method is mostly used for the estimation of effort required to complete a task. This method may introduce some challenges to the new team members, in estimating the amount of work required for each story point. Semenkovich *et al.* [31] proposed an extended scrum model using fuzzy logic approach that overcomes these issues. Their proposed model estimates the user story points based on a fuzzy inference system using team members' opinions, such as complexity of work and the amount of work with different values (very easy, easy, normal, difficult, very difficult). The overall results showed that the system predicts more accurately for new scrum team members but not suitable for experienced scrum team members. Similarly, Stupar *et al.* [32] proposed a decision support system for the scrum process based on fuzzy logic. Their system minimizes the requirement uncertainty and helps the scrum master in efforts estimation. Their system does not predict the cost or recommend appropriate human resources according to the project requirements. They also suggested that more tools are required in the scrum context to complete the product owner's requirements within allocated time and cost.

Furthermore, Çetin and Onay Durdu [33] conducted a case study in two IT organizations in Turkey to determine the use of scrum model. After a case study, they also conducted a survey from different IT professionals to add a new role of project leader in the scrum model to handle the issues among senior management and the scrum team. Their proposed blended model handles the issues of prediction and recommendations based on the new role instead of historical data or best practices. In a similar vein, Soni and Kohli [34] proposed a model for prediction of web-based applications using an algorithmic approach. Their model used function points for the estimation of user story instead of user story points. However, the model only predicts the cost of web application but did not recommend the human resources according to the requirements of the product owner.

The above literature indicates that scrum process is mostly used by software companies for the development of small to medium size software projects. In the scrum process, the scrum master predicts time and cost at an early stage that determines the overall output of the software project. In case, the expert of the same domain is not available, the software prediction is less accurate that results in exceeded time and cost of software projects. Further, the literature also showed that there is a need to develop a system that can appropriately recommend human resources at an early stage and predict the time and cost of scrum projects based on historical data. The inappropriate estimate of time and cost profoundly impacts the outcome of software projects [3], [35]–[38].

III. RESEARCH METHODOLOGY

Design science research (DSR) [39] approach has been used in applied research which combines the design of mixed method and evaluation cycle methodology [39]. The following three phases were considered in the DSR methodology and are mapped to our research problem.

a) The first phase was problem identification and motivation to examine and articulate the research question.
b) The second phase (prototype design and development) consists of three activities: In the first activity, an algorithm was developed for intelligent recommender and decision support system while in the second activity, the formal specification of the proposed algorithm is devised in Z language. In the third activity, an intelligent recommender and decision support system was developed. The system helps the scrum master to make better estimation of cost and time to meet customers' expectations and time-to-market.
c) The third phase (prototype demonstration and evaluation) consists of two further activities: In the first activity, the evaluation was performed to validate the algorithm on new software projects. In the second activity, the research question was answered.

The flow diagram of the adapted methodology is shown in Figure 1.

A. PROBLEM IDENTIFICATION PHASE

In the problem identification phase, a search strategy was defined. The purpose of defining the search strategy was to find an extensive and unbiased method for gathering of research material related to the research question. The search strategy was developed to maximize the probability to search the relevant studies in the problem domain. For this purpose, popular databases in the software engineering research domain, as described in [40] were used, including IEEE Digital Library, ACM Digitial Library, SpringerLink, and ScienceDirect. These databases were used along with Google Scholar for verifying the collected results and performing some meta-investigations. The search strings used to explore these databases were based on words listed in Table 1.

Our search keys stem from the research question, while the Boolean operator "AND" was utilized to join the category 1 and category 2 keywords. Through this search, all the articles that focus on scrum and estimation process were retrieved. Inclusion and exclusion criteria to address the

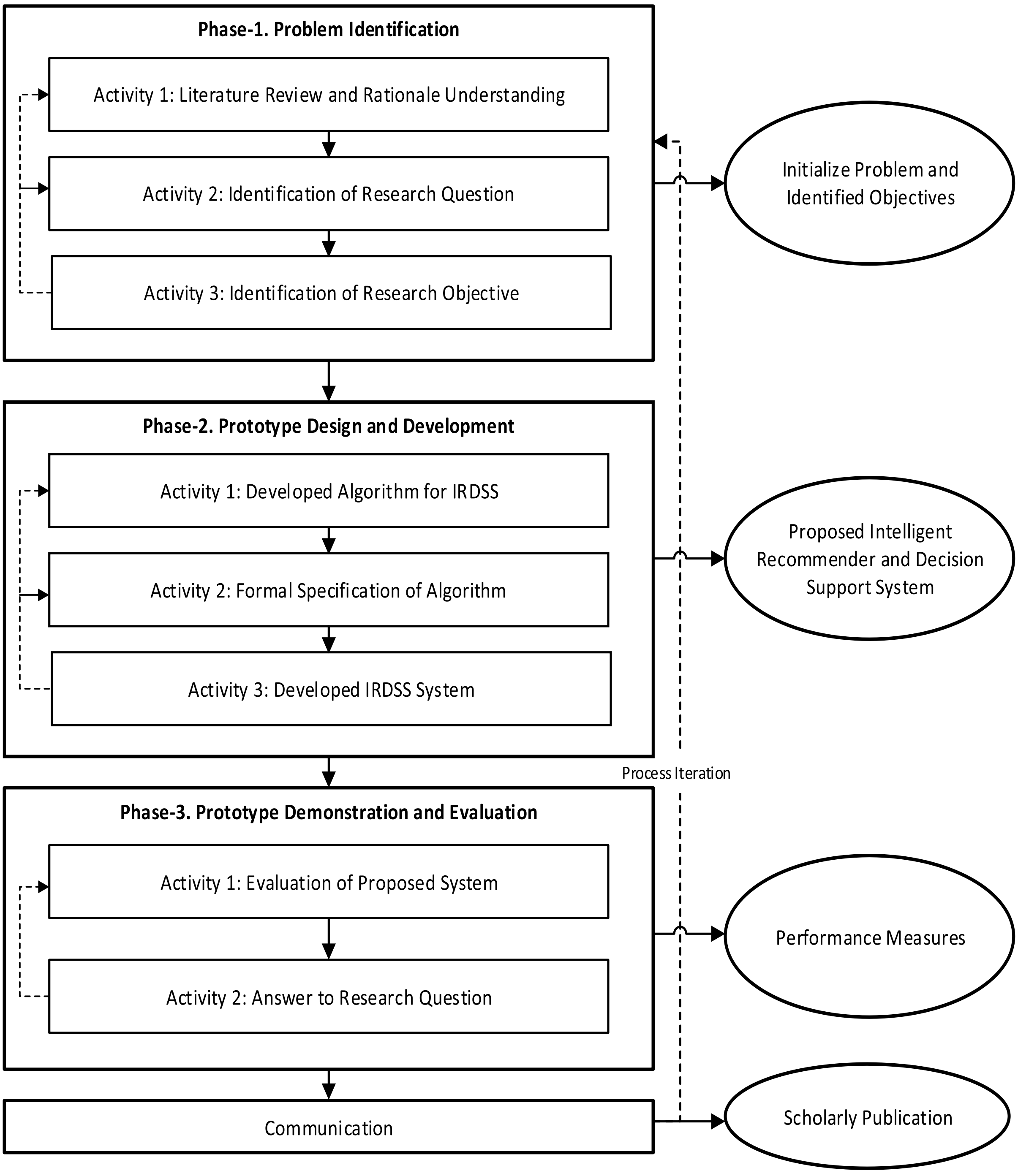

FIGURE 1. Design science research workflow for intelligent recommender and decision support system.

research question of the current investigation are shown in Table 2.

Based on the findings of the literature, a research question was formulated and the objective of the research was articulated.

B. PROTOTYPE DESIGN AND DEVELOPMENT PHASE

The second phase consists of three activities. In the first activity, an algorithm was developed for the estimation (time, cost) and human resource recommendation. In the second activity, a formal specification of the proposed algorithm is devised.

Petri nets and Z language [41] are mostly used for formally modeling the behavior of a system. Petri nets model of a system can be simulated on a paper and the behavior of the model can be analyzed and verified, while Z language is suitable to describe data types, sets of operation, labeling, and constraints of a system.

TABLE 1. Search terms used in the review.

	Category	Keywords
1.	Use of scrum practices	Scrum, Scrum practice, Scrum method, Scrum methodology, Scrum process, Scrum model, Scrum technique
2.	Estimation process	Issue in the estimation process Cost estimation, Time estimation, Effort estimation, Problem in the estimation process, Improvement in the estimation process

TABLE 2. Inclusion and exclusion criteria used in this investigation.

Type	Description
1.	Only those papers were considered which, 1) Belong to popular computing research databases, including ACM, IEEE, Elsevier, Springer and Google Scholar. 2) Discussed estimation process in scrum, their issues and future improvements. 3) Investigated the factors influencing software estimation in scrum process. 4) Provided evidence of estimation on software project estimation failure / success through case studies, experience reports, or field studies. 5) Are published during the years 2010 to 2019.
2.	Exclude the papers, which 1) Belong to non-indexed journals, books, master or doctoral dissertations and the papers that did not undergo a proper peer-review process. 2) Belong to hardware or other fields rather than software engineering. 3) Are not relevant to the research questions. 4) Concern news issues or related experience in software project failure. 5) Are not written in the English language. 6) Are duplicate (select the latest version).

However, the use of Z language has several advantages over Petri nets. For instance, data abstraction and modification cannot be observed in Petri nets as it has a graphical representation. Further, the Petri nets model specifies sets of operations and defines operations occurrence in a certain order. On the other hand, Z language defines the set of operations without enforcing any order. Due to these advantages of Z language over Petri nets, the proposed algorithm was formulated in Z language.

Based on the formal proof using Z language, the IRDSS was developed in the third activity. The IRDSS has the potential to assist the scrum master during decision making and estimation of time and cost for upcoming software projects or sprint.

C. PROTOTYPE DEMONSTRATION AND EVALUATION PHASE

In this phase, the experimental interfaces for IRDSS were designed for evaluation. Based on this prototype, an experiment on fifteen web projects was conducted. Mean Magnitude Relative Error (MMRE) and Percentage Relative Error Deviation (PRED) (prediction at level x) [42] estimation evaluation criteria were also applied, which are commonly used in software estimation. MMRE and PRED are most widely used to measure the accuracy of software project estimation models calculated through the Magnitude of Relative Error (MRE) measurement. MMRE is calculated by averaging the MRE of each project in the data set. MMRE is an evaluation technique that is used to assess the efficiency of the effort to be estimated. PRED is an aggregate of the percentage of MRE which is less than or equal to 0.25. In the next activity, the answers to research questions were discussed.

IV. PROTOTYPE DESIGN AND DEVELOPMENT

The prototype design and development phase contains three activities. In the first activity, an algorithm was proposed for IRDSS. In the second activity, the formal specification of the algorithm was devised, while in the third activity, the actual system was implemented.

A. PROPOSED ALGORITHM FOR IRDSS

In scrum development, it is important to make an appropriate estimate of the time and cost required to complete a user story within the allocated budget. The most commonly used estimation methods in scrum are estimation by analogy, Delphi method, Planning Poker, and hybrid approach. In the analogy method, the scrum master makes estimations based on the historical data of already developed similar software projects. However, the estimation can go wrong if the data of existing software projects were not properly saved. In the Delphi method, a group of experts decides the required time and cost to complete the software project. However, the availability of a group of experts of the same domain may not be feasible for some companies. Also, if an expert leaves a company all the tacit knowledge is lost. In the Planner Poker method, the scrum team estimates the user story based on the project's requirement. Planning Poker is mostly used in the scrum process for estimation but it has some drawbacks as it does not consider other factors such as seniority of team members, task dependencies, priority level, and domain or project-specific requirements. The hybrid method is a combination of the above techniques. In this paper, the hybrid approach was used to develop the algorithm to estimate the upcoming software projects. The mathematical model of the IRDSS works as follows.

$$X_{IRA} = \mathbf{F}(Y_{LA}, Z_{PA}, H_{FA}, G_{HRA});$$
$$Y_{LA}, Z_{PA}, H_{FA}, G_{HRA} >= 0;$$

where;

$$\begin{aligned} Y_{LA} &= \sum\nolimits_{i=1}(Z_{PA}); \\ Z_{PA} &= \sum\nolimits_{i=1}(H_{FA} + Y_{LA}); \\ H_{FA} &= \sum\nolimits_{i=1}(G_{HRA} + Z_{PA}); \\ G_{HRA} &= \sum\nolimits_{i=1}(H_{FA}); \end{aligned}$$

The above mentioned variables are defined as:

Y_{LA} = (Learning Agent Variable)
Z_{PA} = (Planning Agent Variable)
H_{FA} = (Function Agent Variable)
G_{HRA} = (Human Resource Agent Variable)
X_{IRA} = (Intelligent Recommender Agent Variable)

Intelligent recommender agent variable (X_{IRA}) contains the result of the learning agent, planning agent, function agent, and human resource agent. It further indicates that the values of learning agent, planning agent, function agent, and human resources agent are non-negative in the knowledge base. The product backlog phase contains all the requirements (user stories) of the project. On the request of the scrum master, the intelligent recommender agent indicates the other agents to fetch the required information from the knowledge base.

Learning agent variable (Y_{LA}) contains the product backlog information through planning agent variable (Z_{PA}) and interacts with the knowledge base. It takes the decision according to the requirement specification from the scrum master and quickly recommends the latest relevant lessons learned.

Planning agent variable (Z_{PA}) interacts with the learning agent and function agent. The function agent (H_{FA}) combines the high priority user stories in the form of functions to be executed in the upcoming sprint and the learning agent suggests the latest learned lessons from the previous sprint of the same domain to the scrum master.

Function agent variable (H_{FA}) contains the result of the planning agent and human resource agent. It indicates the average team size and iteration length of a sprint.

Human resource agent variable (G_{HRA}) interacts with the function agent that indicates an average number of functions to be executed in the upcoming sprint.

Finally, the intelligent recommender agent variable (X_{IRA}) compiles the results and responds to the scrum master.

B. FORMAL SPECIFICATION OF ALGORITHM FOR IRDSS

The formal specification in Z language allows expressing a systems model to help communicate in a mathematical and precise way. Such communication helps to specify details and associated complications into simpler ones using abstraction. Also, such formal specification of a system holds the essential properties of the notions and provides the facility of mathematical proof. For further details, readers are referred to [41] for the terms, notations, and concepts in the Z language. The proposed algorithm explains various steps involved in the estimation of projects in the scrum environment. The following assumptions are made for formal specification of the proposed algorithm in Z language:

- The amount of time is based on the discrete time units.
- The cost is also in the form of discrete points on non-negative integer line.
- Knowledgebase is a set of tuples where each tuple contains the project types, category, number of team members, time, and cost to complete a project.

In guarded command language [41], the algorithm for IRDSS is defined as the sequence of steps, described below:

IRDSS ⊑ ProjectType; GetStoryPoint; SelectDataValue; RemoveOutlier; TeamMemberOutlier

where '⊑' is the notation used for 'is defined as' and *ProjectType*, *GetStoryPoint*, *SelectDataValue*, *RemoveOutlier,* and *TeamMemberOutlier* are the names of schemas that specify the sequence of steps in the algorithm.

Further, to specify each step of the algorithm in Z language, following are the declarations of data types used in the formal specification:

[ProjectType] is the type of having all project types.

Category ::= web | mobile | desktop

where Category is a free type definition having three values, viz. *web, mobile,* and *desktop*.

KNWGBASE ⊑ ProjectType × Category × Tmember × Time × Cost

where *Tmember, Time,* and *Cost* are the sets of non-negative integers.

To define the schema structure for each step of the algorithm for IRDSS, following non-negative integer values are declared (Figure 2).

$max\_spv : \mathbb{N}$
$max\_time : \mathbb{N}$
$max\_cost : \mathbb{N}$
$max\_tmember : \mathbb{N}$

FIGURE 2. Integer declaration schema.

ProjectType
record:ℙKNWGBASE
prjtype:Project
ctgry:Category

prjtype ∈ dom *record*
ctgry ∈ second*(record)*

FIGURE 3. Schema of defining project type.

Figure 3 represents the global state of the IRDSS and specifies the selection of the project type using the variable *record*.

Figure 4 explains the procedure of getting the story point value.

The next step of the algorithm viz. selecting the data value, is specified in Figure 5. The figure explains selecting integer values of time, cost, and team members. Further, these values are considered as input values in the Z language.

The step of removing the outlier is specified in Figure 6. The schema explains the procedure of defining the outlying values, where LCB is the lower class boundary value calculated through the lower limit of the confidence interval for the sample mean using t-distribution. Whereas, UCB is the

GetStoryPoint

$\Xi ProjectType$
$spv!: 0..max\_spv$

$spv! > 0 \wedge spv! \leqslant max\_spv$

FIGURE 4. Schema of defining get story point.

SelectDataValue

$\Xi ProjectType$
$time?: 0..max\_time$
$cost?: 0..max\_cost$
$tmember?: 0..max\_tmember$

$time? > 0 \wedge time? \leqslant max\_time$
$cost? > 0 \wedge cost? \leqslant max\_cost$
$tmember? > 0 \wedge tmember? \leqslant max\_cost$

FIGURE 5. Schema for selecting data value.

RemoveOutlier

$\Delta ProjectType$
$LCB, UCB: \mathbb{N}$
$ave\_time: 0..\ max\_time$
$ave\_cost: 0..\ max\_cost$

$ave\_time < LCB \vee ave\_time > UCB \Rightarrow outlier$
$ave\_cost < LCB \vee ave\_cost > UCB \Rightarrow outlier$

FIGURE 6. Schema of removing outlier.

upper class limit of the confidence interval form mean value using t-distribution. Further, lower and upper limits of the confidence interval are rounded to the integer values to get LCB and UCB respectively.

Finally, the step of deciding the team member outlier is specified in Figure 7. The time for each team member is counted, in the set *tm*, and is considered as an outlier if is less than LCB or greater than the UCB. Similarly, if the counted cost for a member in the set *tm* is less than LCB or greater than UCB, it is considered as an outlier and removed from the set *tm*.

C. IMPLEMENTATION

The proposed IRDSS becomes activated by the scrum master's request for the allocation of human resources and estimation of time and cost for an upcoming user story. The proposed system has two main components. The first component, decision support system, gives the decision to the scrum master regarding how much time and cost is required for completing a user story. The scrum master specifies the

TeamMemberOutlier

$\Delta ProjectType$
$tm: \mathbb{P}0..max\_tmember$

$\forall m: tm \bullet Count(time) \bullet\ time < LCB \vee time > UCB$
$\forall m: tm \bullet Count(cost) \bullet\ cost < LCB \vee cost > UCB$
$tm' = tm - \{m\}$

FIGURE 7. Schema for team member outlier.

software domain, set of already completed relevant projects, story point values, priority type and total team members to initiate the estimation process. The decision support system then analyzes the current knowledge base as per the scrum master's requirements. The second component, intelligent recommender, helps the scrum master regarding the appropriateness of team members and their seniority level as required to develop a particular user story to complete in optimal cost and time. It also checks the priority level of user story. If the priority is high, intelligent recommender chooses the team members that have experience with minimum time and overlooks the cost factor, otherwise, it shows the average time and cost. Finally, the algorithm compiles an estimation report for scrum master as per their request. The pseudocode of the IRDSS algorithm indicates that the scrum master selects the category, relevant project type, story point, team member, and priority. Based on these parameters, the algorithm selects the time, cost, and team members from a knowledge-base. The pseudocode of the IRDSS algorithm is given below.

Input: category, project type, story point, team member, priority.
Output: time, cost, team members.

Begin:

Get **category**, **projectType**, **storyPoints,** **teamMember** and **priority** from scrum master.
scrumMasterResult = Select **time**, **cost**, **team members** from **ScrumMasterTable**
where **category=?** and **projectType=?** and **storyPoints=?** and **teamMember=?** and **priority=?**
K-means (**scrumMasterResult**) and return **scrumMasterFiltered**
    scrumMasterFiltered = **K-MeanAlgorithm** (**scrumMasterResult**)
        Search (nearest centroid point)
        OutlinerIdentification (point to the cluster of time and cost)
        CalculateMean (all the assigned points in a cluster)
    Store time and cost as got after applying **K-Mean algorithm** in a **scrumMasterFiltered**
        Map Definition

```
11.        Map <Integer, List> scrumMasterFiltered
            = new HashMap <Integer, List> ();
12.        List list = {time, cost, teamMembers}
13.        scrumMasterFiltered.put (teamMembers,
            list)
14.        If IsNotEmpty (scrumMasterFiltered)
15.             return scrumMasterFiltered
16.  K-NN (scrumMasterResult) and return
     scrumMasterRecommendations
17.     scrumMasterRecommendations = K-NN
         Algorithm (scrumMasterResult)
18.        CalculateTimeAndCost (scrumMasterResult)
19.        initialValue {} = FindMinimum (time and cost
            in k-means)
20.        For each point in initialValue
21.             Search (EuclideanDistance (observed
                 points and outliers))
22.             StoreInMap (observed points and
                 outliers)
23.             SortMap (clusters)
24.             Minimum (nearest neighbors)
25.     StoreInMap (time and cost after applying K-NN
         algorithm)
26.         List list = {time, cost, teamMembers}
27.         scrumMasterRecommendations.put
            (teamMembers, list)
28.     If IsNotEmpty (scrumMasterRecommendations)
29.     return scrumMasterRecommendations
```

End

Further, k-mean [43], an unsupervised clustering method, is applied on scrum Master Result to pick the team members clusters from the knowledge base and save in a scrumMasterFiltered, based on the above-mentioned parameters (category, relevant project type, and story point value). The nearest centroid point is selected and assigned to the cluster for the identification of time and cost outliers. It then calculates the mean of all the assigned points in a cluster. Besides, it removes the outliers of time and cost using a t-distribution method from scrumMasterFiltered. Moreover, it calculates the average cost and time of user stories which concerns team members and shows the result to the scrum master. Additionally, a supervised learning algorithm, the k-nearest neighbors (k-NN) method [44], [45], is applied on scrumMasterResult, to find the nearest neighbors. The outliers of time and cost are counted separately which concern team size and further the high category of outliers are removed from the scrumMasterResult. Furthermore, it stores all recommendations having total team size along with time and cost in scrumMasterRecommendations map and shows it to the scrum master.

Moreover, Figure 8 shows an interface that explains the output of the system. The figure shows that if a scrum master enters story point value of 8 with 5 team members, the system estimates the required time and cost as 4 days and 16,175 PKR. Figure 8 also shows that when the user story point priority is high then the system recommends the scrum master to go with 8 team members instead of 5 team members.

V. DEMONSTRATION AND EVALUATION

The demonstration and evaluation phase contains two activities: In the first activity, the experiment and results were discussed, while in the second activity, the research question was answered.

A. EXPERIMENT AND RESULTS

A team of five professionals (one analyst, one designer, one quality engineers, and two developers) having at least five-year experience from a reputed software company working on scrum process was chosen to perform an experiment. To perform the experiment, fifteen web-based projects using SQL and C# technologies from an e-commerce domain were selected by the team. These projects were selected because the said team developed these projects in the last 5 years using these technologies. The team labeled the projects starting from P1 to P15, each having a product backlog of twenty to thirty features. The team was requested to measure the time in person-day unit and cost in PKR using multiple methods (Planning Poker and Delphi) along with our proposed approach. In the case of Delphi estimation method, two experts from the said team were selected to predict the e-commerce web project average estimates.

Further, the said team conducted a Planner Poker method in front of all stakeholders of the project and evaluated on the base of chosen card value ranging from 0 – 200.

The project owner selected a user story from project backlog phase and read that to scrum team for estimation. The scrum team member carried out a discussion on the selected user story and asked related questions. Each team member then secretly picked up one card to represent his/her estimate. Once all the team members were done with the estimation, then they are exposed to all the other members.

If everyone selected the same card, they already have consensus, and that number became the project backlog estimate. If the estimates were different, the team had a discussion to provide a justification and re-estimate until a consensus was reached.

For the required cost and time for all web projects, the proposed approach was carried out. Table 3 represents the computed values of estimated time calculated through Planning Poker, Delphi, and the proposed method. Whereas, visual comparison of Table 3 is portrayed in Figure 9.

The actual cost, as well as the detailed cost for the determination of estimated value by using different estimation techniques, were also collected and examined in the experiment. Table 4 represents the comparison between the actual cost with the detailed estimated cost, while the visual comparison of Table 4 is portrayed in Figure 10. According to Conte *et al.* [42], a software estimation approach is considered accurate when MMRE ≤ 0.25 and PRED $(0.25) \geq 0.75$.

Further, we have used PRED (x) and MMRE measures to analyze the results. The measure MRE is defined as MRE

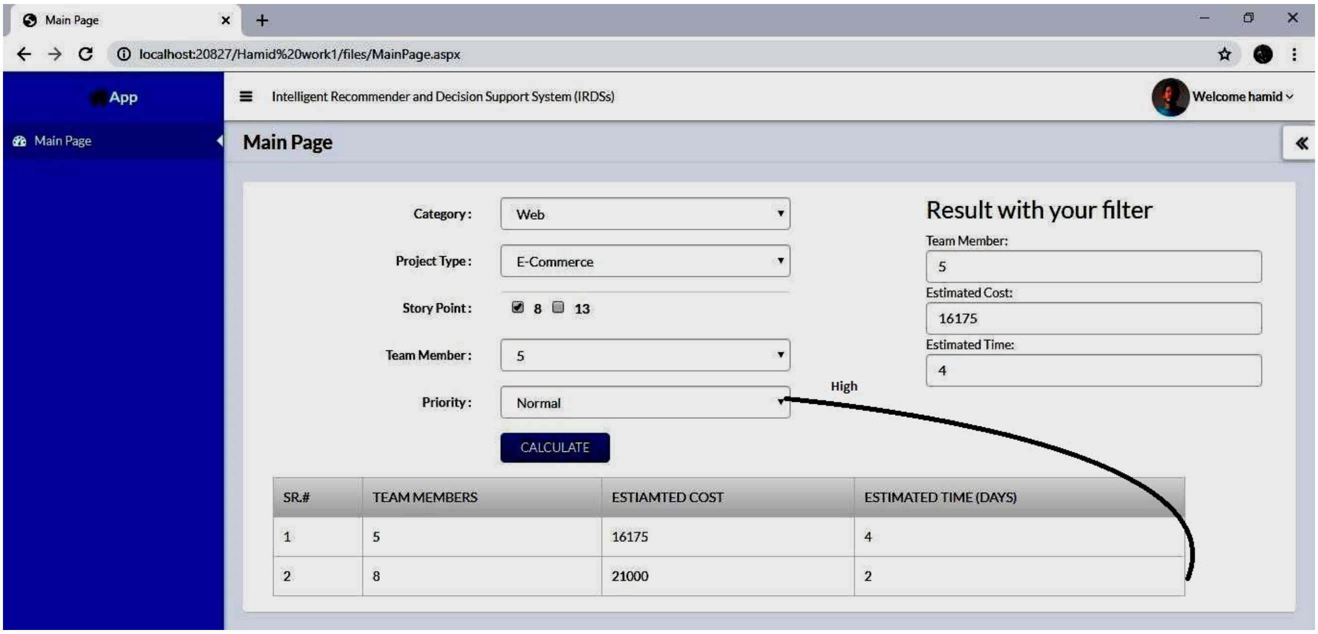

FIGURE 8. Interface showing input of scrum master and proposed reliable output of IRDSS.

TABLE 3. Comparative evaluation of time predicted by estimation techniques using magnitude of relative error.

	Estimated Time (Person Days)				Magnitude of Relative Error (MRE)		
Project id	Proposed Approach	Delphi	Planning Poker	Actual Time	Proposed Approach	Delphi	Planning Poker
P1	150	100	152	142	0.056	0.296	0.07
P2	203	191	210	161	0.261	0.186	0.304
P3	128	129	130	109	0.174	0.183	0.193
P4	107	82	104	126	0.151	0.349	0.175
P5	143	110	147	130	0.1	0.153	0.131
P6	132	144	128	135	0.022	0.067	0.052
P7	158	170	153	162	0.025	0.05	0.056
P8	142	153	139	146	0.027	0.048	0.048
P9	126	131	131	124	0.016	0.056	0.056
P10	135	103	129	139	0.0288	0.259	0.0712
P11	81	79	80	88	0.08	0.102	0.091
P12	119	115	113	125	0.048	0.08	0.096
P13	73	89	65	80	0.088	0.113	0.186
P14	83	82	71	99	0.162	0.172	0.283
P15	128	136	132	124	0.0322	0.097	0.065

$= (AT - ET)/AT$, where ET represents the estimated time and AT gives the actual times respectively, while PRED $(x) = L/N$, where 'L' is the number of observations where MRE $\leq x$ and 'N' is the total number of observations in the set [46], [47]. Similarly, for cost estimation, the measure is defined as MRE $= (AC - EC)/AC$, where EC represents the estimated cost and AC gives the actual cost respectively.

The results indicated that one MRE value (0.25) of proposed approach is greater against project P2, while two MRE values are greater than against two projects labeled as P2 and P14 in the Planning Poker method. Further, in the Delphi method, three MRE value (0.25) is greater than Project P1, P4, and P10. The results indicate that the proposed approach gives better accuracy in prediction as compared to both the

TABLE 4. Comparative evaluation of cost predicted by estimation techniques using magnitude of relative error.

	Estimated Cost (In PKR)				Magnitude of Relative Error (MRE)		
Project id	Proposed Approach	Delphi	Planning Poker	Actual Cost	Proposed Approach	Delphi	Planning Poker
P1	198000	120000	210000	200000	0.01	0.4	0.05
P2	285000	232000	290000	226000	0.261	0.027	0.28
P3	150000	180000	165000	152000	0.013	0.184	0.086
P4	160000	125000	172000	170000	0.059	0.265	0.012
P5	171000	161000	179000	173000	0.012	0.069	0.035
P6	177500	182000	175000	178000	0.003	0.022	0.017
P7	220500	250000	210000	220000	0.002	0.136	0.045
P8	250000	216000	220000	210000	0.190	0.029	0.048
P9	162000	163000	170000	160000	0.013	0.019	0.063
P10	130000	128000	131000	180000	0.278	0.289	0.272
P11	129000	115000	129100	130000	0.008	0.115	0.007
P12	159000	155000	160000	161000	0.012	0.037	0.006
P13	126000	133000	123000	128000	0.016	0.039	0.039
P14	121000	117000	98000	135000	0.104	0.133	0.274
P15	167000	210000	168000	160000	0.044	0.313	.05

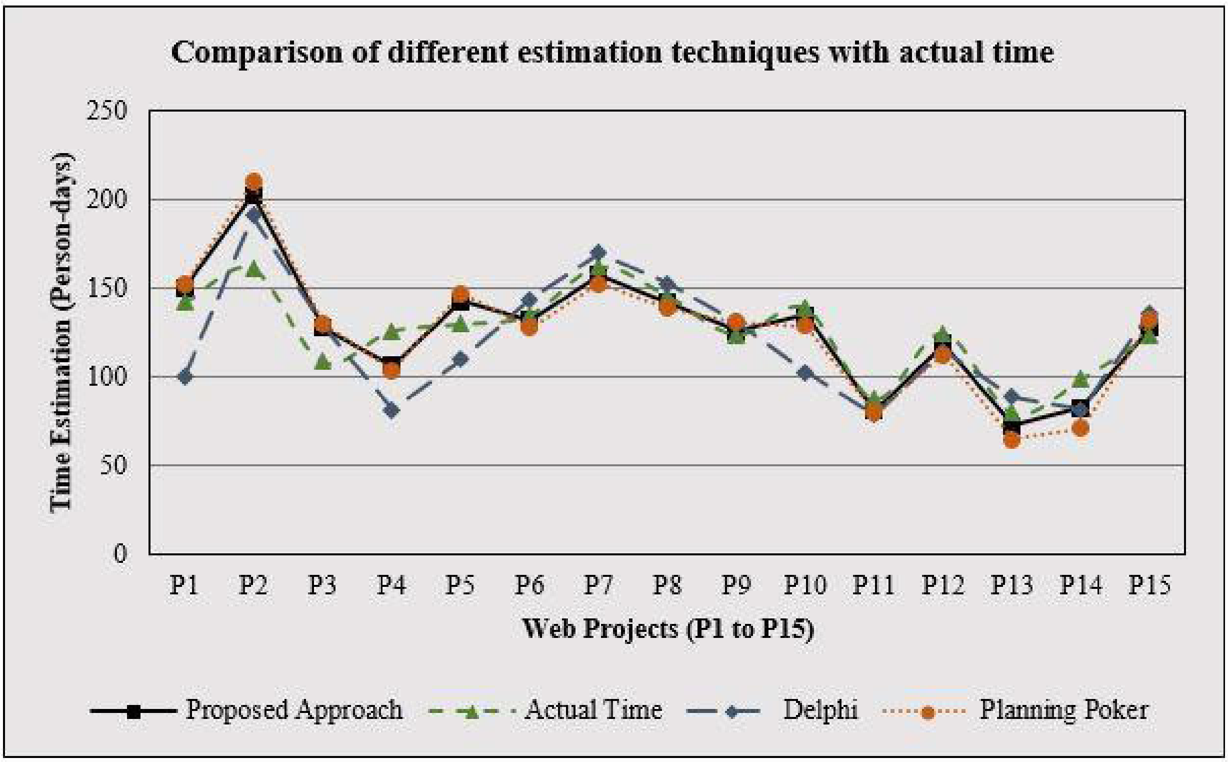

FIGURE 9. Comparison of different estimation techniques with actual time.

other methods as per the recommendation of PRED (25) and MMRE measure.

The detailed results are given in Table 5 while the complete comparison is demonstrated in Figure 11. The MMRE value of the proposed approach is less than Delphi and Planning Poker methods (i.e., 0.08 < 0.147 and 0.08 < 0.125, respectively). Moreover, the PRED (25) value of proposed approach is 0.93 > 0.80 suggest it as a perfect prediction model as per the PRED (25) measure. For PRED (25) of both the Delphi and Planner Poker method, both the values 0.80 and 0.86 are less than 0.93.

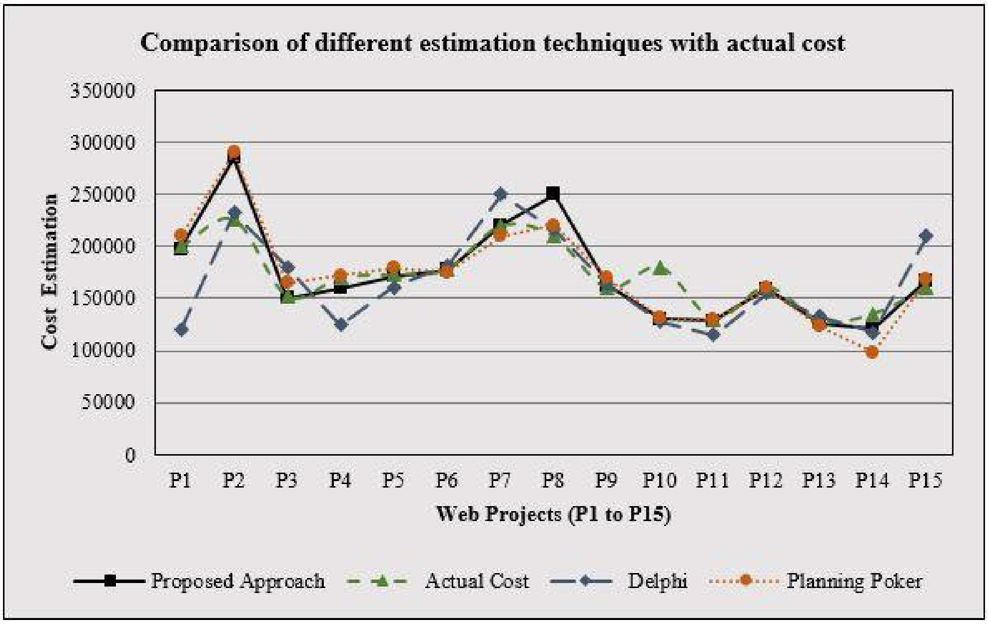

FIGURE 10. Comparison of different estimation techniques with actual cost.

TABLE 5. Applying evaluation measures and obtained results for time estimation.

Methods	MMRE	PRED (25)	Estimation Accuracy
Proposed Approach	0.08	0.93	93 %
Delphi	0.147	0.80	80 %
Planning Poker	0.125	0.86	86 %

It indicates that there is a 93% chance of accurate prediction by using the proposed approach, in the forecasting of the assigned web estimation. The results conclude that the proposed approach carries more accurate estimates of time than Delphi and Planning Poker methods.

TABLE 6. Applying evaluation measures and obtained results for cost estimation.

Methods	MMRE	PRED (25)	Estimation Accuracy
Proposed Approach	0.07	0.87	87 %
Delphi	0.14	0.73	73 %
Planning Poker	0.09	0.80	80 %

In the following, we have used the PRED (x) and MMRE measures to analyze the results for the cost. The results indicate that two MRE values (0.25) of the proposed approach are greater against project P2 and P10, while three MRE values of 0.25 are greater against three projects labeled as P2, P10, and P14 in Planning Poker method. In the Delphi method, four MRE values (0.25) are greater against Project P1, P4, P10 and P15. The results indicate that the proposed approach gives better accuracy in prediction compared to both of the other methods. The detailed results are given in Table 6, while the complete comparison is demonstrated in Figure 12.

The MMRE value of proposed approached is less than Delphi and Planning Poker methods (i.e., $0.07 < 0.14$ and $0.08 < 0.09$, respectively). Moreover, the PRED (25) proposed approach value of $0.87 > 0.73$ suggests it as a perfect prediction model. For PRED (25) of both the Delphi and Planner Poker method, the value 0.87 is greater than 0.80 and 0.73. It indicates that there is an 87% chance of accurate prediction by using the proposed approach in the forecasting of the assigned web estimation. The results reflect that the proposed approach carries more accurate estimates of a cost than Delphi and Planning Poker methods.

B. ANSWER TO RESEARCH QUESTION

The answer of research question formulated at the beginning of this research *"Can we develop a system to assist the scrum master in project estimation and decision making?"* starts from Section II (literature review) which shows that the scrum master face different issues during software project development using scrum process. The major influencing issues are estimation and inappropriate decision making. The proposed system (IRDSS) improves the decision making and estimation process by giving a quality prediction for better project completion. It assists the scrum master during project estimation and decision making based on the best practices of successful managers and historical data. In the scrum process the tacit knowledge is usually lost when the scrum master leave the company.

IRDSS will save the lessons learned (tacit knowledge) from developed projects in a well-organized manner that can be further utilized in the estimation of upcoming software projects. Furthermore, the IRDSS has used the mechanism of data mining technique k-NN and k-mean clustering algorithm for prediction. It helps the scrum master in effective

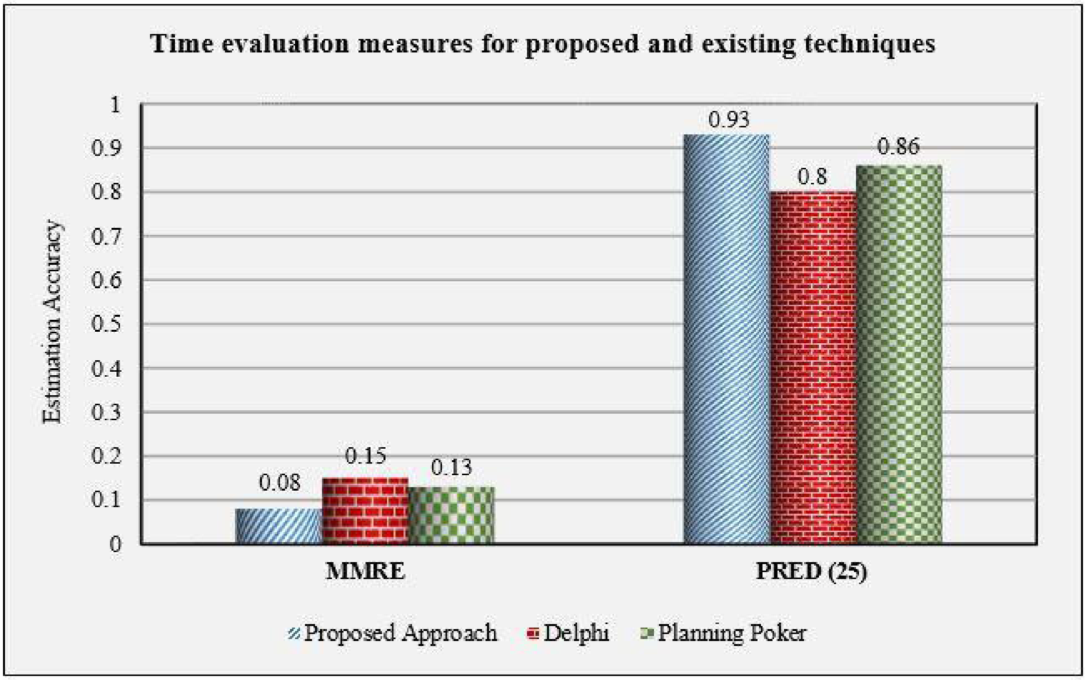

FIGURE 11. Time evaluation measures for proposed and existing techniques.

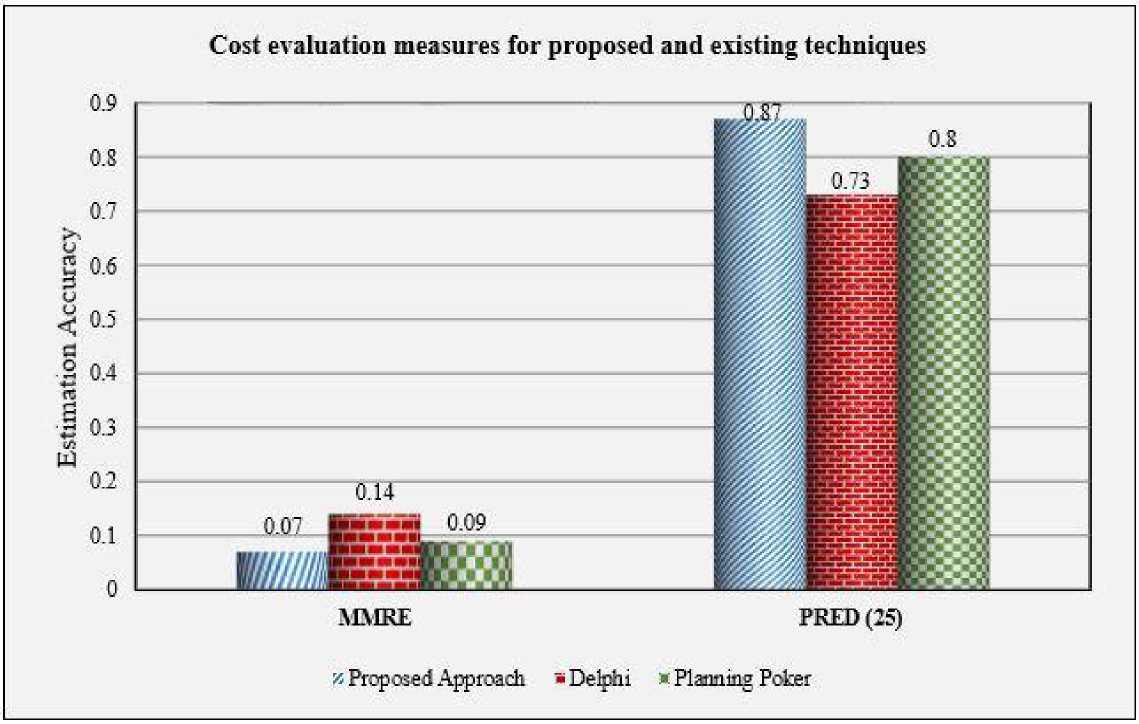

FIGURE 12. Cost evaluation measures for proposed and existing techniques.

predication of time and cost using various attributes. In this system, various attributes are taken as input such as category, project type, story point, team member, and priority. For the grouping of various attributes of time and cost, it uses the k-mean algorithm. Further, for classification and recommendation of human resources, it used k-NN algorithm.

VI. CONCLUSIONS AND FUTURE WORK

This paper aimed to develop an intelligent recommender and decision support system for the scrum process for effective handling and management of software projects. The proposed system assumes that the scrum master is the main user of the proposed approach. The scrum master initiates a request that describes business domain requirements related to which they estimate. These requests are forwarded to the system that activates and assigns the work to the relevant agents. Agents start analyzing the knowledge base as per the scrum master's requirements. Once all the agents accomplish their assigned goals, their work is compiled together in the form of an estimate and presented to the scrum master as per their respective requests.

Further, an algorithm has been developed for these tasks. The proposed algorithm explains various steps involved in the recommendation of human resources and estimation of projects in a scrum environment. Furthermore, formal specification of IRDSS is also performed using the formalism known as Z language.

Moreover, the IRDSS was developed for the allocation of human resources and estimation of time and cost for upcoming user stories. The proposed system has two main components: The first is decision support system, which gives the decision to the scrum master regarding how much time and the cost is required for completing a user story. The second component is intelligent recommender, which helps the scrum master regarding how much appropriate team members, along with their seniority level, are required to develop a particular user story to complete on optimized cost and time.

An experiment on fifteen projects was performed to check the validation of the proposed approach, with Delphi and Planning Poker estimation methods. The overall results by applying MMRE and PRED as evaluation measures indicate that the proposed approach produce better estimation results than Planning Poker and Delphi methods.

There are several directions where this research can be extended, for instance, the current scrum tool does not reuse the already developed user stories in new projects. Also, the proposed system can be enhanced through a machine learning algorithms and ontology knowledge-based. Furthermore, many other functionalities can be incorporated in our system, such as scope management, risk management, security, and reliability for better estimation of upcoming software projects. The enhancement of these features will positively influence geographically distributed development teams.

REFERENCES

[1] R. Popli and N. Chauhan, "A sprint point based tool for agile estimation," in *Software Engineering*. Singapore: Springer, 2019, pp. 63–72.

[2] S. Tahsin, A. Munim, M. Hasan, N. Nahar, and M. Rokonuzzaman, "Market analysis as a possible activity of software project management," in *Proc. IEEE 16th Int. Conf. Softw. Eng. Res., Manage. Appl. (SERA)*, Jun. 2018, pp. 180–183.

[3] H. Rastogi, S. Dhankhar, and M. Kakkar, "A survey on software effort estimation techniques," in *Proc. 5th Int. Conf.-Confluence Next Gener. Inf. Technol. Summit (Confluence)*, Sep. 2014, pp. 826–830.

[4] S. Dhir, D. Kumar, and V. Singh, "Success and failure factors that impact on project implementation using agile software development methodology," in *Software Engineer*. Singapore: Springer, 2019, pp. 647–654.

[5] T. Menzies, Z. Chen, J. Hihn, and K. Lum, "Selecting best practices for effort estimation," *IEEE Trans. Softw. Eng.*, vol. 32, no. 11, pp. 883–895, Nov. 2006.

[6] M. Jørgensen, "A review of studies on expert estimation of software development effort," *J. Syst. Softw.*, vol. 70, nos. 1–2, pp. 37–60, Feb. 2004.

[7] M. Usman, R. Britto, L.-O. Damm, and J. Börstler, "Effort estimation in large-scale software development: An industrial case study," *Inf. Softw. Technol.*, vol. 99, pp. 21–40, Jul. 2018.

[8] P. Sentas, L. Angelis, I. Stamelos, and G. Bleris, "Software productivity and effort prediction with ordinal regression," *Inf. Softw. Technol.*, vol. 47, no. 1, pp. 17–29, Jan. 2005.

[9] A. Panda, S. M. Satapathy, and S. K. Rath, "Empirical validation of neural network models for agile software effort estimation based on story points," *Procedia Comput. Sci.*, vol. 57, pp. 772–781, Jan. 2015.

[10] F. Sarro, A. Petrozziello, and M. Harman, "Multi-objective software effort estimation," in *Proc. 38th Int. Conf. Softw. Eng. (ICSE)*, 2016, pp. 619–630.

[11] M. Jorgensen and M. Shepperd, "A systematic review of software development cost estimation studies," *IEEE Trans. Softw. Eng.*, vol. 33, no. 1, pp. 33–53, Jan. 2007.

[12] F. Collopy, "Difficulty and complexity as factors in software effort estimation," *Int. J. Forecasting*, vol. 23, no. 3, pp. 469–471, Jul. 2007.

[13] E. Kocaguneli, T. Menzies, and J. W. Keung, "On the value of ensemble effort estimation," *IEEE Trans. Softw. Eng.*, vol. 38, no. 6, pp. 1403–1416, Nov. 2012.

[14] K. Chari and M. Agrawal, "Impact of incorrect and new requirements on waterfall software project outcomes," *Empirical Softw. Eng.*, vol. 23, no. 1, pp. 165–185, Feb. 2018.

[15] A. Srivastava, S. Bhardwaj, and S. Saraswat, "SCRUM model for agile methodology," in *Proc. Int. Conf. Comput., Commun. Autom. (ICCCA)*, May 2017, pp. 864–869.

[16] H. F. Cervone, "Understanding agile project management methods using Scrum," *OCLC Syst. Services, Int. Digit. library Perspect.*, vol. 27, no. 1, pp. 18–22, Feb. 2011.

[17] M. Hamid, R. Yousaf, and S. Munawar, "Extended Scrum process model using software reliability engineering concerns," *J. Inf. Commun. Technol. Robotic Appl.*, vol. 8, no. 2, pp. 1–10, Dec. 2018.

[18] G. Lucassen, M. Robeer, F. Dalpiaz, J. M. E. M. van der Werf, and S. Brinkkemper, "Extracting conceptual models from user stories with visual narrator," *Requirements Eng.*, vol. 22, no. 3, pp. 339–358, Sep. 2017.

[19] M. Usman, E. Mendes, F. Weidt, and R. Britto, "Effort estimation in agile software development: A systematic literature review," in *Proc. 10th Int. Conf. Predictive Models Softw. Eng. (PROMISE)*, 2014, pp. 82–91.

[20] M. Hamid, F. Zeshan, A. Ahmad, S. Munawar, E. Aimeur, S. Ahmed, M. A. Elsoud, and M. Yousif, "An intelligent decision support system for effective handling of IT projects," *J. Intell. Fuzzy Syst.*, vol. 38, no. 3, pp. 2635–2647, Mar. 2020.

[21] D. Jamieson, K. Vinsen, and G. Callender, "Agile procurement to support agile software development," in *Proc. INDIN. 3rd IEEE Int. Conf. Ind. Informat.*, Aug. 2005, pp. 419–424.

[22] R. K. Gupta, P. Manikreddy, and K. Arya, "Pragmatic Scrum transformation: Challenges, practices & impacts during the journey a case study in a multi-location legacy software product development team," in *Proc. 10th Innov. Softw. Eng. Conf.*, 2017, pp. 147–156.

[23] M. Adnan and M. Afzal, "Ontology based multiagent effort estimation system for Scrum agile method," *IEEE Access*, vol. 5, pp. 25993–26005, 2017.

[24] R. Colomo-Palacios, I. González-Carrasco, J. L. López-Cuadrado, and Á. García-Crespo, "ReSySTER: A hybrid recommender system for Scrum team roles based on fuzzy and rough sets," *Int. J. Appl. Math. Comput. Sci.*, vol. 22, no. 4, pp. 801–816, Dec. 2012.

[25] Y. Lin, P. Descamps, N. Gaud, V. Hilaire, and A. Koukam, "Multi-agent system for intelligent Scrum project management," *Integr. Comput.-Aided Eng.*, vol. 22, no. 3, pp. 281–296, Jun. 2015.

[26] H. Yuliansyah, S. Qudsiah, L. Zahrotun, and I. Arfiani, "Implementation of use case point as software effort estimation in Scrum framework," in *Proc. IOP Conf. Ser., Mater. Sci. Eng.*, 2018, Art. no. 012085.

[27] V. S. Nepomuceno and M. E. Fontana, "Decision support system to project software management," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2013, pp. 964–969.

[28] S. Bhalerao and M. Ingle, "Incorporating vital factors in agile estimation through algorithmic method," *IJCSA*, vol. 6, no. 1, pp. 85–97, 2009.

[29] A. S. Alhazmi, "A decision support system for sprint planning in Scrum practice," College Eng. Comput. Sci., Florida Atlantic Univ., Boca Raton, FL, USA, Tech. Rep., 2018.

[30] F. Ramos, A. A. M. Costa, M. Perkusich, H. Almeida, and A. Perkusich, "A non-functional requirements recommendation system for Scrum-based projects," in *Proc. 30th Int. Conf. Softw. Eng. Knowl. Eng.*, Jul. 2018, pp. 148–149.

[31] S. A. Semenkovich, O. I. Kolekonova, and K. Y. E. Degtiarev, "A modified Scrum story points estimation method based on fuzzy logic approach," in *Proc. ISP RAS*, vol. 29, 2017, pp. 19–38.

[32] M. Stupar, P. Milosevic, and B. Petrovic, "A fuzzy logic-based system for enhancing Scrum method," *Management, J. Sustain. Bus. Manage. Solutions Emerg. Economies*, vol. 22, no. 1, pp. 47–57, May 2017.

[33] E. Çetin and P. O. Durdu, "Blended Scrum model for software development organizations," *J. Softw., Evol. Process*, vol. 31, no. 2, p. e2147, Feb. 2019.

[34] D. Soni and P. Kohli, "Cost estimation model for Web applications using agile software development methodology," *Pertanika J. Sci. Technol.*, vol. 25, pp. 931–938, Jul. 2017.

[35] S. K. Sehra, J. Kaur, Y. S. Brar, and N. Kaur, "Analysis of data mining techniques for software effort estimation," in *Proc. 11th Int. Conf. Inf. Technol., New Gener.*, Apr. 2014, pp. 633–638.

[36] I. Ali, M. Asif, M. Shahbaz, A. Khalid, M. Rehman, and A. Guergachi, "Text categorization approach for secure design pattern selection using software requirement specification," *IEEE Access*, vol. 6, pp. 73928–73939, 2018.

[37] M. Asif, I. Ali, M. S. A. Malik, M. H. Chaudary, S. Tayyaba, and M. T. Mahmood, "Annotation of software requirements specification (SRS), extractions of nonfunctional requirements, and measurement of their tradeoff," *IEEE Access*, vol. 7, pp. 36164–36176, 2019.

[38] A. U. R. Butt, M. Asif, S. Ahmad, and U. Imdad, "An empirical study for adopting social computing in global software development," in *Proc. 7th Int. Conf. Softw. Comput. Appl. (ICSCA)*, 2018, pp. 31–35.

[39] V. K. Vaishnavi and W. Kuechler, *Design Science Research Methods and Patterns: Innovating Information and Communication Technology*. Boca Raton, FL, USA: CRC Press, 2015.

[40] H. Zhang, M. A. Babar, and P. Tell, "Identifying relevant studies in software engineering," *Inf. Softw. Technol.*, vol. 53, no. 6, pp. 625–637, Jun. 2011.

[41] J. B. Wordsworth, *Software Development With Z: A Practical Approach to Formal Methods in Software Engineering*. Reading, MA, USA: Addison-Wesley,1992.

[42] S. D. Conte, H. E. Dunsmore, and Y. Shen, *Software Engineering Metrics and Models*. San Francisco, CA, USA: Benjamin-Cummings, 1986.

[43] K. Krishna and M. N. Murty, "Genetic K-means algorithm," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 29, no. 3, pp. 433–439, Jun. 1999.

[44] Z.-G. Liu, Z. Zhang, Y. Liu, J. Dezert, and Q. Pan, "A new pattern classification improvement method with local quality matrix based on K-NN," *Knowl.-Based Syst.*, vol. 164, pp. 336–347, Jan. 2019.

[45] M. Asif, A. Ishtiaq, H. Ahmad, H. Aljuaid, and J. Shah, "Sentiment analysis of extremism in social media from textual information," *Telematics Informat.*, vol. 48, May 2020, Art. no. 101345.

[46] F. Ferrucci, C. Gravino, R. Oliveto, F. Sarro, and E. Mendes, "Investigating tabu search for Web effort estimation," in *Proc. 36th EUROMICRO Conf. Softw. Eng. Adv. Appl.*, Sep. 2010, pp. 350–357.

[47] S. Abrahão, J. Gómez, and E. Insfran, "Validating a size measure for effort estimation in model-driven Web development," *Inf. Sci.*, vol. 180, no. 20, pp. 3932–3954, Oct. 2010.

[48] F. Zeshan, R. Mohamad, M. N. Ahmad, M. B. Othman, and I. Babar, "Context-aware ontology and Web services discovery for distributed embedded real-time systems," *Malaysian J. Comput. Sci.*, vol. 32, no. 3, pp. 186–208, Jul. 2019.

MUHAMMAD HAMID received the M.Sc. degree in information technology, and the M.Phil. and Ph.D. degrees in computer science. His Ph.D. dissertation focuses on increasing the software exports of Pakistan by overcoming the most reoccurred problems using Artificial Intelligence (AI). During his Ph.D., he did research work at the Department of Computer Science and Operations Research, University of Montreal, Canada. He has more than nine years of administrative, research, and teaching experience with the University of Veterinary and Animal Sciences, Lahore. He has authored a number of research papers in reputed journals and conferences. His research interests include software engineering, software project management, and AI.

FURKH ZESHAN received the Ph.D. degree in computer science from the Department of Software Engineering, University Teknologi Malaysia (UTM). His research interests include service-oriented computing, embedded and real-time computing, agent-oriented software engineering, software project management, and self-organizing systems.

ADNAN AHMAD received the B.S. (Hons.), M.S., and Ph.D. degrees in computer science from GCU, Pakistan, LUMS, Pakistan, and Massey University, New Zealand, respectively. He has around thirteen years of teaching and research experience at various educational and research institutes. He has served as the founding in-charge of the Center of Advanced Research in Distributed Systems and Security (CARDS) Research Group, COMSATS University Islamabad, Pakistan. He is currently performing his duties as an Assistant Professor at COMSATS University Islamabad, Lahore Campus, Pakistan. More than 40 peer-reviewed articles in renowned conferences and journals are on his credit, including Trustcom, Computers & Security, and *IEEE Communication Magazine*. He has supervised five Ph.D., sixteen M.S. thesis students, and several startup projects. He has also coauthored a book on socio-technical design. His research interests include distributed systems, cybersecurity, software engineering, and the Internet of Things (IoT). He is a reviewer of various IEEE, Elsevier, IET, and springer journals, and has been a part of TPC in several prestigious conferences.

FAROOQ AHMAD has been working as an Associate Professor with the Department of Computer Science, COMSATS University Islamabad at Lahore Campus, Pakistan. His research interests include Petri net theory and its applications, formal methods, simulation, and modeling.

MUHAMMAD ALI HAMZA received the Ph.D. degree in business administration from the National College of Business Administration and Economics (NCBA&E). He is currently serving as an Assistant Professor (TTS) at the UVAS Business School, University of Veterinary and Animal Sciences, Lahore. His research interests are in the field of HRM, technology-oriented management, transparency and information systems, organizational behavior, educational integrity, and training.

ZUHAIB ASHFAQ KHAN received the bachelor's degree (Hons.) in electronics engineering from Air Force University, Islamabad, Pakistan, and the M.Eng. and Ph.D. degrees in telecommunications engineering from the School of Engineering and Technology, Asian Institute of Technology, Thailand, in 2006, 2009, and 2014, respectively. He is currently an Assistant Professor of electrical and computer engineering at COMSATS University Islamabad (CUI) Attock Campus, Pakistan. His research interests are in the general area of broadband wireless networks and signal processing for communication systems, interference mitigation protocols for femtocells networks, MIMO network coding, cooperative systems, and analysis of future communication systems in the Internet of Things (IoT) and the Internet of Vehicles (IoV). He is a member of the IEEE Society and a reviewer of many prestigious journals. Recently, he has completed his postdoctoral research fellowship program at The University of Sydney, Australia, under 'Endeavour Fellowship' in 2016. He has won many prestigious research fellowship programs, exchange attachment, and pursued his research at Yuan Ze University (YZU), Taiwan, in the lab of Communications Department (YZU-CN) and KUL, Belgium, on Erasmus, in 2013 and 2014, respectively.

SAIMA MUNAWAR received the Ph.D. degree from the School of Computer Science, NCBA&E, Lahore, Pakistan. She has several publications on various technical subjects in international and national HEC recognized journals. Her current research interests are software engineering, AI, and education technology. She is currently performing her duties as the Instructor in the Department of Computer Science, VU of Pakistan.

HANAN ALJUAID received the B.S. degree from KAU University, and the M.S. and Ph.D. degrees in computer science from UTM University, in 2014. She is currently with the Computer Sciences Department, College of Computer and Information Sciences, Princess Nourah bint Abdul Rahman University (PNU), Saudi Arabia. She has published numerous articles in pattern recognition, the IoT, and data science. Her research interests span both computer vision and NLP. Much of her work has been on improving the understanding, design, and performance of pattern recognition, mainly through the application of data mining and machine learning. She has given numerous invited talks and tutorials.

• • •