Jestr

Journal of Engineering Science and Technology Review 10 (4) (2017) 51-64

Review Article

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

A Systematic Review on Software Cost Estimation in Agile Software Development

Saurabh Bilgaiyan, Santwana Sagnika, Samaresh Mishra and Madhabananda Das

School of Computer Engineering, KIIT University, Bhubaneswar, Odisha, India-751024

Received 17 July 2017; Accepted 18 September 2017

Abstract

In the last few years, the size and functionality of software have experienced a massive growth. Along with this, cost estimation plays a major role in the whole cycle of software development, and hence, it is a necessary task that should be done before the development cycle begins and may run throughout the software life cycle. It helps in making accurate estimation for any project so that appropriate charges and delivery date can be obtained. It also helps in identifying the effort required for developing the application, which assures the project acceptance or denial. Since late 90's, Agile Software Development (ASD) methodologies have shown high success rates for projects due to their capability of coping with changing requirements of the customers. Commencing product development using agile methods is a challenging task due to the live and dynamic nature of ASD. So, accurate cost estimation is a must for such development models in order to fine-tune the delivery date and estimation, while keeping the quality of software as the most important priority. This paper presents a systematic survey of cost estimation in ASD, which will be useful for the agile users to understand current trends in cost estimation in ASD.

Keywords: Software cost estimation, software effort estimation, software project management (SPM), agile software development (ASD), software measurement and metrics.

1. Introduction

Software engineering broadly encompasses the standard procedures that cover various aspects of software project development, right from its inception to its post-release maintenance. These standards ensure timely and efficient completion of a project while maintaining the quality at the highest level [1]. The specific course of action for handling a project is termed as software project management (SPM). The process of SPM includes the activities of planning, supervising and integrating which leads to production of the project. Planning starts well before the commencement of the actual development, and it includes formulation of a road-map for the activities that needs to be performed with the resources which are required in the process. Supervising is the task of monitoring the progress against the defined milestones. It is the responsibility of project manager to keep track of different phases and their timely completion. During the development process, it is necessary to maintain co-ordination among the multiple activities. It enables the system to perform as per the requirements as a single working unit. This comes under the integrating process of SPM [2, 3].

The planning phase results in a set of products, among which the cost estimation of the software is a major task. Cost estimation is basically a forecasting of expected time, effort, manpower and finance that are needed to complete the development of a project [1]. During the initial phases of software development, it is difficult to predict the size of the project, since the requirements are inaccurate and incomplete due to change in requirement during the development cycle. The dynamic changes result in cascading effect on the cost estimation of the software [4]. An efficient cost estimation mechanism should provide certain benefits. Firstly, it should help in analysis and reuse of warehouse data to reduce estimation time. Secondly it should be simple and understandable for end users. Also, it should take care of incomplete and vague input data. Finally, it should enumerate the various factors that influence the overall cost of the project [5]. Keeping such objectives in view, cost estimation has been performed through various mechanisms, popular among which, are estimation by analogy, expert judgment, function points, regression, work breakdown, Bayesian method, classification and regression trees, etc. [6].

As the scope of software has widened to various fields, it has become difficult to specify all requirements at the start of the project, depending on the application area. In such cases, the requirements are updated and modified along with the duration of the project. Since most traditional methods of development, though effective, need to clearly define the requirements as a prerequisite, it becomes difficult to apply those methods to the current projects. Hence, the need arises to implement newer mechanisms that can adapt to changing requirements dynamically. This has led to the advent of the agile methodology of software development, which is receptive to the changes in requirements that arise during the development period [7]. Agility can be termed as the capability of quick and successful adaptation to continuously changing surroundings, complying with customer needs. ASD involves practices that have been designed by

*E-mail address: saurabh.bilgaiyanfcs@kiit.ac.in
ISSN: 1791-2377
doi:10.25103/jestr.104.08

experienced experts, involving detailed planning and reusable coding for development, to make the process a flexible and rational one. This means that ASD can easily accept and implement ever-changing requirements of users [8, 9]. Various mechanisms have evolved that follow the agile technique of developing software. Some of the key agile development methods involve Crystal methodologies, Scrum, Extreme Programming (XP, XP2), Dynamic Software Development Method (DSDM), Feature-driven development, Lean software development, etc. [9, 10].

Cost estimation for ASD methods is a challenging task. It is so because the agile technique is based on unconventional concepts that are not suited for any of the traditional estimation methods [7]. The application of such estimation methods to agile development mechanisms results in considerable inaccuracies, due to the absence of critical factors like expert opinion and historical data [11]. Since the agile process is much simpler and requires faster work along with more customer involvement, hence this change in approach also requires a change in the cost estimation process. As a result, recent research has focused on developing estimation techniques (ETs) that are compatible with agile methods [12].

2. Review Strategy

The paper presents a systematic literature review that has been performed as per specific guidelines to gather and process the data available from various sources. The guidelines are defined as follows with reference to some standard existing research performed on this topic [6, 13, 14].

A. *Research Issues*

RI-1: What are the various estimation mechanisms explored for agile software development methods and what are the parameters that define their accuracy?

RI-2: What are the comparative accuracies achieved by different estimation techniques?

RI-3: What are the suitable circumstances to which the various estimation techniques can be applied and what problems can be faced while applying different estimation techniques?

RI-4: What are the success/failure rate of the project and their statistics over last decade. What are the major project cost items that are to be included and excluded?

RI-5: What is the popularity of agile software development in current software development environment? What are the frequency of using different agile methods?

To figure axis labels, use words rather than symbols. Do not label axes only with units. Do not label axes with a ratio of quantities and units. Figure labels should be legible, about 9-point type.

Color figures will be appearing only in online publication. All figures will be black and white graphs in print publication.

B. Inclusion and Exclusion Criteria

The current study has included various papers mainly concerning on cost estimation using ASD. Along with this, there are few more papers involving effort and cost estimation using other development techniques, which satisfy the ultimate purpose of faster and more accurate estimation. The study considers various papers from different journals and conferences. Papers that do not serve the specified purpose or are not based on factual and empirical data have not been considered.

C. Data Gathering and Refinement Process

Data gathering

The online databases of IEEE Digital Library, Elsevier, Springer, Science Direct, ACM, IET Library, etc. have been accessed and full text papers have been taken for analysis. Keywords like estimation, cost, effort, agile and prediction have been used as search strings to look for papers. Table 1 gives the detail about the number of papers found in various journals and conferences and the papers found for the respective search strings.

Table 1. Search string based data gathering

SI. No.	Search Strings	Papers from journals	Papers from conferences
1	Software effort estimation	[15-20, 22-27, 29, 30, 158, 159, 162]	[10, 21, 28, 31, 163]
2	Software cost estimation	[6, 12, 18, 22, 24, 32, 156, 160, 161]	[33-35, 155]
3	Agile software development	[9, 37-39, 41, 45-49, 154, 164]	[36, 40, 42-44, 50, 157]
4	Software measurement and metrics	[52-55, 57, 59]	[51, 56, 58]
5	User stories in agile	[61, 63, 68, 72]	[60, 62, 64-67, 69-71]
6	Extreme Programming	[74, 76, 77, 79, 81-83, 87]	[73, 75, 78, 84-86, 88, 89]
7	Scrum	[90, 91, 94-98, 100]	[92, 93, 99, 101-103]

The search strings “software effort estimation” and “software cost estimation” were selected pertaining to the core subject that is dealt with in this paper. “Agile software development” was selected to discover all relevant work regarding the use of agile technology for developing software. “Software measurement and metrics” represented the various parameters that were used for analysis of software costs. “User stories in agile” refers to the basic requirement gathering approach in agile technology that lends it its dynamism. “Extreme Programming” and “Scrum” are the common mechanisms of ASD. Other mechanisms like Crystal, Dynamic System Development Method, Adaptive Software Development, Lean Software Development, etc. were not taken as search strings since the related papers were already found using the previously defined search strings.

The following distribution contains relevant research work published from 2006 to 2015 as given in Table 2.

Table 2. Distribution of research articles over the last decade

Source of Article	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	Total
Science Direct (Elsevier)	0	0	0	0	0	0	2	0	2	1	0	3	1	1	0	2	12
Springer	0	0	0	0	1	1	0	2	0	0	2	0	1	1	1	0	9
ACM	1	0	0	0	0	1	2	1	1	1	1	4	1	2	0	3	18
IEEE	0	0	0	2	0	0	0	1	1	3	1	4	1	1	1	1	16
Others	0	0	0	1	0	1	0	0	4	2	1	7	7	10	10	3	46
Total	1	0	0	3	1	3	4	4	8	7	5	18	11	15	12	9	101

The distribution represented in Table 2. doesn't consider the research articles which are unrelated to the scope of our survey. Hence, articles are considered under a refined process, which satisfies our survey objective. Elsevier, Springer, ACM, IEEE are well known and recognized journals while the category "other" combined some more good journals/conferences. The overall distribution shows that the number of publications have increased over years.

Refinement Process

A large number of relevant research papers were found using the selected search strings (as discussed earlier). These papers were shortlisted and classified according to their publications in journals or conferences. The shortlisting process was performed according to the following steps:

a. Shortlisting by paper title and abstract

The first level of refinement was done on the basis of the paper title and abstract. The paper title was checked for the presence of the respective search strings or related terms within the scope of the survey. The abstract gave an overall idea of whether the content of the paper was relevant to our study or not. Out of a total of 103 papers, 28 relevant papers were selected using this criterion.

b. Shortlisting by relevance to survey

The next level of selection was done by taking into consideration the overview of the contents of the paper. The names of the various sections were checked with their overall relevance. In this step, we selected the papers which were either review papers or papers that dealt with the particular topics in a generic manner. The papers which were focused towards specific methodologies or individual case studies were not taken into account. A total of 49 papers were selected using such criteria.

c. Shortlisting by full paper content

At the final level, the papers were read throughout, and the relevance of their content was analysed. Under this refinement strategy, 26 papers were selected whose entire content was found contributed, such as newly-proposed estimation methods, large-scale case studies, etc. The remaining papers were selected after reading their complete content.

3. Exploring Research Issues

RI-1: *What are the various estimation mechanisms explored for agile software development methods and what are the parameters that define their accuracy?*

Research suggests that the cost incurred in around 60% of projects run beyond their initial estimation and 15% projects fail due to extremely wrong estimations. Hence, it becomes crucial to perform a proper and accurate analysis of the expected costs and plan accordingly. This has given rise to a special field of research in the form of cost and effort estimation, which provides a basis for proper planning and managing the development process [104, 105]. Estimation is basically a forecasting of expected time, effort, manpower and finances that are needed to complete the development of a project. Hence, the main aim is to strike a good balance between the costs incurred and the quality obtained. It is also necessary to assess the accuracy of the estimation technique, so that estimation comes as close to the actual cost as possible. This prevents mis-utilization of resources and unnecessary time and money constraints that can affect the resultant quality [106, 107]. In this study, various existing techniques for estimation in ASD are assessed, and their accuracy parameters get identified. In agile development, the effort estimation also includes size estimation factors, hence, such papers are also considered in this study [13, 108]. Different accuracy parameters used by different estimation techniques are represented in Table 3.

Table 3. Accuracy parameters used by different estimation mechanisms

Estimation Mechanisms	Papers	Accuracy Parameters
NN	[109-123]	MRE, MMRE, PRED(X), MAPE, MdMRE, R^2, MSE, MMER
EJ	[124, 126-129]	MRE, MdMRE
PP/Da	[124, 130-132]	MRE, BRE
UCP	[111, 125, 133, 134]	MRE, MMRE, MMER, PRED(X), R^2, MSE
MUCP	[133, 134, 138, 139]	MRE, MMRE, MdMRE, PRED(X)
LR	[135, 136]	MRE, MMRE, MMER, PRED(X), MdMRE
Wd	[124, 132]	MRE
BU/TD	[129, 137]	MRE

Abbreviations used in Table 3 are defined in the Table 4.

Table 4. Abbreviations used for *Table 3.*

For Estimation Mechanism	
NN	Neural Network
EJ	Expert Judgment
PP	Planning Poker
Da	Disaggregation
UCP	Use Case Point
MUCP	Modified Use Case Point

LR	Linear Regression
Wd	Wideband Delphi
BU	Bottom-up
TD	Top-down
For Accuracy Parameters	
MRE	Magnitude of Relative Error
MMRE	Mean Magnitude of Relative Error
PRED (X)	Prediction
MAPE	Mean Absolute Percentage Error
MdMRE	Median Magnitude of Relative Error
R^2	Squared Correlation Coefficient
MSE	Mean Squared Error
MMER	Mean Magnitude of Error Relative
BRE	Balance Relative Error

The Table 3 shows the most popular estimation mechanisms used for cost estimation. A brief description of each method is as follows:

(i). Neural Networks-
NN(s) are an artificial simulation of the biological nervous systems which are used for approximation and classification of various functions, usually based on certain training data. Such networks use an activation function that operates on the weighted inputs to give the final output using the general formula i.e., where a is the output of the network, f () represents the activation function, w is the weight of input p, and b is the input bias. A special type of neural network is the feed-forward network that allows a uni-directional flow of data. Various feed-forward networks include Radial Basis Function Neural Network (RBFNN), Multilayer Perceptron (MLP), General Regression Neural Network (GRNN), Wavelet Neural Network (WNN), etc. Most of these models have been applied for cost estimation with certain customizations like Input Sensitivity Analysis [115]. There is a high dependency of software cost estimation on various factors that are not linearly related. Hence, most dominant factors need to be identified and provided as inputs to the NN model being used. NN(s) make the best use of the uncertain nature of these inputs, and the various layers work to give a considerably accurate estimation by minimizing error rate [116].

(ii). Expert Judgment-
The most experienced and senior people working in software industries have gained a lot of experience regarding the estimation of various projects, having been involved in different projects throughout their work period. They have sound knowledge, which prominently affect the development time and cost, as well as it can cause digression of the actual cost from the estimates. Hence, experts' opinion is quite often sought for analyzing cost and time for projects [128]. Nevertheless, it is also evident that experts' judgment method is influenced by various subjective factors like bias, influence of work environment, type of projects handled by the concerned experts, as well as human errors. Even so, it is an efficient and useful method for estimating small and medium-sized projects [124]. Expert estimation can follow the top-down approach, i.e. estimation without decomposition, or bottom-up approach, where estimation takes place for each of the divided sub-activities [129].

(iii). Planning Poker-
This technique involves discussions among the team members regarding the estimate. For each user requirement, all involved members make their estimates and compare them to reach an agreement. If any discrepancy occurs, then discussions take place to find consensus, otherwise the agreed estimation is set down as finalized, and the next user requirement is taken into consideration [131, 132]. This technique avoids anchoring by concurrently revealing all estimates. Here, anchoring refers to the impact of initial estimate on the subsequent estimates. It also makes the whole process more interactive and ensures involvement of all team members [130].

(iv). Use Case Points-
This method is a customization of the traditional Function Point method which is suitable for the object-oriented software development method. The use cases and the number of actors as per the use-case diagram are used to calculate unadjusted UCP. The environmental adjustment and technical complexity factors are used to find the adjusted UCP. The final calculation gives an estimation of the number of staff hours per UCP [125, 133]. In case of agile projects, the method keeps a count of the number of transactions as per the defined use cases. Experimental records show that the estimates reached by UCP method are very close to the actual effort, without involving substantial adjustments at a very early stage of development [125].

(v). Modified Use Case Points-
This is an extension of the UCP method, which includes use case narratives, and hence focuses more on the relationships between use cases and actors. As a result, the detailed internal details are taken into consideration. The method focuses on the impacts of the particular use-case activities instead of considering the overall system functionality [138, 139].

(vi). Linear Regression-
Regression refers to finding out the relationships between dependent and independent variables. For cost estimation, the existing data can be used for building regression models. Since the cost of development depends on multiple and conflicting factors, hence multiple linear regression models can be used for representing their interrelationships. This technique is feasible when the cases are much more than the parameters required to be estimated. The missing data should be few, and the behaviour of data should be stable. The assumptions based on past data need to be verified before being applied to the case in hand [135, 140].

(vii). Wideband Delphi-
This method focuses on the interaction between team members and strives to reach a consensus among them. A team is selected to deal with the project, which meets to decide a breakdown structure of the whole work and the necessary assumptions. Then the individual members perform their estimation for various tasks, which are then distributed among everyone for further discussion and revision. The tasks and corresponding estimation are integrated to reach an agreement, which is reviewed finally for relevance. This method also indirectly assures the quality

level. This method is a quite reliable and simple by considering the fact that it takes experts' opinions as well as reaches agreement among all people those who are involved in the system [124].

(viii). Top-down/ Bottom-up Approach-
The top-down approach focuses on estimating the cost required for the whole project without decomposing it into smaller parts. The estimation is then distributed among various activities in proportion. This method can be followed when there is some historical data available about similar projects. In the bottom-up approach, the project is first decomposed into various activities, and then the estimation for each one of those activities is generated by summing up all the individual estimation. Both strategies are equally useful, depending upon the variety of projects [129, 132, 137].

There are quite a few accuracy parameters used as performance evaluation criteria for different estimation techniques. Following are short descriptions of the accuracy parameters used:

(i) Magnitude of Relative Error- MRE is the most common and traditional measurement criteria for estimation techniques. MRE individually assesses every project in a dataset. Following "Eq. (1)" is the mathematical representation for MRE [110, 117, 137, 138]:

$$MRE_i = \frac{\left| Actual\ Effort_1 - Estimated\ Effort_1 \right|}{Actual\ Effort_1} \quad (1)$$

(ii) Mean Magnitude of Relative Error- MMRE is another popular performance measurement parameter for estimation techniques. It measures percentage values of relative errors. The calculated percentage value is the average value over the N items. The overall calculation is based on the MRE and can be rewritten as follows in "Eq. (2)":

$$MMRE_i = \frac{1}{N}\sum_{i=1}^{N} MRE_i \quad (2)$$

MMRE aggregates multiple projects or model having lowest MMRE value which is said to be best among all [109, 112, 117, 134, 138].

(iii) Median Magnitude of Relative Error- MRE has a disadvantage of being sensitive at outliers whereas MdMRE overcomes this by accepting different criteria that are not sensitive to outliers. It measures the Median for all MRE(s). This can be written as follows in "Eq. (3)" [127, 136, 141]:

$$MdMRE_i = Median(MRE_i) \quad (3)$$

(iv) Mean Magnitude of Error Relative- MMER is one of the measures used for assessing the performance of cost estimation models. It is claimed that sometimes MMER is more accurate than MMRE. MMER is calculated from median of Magnitude of Error Relative (MER) and it can be represented as follows in "Eq. (4, 5)" [142]:

$$MRE_i = \frac{\left| Actual\ Effort_1 - Estimated\ Effort_1 \right|}{Actual\ Effort_1} \quad (4)$$

$$MMER_i = \frac{1}{N}\sum_{i=1}^{N} MER_i \quad (5)$$

(v) Mean Absolute Percentage Error- For absolute determination of accuracy for different estimation models MAPE is quite popular. The term absolute is considered as the assessment of the cost estimations from the actual recognized costs. MAPE can be written as follows in "Eq. (6)" [143, 144]:

$$MAPE_i = \frac{1}{N}\sum_{i=1}^{N} \left| \frac{Actual\ Effort_i - Estimated\ Effort_i}{Actual\ Effort_i} \right| * 100 \quad (6)$$

For the calculation of absolute value, first the summation is done for each estimated point of time and then it is divided by the number of suitable points N. MAPE is complained against for returning distorted values if the actual value is close to 0 and 1.

(vi) Mean Squared Error- For calculating MSE, the following "Eq. (7)" is used [123, 141]:

$$MSE_i = \frac{1}{N}\sum_{i=1}^{N} (Actual\ Effort_i - Estimated\ Effort_i)^2 \quad (7)$$

where, N is the total number of data in the dataset. Accuracy of the estimation method is directly proportional to the PRED *(X) and contrariwise to MSE, MdMRE and MMRE parameters.

(vii) Balance Relative Error- Previous research shows that BRE is a quite popular measurement factor. BRE is more balanced than MRE in terms of underestimation and overestimation and can be calculated as follows in "Eq. (8)" [131,132]:

$$BRE = \frac{|Actual\ Effort_i - Estimated\ Effort_i|}{min(Actual\ Effort,\ Estimated\ Effort)} \quad (8)$$

(viii) Squared Correlation Coefficient- R^2 is also known as coefficient of determination. R^2 is defined as the assessment of the effectiveness of a regression. It can be represented as in "Eq. (9)" [111]:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(Actual\ Effort_i - Estimated\ Effort_i)^2}{\sum_{i=1}^{N}(Actual\ Effort_i - Mean(Actual\ Effort))^2} \quad (9)$$

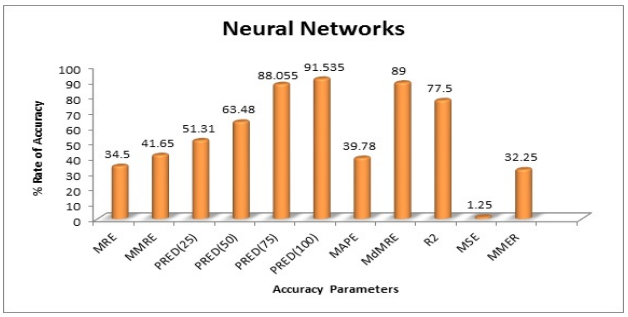
(ix) Prediction (PRED (x))- The percentage of MRE which is less than or equal to value x for all projects is considered as PRED (x) and is calculated as follows in "Eq. (10)" [111, 117]:

$$PRED\ (x) = \frac{K}{N} \quad (10)$$

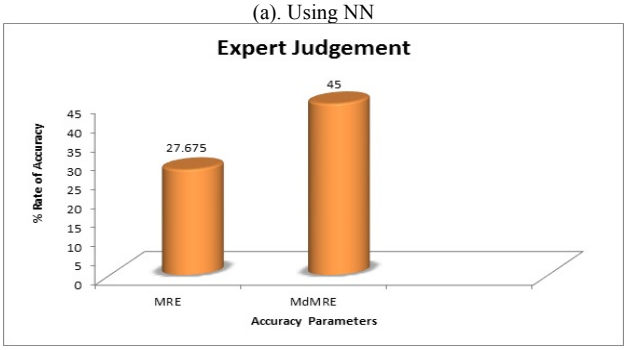
where, K is the number of observations whose MRE is less than or equal to x; N represents the total number of observations. the value of x can be either 0.25, 0.50. 0.75 or 1.0. If a common value of x is 0.25, then PRED (0.25) refers to the percentage of projects whose MRE is less than equal to 25%.

RI-2: *What are the comparative accuracies achieved by different estimation techniques?*
A comparative analysis of accuracies achieved by different accuracy parameters for different estimation techniques, as discussed earlier in this article, is represented in terms of graphs.

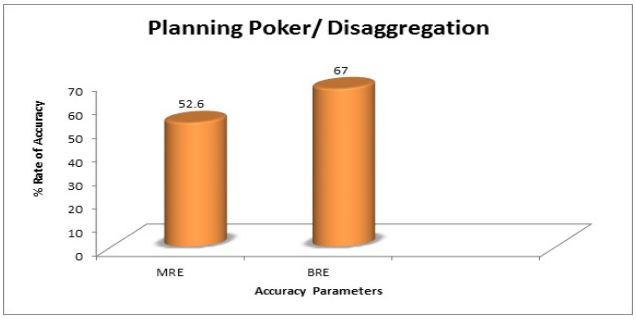
The accuracy rates considered in Fig.1 (a) to Fig.1 (h), is obtained by analysing relevant research articles where an approximate average is taken from the obtained output values for different accuracy parameters applied through different estimation mechanisms. The estimation techniques that use specific methodologies are not considered due to variation in working environment.

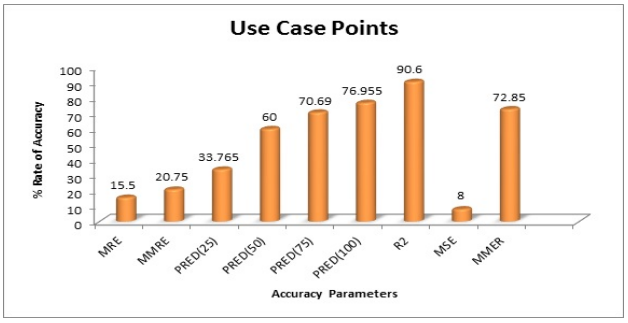


(a). Using NN

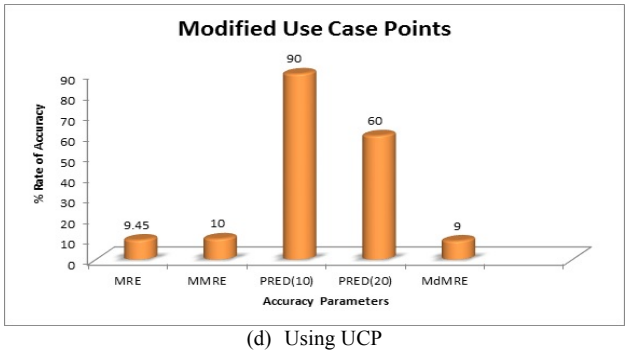


(b). Using EJ

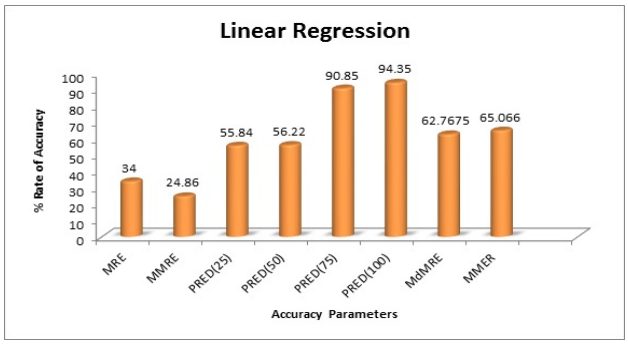




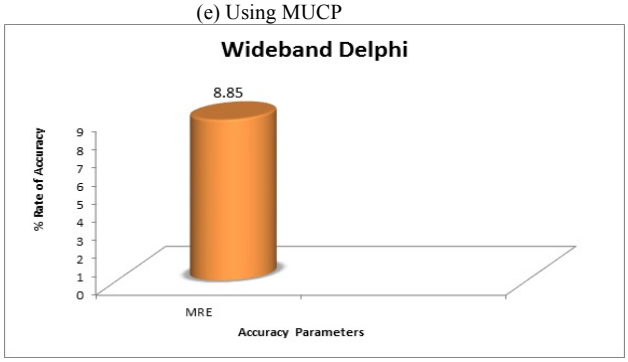
c) Using PP/Da



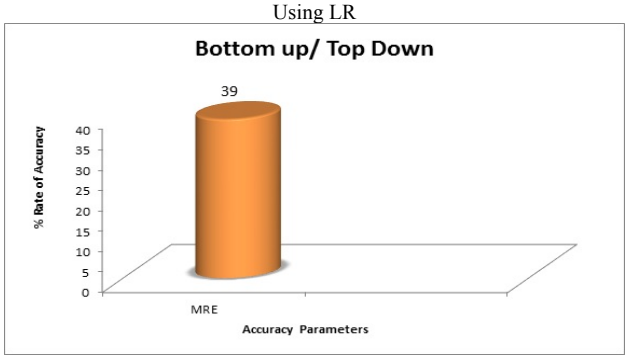
(d) Using UCP



(e) Using MUCP



Using LR



Using TD/BU

Fig. 1. Percentage of accuracy achieved by different accuracy parameters through different estimation techniques

RI-3: *What are the suitable circumstances to which the various estimation techniques can be applied and what problems can be faced while applying different estimation techniques?*

Table 5. Working circumstances for different estimation methods

ET	Suitable circumstances	Difficulties Faced	Source
NN	NN based techniques perform well and produce accurate predictions when input data is distorted by the high noise level, and there is a complex relationship between inputs and output values.	The predicted outcome varies as the network architecture and its parameters are adjusted. Also, NN does not have any explanation facility. It can produce an accurate prediction for a given problem, but it cannot generalize the solution when conditions change. As NN does	[114, 116, 119]

		not support recalibration, it lacks in performance as compare to expert judgment and statistical models for different development environments. It is well known that the accuracy of NN is based on the nature of data. Researchers have faced lots of problems at the time of defining proper datasets from the real-world environment as less research is done in removing outliers and non-dominant variables from data, in the field of cost estimation.	
EJ	EJ is said to be a more accurate applied estimation strategy when there is enough historical data related to the project, and experienced experts are available to deal with such data. EJ leads to more accurate predictions when the essential domain information is included in the models.	EJ method is not much reliable because the accuracy of the prediction has a big influence of the experience and skill of the experts (same information possessed by the same expert at different working occasions leads to different estimation predictions).	[126-128]
PP/Da	PP is applied to situations when every member of the development team irrespective of his position in the team (mostly in XP and Scrum-based projects) participates in group discussions. PP sometimes leads to more accurate predictions than EJ because an individual may exclude some of the important information and it leads to over-optimism.	Less research has been done, and little empirical evidence is available regarding accuracies of PP.	[131, 132]
UCP	UCP is a popular estimation method for predicting initial estimates and it is the most appropriate estimation approach for software development projects applied with rational unified process and unified modeling language.	Researchers faced complications while using UCP in agile software development projects where the product backlog does not satisfy some conditions in the use case documentation.	[125, 133]
MUCP	The suitable circumstances for MUCP are the same as that of UCP. MUCP differs from UCP when it is refined, based on the requirements of UCP based models, project manager, consultants, input from software cost estimators, etc.	Same problem is faced by the researchers as with UCP when dealing with the agile software development projects.	[125, 138]
LR	Suitable for the problems where existing data can be used for the estimation process. LR seems to be more feasible when the number of cases in the estimation process is more than the parameters required to be estimated. There will be more accurate prediction by LR if the missing data will be less and the behavioral data will be stable.	Missing historical data leads to decrease in the accuracy of overall prediction.	[136, 140]
Wd	WD is said to be a suitable estimation strategy when work breakdown structure is the basis for the estimation. It can be useful for the estimation of agile software development projects where the interaction between participants is more than the classical approaches of development.	Accuracy of the estimation is directly affected by the experience of the development team. Also, requires the agreement and management cooperation among the development team. Quite less empirical evidence is available regarding the accuracy of the WD estimation approach.	[124, 132]
BU/TD	Bu/Td is said to be one of the appropriate methods for estimation if there is enough historical data is available about similar projects. There are certain conditions that favor the Td and Bu methods, e.g. if there is a requirement of early stage estimation on the basis of vague requirements specification that do not allow detailed breakdown of the development process, then Bu estimation will be a favorite, and if there is a requirement of re-estimation of the remaining actions then Bu will be a suitable method.	There is not enough empirical evidence about the goodness of the methods as researchers have observed that Td is more accurate over Bu in one experiment, while, in other experiments, no improvements were found.	[129, 132]

There are certain conditions best suited for a specific estimation technique for obtaining an optimum range of accuracy. Also, there are some difficulties faced by the researchers while dealing with such conditions. Table V. shows the favourable conditions with flaws for different estimation techniques.

RI-4: *What are the success/failure rate of the project and their statistics over last decade. What are the major project cost items that are to be included and excluded?*

The statistical analysis of project success/failure rates and other important empirical data over the past few years is done by the Standish Group in the United States. The analysis is not based on primary studies. The survey includes the large, medium and small companies across major industry sections, e. g. Manufacturing, banking, health-care,

securities, insurance, federal organizations, local states and services, wholesale and retail services. The above analysis is classified in the Table 6. [145, 146, 147, 148, 149].

Table 6. Characterization of projects

Resolution Type	Description
Resolution Type-1	The projects which are completed on-budget and on-time with all functions stated by the customer in initially specified features, fall under this category.
Resolution Type-2	The projects which are completed and are operational but their completion time and budget are exceeded, fall under this category. Also, the functions specified in the initially specified requirements are not fully covered.
Resolution Type-3	The projects which are canceled at any point during the development cycle, fall under this category.

By analysing over the years, Standish group finds the following statistics as given in the Table 7:

Table 7. Project success rate over years

Resolution Type	1994	1996	1998	2000	2002	2004	2006	2008	2010	2012
Resolution Type-1	16.2%	27%	26%	28%	34%	29%	35%	32%	37%	39%
Resolution Type-2	52.7%	40%	28%	49%	51%	53%	46%	44%	42%	43%
Resolution Type-3	31.1%	33%	46%	23%	15%	18%	19%	24%	21%	18%

The above data has been collected from the Standish Group Report "CHAOS" reprinted version of 1995, Project resolution from "CHAOS" research (2013), Software projects from 2003 to 2012 within the "CHAOS" database published in Big Bang Boom report (2014).

Further, the success factor in agile vs waterfall models projects (2003-2012) is compared in the "CHAOS MANIFESTO", 2013 which is represented in Fig.2 (a) and Fig.2 (b) as follows:

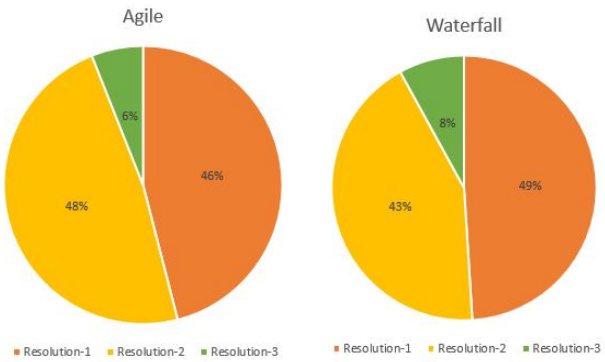
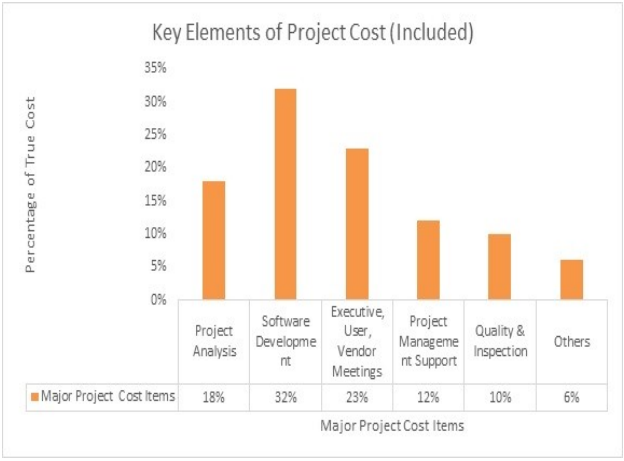
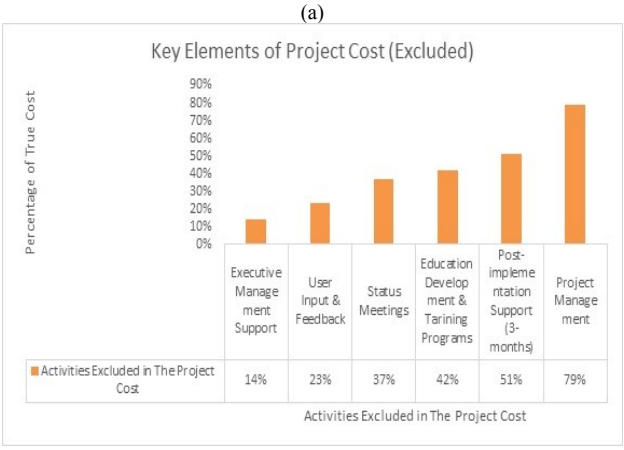


Fig. 2. Success Rate for Projects developed with ASD and Waterfall Model. (a). For Agile and (b) for Waterfall Model

Further, there are certain key elements that are considered as the true elements of a project which is required for its true cost estimation. Cost assessed by such elements is said to be the true cost of a project. Likewise, there are certain activities in the project that are not considered as true cost elements and hence, those are excluded at the time of estimation of the project cost. The exclusion of activities is done due to rarity of properly budgeting for these costs. Fig.3 (a) and Fig.3 (b) show the key activities that are to be included and excluded respectively at the time of the true cost calculation of a project. Data for the Table VI., Table VII., Fig.2 and Fig.3 is taken from the Chaos report [145, 146, 147, 148, 149]



(a)



(b)

Fig. 3. Key Elements of Project to be Included/Excluded for Assessing the True Cost. (a) Included Elements and (b) Excluded Elements

RI-5: *What is the popularity of agile software development in current software development environment? What is the frequency of using different agile methods?*

According to the current survey, approximately 38% of the organizations are using ASD methods frequently [150, 151]. Within these organizations, 75% of them which are highly agile dependent software firms met their goals, 67% finished on budget while 65% successfully completed their projects within time. But according to the law of agility, the analysis is different. Only 56% of development firms meet their

business goals while 45% and 40% of the projects completed on-budget and on-time respectively [150, 151]. The following *Fig.4* shows the current popularity of different agile software development tools [152]:

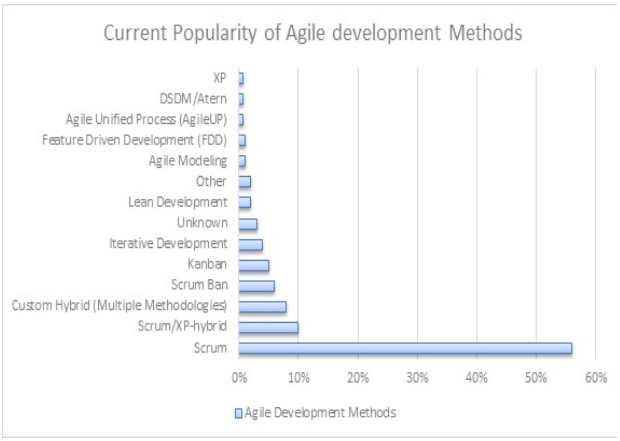


Fig. 4. Popularity of Different Agile Software Development Based Tools

4. Conclusion and Future Work

This paper presents a systematic survey about software cost estimation in agile software development. The paper deals with the current estimation schemes used in software development other than agile estimation, so that these schemes may be useful in the agile development environment. Following are the findings for our research article:

A. Findings:

Our study and analysis show that agile software development is becoming one of the frequently used software development tools, which is widely adopted by the different researchers as well as software development organizations. Accurate estimation of cost and effort for a software project plays a major role in the success of that project. A significant increase in the success rate of the different types of software projects is achieved by improvement in the accuracy of estimation environment. Till date, very less work has been done in the field of cost and effort estimation for agile software development. Due to the lack of empirical evidence for the used techniques, performance can't be guaranteed. Owing to their capability of adaptation to dynamic requirement changes, agile methods are quite popular but complex in nature too. Hence, estimation for such dynamic and complex development paradigms becomes difficult. In the existing literature, we found that NN, EJ, PP/Da, UCP, MUCP, LR, Wa and BU/TD are the estimation techniques applied over ASD and other development environments. NN and expert estimation are the more popular ones among the existing conventional methods of estimation for ASD. In this survey, we identified the important elements that are included/excluded in the estimation process for calculation of the true cost of a project. The work has certain limitations as authors did not perform any test to evaluate the inter-rater agreement between reviewers in the review activity. Some recent and relevant publications might be missing in the review process.

B. Future Findings:

There are more possibilities in the field of software effort and cost estimation using soft computing techniques (especially using swarm-based algorithms like particle swarm optimization (PSO), bat swarm optimization, ant colony optimization (ACO), bee colony optimization (BCO), etc.). In addition to this, there is the possibility of optimizing the existing estimation techniques with more empirical outcomes on different existing test environments.

References

[1]. Rd. Lemos et al., "Software Engineering for Self-Adaptive Systems: A Second Research Roadmap", Software Engineering for Self-Adaptive Systems II, Lecture Notes in Computer Science, Springer, Vol. 7475, No. 1, pp. 1-32, 2013.

[2]. Ch. V. M. K Hari, T.S. Sethi and M. Jagadeesh, "SEEPC: A Toolbox for Software Effort Estimation using Soft Computing Techniques", International Journal of Computer Applications, Vol. 31, No. 4, pp. 12-19, 2011.

[3]. I. Attarzadeh and S. H. Ow, "Proposing a New Software Cost Estimation Model Based on Artificial Neural Networks", 2nd International Conference on Computer Engineering and Technology, IEEE, Chengdu, Vol. 3, No. 1, pp. V3-487-V3-491, 2010.

[4]. I. Attarzadeh and S.H. Ow, "A Novel Algorithmic Cost Estimation Model Based on Soft Computing Technique", Journal of Computer Science, Vol. 6, No. 2, pp. 117-125, 2010.

[5]. A. Idri and A. Zahi, "Software Cost Estimation by Classical and Fuzzy Analogy for Web Hypermedia Applications: A replicated study", IEEE Symposium on Computational Intelligence and Data Mining (CIDM), Singapore, pp. 207-213, 2013.

[6]. M. Jorgensen and M. Shepperd, "A Systematic Review of Software Development Cost Estimation Studies", IEEE Transactions on Software Engineering, Vol. 33, No. 1, pp. 33-53, 2007.

[7]. S. Ziauddin, S.K. Tipu and S. Zia, "An Effort Estimation Model for Agile Software Development", Advances in Computer Science and its Applications (ACSA), Vol. 2, No. 1, pp. 314-324, 2012.

[8]. A. Gunasekaran: Agile Manufacturing, "A Framework for Research and Development", International Journal of Production Economics, Elsevier, Vol. 62, No. (1-2), pp. 87-105, 1999.

[9]. T. Dyba and T. Dingsoyr, "Empirical Studies of Agile Software Development: A Systematic Review", Information and Software Technology, Elsevier, Vol. 50, No. (9-10), pp. 833-859, 2008.

[10]. A. Schmietendorf, M. Kunz and R. Dumke, "Effort estimation for Agile Software Development Projects", In Proceedings of 5th Software Measurement European Forum, Milan, Italy, pp. 113-126, 2008.

[11]. R. Popli and N. Chauhan, "Cost and Effort Estimation in Agile Software Development", International Conference on Optimization, Reliabilty, and Information Technology (ICROIT), Faridabad, India, pp. 57-61, 2014.

[12]. Z. Mansor et al., "Review on Traditional and Agile Cost Estimation Success Factor in Software Development Project", International Journal on New Computer Architectures and Their Applications, IEEE, Vol.1, No.3, pp. 942-952, 2011.

[13]. M. Usman, E. Mendes, F. Weidt and R. Britto, "Effort Estimation in Agile Software Development: A Systematic Literature Review", 10th International Conference on Predictive Models in Software Engineering, ACM, pp. 82-91, 2014.

[14]. D. Nguyen-Cong and D. Tran-Cao, "A Review of Effort Estimation Studies in Agile, Iterative and Incremental Software Development", IEEE RIVF International Conference on Computing and Communication Technologies – Research, Innovation and Vision for the Future (RIVF), Hanoi, Vietnam, pp. 27-30,2013.

[15]. J. G. Borade and V. R. Khalkar, "Software Project Effort and Cost Estimation Techniques", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, No. 8, pp. 730-739,2013.

[16]. S. Malathi and S. Sridhar, "A Classical Fuzzy Approach for Software Effort Estimation on Machine Learning Technique", International Journal of Computer Science Issues (IJCSI), Vol. 8, No. (6-1), pp. 249-253,2011.

[17]. A. F. Sheta and S. Aljahdali, "Software Effort Estimation Inspired by COCOMO and FP Models: A Fuzzy Logic Approach", International Journal of Advanced Computer Science and Applications, Vol. 4, No. 11, pp. 192-197,2013.

[18]. P. K. Suri and P. Ranjan, "Comparative Analysis of Software Effort Estimation Techniques", International Journal of Computer Applications (IJCA), Vol. 48, No. 21, pp. 12-19,2012.

[19]. P. Rijwani, S. Jain and D. Santani, "Software Effort Estimation: A comparison based Perspective", International Journal of Application or Innovation in Engineering and Management (IJAIEM), Vol. 3, 2014, No. 12, pp. 18-29,2014.

[20]. N. Saini and B. Khalid, "Empirical Evaluation of Machine Learning Techniques for software Effort Estimation", IOSR Journal of Computer Engineering (IOSR-JCE), Vol. 60, No. 1, pp. 34-38, 2014.

[21]. K. Molokken and M. Jorgensen, "A Review of Surveys on Software Effort Estimation", International Symposium on Empirical Software Engineering, pp. 223-230, 2003.

[22]. N. Grimstad, M. Jorgensen and K. Molokken-Ostvold, "Software Effort Estimation Terminology: The Tower of Babel", Information and Software Technology, Elsevier, Vol. 48, No.4, pp. 302-310, 2006.

[23]. E. Khatibi and R. Ibrahim, "Efficient Indicators to Evaluate the Status of Software Development Effort Estimation inside the Organizations", International Journal of Managing Information Technology (IJMIT), Vol. 4, No. 3, pp. 23-32, 2012.

[24]. A. Zaid et al., "Issues in Software Cost Estimation", International Journal of Computer Science and Network Security, Vol. 8, No. 11, pp. 350-356, 2008.

[25]. K. Borte and M. Nerland, "Software Effort Estimation as Collective Accomplishment", Scandinavian Journal of Information Systems, Vol. 22, No. 2, pp. 65-98, 2010.

[26]. L. McClain et al., "Comparative Assessment of Software Development Project Effort Estimation Tools", Annals of the Master of Science in Computer Science and Information Systems at UNC Wilmington, Vol. 5, No. 2, pp. 1-4, 2011.

[27]. J. Popovic and D. Bojic, "A Comparative Evaluation of Effort Estimation Methods in the Software Life Cycle", Computer Science and Information Systems, Vol. 9, No. 1, pp. 455-484,2012.

[28]. D. Toka and O. Turetken, "Accuracy of Contemporary Parametric Software Estimation Models: A Comparative Analysis", 39th IEEE Euromicro Conference Series on Software Engineering and Advanced Applications, Santander, UK, pp. 313-316, 2013.

[29]. O. Tailor, J. Saini and P. Rijwani, "Comparative Analysis of Software Cost and Effort Estimation Methods: A Review", International Journal of Computer Science and Mobile Computing, Vol. 3, No. 4, pp. 1364-1374, 2014.

[30]. Y. Ahn, J. Suh, S. Kim and H. Kim, "The Software Maintenance Project Effort Estimation Model Based on Function Points", Journal of Software Maintenance and Evolution: Research and Practice, Vol. 15, No. 2, pp. 71-85, 2003.

[31]. R. Popli and N. Chauhan, "An Agile Software Estimation Technique based on Regression Testing Efforts", 13th Annual International Software Testing Conference in India, Bangalore, India, pp. 1-9, 2013.

[32]. C. F. Kemerer, "An Empirical Validation of Software Cost Estimation Models", Communications of the ACM, Vol. 30, No. 5, pp. 416-429, 1987.

[33]. T. R. Benala, S. Dehuri and R. Mall, "Computational Intelligence in Software Cost Estimation: An Emerging Paradigm", ACM SIGSOFT Software Engineering Notes, Vol. 37, No. 3, pp. 1-7, 2012.

[34]. D. Kashyap and A. K. Misra, "Software Development Cost Estimation using Similarity Difference between Software Attributes", International Conference on Information Systems and Design of Communication, pp. 1-6, 2013.

[35]. J. Keung, "Software Development Cost Estimation using Analogy: A Review", In Proceedings of IEEE Australian Software Engineering Conference, ASWEC, Gold Coast, QLD, Australia, pp. 327-336, 2009.

[36]. A. Begel and N. Nagappan, "Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study", International Symposium on Empirical Software Engineering and Measurement (ESEM), ACM, pp. 255-264, 2007.

[37]. A. B. M. Moniruzzaman and S. A. Hossain, "Comparative Study on Agile Software Development Methodologies", Global Journal of Computer Science and Technology, Vol. 13, No. 7, pp. 1-14, 2013.

[38]. K. N. Rao, G. K. Naidu and P. Chakka, "A Study of the Agile Software Development Methods, Applicability and Implications in Industry", International Journal of Software Engineering and the Applications, Vol. 5, No. 2, pp. 35-46, 2011.

[39]. M. Hneif and SH. Ow, "Review of Agile Methodologies in Software Development", International Journal of Research and Reviews in Applied Sciences, Vol.1, pp. 1-8, 2009.

[40]. O. Salo and P. Abrahamsson, "Empirical Evaluation of Agile Software Development: The Controlled Case Study Approach", In Proceedings of the 5th International Conference on Product Focused Software Process Improvement, Springer, Kansai Science City, Japan, Vol. 3009, No. 1, pp. 408-423, 2004.

[41]. T. Dingsoyr, S. Nerur, V. Balijepally and N. B. Moe, "A decade of agile methodologies: Towards explaining agile software development", Journal of Systems and Software, Elseiver, Vol. 85, No. 6, pp. 1213-1221, 2012.

[42]. P. Abrahamsson, J. Warsta, M. T. Siponen and J. Ronkainen, "New Directions on Agile Methods: A Comparative Analysis", 25th IEEE International Conference on Software Engineering, Portland, Oregon, pp. 244-254, 2003.

[43]. S. Jalali and C. Wohlin, "Agile Practices in Global Software Engineering – A Systematic Map", In Proceedings of 5th IEEE International Conference on Global Software Engineering, Princeton, NJ, pp. 45-54, 2010.

[44]. TD. Hellman, A. Sharma, J. Ferreira and F. Maurer: Agile Testing, "Past, Present, and Future – Charting a Systematic Map of Testing in Agile Software Development", IEEE Agile Conference (AGILE), Dallas, TX, pp. 55-63, 2012.

[45]. H. Flora and S. V. Chande, "A Systematic Study on Agile Software Development Methodologies and Practices", International Journal of Computer Science and Information Technologies, Vol. 5, No. 3, pp. 3626-3637, 2014.

[46]. A. S. Alqahtani and J. D. Moore et al., "The Challenges of Applying Distributed Agile Software Development: A Systematic Review", International Journal of Computer Science and Information Technologies, Vol. 5, No. 2, pp. 23-36, 2013.

[47]. T. Chow and D. B. Cao, "A Survey Study of Critical Success Factors in Agile Software Projects", The Journal of Systems and Software, Elsevier, Vol. 81, No. 6, pp. 961-971, 2008.

[48]. S. C. Misra, V. Kumar and U. Kumar, "Identifying Some Important Success Factors in Adopting Agile Software Development Practices", The Journal of Systems and Software, Elsevier, Vol. 82, No. 11, pp. 1869-1890, 2009.

[49]. L. Cao: Estimating Agile Software Project Effort, "An Empirical Study", In Proceedings of Americas Conference on Information Systems, pp. 1907-1916, 2008.

[50]. M. Kunz, R. Dumke and N. Zenker, "Software Metrics for Agile Software Development", In Proceedings on 19th IEEE Australian Conference on Software Engineering, Perth, WA, Australia, pp. 673-678, 2008.

[51]. T. Honglei, S. Wei and Z. Yanan, "The Research on Software Metrics and Software Complexity Metrics", In Proceedings of IEEE International Forum on Computer Science-Technology and Applications, Chongqing, China, pp. 131-136, 2009.

[52]. N. Fenton: Software Measurement, "A Necessary Scientific Basis", IEEE Transactions on Software Engineering, Vol. 20, No. 3, pp. 199-206, 1994.

[53]. S. Dhawan and N. Juneja, "Relevance of Software Metrics for a Software Project", International Journal of Advanced Research in Engineering and Applied Sciences, Vol. 2, No. 10, pp. 77-87, 2013.

[54]. S. Pasupathy and R. Bhavani, "Object Oriented Metrics Estimation and Performance Matching", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 4, No. 1, pp. 1242-1248, 2014.

[55]. G. Kaur and K. Bahl, "Software Reliability, Metrics, Reliability Improvement Using Agile Process", International Journal of Innovative Science, Engineering & Technology, Vol. 1, No. 3, pp. 143-147, 2014.

[56]. N. E. Fenton and M. Neil: Software Metrics, "Roadmap", In Proceedings of the ACM Conference on the Future of Software Engineering, pp. 357-370, 2000.
[57]. N. Vanith and R. ThirumalaiSelvi, "A Report on the Analysis of Metrics and Measures on Software Quality Factors- A Literature Study", International Journal of Computer Science and Information Technologies, Vol. 5, No. 5, pp. 6591-6595, 2014.
[58]. J. Heidenberg et al., "A Metrics Model to Measure the Impact of an Agile Transformation in Large Software Development Organizations", Agile Processes in Software Engineering and Extreme Programming, Lecture Notes in Business Information Processing, Springer, Vol. 149, No. 1, pp. 165-179, 2013.
[59]. M. Tihinen, P. Parviainen, R. Kommeren and J. Rotherham, "Matrics and Measurements in Global Software Development", International Journal on Advances in Software, Vol. 5, No. (3 & 4), pp. 278-292, 2012.
[60]. R. Popli, N. Chauhan and H. Sharma, "Prioritising User Stories in Agile Environment", In Proceedings of IEEE International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), Ghaziabad, India, pp. 515-519, 2014.
[61]. K. M. Saravana, G. N. Basavaraj, Rajkumar and A. Kovalan, "A Case Study on Agile User Stories Prioritization Using Imaginative Standard", International Journal of Engineering Research and Applications, Vol. 2, No. 5, pp. 472-480, 2012.
[62]. J. M. Desharnais et al., "Using the COSMIC Method to Evaluate the Quality of the Documentation of Agile User Stories", IEEE Joint Conference of the 21st International Workshop on Software Measurement and 6th International Conference on Software Process and Product Measurement, pp. 269-272, 2011.
[63]. Anita and N. Chauhan, "A Risk Based Story Prioritization Technique in An Agile Environment", International Journal of Advance Foundation and Research in Computer (IJAFRC), Vol. 1, No. 7, pp. 16-25, 2014.
[64]. A. Jaqueira et al., "Using i* Models to Enrich User Stories", In Proceedings of the 6th International Workshop, Valencia, Spain, pp. 55-60, 2013.
[65]. A. M. Moreno and A. Yague, "Agile User Stories Enriched with Usability", Agile Processes in Software Engineering and Extreme Programming, Lecture Notes in Business Information Processing, Springer, Vol. 111, No. 1, pp. 168-176, 2012.
[66]. V. Schetinger et al., "User Stories as Actives for Game Development", Brazilian Symposium on Games and Digital Entertainment, pp. 1-4, 2011.
[67]. C. O'hEocha and K. Conboy, "The Role of the User Story Agile Practice in Innovation", Lean Enterprise Software and Systems, Lecture Notes in Business Information Processing, Springer, Vol. 65, No. 1, pp. 20-30, 2010.
[68]. E. Miranda et al., "Sizing User Stories using Paired Comparisons", Journal of Information and Software Technology, ACM, Vol. 51, No. 9, pp. 1327-1337, 2009.
[69]. DX. Houston et al., "Modeling User Story Completion of an Agile Software Process", ACM Int. Conference on Software and System Process, pp. 88-97, 2013.
[70]. P. Abrahamson, I. Fronza, R. Moser, J. Vlasenko and W. Perdrycz, "Predicting Development Effort from User Stories", ACM International Symposium on Empirical Software Engineering and Measurement, Banff, Canada, pp. 400-403, 2011.
[71]. L. Ludlow, "The Application of User Stories for Strategic Planning", In Proceedings of the 8th International Conference on Agile Processes in Software Engineering and Extreme Programming, LNCS, Springer, Vol. 4536, No. 1, pp. 198-202, 2007.
[72]. M. Qasaimeh and A. Abran, "Extending Extreme Programming User Stories to Meet ISO 9001 Formality Requirements", Journal of Software Engineering and Applications, Vol. 4, No. 11, pp. 626-638, 2011.
[73]. G. V. Valkenhoef et al., "Product and Release Planning Practices for Extreme Programming", In Proceedings of the 11th International Conference on XP, Lecture Notes in Business Information Processing, Springer, Vol. 48, No. 1, pp. 238-243, 2010.
[74]. H. Mcheick, "Improving and Survey of Extreme Programming Agile Methodology", International Journal of Advanced Computing, Vol. 3, No. 3, pp. 87-92, 2011.
[75]. Y. Dzhurov, I. Krasteva and S. Ilieva, "Personal Extreme Programming – An Agile Process for Autonomous Developers", In Proceedings of International Conference on Software, Services \& Semantic Technologies, Bulgaria, pp. 252-259, 2009.
[76]. J. Erickson, K. Lyytinen and K. Siau, "Agile Modeling, Agile Software Development and Extreme Programming: The State of Research", Journal of Database Management, Vol. 16, No. 4, pp. 88-100, 2005.
[77]. T. Goto et al.: EPISODE, "An Extreme Programming Method for Innovative Software Based on Systems Design and its Practical Study", International Journal of Software Engineering & Applications, Vol. 5, No. 5, pp. 1-13, 2014.
[78]. K. Braithwaite and T. Joyce, "XP Expanded: Distributed Extreme Programming", In Proceedings of the 6th International Conference on XP, Extreme Programming and Agile Processes in Software Engineering, Springer, Vol. 3556, pp. 180-188, 2005.
[79]. S. Wood, G. Michaelides and C. Thomson, "Successful Extreme Programming: Fidelity to the Methodology or Good Teamworking?", Information and Software Technology, Elsevier, Vol. 55, No. 4, pp. 660-672, 2013.
[80]. W. A. Wood and W. L. Kleb, "Extreme Programming in a Research Environment, Extreme Programming and Agile Methods – XP/ Agile Universe", LNCS, Springer, Vol. 2418, No. 1, pp. 89-99, 2002.
[81]. L. Layman, L. Williams, D. Damian and H. Bures, "Essential Communication Practiced for Extreme Programming in a Global Software Development Team", Information and Software Technology, Elsevier, Vol. 48, No. 9, pp. 81-794, 2006.
[82]. R. Fojtik, "Extreme Programming in Development of Specific Software", World Conference on Information Technology, Procedia Computer Science, Elsevier, Vol. 3, No. 1, pp. 1464-1468, 2011.
[83]. G. V. Valkenhoef, T. Tervonen, Bd. Brock and D. Postmus, "Quantitative Release Planning in Extreme Programming", Information and Software Technology, Elsevier, Vol. 53, No. 11, pp. 1227-1235, 2011.
[84]. X. Ge et al., "Extreme Programming Security Practices", In Proceedings of the 8th International Conference on Agile Processes in Software Engineering and Extreme Programming, LNCS, Springer, Como, Italy, Vol. 4536, No. 1, pp. 226-230, 2007.
[85]. R. Agarwal and D. Umphress, "Extreme Programming for a Single Person Team", In Proceedings of the 46th Annual Southeast Regional Conference on XX, ACM, Auburn, Alabama, pp. 82-87, 2008.
[86]. P. S. Grisham and D. E. Perry, "Customer Relationships and Extreme Programming", Workshop on Human and social factors of software engineering, ACM SIGSOFT Software Engineering Notes, Vol. 30, No. 4, pp. 1-6, 2005.
[87]. NF. LeJeune, "Teaching Software Engineering Practices with Extreme Programming", Journal of Computing Sciences in Colleges, Vol. 21, No. 3, pp. 107-117, 2006.
[88]. V. B. Misic, "Perceptions of Extreme Programming: An Exploratory Study", ACM SIGSOFT Software Engineering Notes, Vol. 31, No. 2, pp. 1-8, 2006.
[89]. G. Hedin, L. Bendix and B. Magnusson, "Introducing software engineering by means of Extreme Programming", In Proceedings of the 25th IEEE International Conference on Software Engineering, Portland, Oregon, USA, pp. 586-593, 2003.
[90]. C. Malhotra and A. Chug, "Agile Testing with Scrum – A Survey", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, No. 3, pp. 452-459, 2013.
[91]. IR. Lima et al., "Adapting and Using Scrum in Software Research and Development Laboratory", Salesian Journal on Information Systems, Vol. 9, pp. 16-23, 2012.
[92]. Zg. Hu, Q. Yuan and X. Zhang, "Research on Agile Project Management with Scrum Method", In Proceedings of IEEE IITA International Conference on Services Science, Management and Engineering (SSME), Zhangjiajie, China, pp. 26-29, 2009.
[93]. L. Williams, G. Brown et al., "Scrum+Engineering Practices: Experiences of Three Microsoft Teams", In Proceedings of the International Symposium on Empirical Software Engineering and Measurement, IEEE-ACM, Banff, AB, Canada, pp. 463-471, 2011.
[94]. J. Cho, "Issues and Challenges of Agile Software Development with SCRUM", Issues in Information System, Vol. IX, No. 2, pp. 188-195, 2008.
[95]. M. Agarwal and R. Majumdar, "Tracking Scrum Projects Tools, Metrics and Myths About Agile", International Journal of Emerging Technology and Advanced Engineering, Vol. 2, No. 3, pp. 97-104, 2012.
[96]. N. S. Morampudi and G. Raj, "Evaluating Strengths and Weaknesses of Agile Scrum Framework using Knowledge

Management", International Journal of Computer Applications, Vol. 65, No. 23, pp. 1-6, 2013.
[97]. D. Vijay and G. Ganapathy, "Guidelines to Minimize the Cost of Software Quality in Agile Scrum Process", International Journal of Software Engineering and Applications, Vol. 5, No. 3, pp. 61-69, 2014.
[98]. V. Mahnic, "A Case Study on Agile Estimating and Planning using Scrum", Electronics and Electrical Engineering, Vol. 5, No. 111, pp. 123-128, 2011.
[99]. H. Guang-yong, "Study and Practice of Import Scrum Agile Software Development", In Proceedings of the 3rd IEEE International Conference on Communication Software and Networks (ICCSN), Xi'an, China, pp. 217-220, 2011.
[100]. K. Vlaanderen, S. Jansen, S. Brinkkemper and E. Jaspers, "The Agile Requirements Refinery: Applying SCRUM Principles to Software Product Management", Information and Software Technology, Elsevier, Vol. 53, No. 1, pp. 58-70, 2011.
[101]. Hossain et al., "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 (ICSSP), ACM, pp. 110-119, 2011.
[102]. P. Agarwal, "Continuous SCRUM: Agile Management of SAAS products", In Proceedings of the 4th India Software Engineering Conference (ISEC), ACM, Thiruvananthapuram, Kerala, India, pp. 51-60, 2011.
[103]. O. Ktata and G. Levesque, "Designing and Implementing a Measurement Program for Scrum Teams: What do Agile Developers Really Need and Want?", In Proceedings of the Third C* Conference on Computer Science and Software Engineering, ACM, Montréal, Quebec, Canada, pp. 101-107, 2010.
[104]. O. Adalier, A. Ugur, S. Korukoglu and K. Ertas, "A New Regression Based Software Cost Estimation Model Using Power Values", Intelligent Data Engineering and Automated Learning (IDEAL), LNCS, Springer, Birmingham, UK, Vol. 4881, No. 1, pp. 326-334, 2007.
[105]. P. S. Rao and R. K. Kumar, "Software Effort Estimation through a Generalized Regression Neural Network", Emerging ICT for Bridging the Future – Proceedings of the 49th Annual Convention of the Computer Society of India (CSI), Advances in Intelligent Systems and Computing, Springer, India, Vol. 337, No. 1, pp. 19-30, 2015.
[106]. V. S. Dave and K. Dutta, "Neural Network Based Models for Software Effort Estimation: A Review", Artificial Intelligence Review, Springer, Vol. 42, pp. 295-307, 2012.
[107]. J.N.V.R. Swarup Kumar et al., "A Novel Model for Software Effort Estimation Using Exponential Regression as Firing Interval in Fuzzy Logic", Second International Conference on Computer Networks and Information Technologies, Communications in Computer and Information Science, Springer, Vol. 142, pp. 118-127, 2011.
[108]. M. Cohn, "Agile Estimating and Planning", in Prentice Hall PTR, 2005.
[109]. M. Shin and A. L. Goel, "Empirical Data Modeling in Software Engineering Using Radial Basis Functions", IEEE Transactions on Software Engineering, Vol. 26, No. 6, pp. 567-576, 2000.
[110]. K. Srinivasan and D. Fisher, "Machine Learning Approaches to Estimating Software Development Effort", IEEE Transactions on Software Engineering, Vol. 21, No. 2, pp. 126-137, 1995.
[111]. B. Lo and X. Gao, "Assessing Software Cost Estimation Models: criteria for accuracy, consistency and regression", Australasian Journal of Information Systems, Vol. 5, No. 1, pp. 30-44, 1997.
[112]. T. Foss, E. Stensrud, B. Kitchenham and I. Myrtveit, "A Simulation Study of the Model Evaluation Criterion MMRE", IEEE Transactions on Software Engineering, Vol. 29, No. 11, pp. 985-995, 2003.
[113]. K. K. Shukla, "Neuro-genetic Prediction of Software Development Effort", Information and Software Technology, Elsevier, Vol. 42, No. 10, pp. 701-713, 2000.
[114]. G. R. Finnie, G. E. Wittig and J. M. Desharnais, "A Comparison of Software Effort Estimation Techniques: Using Function Points with Neural Networks, Case-based Reasoning and Regression Models", Journal of Systems and Software, Elsevier, Vol. 39, No. 3, pp. 281-289, 1997.
[115]. A. Heiat, "Comparison of Artificial Neural Network and Regression Models for Estimating Software Development Effort", Information and Software Technology, Elsevier, Vol. 44, No. 15, pp. 911-922, 2002.
[116]. H. Park and S. Baek, "An Empirical Validation of a Neural Network Model for Software Effort Estimation", Expert Systems with Applications, Elsevier, Vol. 35, No. 3, pp. 929-937, 2008.
[117]. K. V. Kumar, V. Ravi, M. Carr and N. R. Kiran, "Software Development Cost Estimation using Wavelet Neural Networks", The Journal of Systems and Software, Elsevier, Vol. 81, No. 11, pp. 1853-1867, 2008.
[118]. G. Wittig and G. Finnie, "Estimating Software Development Effort with Connection Models", Information and Software Technology, Vol. 39, No. 1, pp. 469-476, 1997.
[119]. B. T. Rao, B. Sameet, G. K. Swathi, K. V. Gupta, Ch. RaviTeja and S. Sumana, "A Novel Neural Network Approach for Software Cost Estimation Using Functional Link Artificial Neural Network (FLANN)", International Journal of Computer Science and Network Security, Vol. 9, No. 6, pp. 126-131, 2009.
[120]. A. Idri and A. Zahi, "Emilia Mendes and Abdelali Zakrani, Software Cost Estimation Models Using Radial Basis Function Neural Networks", International Conference on Software Process and Product Measurement, Springer, Vol. 4895, pp. 21-31, 2008.
[121]. P. V. G. D. Prasad Reddy, K. R. Sudha, P. Rama Sree and S. N. S. V. S. C. Ramesh, "Software Effort Estimation using Radial Basis and Generalized Regression Neural Networks", Journal of Computing, Vol. 2, No. 5, pp. 87-92, 2010.
[122]. R. Strba et al., "Intelligent software support of the SCRUM process", Information Modeling and Knowledge Bases, Vol. 272, No. 1, pp. 408-416, 2014.
[123]. A. Panda, S. M. Satapathy and S. K. Rath, "Neural Network Models for Agile Software Effort Estimation based on Story Points", In Proceedings of the International Conference on Advances in Computing, Control and Networking, pp. 26-30, 2015.
[124]. T. J. Gandomani, K. T. Wei and A. K. Binhamid, "A Case Study Research on Software Cost Estimation Using Experts Estimates, Wideband Delphi, and Planning Poker Technique", International Journal of Software Engineering and Its Applications, Vol. 8, No. 11, pp. 173-182, 2014.
[125]. Z. C. Ani and S. Basri, "A Case Study of Effort Estimation in Agile Software Development using Use Case Points", In Proceedings of Special Issue- Agile Symposium, Science International, malaysia, Vol. 25, No. 4, pp. 1111-1115, 2013.
[126]. M. Jorgensen and S. Grimstad, "Software Development Effort Estimation: Demystifying and Improving Expert Estimation", Simula Research Laboratory, Springer, pp. 381-403, 2010.
[127]. S. Grimstad and M. Jorgensen, "Inconsistency of Expert Judgment-based Estimates of Software Development Effort", Journal of Systems and Software, Elsevier, Vol. 80, No. 11, pp. 1770-1777, 2007.
[128]. M. Jorgensen, "A Review of Studies on Expert Estimation of Software Development Effort", Journal of Systems and Software, Elsevier, Vol. 70, pp. 37-60, 2004.
[129]. M. Jorgensen, "Top-down and Bottom-up Expert Estimation of Software Development Effort", Information and Software Technology, Vol. 46, No. 1, pp. 3-16, 2004.
[130]. N. C. Haugen, "An Empirical Study of Using Planning Poker for User Story Estimation", In Proceedings of IEEE Agile Conference, Minneapolis, pp. 1-9, 2006.
[131]. V. Mahnic and T. Hovelja, "On Using Planning Poker for Estimating User Stories", The Journal of Systems and Software, Elsevier, Vol. 85, No. 9, pp. 2086-2095, 2012.
[132]. K. Molokken-Ostvold, N. C. Haugen and H. C. Benestad, "Using Planning Poker for Combining Expert Estimates in Software in Software Projects", The Journals of Systems and Software, Elsevier, Vol. 81, No. 12, pp. 2106-2117, 2008.
[133]. H. O. Farahneh and A. A. Issa, "A Linear Use Case Based Software Cost Estimation Model", International Scholarly and Scientific Research and Innovation, Vol. 5, No. 1, pp. 385-389, 2011.
[134]. S. M. Satapathy et al., "Story Point Approach based Agile Software Effort Estimation using Various SVR Kernel Methods", In Proceedings of 26th International Conference on Software Engineering & Knowledge Engineering, pp. 304-307, 2014.
[135]. A. L. I. Oliveira, "Estimation of Software Project effort with Support Vector Regression", Neurocomputing, Elsevier, Vol. 69, No. (13-15), pp. 1749-1753, 2006.
[136]. N. Mittas and E. Papatheocharous et al., "Integrating Non-Parametric Models with Linear Components for Producing Software Cost Estimations", The Journal of Systems and Software, Elsevier, Vol. 99, No. 1, pp. 120-134, 2015.

[137]. T. C. D. Dean, "Decomposed Versus Holistic Estimates of Effort Required for Software Writing Tasks", Management Science, Vol. 43, No. 7, pp. 1029-1045, 1997.
[138]. M. M. Kirmani and A. Wahid, "Impact of Modification Made in Re-UCP on Software Effort Estimation", Journal of Software Engineering and Applications, Vol. 8, No. 6, pp. 276-289, 2015.
[139]. A.W.M.M. Parvez, "Efficiency Factor and Risk Factor Based User Case Point Test Effort Estimation Model Compatible with Agile Software Development", In Proceedings on Information Technology and Electrical Engineering, pp. 113-118, 2013.
[140]. O. Fedotova, L. Teixeira and H. Alvelos, "Software Effort Estimation with Multiple Linear Regression: Review and Practical Application", Journal of Information Science and Engineering, Vol. 29, No. 1, pp. 925-945, 2013.
[141]. W. L. Du et al., "A Hybrid Intelligent Model for Software Cost Estimation", Journal of Computer Science, Vol. 9, No. 11, pp. 1506-1513, 2013.
[142]. O. Chouseinoglou and O. M. Aydın, "A Fuzzy Model of Software Project Effort Estimation", Turkish Journal of Fuzzy Systems, Vol. 4, No. 2, pp. 68-76, 2013.
[143]. M. Kaiser and C. Ullrich, "Estimation Accuracy in Large is Programs Insights from A Descriptive Case Study", Twenty Second European Conference on Information Systems, Tel Aviv, Israel, pp. 1-14, 2014.
[144]. R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy", International Journal of Forecasting, Vol. 22, No. 4, pp. 679-688, 2006.
[145]. The Standish Group. (1994) CHAOS Report. [Online]. Availaible on:https://www.standishgroup.com /sample_research_ files/chaos_report _1994. pdf,.
[146]. The Standish Group. (1995) CHAOS Report. [Online]. Availaible on: https://net.educause.edu/ ir/library/pdf/NCP08083B.pdf
[147]. The Standish Group. (2001) CHAOS Report. [Online]. Availaible on: http://www.cin.ufpe.br/~gmp /docs/papers/extreme _chaos2001.pdf
[148]. J. L. Eveleens and C. Verhoef, "The Rise and Fall of the Chaos Report Figures", Software, IEEE, Vol. 27, No. 1, pp. 30-36, 2009.
[149]. The Standish Group. (2009) CHAOS Summary. [Online]. Availaible on:https://www.classes.cs.uchicago. edu/archive/2014/fall/512101/required.reading/Standish.Group.Cha os.2009.pdf
[150]. The Standish Group. (2013) CHAOS Manifesto. [Online]. Availaible on: https://larlet.fr /static/david/stream/ChaosManifesto2013.pdf
[151]. The Standish Group. (2014) Big Bang Boom. [Online]. Availaible on: https://www.standishgroup.com /sample_research _files/BigBangBoom.pdf
[152]. The Standish Group. (2015) Haze. [Online]. Availaible on: https://www.standishgroup.com/ sample _research_files/Haze4.pdf
[153]. 9th Annual State of Agile Survey. (2015) [Online]. Availaible on: https://www.versionone.com/pdf/state-of-agile-development-survey-ninth .pdf
[154]. Y. I. Alzoubi, A. Q. Gill and A. Al-Ani, "Empirical Studies of Geographically Distributed Agile Development Communication Challenges: A Systematic Review", Information & Management, Elsevier, Vol. 53, No. 1, pp. 22-37, 2016.
[155]. S. Sequeira et al., "Simple Method Proposal for Cost Estimation from Work Breakdown Structure", Conference on ENTERprise Information Systems/International Conference on Project MANagement/Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN / HCist , Elsevier, Algarve, Portugal, Vol. 64, No. 1, pp. 537-544, 2015.
[156]. F. S. Gharehchopogh and A. Pourali, "A New Approach based on Continuous Genetic Algorithm in Software cost Estimation", Journal of Scientific Research and Development, Vol. 2, No. 4, pp. 87-94, 2015.
[157]. T. S. D. Silva, M. S. Silveira and F. Maurer, "Usability Evaluation Practices within Agile Development", 48th Hawaii International Conference on System Sciences, IEEE, Kauai, Hawaii, USA, pp. 5133-5142, 2015.
[158]. N. Ghatasheh, H. Faris, I. Aljarah and R. M. H. Al-Sayyed, "Optimizing Software Effort Estimation Models Using Firefly Algorithm", Journal of Software Engineering and Applications, Vol. 8, No. 3, pp. 133-142, 2015.
[159]. M. M. Kirmani and A. Wahid, "Impact of Modification Made in Re-UCP on Software Effort Estimation", Journal of Software Engineering and Applications, Vol. 8, No. 6, pp. 276-289, 2015.
[160]. J. Huang, Y. F. Li and M. Xie, "An Empirical Analysis of Data Preprocessing for Machine Learning-based Software Cost Estimation", Information and Software Technology, Elsevier, Vol. 67, No. 1, pp. 108–-127, 2015.
[161]. S. Kumari and S. Pushkar, "A Genetic Algorithm Approach for Multi-criteria Project Selection for Analogy-Based Software Cost Estimation", Computational Intelligence in Data Mining, Springer, Vol. 3, No. 33, pp. 13-24, 2014.
[162]. P. A. Whigham, C. A. Owen and S. G. Macdonell, "A Baseline Model for Software Effort Estimation", ACM Transactions on Software Engineering and Methodology (TOSEM), Vol. 24, No. 3, pp. 20:1-20:11, 2015.
[163]. M. Usman, E. Mendes and J. Börstler, "Effort Estimation in Agile Software Development: A Survey on the State of the Practice", Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering (EASE), ACM, Nanjing, China, pp. 12:1-12:10, 2015.
[164]. G. S. Matharu, A. Mishra, H. Singh and P. Upadhyay, "Empirical Study of Agile Software Development Methodologies: A Comparative Analysis", ACM SIGSOFT Software Engineering Notes, Vol. 40, No. 1, pp. 1-6, 2015.
[165]. C. Jones: {A Short History of Software Estimation Tools (version 12.0)}. (2003) [Online]. Availaible on:http://namcookanalytics.com /wp-content/uploads/2013/ 08/ESTtoolHISTORY2013.pdf
[166]. R. Hintermann, "Introduction on Software Cost Estimation", Seminar on Software Cost Estimation, 2002, pp. 1-11.

Appendix: History of Software Cost Estimation

- *1940*- It was the beginning of the computer era and towards the digitalization of information.
- *Before 1970*- Simple rules of thumb and some simple algorithms of estimation were used and hence error rate in the prediction was high.
- *Around 1970*- Researchers thought about the automation tools for more accurate and fast prediction of cost and during this period the first automated tool for software cost estimation was built. It was developed by Barry Boehm and named as COCOMO (Constructive Cost Model), and it is given in the book Software Engineering Economics by Bohem.
- *Around 1975*- Function point metric for the estimation was developed.
- *1977*- Frank Freiman developed PRICE-S commercial tool for the United States commercial purposes.
- *1979*- Lawrence H. Putnam introduced the SLIM (Software life cycle management)- second commercial tool for the United States.
- *1981*- B. Bohem emphasized on some useful algorithms for COCOMO model.
- *1982*- In the book "controlling software projects", T. deMarco introduced a functional metric with some inherited features of Albrecht's function point metric.
- *1983*- Ada programming language was developed by the U.S. Department of Defense (DoD) for reducing the cost of developing large military software systems. Also, in the same year, mark-II function point metric was introduced by Charles Symons.
- *1984*- IBM did a major revision in its function point metric.
- *1985*- C. Jones with his colleagues developed the SPQR/20 (software productivity quality and reliability)

tool for estimation. Also, the concept of Function Point was extended by C. Jones, to include the effect of computationally difficult problems.

- *1986*- IFPUG (International Function Point Users Group) was introduced in Toronto, Canada.
- *1990*- Barry Boehm, at the University of Southern California, started extending and revising the existing COCOMO model.
- *1991*- Hans Koolen and Michel van Genuchten contributed to the field of cost estimation by introducing some of useful tools and methods for software cost estimation.
- *1992*- R. Betteridge used the mark-II function points metrics to predict the cost of some projects.
- *1993*- New version of COCOMO model which is COCOMO 2.0, appeared in 1994.
- *1997*- Some of the existing models were revised for increasing their accuracy rate.
- *1998*- A new model called MARCS is introduced by Chatzoglou, for predictions of software effort.
- *1999*- J. J. Dolado applied genetic programming for predicting cost functions.
- *Around 2008*- C. Jones filed several patent applications for introducing a new method of high-speed software sizing based on pattern matching. The advantage of this method is its applicability before requirements. This new method is embedded with a new cost estimation tool called Software Risk Master (SRM).
  In the last decade- COCOMO II, CostXpert, SLIM, KnowledgePlan, SEER, SRM and True Price are the tools and methods have emerged and become popular among researchers and software industries. A number of researchers have successfully applied NN, EJ, PP, UCP, MUCP, LR, etc. methods for estimating small as well as large complex software projects according to different suitable working conditions (discussed earlier in this paper) [165, 166].