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 coordination 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
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

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

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.
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

<table>
<thead>
<tr>
<th>Estimation Mechanisms</th>
<th>Papers</th>
<th>Accuracy Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN</td>
<td>[109-123]</td>
<td>MRE, MMRE, PRED(X), MAPE, MdMRE, R², MSE, MMER</td>
</tr>
<tr>
<td>EJ</td>
<td>[124, 126-129]</td>
<td>MRE, MdMRE</td>
</tr>
<tr>
<td>PP/DA</td>
<td>[124, 130-132]</td>
<td>MRE, BRE</td>
</tr>
<tr>
<td>UCP</td>
<td>[111, 125, 133, 134]</td>
<td>MRE, MMRE, MMER, PRED(X), R², MSE</td>
</tr>
<tr>
<td>MUCP</td>
<td>[133, 134, 139]</td>
<td>MRE, MMRE, MdMRE, PRED(X)</td>
</tr>
<tr>
<td>LR</td>
<td>[135, 136]</td>
<td>MRE, MMRE, MMER, PRED(X), MdMRE</td>
</tr>
<tr>
<td>WD</td>
<td>[124, 132]</td>
<td>MRE</td>
</tr>
<tr>
<td>BU/TD</td>
<td>[129, 137]</td>
<td>MRE</td>
</tr>
</tbody>
</table>

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

### Table 4. Abbreviations used for Table 3.

<table>
<thead>
<tr>
<th>For Estimation Mechanism</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NN</td>
<td>Neural Network</td>
<td></td>
</tr>
<tr>
<td>EJ</td>
<td>Expert Judgment</td>
<td></td>
</tr>
<tr>
<td>PP</td>
<td>Planning Poker</td>
<td></td>
</tr>
<tr>
<td>Da</td>
<td>Disaggregation</td>
<td></td>
</tr>
<tr>
<td>UCP</td>
<td>Use Case Point</td>
<td></td>
</tr>
<tr>
<td>MUCP</td>
<td>Modified Use Case Point</td>
<td></td>
</tr>
</tbody>
</table>

---

**Source of Article**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Science Direct</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>Elsevier</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Springer</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>ACM</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>IEEE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>Others</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>7</td>
<td>10</td>
<td>10</td>
<td>3</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td>7</td>
<td>5</td>
<td>18</td>
<td>11</td>
<td>15</td>
<td>12</td>
<td>9</td>
<td>101</td>
</tr>
</tbody>
</table>
Even so, it is an efficient and useful method for estimating handled by the concerned experts, as well as human errors. Nevertheless, it is also evident that experts’ judgment method is influenced by various subjective factors like bias, influence of work environment, type of projects being used. NN(s) make the best use of the uncertain nature of these inputs, and the various layers work customize the activation function, \( w \) is the weight of input \( p \), and \( b \) is the input bias. A special type of neural network is feed-forward network that allows a uni-directional flow of data. Various feed-forward networks include Radial Basis Function Neural Network (RBFNN), Multi-layer 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].

**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, \( f(\cdot) \) 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), Multi-layer 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]:

\[
MRE_i = \frac{\text{Actual Effort}_i - \text{Estimated Effort}_i}{\text{Actual Effort}_i}
\]  

(ii) Mean Magnitude of Relative Error- MMRE is another popular performance measurement factor 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)”:  

\[
\text{MMRE}_i = \frac{1}{N} \sum_{i=1}^{N} MRE_i
\]  

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 to 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]:

\[
\text{MdMRE}_i = \text{Median}(MRE_i)
\]

(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 (MRE) and it can be represented as follows in “Eq. (4, 5)” [142]:

\[
\text{MMER}_i = \frac{1}{N} \sum_{i=1}^{N} MRE_i
\]

(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]:

\[
\text{MAPE}_i = \frac{\sum_{i=1}^{N} |\text{Actual Effort}_i - \text{Estimated Effort}_i|}{\text{Actual Effort}_i} \times 100
\]

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]:

\[
\text{MSE}_i = \frac{1}{N} \sum_{i=1}^{N} (\text{Actual Effort}_i - \text{Estimated Effort}_i)^2
\]

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]:

\[
\text{BRE} = \frac{\sum_{i=1}^{N} (\text{Actual Effort}_i - \text{Estimated Effort}_i)}{\text{min(Actual Effort, Estimated Effort)}}
\]

(viii) Squared Correlation Coefficient- R² is also known as coefficient of determination. R² is defined as the assessment of the effectiveness of a regression. It can be represented as in “Eq. (9)” [111]:

\[
R^2 = 1 - \frac{\sum_{i=1}^{N} (\text{Actual Effort}_i - \text{Estimated Effort}_i)^2}{\sum_{i=1}^{N} (\text{Actual Effort}_i - \text{Mean}(\text{Actual Effort}))^2}
\]

(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]:

\[
\text{PRED} (x) = \frac{K}{N}
\]

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.

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>
<thead>
<tr>
<th>ET</th>
<th>Suitable circumstances</th>
<th>Difficulties Faced</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN</td>
<td>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.</td>
<td>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</td>
<td>[114, 116, 119]</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Flaws</th>
</tr>
</thead>
<tbody>
<tr>
<td>EJ</td>
<td>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.</td>
<td>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).</td>
</tr>
<tr>
<td>PP/Da</td>
<td>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.</td>
<td>Less research has been done, and little empirical evidence is available regarding accuracies of PP.</td>
</tr>
<tr>
<td>UCP</td>
<td>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.</td>
<td>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.</td>
</tr>
<tr>
<td>MUCP</td>
<td>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.</td>
<td>Same problem is faced by the researchers as with UCP when dealing with the agile software development projects.</td>
</tr>
<tr>
<td>LR</td>
<td>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.</td>
<td>Missing historical data leads to decrease in the accuracy of overall prediction.</td>
</tr>
<tr>
<td>WD</td>
<td>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.</td>
<td>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.</td>
</tr>
<tr>
<td>BU/TD</td>
<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.</td>
<td>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.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Resolution Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution Type-1</td>
<td>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.</td>
</tr>
<tr>
<td>Resolution Type-2</td>
<td>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.</td>
</tr>
<tr>
<td>Resolution Type-3</td>
<td>The projects which are canceled at any point during the development cycle, fall under this category.</td>
</tr>
</tbody>
</table>

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

Table 7. Project success rate over years

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution Type-1</td>
<td>16.2%</td>
<td>27%</td>
<td>26%</td>
<td>28%</td>
<td>34%</td>
<td>29%</td>
<td>35%</td>
<td>32%</td>
<td>37%</td>
<td>39%</td>
</tr>
<tr>
<td>Resolution Type-2</td>
<td>52.7%</td>
<td>40%</td>
<td>28%</td>
<td>49%</td>
<td>51%</td>
<td>53%</td>
<td>46%</td>
<td>44%</td>
<td>42%</td>
<td>43%</td>
</tr>
<tr>
<td>Resolution Type-3</td>
<td>31.1%</td>
<td>33%</td>
<td>46%</td>
<td>23%</td>
<td>15%</td>
<td>18%</td>
<td>19%</td>
<td>24%</td>
<td>21%</td>
<td>18%</td>
</tr>
</tbody>
</table>

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]

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 Development Methods Based Tools](image)

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.

This is an Open Access article distributed under the terms of the Creative Commons Attribution Licence

References


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 Boehm.
- **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. Boehm 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 that 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].