

A Review on Software Cost and Effort Estimation Techniques for Agile Development Process

Manju Vyas^[1], Amit Bohra^[2], Dr. C.S. Lamba^[3] and Abhilasha Vyas^[4]

¹ Assistant Professor, Department of Information Technology, JECRC, Jaipur

¹ Assistant Professor, Department of Information Technology, GIT, Jaipur

¹ Professor, Manipal University, Jaipur

¹ Assistant Professor, G D Goenka University, Gurgaon

Abstract- In software industry, accurate estimation of cost and effort is a very crucial and important task for successful project development. A correct estimate results in timely completion of project and within the estimated budget, while if the estimated cost and effort are not accurate then it may result in project failure in terms of budget and delivery time. There are various development models which have been used in software industry based on organization's capabilities and project requirements. Recent advancements and need of adaptation to changes in requirements resulted in the agile development methodology which gained a lot of popularity in software industries. In this framework, customer plays an active role in the development process. Hence changes in the requirements can be accommodated in any phase of development process. Various techniques are used for software cost and effort estimation including non algorithmic, algorithmic and machine learning techniques. This paper gives a review of estimation techniques used by categorising the techniques into basic estimation techniques, the techniques that are technically applicable for the estimation of projects developed using agile methodology and machine learning based techniques for estimation.

Keywords- Estimation, Agile, Expert based, Machine Learning.

I INTRODUCTION

Estimation is defined as the methodology for anticipating the practical measures which are effort and cost measured in individual hours and capital in the context of software estimation. In software project management area, estimation of effort and cost has become a primary activity as it has a major role to play both at the client and the developer side. With the optimised estimation, the developer side can plan the development as well as monitor the progress and the client side can negotiate contracts as well as set the completion dates. Software project estimation involves estimation of effort, size, staffing, time and cost. For any software project, first the size of the software is estimated and then the effort needed is identified by considering the size as input parameter. Project cost and completion time are calculated by effort estimated.

1.1 Introduction to Agile Methodology

In modern times most of the software companies have shifted to agile methodology since its emergence in 2001. Agile software engineering works on certain guidelines and policies. The major intention is to enhance customer satisfaction as well as timely delivery of software in increments using small and dedicated teams; informal methods; minimal planning in software engineering phases, with simplicity in overall development. The guidelines emphasize delivery over analysis and design and motivates a continuous communication between the customers and the development teams. Software engineers and other project stakeholders like customers, managers and end-users work together on an agile team. It is

important because it meets the challenge of changing requirements which is a feature of today's modern business environment

1.2 Estimation practices in Agile Technology

In traditional development processes the manager determines the workload capacity of every team member by estimating the time required for completion of certain tasks and then work is assigned to every member of the team by considering their total available time. Agile methodology uses a different approach to determine a member's capacity. It assigns work to an entire team and not an individual. Under an agile development process there is an incremental development of the software which results in small iterations and after every iteration the customer feedback is taken which serves as an input for next iterations. This means that the planning and estimation process must be done progressively. The estimation techniques used in Agile are Expert opinion, analogy and disaggregation, Planning Poker and few more. A new estimation technique based on Use Case Points Method has also been proposed and further modified.

The accuracy metrics most frequently used by the existing estimation techniques are Mean Magnitude of Relative Error (MMRE) and Magnitude of Relative Error (MRE).

The size metrics used in agile framework are story points which measures user stories and use case points which are derived from use case diagrams.

SCRUM, Adaptive Software Development, Extreme Programming, Crystal, Feature Driven Development, Dynamic Service Development Method are the models which

are used in agile technology for the estimation of cost and effort.

II LITERATURE REVIEW

The software estimation techniques can be classified into different categories depending on the features of these techniques but there is no universally accepted classification. Moløkken-Østvold *et al.*, 2004 after comparing various surveys grouped the estimation techniques into three categories: expert judgement models, algorithmic model based methods and others. Shepperd and Schofield, 1996 proposed three categories: expert judgment, algorithmic models and analogy methods. Later some researchers suggested machine learning to be the third major category, as Wen *et al.*, 2012.

2.1 Survey on Basic Estimation techniques

Putnam, (1978) proposed a model for effort estimation which was named SLIM. Then a function point based model was given by Albrecht in 1979 which used function point as size metric in place of KLOC which measures the physical component of software. The COCOMO model proposed by Boehm in 1981 is an algorithmic model used to estimate software cost.

Hughes, (1996) proposed an expert judgement model which is based on the expertise of an expert for estimation. The expert judgment model has a major drawback of opinion bias. Similarly FP and LOC based techniques also suffers from the drawback of aligning every model to different estimation environments. Another approach is proposed by Shepperd *et al.*, 1995 which is an analogy based technique and the work evaluated the model with distinct datasets collected from various environments. The main disadvantage of the analogy based technique is the performance of similarity function which is used to find the analogies from the case repository.

Greening, (2002) proposed another estimation technique named Planning Poker and later in 2005 M. Cohn suggested that this method is suitable for estimation of the projects developed using Agile methodology as it can be applicable with the customer-oriented approach of Agile software development.

Jogersen M. *et al.*, (2007) suggested further scope for improvement in software estimation using a systematic literature review of published studies. The review considered 304 software effort and cost estimation papers published in 76 journals. The SLR then categorized the papers according to topic of research, study context estimation technique, and data set. Based on the review, the paper suggested scope for further research on basic cost estimation techniques and experimental evaluation of the existing estimation techniques with the use of more relevant data-sets.

2.2 Survey on Estimation techniques used in agile development

Keaveney and Conboy, (2006) examined the application of traditional estimation techniques like function point based techniques and others for effort estimation in agile framework using various case studies.

Coelhlo and Basu, (2012) have described the steps defined in story point based method used for cost and effort estimation of agile software and identified the areas for further research. Ziauddin *et al.*, (2012) have developed a regression based model for effort estimation of projects developed using agile methodology. The model was then calibrated and tuned using the empirical data set which was collected from 6 project houses with 21 software projects. The results showed that the model had good estimation accuracy in terms of the accuracy metrics MMRE and Pred (n).

Usman *et al.*, (2014) conducted a systematic literature review in which they provided a detailed introduction of the recent trends in the area of effort estimation in agile software development. They reviewed 25 published studies and identified various research gaps relating to the agile methods, size metrics, and accuracy metrics and cost drivers, thus suggesting possibilities for future research areas.

Power, (2011) identified the challenges of implementation of planning poker technique for sizing user stories and proposed a technique called Silent Grouping that can be used to compliment Planning Poker. The paper used a dataset of experiences of seven Scrum teams from Cisco's Unified Communication Business Unit. The paper also described the application of this technique with co-located teams and distributed teams.

Parvez, (2003) investigated the use case point estimation method and concluded that the use case point method provides the estimation depending on the features of the project without considering team properties. So he developed a new model by adding a new layer of efficiency and risk factor to the existing method to increase the effectiveness and performance of the use case point method.

Hussain *et al.*, (2013) proposed an approach which uses function point as the size metrics in agile development and the approach helps in removing problems like formalised user requirements.

Cohn, (2005) suggested that Planning poker is a suitable technique for estimation in agile development process as it supports the person oriented approach of Agile. In this technique estimates are represented by certain numbers written on index cards. Each member holds a deck of cards with values 1,2,3,5,8,13,21... Each value represents the team estimates which are given by either number of story points or ideal days, or other units. Story point is a unit for measurement of user stories which are requirements of the customers for specifying the functionalities. The team of estimators along with the customer representative finalises the project functionalities. After the discussion, each estimator selects one card to show his or her estimate. All cards are then showed at the same time. If the same value is selected, then it is considered as the final estimate. Else, the highest and lowest estimators discuss their reasons and after further discussion a re-estimate is done until a mutual agreement is achieved.

Gandomani *et al.*, (2014) compared the planning poker technique and the Wideband Delphi technique by using two cases. They concluded that Planning poke gave better estimation accuracy than expert estimation and Delphi technique. Planning Poker also reduced the total financial risks of projects. The study examined data from two companies and showed that the average estimation error in

terms of magnitude relative error was 14.8% for expert estimation, 7.6% for Delphi technique and 7.1% using Planning poker.

Hamouda, (2014) proposed a technique which measures the software size in relative terms using story points, which was then applied on different project of level three CMMI.

Ungan *et al.*, (2014) compared the story point based technique and planning poker technique by using various regression models and artificial neural network and concluded that the estimation accuracy increases with the use of these techniques.

Tanveer *et al.*, (2016) conducted a case study research in which data was collected from team of a German multinational software company which used Agile development framework. The results identified that factors like developer's experience and knowledge as well as the impact of the system's changes affect the estimation accuracy. Further they concluded that tool must be developed which includes expert experience and knowledge and considers the various cost drivers so that the effectiveness of effort estimation process is improved.

2.3 Survey on Software Estimation using Machine Learning Techniques

Wen *et al.*, (2012) performed a systematic literature review of published studies on machine learning models. They analysed the studies on the basis of four factors which are type of Machine Learning technique, accuracy of the technique, model used and estimation metrics. They identified 84 studies and observed that eight types of Machine Learning techniques are used in estimation models and concluded that the estimation accuracy of Machine Learning models is better than non Machine Learning models. Also the study concluded that different Machine Learning models have different strengths and weaknesses.

Baskeles *et al.*, (2007) reported that higher confidence and accuracy can be achieved with well defined and structured use cases. The paper also suggested that the algorithmic methods should not replace the expert judgment methods but should be used in conjunction with expert methods.

Braga *et al.*, (2008) proposed a new technique which used genetic algorithm for selection of an optimal feature subset and optimized regression parameters so as to improve the effort estimation accuracy. They validated the approach by experimenting it using two software projects datasets in which the simulation showed that the proposed Genetic Algorithm based approach improved the effort estimation accuracy.

Satapathy *et al.*, (2014) used stochastic Gradient Boosting technique which is one of the machine learning techniques for improving estimation prediction. In this paper they used the class point approach for calculation of effort and then further optimised the parameters using SGB technique to improve the prediction accuracy. They further compared the performance of the proposed model with Radial Basis Function Network to highlight the increase in accuracy.

Satapathy S. and Rath S., (2014) also proposed Support Vector Regression based technique which is one of the machine learning techniques for optimising effort estimation accuracy. The basic concept of Support Vector Regression is based upon the computation of linear regression function. Here use case point approach is used for calculation of effort

and then various SVR kernels are applied for optimising the results.

Peter Hearty, *et al.*, (2009) proposed the extension of Bayesian Network by including a temporary dimension to the network model. The model was built using AgenaRisk tool and then was applied on an industrial project where the model learns from initial data entered by the parameter learning. The results showed that the model gave extremely accurate effort predictors for XP based projects.

Abrahamsson *et al.*, (2011) proposed a technique for estimating effort using user stories. The technique extracted certain predictors from the specified user stories. They validated the proposed technique by applying it to two software projects developed using Agile. The paper concluded that accuracy of effort estimation is good if structuring of the user stories is done properly.

III COMPARATIVE ANALYSIS OF THE EXISTING TECHNIQUES

Table 1. Advantages and disadvantages of the estimation techniques

	Non-Algorithmic/ Expert Judgmental	Machine Learning Based techniques
Advantages	<ul style="list-style-type: none"> ➤ These techniques do not require previous measurement data of any project. 	<ul style="list-style-type: none"> ➤ The estimation accuracy is increased as supported by various studies.
Disadvantages	<ul style="list-style-type: none"> ➤ Every time the estimation is done there is a requirement for expert. ➤ The accuracy and the reliability is dependent on the expert involved. 	<ul style="list-style-type: none"> ➤ Since it is data driven, so its accuracy depends on availability of historical project data.

IV CONCLUSION

Most of the algorithmic techniques are data intensive and the major drawback of these techniques is that large data sets are required while in current industrial scenario the data available is mostly incomplete and inconsistent. Also these techniques are complicated to use and no study proves these techniques to be superior or more accurate as compared to non algorithmic or expert judgment techniques. In contrast to algorithmic or data-centric methods, non algorithmic techniques or expert based estimation does not require any project measurement data. Although these techniques are criticised as there is a requirement for expert every time the estimation has to be performed and the accuracy and reliability of the technique depends on the expertise of the expert.

REFERENCES

- [1]. Abrahamsson P., and Koskela J., "Extreme Programming: A Survey of Empirical Data from a

- Controlled Case Study”, Proceedings of International Symposium on Empirical Software Engineering, pp 73-82, 2004
- [2]. Abrahamsson P., Fronza I., Moser R., Vlasenko J., Pedrycz, W., “Predicting development Effort from User Stories” Empirical Software Engineering and Measurements (ESEM) International Symposium, pp 400-403, 2011
- [3]. Basten D., Mellis W., “A current assessment of software development effort estimation.” International symposium on empirical software engineering and measurement, pp 235–244, 2011
- [4]. Beck K., “Extreme Programming Explained”, Reading MA, Addison Wesley.
- [5]. Boehm B., Abts C., Chulani S., “Software development cost estimation approaches—A survey” Annals of Software Engineering, vol 10, pp 177–205, 2001
- [6]. Britto R., Freitas V., Mendes E., Usman M., “Effort Estimation in Global Software Development: A Systematic Literature Review” 2014 IEEE 9th International Conference on Global Software Engineering, pp 135–144, 2014
- [7]. Britto R., Mendes E., Börstler J., “An Empirical Investigation on Effort Estimation in Agile Global Software Development” 2015 IEEE 10th International Conference on Global Software Engineering, pp 38–45, 2015
- [8]. Choetkiertikul M., Dam H.K., Tran T., Pham T., Ghose A., Menzies T., “A deep learning model for estimating story points” Cornell University Library, 2016
- [9]. Coelho E., Basu, A., “Effort Estimation in Agile Software Development using Story Points” International Journal of Applied Information Systems, Foundation of Computer Science, New York, USA, vol 3(7), 2012
- [10]. Cohn M., “Agile estimating and planning”, Pearson Education 2005
- [11]. Grenning J., “Planning Poker or how to avoid analysis paralysis with release planning” Hawthorn Woods: Renaissance Software Consultancy, vol 3, 2002
- [12]. Grimstad S., Jørgensen M., Moløkken-Østfold K., “The clients’ impact on effort estimation accuracy in software development projects” Proceedings - international software metrics symposium 2005, pp 5–14, 2005
- [13]. Hamouda A. E. D., “Using Agile Story Points as an Estimation Technique i0n CMMI Organizations” Agile Conference, pp 16-23, IEEE 2014
- [14]. Haugen N. C., ”An empirical study of using planning poker for user story estimation” Proceedings - AGILE Conference, pp 23–31, 2006
- [15]. Hussain I., Kosseim L., Ormandjieva O., ”Approximation of Cosmic functional size to support early effort estimation in Agile” Data and Knowledge Engineering, 85, pp 2-14 Elsevier 2013
- [16]. Jørgensen M., “A review of studies on expert estimation of software development effort” Journal of Systems and Software, vol 70(1-2), pp 37–60, 2004
- [17]. Jørgensen M., Shepperd M., “Systematic Review of Software Development Cost Estimation Studies” IEEE Transactions on Software Engineering, vol 33(1), pp 33–53, 2007
- [18]. Kang S., Choi S., Baik J., “Model Based Estimation and Tracking Method for Agile Software Project”, International Journal of Human Capital and IT Professionals, vol 3(2), pp 1-15, 2012
- [19]. Kitchenham B., ”Preliminary guidelines for empirical research in software engineering”, IEEE Transactions on Software Engineering, pp 721-733, 2002
- [20]. Magazinius A., Feldt R., “Confirming distortional behaviours in software cost estimation practice” Proceedings - 37th euromicro conference on software engineering and advanced applications, pp 411, 2011.
- [21]. McDaid K.,” ‘Managing Uncertainty in Agile Release Planning”, 18th International Conference on Software Engineering & Knowledge Engineering, pp 138-143, 2006
- [22]. Moløkken K., Jørgensen M., “A review of surveys on software effort estimation” Proceedings of the 2003 International Symposium on Empirical Software Engineering pp 1325, 2003
- [23]. Moløkken-Østfold K., Jørgensen M., Taniklan S. S., Gallis H., Lein A.C., Hove S. E., “A survey on software estimation in the Norwegian industry” Proceeding-International Software Metrics Symposium, pp 208-219, 2004
- [24]. Moløkken-Østfold K., Haugen N. C., “Combining estimates with planning poker - An empirical study” Proceedings of the Australian Software Engineering Conference, ASWEC, pp 349–358, 2007
- [25]. Özkaya A., Ungan E., Demirörs O., “Common practices and problems in effort data collection in the software industry” Proceedings - joint conference of the 21st international workshop on software measurement, pp 308, 2011
- [26]. Parvez A. W. M. M., “Efficiency factor and risk factor based user case point test effort estimation model compatible with agile development”, Proceedings of International conference on Information Technology and Electrical Engineering, Yogyakarta, Indonesia, pp 113-118. 2013
- [27]. Power K., “Using Silent Grouping to Size User Stories”, 12th International Conference on Agile Processes in Software Engineering and Extreme Programming, XP 2011, LNBIP, 60-72. Madrid, Spain, Springer Verlag, 2011
- [28]. Santana C., “Using Function Points in Agile Projects”, Agile Processes in Software Engineering and Extreme Programming, Berlin, Springer-Verlag Berlin, pp 176-191, 2011
- [29]. Satapathy S. M., Panda, A., Rath S.K., “Story point approach based agile software effort estimation using various SVR kernels” Twenty Sixth International

- Conference on Software Engineering & Knowledge Engineering, pp 304-307, SEKE, 2014
- [30]. Shepperd M., Schofield, C., "Estimating software project effort using analogies" *IEEE Transactions on Software Engineering*, vol **23(11)**, pp 736-743, 1996
- [31]. Shinji K., Schigeo H., "Estimating Effort by Use Case Points: Method, Tool and Case Study", *Proceedings of the 10th International Symposium on Software Metrics (METRICS'04)* 1530-1435/04 IEEE, 2004
- [32]. Shepperd, M., Schofield, C., Kitchenham, B., "Effort Estimation Using Analogy" *Proceedings of the 18th International Conference on Software Engineering*, pp 170, 1995
- [33]. Tanveer B., Guzman L., Engel U.M., "Understanding and improving effort estimation in agile software development – an industrial case study", *International Conference on Software and System Processes*, ACM 2016
- [34]. Trendowicz A., Münch J., Jeffery R., "State of the practice in software effort estimation: a survey and literature review" *Software Engineering Techniques*, 4980 LNCS, pp 232-245, 2011
- [35]. Ugan E., Cizmeli N., Demirors O., "Comparison of functional size based estimation and story points, based on effort estimation effectiveness in scrum projects" *Software Engineering and Advanced Applications 4th Euromicro conference*, pp 77-80 IEEE 2014
- [36]. Usman M., Mendes E., Weidt F., Britto R., "Effort Estimation in Agile Software Development: A Systematic Literature Review" *Proceedings of the 10th international conference on predictive models in software engineering –promise 14* , ACM Press. pp 82-91, 2014
- [37]. Wen J., Li S., Lin Z., Hu Y., Huang C., "Systematic literature review of machine learning based software development effort estimation models" *Information and Software Technology*, vol **54(1)**, pp 41-59, 2012