Cost and Effort Estimation in Agile Software Development

Rashmi Popli
Department of Computer Engineering
YMCA University of Science & Technology
Faridabad, India
rashmimukhija@gmail.com

Naresh Chauhan
Department of Computer Engineering
YMCA University of Science & Technology
Faridabad, India

Abstract— **Projects that are over-budget, delivered late, and fall short of user's expectations have been a common problem area for software development efforts for years. Agile methods, which represent an emerging set of software development methodologies based on the concepts of adaptability and flexibility, are currently touted as a way to alleviate these reoccurring problems and pave the way for the future of development. The estimation in Agile Software Development methods depends on an expert opinion and historical data of project for estimation of cost, size, effort and duration. In absence of the historical data and experts the previous method like analogy and planning poker are not useful. This paper focuses on the research work in Agile Software development and estimation in Agile. It also focuses the problems in current Agile practices thereby proposed a method for accurate cost and effort estimation.**

Key Words — **Agile Software Development, User-Stories;Story-points;Estimation;Traditional Estimation methods,Cost,Effort per Person.**

I. INTRODUCTION

Agile software methodologies are widely used in a variety of industrial projects because of their volatile and flexible nature. The formation of the Agile Alliance in 2001 and the publication of the Agile Manifesto formally introduced agility to the field of software development. The Agile Manifesto presented an industry-led vision for a profound shift in software development. In Agile, software is developed in small iterations and the new changes are always welcomed. Due to the dynamic nature of Agile it becomes very difficult to estimate cost and time in Agile environment [1,10]. By literature survey it has been observed that current estimation methods which are used in agile are not efficient because they do not consider any mathematical formula for accurate calculation of effort and cost. In this paper to support flexible and changing nature of Agile, a mathematical estimation technique has been proposed that estimates the more accurate release date, cost, effort and duration for the project.

This paper is a step towards understanding the causes of inaccurate estimates in Agile software development and problems related to estimation in Agile. The projects will be successful if they are delivered on time with the efficient planning.

In section II the Agile life cycle is described, in section III causes of inaccurate estimates are discussed. In section IV various Agile estimation techniques are discussed, Section V describes related work in estimation field. Section VI describes the proposed estimation technique. Section VIII contains the conclusion and future work.

II. AGILE LIFE CYCLE

Agile methods stress values such as individuals and interactions over processes and tools; working software over comprehensive documentation; customer collaboration over contract negotiation; and responding to change over following a plan. The main methods that fall under the Agile umbrella include XP, Scrum, Crystal, DSDM, FDD and ASD. Agile methods break tasks into small increments with minimal planning, and don't directly involve long-term planning. Iterations are short time frames that typically last from one to four weeks [1].Each iteration is worked on by a team through a full software development cycle including planning, requirements analysis, design, coding, unit testing, and acceptance testing when a working product is demonstrated to stakeholders. This helps minimize overall risk, and lets the project adapt to changes quickly. Stakeholders produce documentation as required. Iteration may not add enough functionality to warrant a market release, but the goal is to have an available release (with minimal bugs) at the end of each iteration. Multiple iterations may be required to release a product or new features as shown in Figure 1.

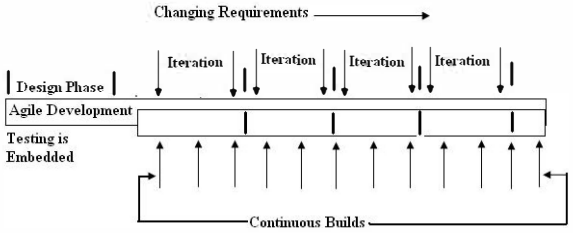

Fig 1: Agile Life Cycle

III. CAUSES OF INACCURATE ESTIMATES IN AGILE SOFTWARE DEVELOPMENT

An estimate is a prediction of how long a project will take. Estimation should be treated as an unbiased, analytical process and planning should be treated as a biased, goal-seeking process.

An effective software estimate provides the information needed to design a workable software development plan. A software estimation process that is integrated with the software development process can help projects establish realistic and credible plans to implement the project requirements and satisfy commitments. It also can support other management activities by providing accurate and timely planning information. Perhaps the most important element in the success or failure of a project are estimates of its scope, in terms of both the time and cost that will be required and the plans based on those estimates. Various causes of inaccurate estimates in Agile software development are discussed below. An effective software estimate provides the information needed to design a workable software development plan. A software estimation process that is integrated with the software development process can help projects establish realistic and credible plans to implement the project requirements and satisfy commitments. It also can support other management activities by providing accurate and timely planning information. Perhaps the most important element in the success or failure of a project are estimates of its scope, in terms of both the time and cost that will be required and the plans based on those estimates. Various causes of inaccurate estimates inAgile software development are discussed below.

A. Methodology

It refers to the estimation process adopted and includes the steps undertaken to produce the estimate and also the means of examining and reviewing the estimates relating to past projects Estimation inaccuracy can be caused from a lack of procedures and policies on how to deal with failures and avoid repeating mistakes by learning from past experiences.

B. Political Forces

Political forces at work within a project or company can often drive estimation inaccuracy. This is usually in the form of managerial pressure to stay within or meet the estimate[10].The estimation process can be impacted negatively by these pressures resulting in time or cost constraints .When estimates are produced simply in order to satisfy managers or customers it will inevitably lead to inaccuracy.

C. Communication between users

User Communication refers to the factors relating to the customers and their changing requirements throughout a system's life-cycle. This is usually the most prominent factor in causing project estimates to be inaccurate [13]. Improper communication between client and customer leads to problems.

D. Management control In Agile Enviornment

Management Control Problems caused by Management control include management reviews, and comparison between estimates and actual. Inaccuracy also occurs when management does not refer to the estimate when conducting performance reviews of estimators and other project personnel [12,13].

E. Uncertainity

It refers to the changing requirements of the customers; if customer changes his perspective about project then it leads to inaccuracy**.** So there is need to gather better knowledge whether by leveraging history, mathematical algorithms and/or project specific information to make better estimates[13].

Integrating agile techniques for knowledge capture in projects are tools for reducing uncertainty.

F. Self Knowledge

Two psychologists, Joseph Luft and Harry Ingham developed a construct to understand personal awareness. The tool named Johari's Windowiii divides personal awareness into four different categories, as represented by its four quadrants: open, hidden, blind and unknown. The lines dividing the four panes are like window shades, which can move as an interaction progresses. The concept is adaptable to teams. Team level blind spots complicate estimation, planning and ultimately performance. Techniques to improve a team's self knowledge include forming stable teams, fostering intimate communications.

IV. AGILE ESTIMATION METHODS(AEM)

Estimation is the process to estimate the cost, effort and the time taken for a software project. The process of estimation starts from the planning phase of the Software development life cycle (SDLC) and this is refined throughout all the phases of the SDLC. Agile methods are volatile in nature and can accept last minute changes. Some approaches of estimation in Agile are as below.

Table 1: Agile Estimation Method

S.No	Approaches of Agile Estimation		
	Name of approach	Working	Problem
	Learning-oriented Approach	This approach is based on the collective learning from previous estimation experience and knowledge of several managers drawn from the results of many specific projects.	This method is not used as it may provide very unrealistic estimates.specific projects.
	Expertise-based Approach	In this technique an expert compare the project with similar past projects based on personal memory.	In this technique an expert compare the project with similar past projects based on personal memory.
	Regression methods	This model is based on regression data and developing regression equations to make estimates.	It does not promote good software engineering practice.
	Bottom-Up	In this method each component of the software system is separately estimated and the results aggregated to produce an estimate for the overall system.	How the system is decomposed into different components

a. Approaches of Agile Estimation

V. RELATED WORK

Research work in Agile Estimation was done by Abrahamsson et al.[13], Cohen et al.[11], and Maurer et al.[3]. These three reports describe the state of the art and state of the practice in terms of characteristics of the various agile methods and lessons learned from applying such methods in industry.

Abrahamsson et al. [13] demonstrated how to collect metrics to measure productivity, quality and schedule estimation, cost and effort estimation for an Agile Software Development project using XP. The author provides evidence that agile methods are "effective and suitable for many situations and environments".

Williams et al. [11] investigated the usage of a subset of XP [3] practices at a group in IBM. The product developed at IBM using XP was found to have significantly better pre-release and post-release quality compared to an older release. The teams using XP reported an improvement in productivity, schedule, cost and effort estimation. In addition, customers were more satisfied with the product developed using XP because the teams delivered more than what the customers had originally asked for.

Maurer et al. [3] studied the development of a web based system by nine full time employees in a small company that used XP and observed substantial productivity gains compared to their pre-XP timeframe.

Heemstra surveyed 364 organizations and found that only 51 used models to estimate effort and that the model users made no better estimate than the non-model users [2]. Also, use of estimation models was no better than expert judgment.

Finnie and Wittig applied artificial neural networks (ANN) and case-based reasoning (CBR) to estimation of effort [4]. Using a data set from the Australian Software Metrics Association, ANN was able to estimate development effort within 25% of the actual effort in more than 75% of the projects

A survey of software development within JPL found that only 7% of estimators use algorithmic models as a primary approach of estimation [4].

VI. PROBLEMS IN EXISTING METHODS

In Agile, Releases are scheduled and tracked at the PBI level. Agile provides an estimation system that enables teams to estimate distant, uncommitted work with a low degree of precision. However, effort and risk estimation is not done by considering all the factors. Agile provides for **two levels** of estimation; one for work (Tasks) to be completed within the current iteration and one for more distant Product Backlog Items (PBIs). But these are not sufficient for calculating Release Date. The proposed work provides a way to quantify the size or efforts required in a release and also perform release risk tracking. Various Problems with Agile Estimation Scheme are as below:

A. Effort Estimation

Estimations are done in units of time while to estimate how much each member can spend for sprint related work is more important[5].The fact is that no one can sit in one place to complete the job without attending meetings, lunch breaks, unexpected breaks, checking emails and phone calls etc.

B. Release Date estimation

The Release planning is the activity to calculate the actual release date so that the final product is handed over into use for the customer. In Scrum Estimation technique a release plan is made but it doesn't consider various factors like velocity, cost-benefit ratio[7,8]

C. Release Risk tracking and estimation

Risk estimation is not done in Agile Estimation. For instance, if there is a risk of deviation of estimates is high then what should be done. It is necessary to track total risk values of the stories in a release which was not considered in previous Agile estimation techniques.

Today, almost no method can estimate software with a high degree of accuracy. In Agile environment, at the initial stage of a project, there is high uncertainty about various project attributes. The estimates produced at early stages are inaccurate, as the accuracy depends highly on the amount of reliable information available to the estimator. Agile estimation methods may lead to the errors in case of inexperienced Agile team. Therefore, there is strong need of analyzing the factors that affect the estimation of the Agile project.

VII. PROPOSED METHOD FOR ESTIMATION IN AGILE

Identify the total number of user-stories as per the requirements of the customer and find out the total story-points. Then determine the initial velocity of the project on the basis of which calculate the effort and cost of the project.

A. Proposed Algorithm for Agile Estimation

- Identify the total number of user-stories(US) and story points(SP) per user story and compute total Story Points (TSP)

  TSP=US*SP
- Compute the Velocity from first iteration as

  V = Story point completed in one iteration/ Story point in one user story.
- Compute the Decelerated Velocity by considering the dynamic changes in Agile environment

  DV= V*VF
- Compute the Estimated development time required for the project

  Estimated Development Time (EDT) = ESP/Velocity (in Days)
- Compute Total Estimated Effort (TEE)

  TEE=TSP*Effort per person
- Compute Total Estimated Cost (TEC)

  TEC=TSP*Cost per story-point

B. Proposed Activity diagram for Agile Estimation

The prioritization diagram in figure 2 explains the various steps involved in estimation in Agile environment like gathering of requirements, velocity calculation, effort calculation and cost calculation etc.

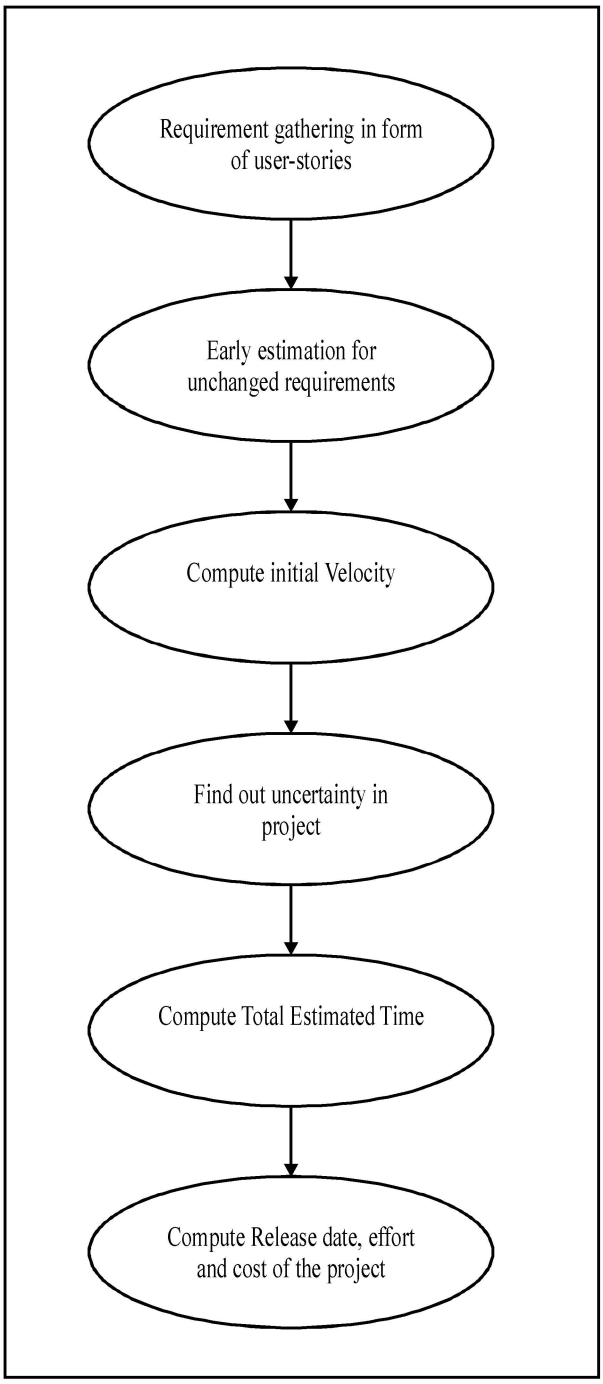

Fig 2: Proposed Estimation Technique

VIII. FEASIBILITY STUDY

In this section the feasibility of our algorithm is shown by a case study in which the total number of story-points is calculated. Then velocity per iteration is calculated. We have considered the user stories of enable quiz which is a lightweight technical quizzing solution; for companies that hire engineers. It will allows them to better screen job candidates and assess their internal talent for skills development. The table II shows the user-stories and total number of story-points in each user-

Table II: User Stories

S.No.	User Story		
	User Story	**Story Points**	**Priority**
2	As a manager, I can make sure I'm subscribed to all the necessary topics for my skills audit.	40	9
3	As a manager, I can add additional technical topics to my quizzes.	40	10
4	As a manager, I want to create a custom quiz bank	30	5
5	As a manager, I want to create a quiz so I can use it with my staff.	40	6
6	As a manager, I want to create a list of students from an Excel file so I can invite them to take the quiz.	30	3
7	As a manager, I want to create a list of	30	2
8	As a manager, I want to invite a set of	20	4
9	As a manager, I want to see which students have completed the quiz.	10	7
10	As a manager, I want to see how the students scored on the test so I can put	10	8

A. Inputs to Proposed Agile Estimation Technique

Inputs:

No. of user stories (US) = 10
Total Story-points=300
No. of story point completed by the team in one iteration = 50
Initial Velocity=300/50=6 SP/day
No. of days in one iteration (iteration time) = 10 days
No. of working days per month = 22 days
No. of working hours per day = 8 hrs.
Cost per story-point=50 Rs.

B. Results of case study

The table III shows the results of the proposed algorithm. The values of total estimated time, total estimated effort andtotal cost are calculated by using formulas of proposed algorithm.

Table III: Outputs

S.No.	Hypothetical values of factors	
	Unadjusted Value	**Label**
1	Total Story-points	300SP
2	Velocity	6 SP/Day
3	Estimated Development Time	50 Working Days
4	Total Estimated Effort	50 Person-days
5	Total Estimated Cost	1500$

IX.CONCLUSION AND FUTURE WORK

Research on estimation has been conducted for decades with immense quantities of models and tools produced. The Agile estimation methods are not suitable for organizations because these provide little details for justifying estimates and the estimates cannot be produced early in the life-cycle. With the proposed method in research work the accurate velocity of the project can be calculated. By this method Cost, effort and duration of small and medium size project can be calculated efficiently. In the future the other factors which affect the estimation most can be added; thereby having makes the estimation more correct and efficient. The benefit of the proposed estimation technique is that it reduces the risk of falling project in chaos by providing realistic figures of estimation.

REFERENCES

[1] Cockburn, "Agile Software Development," Pearson Education, Asia Low Price Edition, 2007.

[2] F. J. Heemstra, "Software cost estimation", Information and Software Technology, vol. 34, no.10, 1992, pp. 627-639.

[3] F. Maurer and S. Martel, "Extreme Programming: Rapid Development for Web-Based Applications", IEEE Internet Computing, 6(1), pp. 86-91, Jan/Feb 2002.

[4] G. R. Finnie, G. E. Wittig, "AI tools for software development effort estimation", Software Engineering and Education and Practice Conference, IEEE Computer Society Press, pp. 346-353, 1996.

[5] J. Hihn and H. Habib-Agahi, "Cost estimation of software intensive projects: a survey of current practices", International Conference on Software Engineering, 1991, pp. 276-287.

[6] Jorgensen, M. "Top-Down and Bottom-Up Expert Estimation of Software Development Effort", Information and Software Technology, 46, 3-16.,2003.

[7] Jorgensen, M., Indahil, U. & Sjoberg, D. (2003) "Software Effort Estimation by Analogy and Regression Toward the Mean". Journal of Tradional Estimation Methods and Software, 68, 253-262.

[8] Keung, J., Jeffery, R. & Kitchenem, B. (2004) The Challenge of Introducing a New Software Cost Estimation Technology into a Small Software Organisation", Proceedings of the 2004 Australian Software Engineering Conference. Australia.

[9] K. McDaid, D. Greer, F. Keenan, P. Prior, P. Taylor, G. Coleman, "Managing Uncertainty in Agile Release Planning", 18th Int. Conference on Software Engineering and Knowledge Engineering (SEKE'06), pp. 138-143, 2006.

[10] Lederer, A. L. & Prasad, "The Validation of a Political Model of Information Tradional Estimation Method Development Cost Estimating", Proceedings of the 1991 conference on SIGCPR,1991

[11] L. Layman, L. Williams, and L. Cunningham, "Motivations and Measurements in an Agile Case Study", Proceedings of ACM SIGSOFT Foundation in Software Engineering Workshop Quantitative Techniques for Software Agile Processes (QTE-SWAP), Newport Beach, CA, 2004.

[12] L. Williams, W. Krebs, L. Layman, A. Antón, and P. Abrahamsson, "Toward a Framework for Evaluating Extreme Programming", Proceedings of Empirical Assessment in Software Eng. (EASE) 2004, Edinburgh, Scot., pp. 11-20, 2004.

[13] P. Abrahamsson, 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.