© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 07 Issue: 02 | 29 | April - June 2016
Paper is available at www.jrps.in | Email : info@jrps.in

Using human opinion for parameter tuning of Software cost estimation in agile software methodology of Prototyping an algorithm

Saurabh[1], Ms. Swita Bishnoi[2]
[1] M. tech. Research Scholar, MDU University, Rohtak, Haryana, India
[2] MDU University, Rohtak, Haryana, India

ABSTRACT

Agile methodologies provide a structure for highly collaborative software development. Rather than adhering to traditionally long periods of upfront requirements gathering and design before software production, agile teams elicit feedback early on in the process, and deal with the complexities of software development by practicing rapid iterative development from project inception. A major cause of failure of many software projects is the lack of accurate and early cost estimation. Barry Boehm proposed Constructive Cost Model also known as, COCOMO Model which used basic regression formula with parameters derived from historical project data and characteristics of the current project for estimating the cost of software. This model is a high risk due to low accuracy and lack of reliability. This is where the need of optimization comes in. Various approaches like Genetic Algorithm have already been applied for tuning of the parameters of COCOMO in order to increase it's accuracy and reliability. Regardless, that humans are the most intelligent social animals, an approach based on crowd dynamics, opinion dynamics, language dynamics is seldom used for optimization. Interaction between humans gives rise to different kind of opinions in a society. The process of opinion formation evolves from collective intelligence emerging from integrative forces of social influence with disintegrative effects of individualization. Opinion dynamics leads to efficient decision making and so, we propose an approach based on human opinion dynamics for effective and accurate software cost estimation.

I. INTRODUCTION

Agile methodologies provide a structure for highly collaborative software development. Developed in the 1990's, the adaptive methodologies were formulated by and for developers in reaction to perceived deficiencies in conventional 'top down' or 'plan driven' methods. Commonly associated with 'lean' engineering (e.g. Poppendieck Poppendieck, 2003), agile software development closely follows the flow of business value, with a focus on activities that directly contribute to the project end goal of quality software.

Accurate software cost estimation has a great significance for both software development team and customers involved in the project [1][2]. Estimating the effort, time plan and staffing levels required to develop a software project is referred as software cost estimation. Standish group reported, in U.S 53% of software projects ran over 189% of the original estimate due to lack of early estimation. But, estimation is definitely not enough, the key lies within accurate estimation. The Constructive Cost Model (COCOMO) first used in 1981, laid a more calculative foundation towards cost estimation but at last suffered lack of accuracy and reliability. Several approaches are already endorsed inspired from agglomeration is physical and behavioral space like Ant Colony Optimization, Particle Swarm Optimization etc for effective estimations. Collective patterns emerging from, as simple as bacterial colonies to as complex as humans; has always been an inspiration to solve complex optimization problems [2]. As, human opinion is one of those parameters which help humans to make effective and smart decisions throughout their life so it becomes evident, that it also provides a stable base for solving such practical optimization problem like software cost estimation. Opinion dynamics is a complex and difficult approach because of its evolutionary nature and impacts of social influence and individualization. Decision making on the basis of human opinion and social structure is referred as opinion consensus. The major challenges faced by opinion dynamics, is the modeling of opinion and the impact of the social structure on each opinion. However, opinion dynamics once modeled and deployed successfully can prove out to be revolutionary for solving complex mathematical optimization problems [3][4][5].

There is a growing body of practitioner evidence to suggest that participants in agile team environments find the experience particularly rewarding; more so than most other software development environments (e.g. Beck, 2004; Jeffries, 2004). A survey by Cockburn and Highsmith (2001, p. 3), for example, found that agile methodologies were rated higher than other methodologies in terms of morale, while Goebel (2002, p.12) found in an informal poll that people who worked in agile teams were reluctant to go back to old methods of development. Although the 'hype' surrounding agile methodologies (Stephens & Rosenberg, 2003) is likely a strong contributor to the enthusiasm observed, there seems to be some basis for the common association of agile practices and teams with the idea of 'project chemistry' (Nicolini, 2001) or positive 'team climate' (Anderson & West, 1994) that can contribute to a high performance (Goebel, 2002).

Accuracy in estimations allows the company to develop appropriate time plan and estimate the most feasible budget and the effort required to build the project. COCOMO Model proposed by Barry Boehm is widely used for estimating the effort and development time using basic regression formula with parameters derived from historical project data and characteristics of the current project for estimating the cost of software. This model is a high risk due to low accuracy and lack of reliability [6][7]. This motivated the software professionals to keep a step forward towards not only estimation but early and accurate estimation and from then on approaches have been proposed but accurate software cost estimation is still a challenge in IT industry [8].

II. Importance of software cost estimation

Software cost estimating has been growing in importance till today. When the computer era began, very few computers were in use and most of the applications were small. As time moved on, computers became widespread the applications is use grew in number, size and importance and along with this costs to develop software grew as well. As a result of the growth, the consequences of errors in software cost estimation became more vulnerable [9]. Even today, a lot of cost estimates of software projects are not very accurate infact, most of them are too low. This is not a surprising that we have to face various difficulties when estimating software costs. There are few costs which are not at all hard to determine and can be estimated in advance and are even fixed sometimes as the hardware or software requirement purchase or the license costs [10]. But also there exists cost which are not easy to be estimated. The by far greatest amount of the total costs of a project arises from the salaries of the personnel. The costs for the human workers are highly correlated to the effort we need to perform the project. Therefore, it becomes necessary to get an accurate enough estimate of the total effort in order to make more precise estimate of the costs. Size and complexity are the basis of estimating effort for the project and both of these are derived from the specification. Because the requirements of the software are likely to change at any given instant, we have to consider it into account too when estimating the effort. The difference in productivity of software developers is a major issue to solve during the estimation process. An experienced developer will have far more productivity than a beginner. But, because each project is unique and uses it's own tools and languages, the experience level of the development team is hard to judge [11]. Another problem appears when humans are estimating. Sometimes, unknowingly we tend to underestimate immaterial things like software which later becomes a problem in the later development phase of the software. Today's world would not be the same if there was no software.

III. LITERATURE SURVEY

The development teams also have on-site customers with substantial domain knowledge to help them better understand the requirements (Abrahamsson, Solo, Ronkainen, & Warsta, 2002). Multiple short development cycles also enable teams to accommodate request for change and provide the opportunity to discover emerging requirements (Highsmith, 2002). The agile approach promotes micro-project plans to help determine more accurate scheduling delivery commitments (Smits, 2006). M Lindvall, V Basili, B Boehm, P Costa, (2002), summarize the working definition of agile methodologies as a group of software development processes that must be iterative (take several cycles to complete), incremental (not deliver the entire product at once), self-organizing (teams determine the best way to handle work), and emergent (processes, principles, and work structures are recognized during the project rather than predetermined). In the paper by (Abrahamsson, Warsta, Siponen & Ronkainen, 2003), in general, characterized agile software development by the following attributes: incremental, cooperative, straightforward, and adaptive. Boehm, B., & Turner, R. (2005), generalize agile methods are lightweight processes that employ short iterative cycles, actively involve users to establish, prioritize, and verify requirements, and rely on a team‘s tacit knowledge as opposed to documentation.

Software cost estimation by Samuel Lee, Lance Titchkosky, Seth Bowen defines software cost estimation as the approximate judgement of the costs for a project. It concludes that it is not very uncommon for the software projects to exceed time and budget which isn’t healthy for software development. This problem is due to the fact that software development is a complex process because of the number of factors involved, including the human factor, and the complexity of the product that is developed. Furthermore, the industry is highly competitive. Software cost estimation is an important part of the development process that requires improvement in adoption and diligence. Data should be gathered throughout the entire life cycle so that the accuracy of the estimates can be improved. Although expert-based estimation is one of the most common methods of estimation because of its lightweight nature, the method suffers from being highly dependent upon competent estimators. So, Any model which is incorporated should be calibrated to the development environment because of difference in all development environments [12][13].

Software Project Cost Estimation: Issues, Problems and Possible Solutions by Adanma C. Eberendu discuss the various issues, problems and possible solutions for software cost estimstion. It states that each of the classical software cost estimation techniques has advantages and disadvantages and so, the best approach is to combine two or more techniques to estimate project cost, thus, hybrid estimation technique is recommended. The lack of accurate and reliable estimation techniques combined with financial, technical, organizational, and social risks of software projects, require a frequent estimation during the development of an application and the use of more than one estimation technique.

IV. Human Opinion Dynamics Algorithm

The study of opinion dynamics and formations is an important area of social physics. Human opinion dynamics algorithm is complex to implement but equally effective too. The four pillars of this algorithm are- Social Structure, Opinion Space, Social Influence and Updating rule.

- **Social Structure:** Social structure lies between individuals or group of individuals. It portrays the way of interaction of individuals from other individuals in their neighborhood. It is a network which ties a number individuals within one structure and reflects a stable pattern of relationship between the entities. A social graph is formed within a social structure in which the individuals forms the nodes of the graph and the neighboring set of individuals from which each individual interact is defined with the edges of the graph.

- **Opinion Space:**

The second pillar of the algorithm is the opinion space. Each individual within a social graph has its own opinion space. Opinion space can be discrete or continuous, where discrete opinions can be as {0,1) continuous opinions can take any real value. Each individual *i* is associated with an opinion vector o(t) at any given time which allows us to search in a multidimensional space.

- **Social Influence:**

Social Influence plays a huge role in opinion dynamics. Decision making process is influenced by one’s own considerations as well as social beliefs in the structure. A disintegrative tendency of individualization goes hand in hand with integrative tendencies of socialization. However, for the sake of simplicity, only local dynamics is taken in account for representing social influence. Therefore, Social influence is formulated using the Social Rank and the distance between the two nodes in the social graph. Social rank is determined from the fitness values which are the output from the objective function that is to be minimized, lesser the fitness value higher will be the social ranking. Distance is actually the Euclidean Distance between the two nodes. The social influence $w_{ij}(t)$ of individual j on individual i is given by equation:

$$w_{ij}(t) = \frac{SR_j(t)}{d_{ij}(t)} \qquad (1)$$

where d_{ij} this the Euclidean distance among the two nodes.

V. Performance Analysis

The performance analysis of the two models is done by calculating the Mean Absolute Relative Error. MARE is calculated as:

a. Mean Absolute relative error

$$\%MARE = \sum \left[\frac{abs(mes.effort - est.effort)}{(mes.effort)}\right] /n$$

where Meas. Effort is the Effort in the NASA datasets, Calc. Effort is the effort calculated from the given model and n is the number of projects.

Using the above formulae the Average error % calculated is given below:

i) MARE% of COCOMO= 39.75%

ii) MARE% of HOD optimized COCOMO= 18.59%

The obtained results, clearly indicates that when the parameters of COCOMO are tuned by HOD the MARE% of COCOMO reduces almost by half. When the mean absolute relative error reduces the accuracy of the model significantly increases showing that HOD optimized COCOMO show far better results than the standard COCOMO model.

The graph of mean square error, convergence and 2-D movement of opinions are given below:

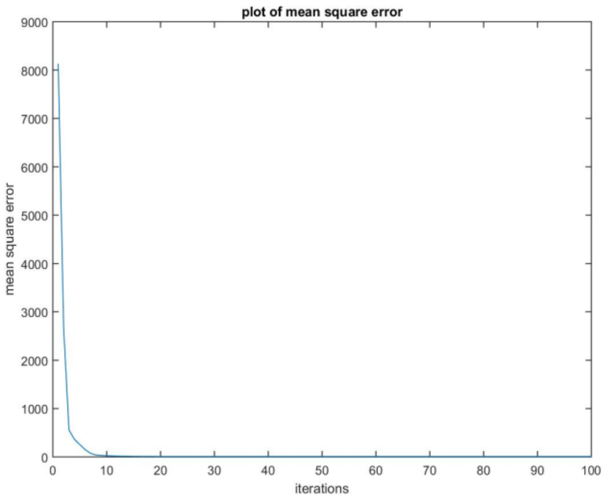

Figure 1: Graph between mean suare Error and iterations

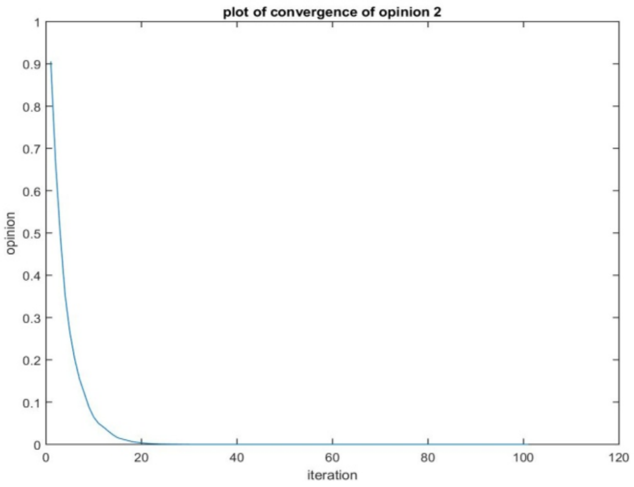

Figure 2: Graph between different opinions and iterations

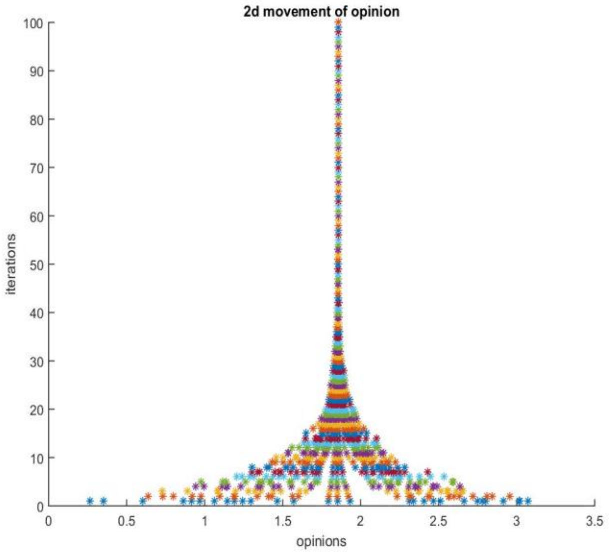

Figure 3: Graph shows 2D Movement of Opinions

VI. CONCLUSION

Here a new approach has been proposed to estimate the software cost for projects using human opinion dynamics. Human opinion dynamics is an efficient algorithm to estimate most of the optimization problems but in some cases it undergoes large computational complexity. Availability of good historical data used by COCOMO coupled with an efficient evolutionary algorithm like human opinion dynamics algorithm generates better results. Modified version of the famous COCOMO model was provided to consider the effect of methodology in effort estimation. According to the results obtained, the proposed model shows good estimation capabilities with a much lower MARE percentage as compared to the COCOMO model. This model clearly exhibits much higher accuracy when the results obtained The developed model is able to provide good estimation capabilities.

REFERENCE

1. Abrahamsson, P., Warsta, J., Siponen, M.T. and Ronkainen, J. New Directions on Agile Methods: A Comparative Analysis, Proceedings of the International Conference on Software Engineering, 2003 (Oregon, USA).
2. L. Williams and A. Cockburn, —Agile Software Development: It‘s about Feedback and Change, IEEE Computer, June 2003, pp. 39-43
3. Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, *48*(5), 72-78.
4. Latané, B., Nowak, A. & Szamrej, J. From private attitude to public opinion: A dynamic theory of social impact. Psychol. Rev. 97, 362–376 (1990).

5. Manifesto for Agile software development; http://agilealliance.com
6. Boehm, B.W. Software Engineering Economics, 1981 (Prentice Hall, Upper Saddle River, New Jersey)
7. Sultan Aljahdali and Alaa Sheta ,Taif University," Evolving Software Effort Estimation Models Using Multigene Symbolic Regression Genetic Programming
8. G.S. Hornby and J.B. Pollack. Creating high-level components with a generative representation for body-brain evolution. Artificial Life, 8(3):223–246, 2002.
9. Ferreira, C., 2001. Gene Expression Programming: A New Adaptive Algorithm for Solving Problems. Complex Systems, Vol. 13, issue 2: 87-129.
10. Novel Meta-Heuristic Algorithmic Approach for Software Cost Estimation by Ruchi Puri and Iqbaldeep kaur
11. Brajesh Kumar Singh, A.K Gupta," Software Effort Estimation by Genetic Algorithm Tuned Parameters of Modified Constructive Cost Model for NASA Software Projects".
12. Vahid Khatibi Bardsiri, Dayang Norhayati Abang Jawawi,Siti Zaiton Mohd Hashim, Elham Khatibi," A PSO-based model to increase the accuracy of software development effort estimation".
13. Zainudin Zukhri and Irving Vitra Paputungan,"A Hybrid Optimization Algorithm Based On Genetic Algorithm And Ant Colony Optimization"
14. Dorigo, M. & Di Caro, G. Ant colony optimization: a new meta-heuristic. in Evol. Comput. 999. CEC. 99. Proc. 1999. Congr. 2, (IEEE, 1999).
15. K. Peterson, A Comparison of Issues and Advantages in Agile and Incremental Development between State of the Art and an Industrial Case. Journal of System and Software. 2009