Communication as an essential factor for effort estimation in GSD

Manoj Ray D
College of Engineering Karunagappally
Kollam Dist. Kerala State, India
e-mail: manojray@yahoo.com

Abstract— **Global Software Development, also known as Distributed Software Development, is originally an outsourcing technique in which the development teams hailing from diverse cultural backgrounds and located at distant remote locations participate in the software development efforts and provide their services through a tightly coupled computer networks like Internet. Accurate software cost and effort estimate are critical to both developers and customers. Communication difficulties of the Global Software Development projects adversely influence project cost which affects the cost and effort estimation of the software development. This paper is a survey on the effort estimation methods and the communication problems in the GSD projects. As a result suggests that the communication factor also consider for the effort estimation in the Global Software Development project, which results in the effective estimation.**

Keywords-Global Software Development; effort estimation; communication

Globalization of innovation and markets has dramatically impacted software development. Today, more software projects are run in geographically distributed environments, and global software development is becoming a norm in the software industry [1]. Global Software Development (GSD), also known as Distributed Software Development, is originally an outsourcing technique in which the development teams hailing from diverse cultural backgrounds and located at distant remote locations participate in the software development efforts and provide their services through a tightly coupled computer networks like Internet [2]. The cost effectiveness as outsourcing organizations usually outsource their contracts to the country where the labor cost is comparatively low is the reason for selecting the technique of outsourcing . The need to globalize software development is to save time, cost and resources. Thus GSD has become more important, in both industry and research, which is due to the benefits that a company distributing its work globally can achieve: global resource pools, attractive cost structures, the possibility of developing around the clock, and presence on local markets[2].

As the importance and the complexities of global software development's growing , robust models, methods, processes, and tools are required to effectively and efficiently organize and execute global software work. Even though significant advances have taken place in this regard, much still needs to be done to assure predictability, efficiency, effectiveness, and consistency in global software development [1].

Effort estimation is a process that estimates necessary effort (cost, time, etc). It is highly important to make accurate/reliable estimates for the project at the beginning to support project planning, and control the project within the budget and schedule. The paper [3] summarizes the estimation knowledge through a review of surveys on software effort estimation. To call a project successful, it has to be on time, in budget and fulfilling the customer's requirements effectively [4]. One of the major challenges of the GSD projects is the communication and coordination problem. The effective effort estimation of GSD projects should also consider the communication as a major factor as the it adversely affect the success of the projects..

II. Effort Estimation Approaches

There are many researches that explain the effort estimation in collocated software development and in global software development.

Accurate software cost estimates are critical to both developers and customers. Researches show many machine learning methods can also be applied for the software effort estimation. The [5] describes two methods of machine learning, which used to build estimators of software development effort from historical data. The method in [6] describes a machine learning method which gives the estimation of effort together with a confidence level for it. The method employs the robust confidence intervals, which do not depend on the form of probability distribution of errors in the training set. The work in [7] describes a neural network system as a soft computing approach to model the effort estimation of the software systems. The work in [8] is an application of machine learning tools and variable section tools to solve the problem of estimating the execution effort of functional tests

The number of studies show that 40 percent of offshore software development projects are flops due to lack of standardized and practicable methods of risk management due to inherent complexities [9]. A characteristic that successful offshore projects had in common was the availability of informal mutual adjustment, which means facilitating the informal communication between the team members in the right way. A major characteristic that unsuccessful projects had in common was improper planning, which has a large influence on the team results in an offshore custom software development project [10] [11]. Soft Coordination and communication become more difficult as the software components are sourced from different places, thus affecting project organization, project control, and product quality. New processes and tools are consequently necessary [12]. Software development effort estimation plays a very important role to achieve this goal of reducing costs. Once the effort required to develop software is a key component of the cost of development, the correct application of effort estimation techniques is crucial for the success of these projects [13]. The risk occurrences, cost analysis and effort estimation in GSD is much more complex than in collocated development [14] [15]. Researches shows that wrong estimation of effort can lead to a project failure. To overcome these failures, the developing tools, techniques and methods, like COCOMO II, SLIM, CoBRA etc have been developed to facilitate the software development industry [16].

The Constructive Cost Model (COCOMO) was introduced in 1981 by Barry Boehm [17]. The enhanced version, COCOMO II, was presented in 1985 and has been further adapted since then. Many researches shows COCOMO II was the most used method for estimating cost of software projects. The study in [18] is based on COCOMO II which can be used to estimate the effort to globally distributed projects than the traditional methods. It will also help predict the outcome of collaborative project by reducing overall risk. It is accepted internationally and in organizations of all sizes. In [19] the additional cost drivers of distributed development are structured and examine the significance of each of these factors as a contributor to the overall cost of a software development. The COCOMO II can be tailored to account for these additional complexities.

Developing the algorithmic cost estimation model requires extensive past project data, which are not always available. To overcome disadvantages of the models, the [20] presented a method for cost estimation that combines aspects of algorithmic and experimental approaches referred as Cost Estimation, Benchmarking and Risk Assessment (CoBRA). The cost estimation model should consider characteristics of distributed development and provide support for effort prediction in GSD projects. The [21] presents an enhanced process for developing CoBRA cost estimation model by systematically including iterative analysis and feedback cycles. The [22] shows a creation of cost estimation model based on CoBRA approach specifically for developing individual cost estimation and project data, i.e. , development of an effort model for distributed and GSD projects. The approach combined expert-driven and data-driven cost estimation method that can be used in organizations where only little data is available. The model estimated the overhead in distributed development projects based on a set of influencing factors, their causal relationships, and a quantification of their impact on effort overhead. The article [23] gives a comparison of software cost estimation methods.

A model using Monte Carlo simulation techniques was developed to estimate software development project effort estimation risks. This model has been implemented in a mid-sized software services company, and is expected to benefit other software companies through more useful ‘confidence’ based effort estimations for software development projects [24]. Monte Carlo simulations are based on detecting the probability distribution functions of the independent input variables, and in simulating mathematically thousands of possible scenarios to derive PDF of the dependent output variable

There are many ways of categorizing estimation approaches, see for example. The top level categories are the following [25]:

- Expert estimation: The quantification step, i.e., the step where the estimate is produced based on judgmental processes.
- Formal estimation model: The quantification step is based on mechanical processes, e.g., the use of a formula derived from historical data.
- Combination-based estimation: The quantification step is based on a judgmental or mechanical combination of estimates from different sources.

A. Factors of Effort Estimation Methods

The effort estimation approaches comprises the finding the factors such as:

- Estimating the size of the development product.
- Estimate the effort in person- hours
- Estimate the schedule in calendar -months
- Estimate the project cost in dollars
- Measuring the size/complexity of the software program is the most elusive aspect in the estimation process. There are different methodologies for arriving at and expressing the size/complexity of the Software Program such as function points, lines of code, feature points, benchmarking.

But any of these approaches considered the communication factor for the effort estimation. The Table 1 shows the examples of estimation approaches within each category[25].

TABLE I. EXAMPLES OF EFFORT ESTIMATION APPROACHES TABLE

Estimation approach	Category	Examples of support of implementation of estimation approach
Analogy-based estimation	Formal estimation model	ANGEL, Weighted Micro Function Points
WBS-based (bottom up) estimation	Expert estimation	Project management software, company specific activity templates
Parametric models	Formal estimation model	COCOMO, SLIM, SEER-SEM
Size-based estimation models[11]	Formal estimation model	Function Point Analysis,[12] Use Case Analysis, SSU (Software Size Unit), Story points-based estimation in Agile software development
Group estimation	Expert estimation	Planning poker, Wideband Delphi
Mechanical combination	Combination-based estimation	Average of an analogy-based and a Work breakdown structure-based effort estimate
Judgmental combination	Combination-based estimation	Expert judgment based on estimates from a parametric model and group estimation

Expert judgment is a commonly employed method for estimating the effort required to complete software projects [26]. An elaborate method of reducing problems in groups related to company politics is to employ the Delphi technique ,which is often recommended in management papers. The Delphi technique does not involve face-to-face discussion, but anonymous expert interaction through several iterations, supervised by a moderator until an agreed-on majority position is attained. A modification of this technique, which includes more estimation group interaction, was developed by Barry Boehm and his colleagues, and labeled the Wideband Delphi technique . This technique is a hybrid of unstructured groups and the traditional Delphi method. As in the Delphi technique, there is a moderator (labeled coordinator), that supervises the process and collects estimates. In this approach, however, the experts meet for group discussions both prior to, and during the estimation iterations.

III. Communication IN GSD

Communication is a mediating factor between coordination and control. Distributed Software Development (DSD) allows team members to be located in various remote sites during the software life cycle, thus making up a network of distant sub- teams. Interaction between members requires the use of technology to facilitate communication and coordination. [14]. Individual team members are required to be able to communicate and work with people who they do not know and whose cultures they may not understand. In [27] found that a very high proportion of engineers are directly involved with GSD. In addition, more than 50% of the respondents regularly collaborate with people more than three time zones away. Engineers also report that communication difficulties around coordination are the most critical, yet difficult to solve issues with GSD. In distributed teams the definition of standard communication paths is critical. The communication paths in the distributed software development are well defined in [28]. In these communication paths there are several central directors, primarily the Program and Project Managers, who ensure that the correct team members are involved in or apprised of any critical communications.

The paper [29] presents a tool that provides synchronous, explicit and formal communication, for a Distributed Software Development Environment. This tool, named VIMEE (Virtual Distributed Meeting), allows relate to CSCW (Computer Supported Cooperative Work) and DSD areas; it defines a common workspace to manage the virtual meeting; it supports the project management and the decision making in group.

One of the main challenges in distributed development is to establish effective communication and coordination mechanisms among the distributed teams to exploit their full potential. The GSD practices is to ensure effective communication among the team members as in most of the cases GSD teams use electronic communication media for correspondence [30] . Large time zone differences, cultural diversities, trust among customers and developers language diversities are the fundamental factors that can cause failure of a project. The factors identified in the literature study as having an impact on communication problems were physical distance, language differences, cultural differences, differences in company culture, and common experiences of working together, infrastructure distance, time shift, and process maturity [31].

Besides differences in language and culture, GSD teams suffer from a lack of informal communication, resulting in low levels of trust and awareness of work and progress at remote sites. In GSD projects, managing cross-cultural issues is important. One class of software development activities directly affected by challenges in communication is requirements related activities, such as requirements definition and negotiation, design, and project management. [32]

When two or more people are separated geographically then the interaction between them is limited. Global Software Development provides a better concept in adopting software development in geographical context. GSD concept tells us about the process, the artifact, and the project management. However, its shortage of communication plans. The lack of communication problem is unavoidable since every team in every world has their own way. In developing country, the lack of good Internet connection makes the mobile phone, or SMS is preferable as a main communication tool. On the other side, a country that has enough broadband inter nets prefers communication tool like a video conferencing, VOIP, or even unified communication platform. The divergence of communication tools that used in certain country will ascend a problem in communication collaboration. For example, when developed country that has sufficient bandwidth collaborate with developing country who has very limited bandwidth [33].

The cost rate and cost overhead on the development costs having impact on productivity. The importance of cost rate depends on whether the work was being distributed globally or not. Cost overhead was described to the practitioners as costs for traveling and for local overhead structures at the sites, such as additional taxes [31]. The solution identified in [30] for the communication problem include using ontologies as communication facilitators, modularization of work, study of the cognitive nature of people and the characteristics of their environment and training on cultural norms.

IV. Observations

The goals identified in the literature for the GSD projects are cost, time utility, resource utilization, proximity to customer and IP protection. The problems in communication, coordination and control which caused due to the impediments in communication adversely influencing factor of the goals GSD project [30]. The cost of the project is increased by the physical distance, time shift which are factors identified as having the impact on communication problem. The cost overhead due to increased communication and decreased productivity also be considered for the GSD projects. The face to face meetings due to the other communication inserts the traveling expenses for the GSD projects causes the increased overhead in the development cost. The project cost, the factor in the effort estimation of a GSD project, is used by the effort estimation approaches. Thus from the researches it is clear that the factors of communication problems adversely affect the effort estimation of the GSD projects. For all these reasons the communication factor should be also be considered in the quantitative analysis of the effort estimation methods. As improving the quality of effort estimation is one of the great challenges for software project management the communication should also be considered as a factor for cost and effort estimation.

V. Conclusion

Coordination and communication become more difficult as the software components are sourced from different places, thus affecting project organization, project control, and product quality. One of the main challenges of GSD practices is to ensure effective communication among the team members. This paper is a survey on the effort estimation methods and the communication problems in the GSD projects. The effort estimation is the GSD projects should consider the communication as a factor since communication problem having a major impact on the project cost. The finding of metric that estimates the effort of GSD`s by measuring the communication factors can be done as future works.

.

References

[1] Daniela Daman, Deependra Moitra, “ Global Software Development How far we have come?” IEEE Software; Sept 2006. 740-745

[2] Samiul Haq,Mushtaq Raza, Asraf Zia, et.al., “ Issues in Global Software Development : A Critical Review.” IJ. Software Engineering & Applications; 20i,4590-595.

[3] Kjetil Molokken, Magne Jorgenson,” A Review of Surveys on Software Effort Estimation.”, Proceedings of the 2003 International Symposium on Empirical Software Engineering (ISESE’03): IEEE, 2003R.

[4] Abid Ai Khan, Zaka Allah Mohammmed, “ Exploring the Accuracy of Existing Effort Estimation Methods for Distributed Software Projects” Master Thesis , Software Engineering; MSE-209-09, June 2009.

[5] Krishnamoorthy Sreenivasan, Douglas Fisher, “ Machine earning Approaches to Estimating Software Development Effort.”, IEEE Transaction on Software Engineering,Vol 21, No. 2. Feb 1995: 126-137.

[6] Petronio . Braga, Adrianao L. I. Oliviera,” Software Effort Estimation using Machine Learning Techniques with Robust Confidence Intervals. “, Seventh International Conference on Hybrid Intelligence Systems, IEEE , 2007: 352- 357.

[7] Jaswinder Kaur, Satwinder Singh, Karanjeet Singh Kahlon, “ Comparative Analysis of Software Effort Estimation Model “,World Academy of Science, Engineering and Technology 46 2008: 485-487.

[8] Daniel G. e Silva, Mario Jino, Bruno T. de Abreu. “Machine learning methods and asymmetric cost function to estimate execution effort of software testing”, Third International Conference on Software Testing, Verification and Validation; IEEE, 2010: 275- 284.

[9] Stefanie Betz, Juho Mäkiö, Rolf Stephan, “Offshoring of Software Development – Methods and Tools for Risk Management.” International Conference on Global Software Engineering, IEEE , 2007:

[10] Fabriek, Matthias, Brand, Brinkkemper, “Reasons For Success And Failure In Offshore Software Development Projects”,

[11] Siffat Ullah Khan, Mahmood Niazi, Rashid Ahmad, “Critical Success Factors for Offshore Software Development Outsourcing Vend, ors: A Systematic Literature Review”, Fourth IEEE International Conference on Global Software Engineering,2009: 307- 316.

[12] Miguel Jim´enez, Mario Piattini, Aurora Vizca´ıno, “Challenges and Improvements in Distributed Software Development: A Systematic Review”, Hindawi Publishing Corporation Advances in Software Engineering, Volume 2009, Article ID 710971, 14 pages.

[13] Carlos Eduardo Lima Peixoto, Jorge Luis Nicolas Audy, Rafael Prikladnicki, “Effort Estimation in Global Software Development Projects Preliminary Results from a Survey. “International Conference on Global Software Engineering, IEEE, 2010: 123- 127.

[14] Quadeem Khan,Shahbaz Gayyur,”Software Risks and Mitigation in Global Software Development.”Journal of Theoretical and Applied Information Technology: 2005-2010.

[15] Stephen G. MacDonell, Martin J. Shepperd, “Comparing Local and Global Software Effort Estimation Models- Reflections on a Systematic Review.”, First International Symposium on Empirical Software Engineering and Measurement, IEEE , 2007: 401- 409.

[16] Mohammad Muhairat, Saleh Aldaajeh, Rafa E. Al-Qutaish, “The Impact of Global Software Development Factors on Effort Estimation Methods”, European Journal of Scientific Research, ISSN 1450-216X Vol.46 No.2 (2010), pp.221-232.

[17] Boehm, B. *Software Engineering Economics*. Prentice-Hall, Englewood Cliffs, NJ, 1981.

[18] Stefanie Betz,Juho Makio, “Amplification of the COCOMO II regarding Offshore Software Projects.”

[19] Patrick Keil, Daniel J. Paulish, Raghvinder S.Sangwan,” Cost Estimation for Global Software Development.” EDSER 06, May 2006.

[20] Lionel C.Briand, Khaled L. Emam, Frank Bomarius, “ COBRA : A Hybrid Method for Software Cost Estimation, Benchmarking and Risk Assessment.” International Software Engineering Research Network Technical Report ISERN-97-24.

[21] Adam Trendowicz, Jens Heidrich, Jürgen Münch, et al. “Development of Hybrid Cost Estimation Model in an iterative Manner. “, ICSE’06, May 20-28, 2006, Shanghai, China: 331- 340.

[22] Ansgar Lamersdorf, Jurgen Munch,Alicia Fernandez-del Viso Torre, et a. “Estimating the Effort Overhead in Global Software Development.” International Conference on Global Software Engineering; IEEE; 2010; 267-276.

[23] Liming Wu, “The Comparison of the Software Cost Estimating Methods.”

[24] J. Dasgupta, G. Sahoo, R. P. Mohanty, “Monte Carlo Simulation Based Estimations: Case from a Global Outsourcing Company.” IEEE Int'l Technology Management Conference, IEEE, 2011: 619-625.

[25] Software Estimation Approaches http://en.wikipedia.org/wiki /Software_ development_effort_estimation

[26] Kjetil Moløkken, Magne Jørgensen ,” Group Processes in Software Effort Estimation”.

[27] Andrew Begel, Nachiappan Nagappan, “Global Software Development: Who Does It?”, International Conference on Global Software Engineering, IEEE, 2008: 195-199.

[28] Waverley Software global solutions, “Communication Paths in Distributed Software Development Environment”.

[29] Daniela de Freitas Guilhermino Trindade, Tania Fatima Calvi Tait, Elisa Hatsue Moriya Huzita, “A Tool for supporting the Communication in Distributed Software Development Environment.”, JCS&T Vol. 8 No. 2 July 2008; 118-124.

[30] Yasir Hassan Shah, Mushtaq Raza, Sani UlHaq, “ Communication Issues in GSD”, International Journal of Advanced Sciences and Technology, Vol 40, March 2012; 69-76.

[31] Ansgar Lamersdorf, Jurgan Munch, “ Studying the Impact of Global Software Development Characterstics on Project Goals: A Causal Model”, The Open Software Engineering Journal, 2010, 4, 2-13.

[32] Lucas Laymana, Laurie Williamsa, Daniela Damianb, Hynek Buresc, “Essential Communication Practices for Extreme Programming in a Global Software Development Team”, Information and software Technology Vol 48, issue 9, September 2006: 781–794.

[33] Paulus Insap Santosa and Ridi Ferdiana, “Email Communication Pattern in Global Software Development”, Asian Transactions on Engineering (ATE ISSN: 2221-4267) Volume 02 Issue 01, March 2012