

## MULTI STAGE SOFTWARE PROJECT EFFORT ESTIMATION

Chetan Nagar , Dr Anurag Dixit<sup>2</sup>



<sup>1</sup>Ph.D Student, Mewar University (Gangrar) Chittodgarh Rajasthan India

callchetan\_nagar@yahoo.com

<sup>2</sup> Professor CSE Galgotia Universities UP

<sup>2</sup>anuradixit@gmail.com

**Abstract**—Software industries have done a lot of advancement in the development of software but the effort required to build a software project is still much difficult to estimate in advance. For the success of the project accurate effort estimation is always essential. In practice; many industries still using ad hoc methods for efforts estimation. Effort estimation is an activity performed in analysis phase of software development. There are several reasons of estimation failure, but unclear requirement is a measure problem of efforts estimation. All the requirements are not exactly known to the customer when he decides to go for software development, because user exactly do not know what he wants and developer do not know what he has to build , a lot of changes take place in the latter phases of development, due to this reason effort estimation failed. This paper is suggests a new approach of effort estimation. In this approach we performed efforts estimation in more than one step of software development and check the difference, if the difference is more then take some necessary to adjust the difference. Use of strong monitoring policies is necessary for the success of a software project [15]. Strong monitor policies allow us to track slippage and take some necessary action to adjust this slippage.

**Keywords**— KLOC (Kilo Line of Code), UCP (Use Case Point), FP (Function Point), these are all unit of software size. Software Efforts estimation, Person-month, Man-Hours these are units of efforts.

### 1. INTRODUCTION

Requirements elicitation is both the hardest and most critical part of software development, since errors at this beginning stage propagate through the development process and are the hardest to repair later. Requirements elicitation is a difficult process in which one has to deal with ambiguity, informality, incompleteness and inconsistency, in which the “knowledge” of the requirements is not clear. Errors in requirements elicitation are, overall, most serious in software development, and harder to repair. 70% of the system errors are due to inadequate system specification [20].

Unclear requirement is the major problem in software effort estimation. It is frequently seen that customer is in too much hurry to start the development; he is not ready to spend sufficient time with the analysis team to understand the requirement completely and unambiguously.

It is another fact, when the project moves from one phase to another phase, both the customer and the developer got the better understanding of project because they become more familiar with the project and in this stage customer wants to change his requirement because now he realize that what is has said wrong. These changes lead to failure to schedule and failure of the project.

Various methods are followed in SDLC for software development and design with different phases. Braude, J. E., (2010) [8], Conger, Sue. (2008) [9], Sommerville, Ian., (2009) [10], Hoffer et al (2009) [11] had discussed the software development life cycle in detail. But the major phase identified common to all these methods are: Planning, Analyzing, Designing, Testing and Implementing and Maintenance. In analysis phase, the user requirements, system requirements, domain requirements, cost requirements and various other requirements of developing and designing software are studied and analyzed.

### 2. METHODS FOR ESTIMATION

In general following methods are more popular for efforts estimation.

- A. COCOMO Model [4] [5][16].
- B. Function Point Based Estimation [5] [1].
- C. Use Case Point Based Estimation [6] [14] [18].
- D. Expert Judgment. [7].
- E. Estimation by Analogy [7] [17].
- F. Parkinson's principle [3].
- G. Software Efforts Estimation using Soft Computing Techniques.
- H. Software efforts estimation using Neural Network Techniques [13].

In examining the assumptions and construction of typical cost estimation methods, two limitations are often the major impediment for a software project to meet the success criteria, i.e. on time, within budget, and with originally-specified functionalities. Effort estimation in practice usually relies on human domain experts, who offer estimations based on their experiences with similar projects in the past, and estimation quality thus depends on personal experiences and subjective judgments, which tend to result in unstable or even poor estimation accuracy [1].

Each method written above has some advantages and disadvantages. It is always a matter of discussion that which method is best. Every industry has its own liking

and disliking for a particular method. It is out of scope of our paper is to discuss the merits and demerits of these methods. Here we are assuming that industry is using best method of estimation which is providing them a good result.

In this paper our concentration on to cover the failure of effort estimation due to change of requirements in latter phases of software development.

### 3. MULTISTAGE EFFORTS ESTIMATION

The first step of software development is requirement analysis. In this phase we have to collect the requirements from the user. A lot of methods (like questionnaires, prototyping, brainstorming, interview and group interviews etc) are adopted for requirement gathering. After completion of requirements analysis we have to estimate the efforts required to build the project. Efforts estimation provides basis for the other software development activities like planning, Scheduling etc. As we know that efforts estimation is not an exact science, and it is just a prediction. After completion of project we can exactly determine the efforts were required to build the project, before that it is just a prediction, but as phases of the developments are passes accuracy of our prediction is increases. In the latter phase of development we can better estimate as compare than early phases of development.

In our approach first time we perform efforts estimation in analysis phase and second time after completion of design and check the difference, if difference is more, than we have to adjust it in remaining development time.

When we have asked regarding to analyst of some private companies, they had told that management and customer both are always in hurry to complete the requirement analysis. Customer is in hurry due to its business need and management is hurry to complete other activities like planning scheduling etc

Every phase of software development required some amount of effort to complete it. CSBSG data analysis shows a waterfall-based phase distribution scheme as: 16.14% for plans and requirements phase, 14.88% for design phase, 40.36% for code phase, 21.57% for test phase, and the other 7.06% for transition phase. In round figure we can write as 16% for plans and requirements phase, 15% for design phase, 40% for code phase, 22% for test phase, and the other 7% for transition phase

Consider an example, we have performed effort estimation for a project A. In analysis phase and found 1000 man-hours required to build that project and we have divided it according CSBSG 160 Man-Hours for plans and

requirements phase, 150 Man-Hours for design phase, 400 Man-Hours for code phase, 220 Man-Hours for test phase, and the other 70 Man-Hours for transition phase.

. Than we move for design phase, after completion of design we again perform efforts estimation and found that 1200 man-hours required building that project. At that time we have completed analysis and design phase that means we have spent 372 Man-Hours instead of 310 Man-Hours.

Now we have to accommodate 828 Man-Hours into 690 Man-Hours. We can not increase the main hours because in early estimation we had promise for 1000 Man Hours and we had quote the cost according to 1000 Man Hours. So we have to reduce the loss by compensating 828 Man-Hours into 690 Man Hours.

Logically it not possible to accommodate 828 Man-hours into 690 Man-Hours, so following actions can be taken to adjust these man-hours:

1. One simple approach for this adjustment is increase working time of your development team.
2. We can increase team size.
3. We may outsource some component.

### 4. RESULTS

In this paper, we have tried to throw a new concept to test and debate, although, it is always criticized that why we have to perform efforts estimation multiple times. But two time estimation is better than a project failure.

Approximate all software projects are facing the problem of change in requirement and we cannot stop it, so one solution (Multistage effort estimation) is suggested in this paper.

It is somewhat difficult to produce the result based on this concept. We are all seen that there is a marginal difference between estimated efforts and actual efforts. One major reason is for it is poor understanding of the project.

We have applied this concept on some projects of a small software industry. First, we had estimated the effort after requirement analysis and then again estimate after designing and we have found some difference. Now what to do next that will depend upon the management, they may take action specified in the previous section. The company was not ready to display their result that is why we are not displaying the result.

## 5. CONCLUSION

Success of project (complete on time and within budget) is depending on so many parameters, accurate efforts estimation is playing a crucial role in the success of a software project. But the successful effort estimation depends on clear and complete requirements, use a mature process for estimation and use of strong monitoring policy. It is very difficult to get clear and complete requirements for the project and all the customers. So we have to use to use some alternative way which is the multistage software effort estimation.

This paper is not suggesting any new model or method for the estimation; we are just giving one concept which can allow you to complete the project on the time. Every model or method in the effort estimation is based on the prediction. As the time passes and we move towards completion of the project, our prediction becomes more accurate, because now we are becoming familiar with most of the facts. This approach will help us to complete the project on the time. Although in this approach we have to perform estimation two times. But multiple time efforts estimation is much better than project slippage or failure.

## REFERENCES

- [1] BINGCHIANG JENGI\*, DOWMING YEH2, DERON WANG3, SHU-LAN CHU2, "A Specific Effort Estimation Method Using Function Point" JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 27, 1363-1376 (2011)
- [2] Anda, B., Benestad, H.C., Hove, S.E.: "A multiple case study of Effort Estimation based on Use Case Points" Empirical Software Engineering, 2005.
- [3] G.N. Parkinson, *Parkinson's Law and Other Studies in Administration*, Houghton-Mifflin Boston, 1957
- [4] Roger E Masse "An Analysis of the Evolution of COCOMO and Function Points "University of Maryland, July, 1997
- [5] Basavaraj M.J, Dr. K.C Shet "Empirical validation of Software development Effort multipliers of Intermediate COCOMO Model" JOURNAL OF SOFTWARE, VOL. 3, NO. 5, MAY 2008.
- [6] Anda, B., Benestad, H.C., Hove, S.E.: "A multiple case study of Effort Estimation based on Use Case Points" Empirical Software Engineering, 2005
- [7] <http://www.scribd.com/doc/51198168/45/Expert-Judgment-Method>
- [8] Braude, J. E., (2010) – Software Engineering – An Object Oriented Perspective – Wiley
- [9] Conger, Sue., (2008) – The New Software Engineering – Global Text
- [10] Sommerville, Ian., (2009) – Software Engineering – 8e – Pearson Education
- [11] Hoffer, J. A., George, J. F., Valacich, J.S., (2009) – Modern Systems Analysis and Design – 5e – Pearson Education
- [12] As Karner G.. "Metrics for Objectory". Diploma thesis, University of Linköping Sweden. No. LiTH-IDA-Ex-9344:21, December 1993
- [13] Effort Ali Bou Nassif, Luiz Fernando Capretz, and Danny Ho. Estimating software effort using an ann model based on use case points. In Machine Learning and Applications (ICMLA), 2012 11th International Conference on, volume 2, pages 42–47. IEEE,2012.
- [14] Clemmons R.: Project Estimation With Use Case Points, The Journal of Defense Software Engineering, 2006
- [15] Chetan Nagar and Dr Anurag Dixit, "Software Project Management with Control Based Monitoring" International Journal of Advances in Science & Technology, October-2011, volume 3 No 4.
- [16] B. W. Boehm, Software engineering economics, Englewood Cliffs, NJ: Prentice-Hall, 1981.
- [17] M. Shepperd and C. Schofield, "Estimating software project effort using analogy", IEEE Trans. Soft. Eng. SE-23:12, 1997, pp. 736-743.
- [18] Karner, Gustav: "Resource Estimation for Objectory Projects Objective Systems " SF AB, 1993
- [19] Mel Damodaran and A Washington. Estimation using use case points. Computer Science Program. Texas–Victoria: University of Houston. Sd, 2002.
- [20] Prasad Rajagopal, Roger Lee1, Thomas Ahlswede,"A New Approach for Software Requirements Elicitation", Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence,Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN'2005)