Journal of Indian Research, Volume 1 No. 1, 2013, 85-90

Computer Science

MULTI STAGE SOFTWARE PROJECT EFFORTS ESTIMATION

Chetan Nagar*, Dr Anurag Dixit**

ABSTRACT

The staff resources or effort required for a software project are notoriously difficult to estimate in advance. Accurate effort estimation is essential for the success of a software project. In practice; many industries still use ad hoc methods for efforts estimation. Efforts estimation is an activity performed in analysis phase of software development. All the requirements are not exactly known in this phase, because user exactly does not know what he wants and developer does not know what is to be built. Lot of changes take place in latter phases of development, due to this reason effort estimation failed. Unclear requirement is a measure problem in efforts estimation. This paper aims to suggest a new approach of multistage efforts estimation. We intend to perform efforts estimation in more than one step of software development and check the difference to establish optimal level of manpower required for a project.

Keywords: KLOC (Kilo Line of Code), UCP (Use Case Point), FP (Function Point), . Software Efforts estimation, Person-month, Man-Hours

INTRODUCTION

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

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.

METHODS FOR ESTIMATION

In general following methods are more popular for efforts estimation.

A. COCOMO Model [4][5].

B. Function Point Based Estimation [5][1].

**Chetan Nagar is Ph.D scholar in Mewar University, Gangrar, Chittodgarh Rajasthan (e-mail- callchetan_nagar@yahoo.com)*

***Dr Anurag Dixit is Dean-Professor (CS/IT) BRCM CET, Bahal Bhiwani (e-mail: anuradixit@gmail.com)*

C. Use Case Point Based Estimation [6][14].

D. Expert Judgment. [7].

E. Estimation by Analogy [7].

F. Parkinson's principle

G. Software Efforts Estimation using Soft Computing Techniques.

H. Software efforts estimation using Neural Network Techniques.

There are lot of methods which can be use for efforts estimation, but an industry wants a simple and accurate way of efforts estimation. We must use more calculative method as compare than more predictive approach. COCOMO and Use Case Point are more calculative approach which is covering many factors that may affect the cost. But Expert judgment and Analogy based estimation is more predictive approach, in which much experience is required. A rich set of old data is required for better estimation.

A. COCOMO (Cost Construction Model)

COCOMO is one of the popular and old models of efforts estimation. In this model we have to estimate the line of code. Counting of line of code is one of the difficult tasks when project is complex and new to us. In such situation we have to divide that project in module and divide that module into sub module to make problem less complex. Most senior person of your team should take responsibility to count KLOC, because we need to estimate KLOC before writing it. We can use old data to predict KLOC, but no project will completely same with previous project, so some intelligence and experience will be required for a better estimation. Here we will use advance COCOMO for estimation.

Efforts = $a*(KLOC)^b$ *EAF

Here a and b are complexity factor.

TABLE I

COMPLEXITY FACTOR

Model	A	B
Organic (simple in terms of size and complexity	3.2	1.05
Semi-ditched (average in terms of size and complexity)	3.0	1.12
Embedded (Complex)	2.8	1.20

In Intermediate COCOMO only 17 EAF are used, but in advance COCOMO we are using 22 EAF. Typical values for EAF range from 0.9 to 1.4.

B. Function Point based Estimation

The steps for counting function points are as following:

1. Identify data functions (External Interface files and Internal Logical Files) and rate them.
2. Identify transaction functions (External Input, External Output and External Inquiry) and determine the complexity.

3. Compute unadjusted function points. Number of EI, EO, EQ, ILF and EIF for each complexity level (Simple, Average and Nominal) is obtained and the corresponding weight for each complexity level is multiplied with the count to finally get the unadjusted function point count. Details of function point count are available in appendices.
4. Determine the ratings of 14 general system characteristics.
5. Calculate value adjustment factor (VAF).

   *VAF = (TDI * 0.01) + 0.65*

   Where, TDI = Total Degree of Influence obtained by multiplying the ratings of general system characteristics.
6. Compute function point counts.

C. Use Case Point Approach

It is another popular and efficient method of efforts estimation. Here we will calculate use case and actors

UUCP=Use cases + Actors

Where UUCP stands for Unadjusted Use Case Point Using the following table1 we can calculate Use Case used in a project

TABLE II

USE CASE CALCULATION

Use case type	Description	Quantity	Weight Factor	Sub total
Simple	3 or fewer transaction		1	
Average	5 to 7 transaction		5	
Complex	Greater than7 transaction		10	
TOTAL				

By using the following table2, we can estimate actors used in a project

TABLE III

ACTOR CALCULATION

Use case type	Description	Quantity	Weight Factor	Sub total
Simple	3 or fewer transaction		1	
Average	5 to 7 transaction		2	
Complex	Greater than 7 transaction		3	
TOTAL				

UCP=UUCP*TCF*EF

TCF is Technical Complexity Factor, which is sum of 13 complexity parameters; each factor is range from 1 to 5 and multiplied by weight.

EF is Experience Factor, which is sum of 08 complexity parameters; each factor is range from 1 to 5 and multiplied by weight.

Effort = UCP *ER (Efforts will be in man hours)

Finally, the UCP is multiplied by a historically collected data representing productivity, such as a factor of 20-staff hours per use case point, to arrive at a project estimate. The result is an estimate of the total number of person hours required to complete the project. Karner provided a detailed table describing how each factor was determined and what value was assigned at each step [18].

D. Expert Judgment

This method involves consulting one or more experts. The experts provide estimates using their own methods and experience. Expert-consensus mechanisms such as Delphi technique or PERT will be used to resolve the inconsistencies in the estimates.

E. Estimation by Analogy

This method requires one or more completed projects that are similar to the new project and derives the estimation through reasoning by analogy using the actual costs of previous projects. Estimation by analogy can be done either at the total project level or at subsystem level. The total project level has the advantage that all cost components of the system will be considered while the subsystem level has the advantage of providing a more detailed assessment of the similarities and differences between the new project and the completed projects. The strength of this method is that the estimate is based on actual project experience. However, it is not clear to what extend the previous project is actually representative of the constraints, environment and functions to be performed by the new system

F. Parkinson's Principal

Using Parkinson's principle "work expands to fill the available volume" [3], the cost is determined (not estimated) by the available resources rather than based on an objective assessment. If the software has to be delivered in 12 months and 5 people are available, the efforts estimated to be 60 person-months. Although it sometimes gives good estimation, this method is not recommended as it may provide very unrealistic estimates. Also, this method does not promote good software engineering practice.

MULTISTAGE EFFORTS ESTIMATION

The first step of software development is requirement analysis. In this phase we have to collect the requirements from the user, 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, 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.

Every phase of software development required some amount of efforts 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 [13]. 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 performed efforts estimation for a project A in analysis phase and found 1000 man-hours required to build that project and we have divided it according to 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.

Subsequently,we move for design phase. After completion of design, we again perform efforts estimation and found that 1200 man-hours are required for 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 anticipated for 1000 Man Hours and we quoted the cost according to 1000 Man Hours.Therefore, we have to reduce the loss by compensating 828 Man-Hours into 690Man Hours.

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

1. Increase working time of development team.
2. Increase team size.
3. Outsource some component.

RESULTS

In this paper, we have tried to throw a new concept to test and debate. Although, it is open to criticism that why did we have to perform efforts estimation multiple times, but in works by R S Pressman, it is suggested for better estimation, delay efforts estimation until it is possible. Actually it is not possible, so the multistage efforts estimation can work as a substitute for it. 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 estimated after designing and we have found some difference. Company was not ready to display their result,the complete data is not being dispalyed .

CONCLUSION

Success of project (complete on time within budget) is dependant on various parameters like first clear and complete requirements, Use of a mature process for estimation and some strong monitoring policy.

This paper does not suggest any new model or method for the estimation. We are just giving one concept which can allow professionals to complete the project on the time. Every model or method in the efforts estimation is based on the prediction .As the time goes by and we move towards completion of the project, our prediction becomes more accurate, because now we are familiar with most of the facts. This approach will help us to complete the project on 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, Dowmimg Yeh, Deron Wang, Shu-Lan Chu. "A *Specific Effort Estimation Method Using Function Point"* Journal Of Information Science And Engineering, Vol.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-Miffin, 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., *Software Engineering – An Object Oriented Perspective* – Wiley (2010)

[9] Conger, Sue., *The New Software Engineering – Global Text* (2008)

[10] Sommerville, Ian., *Software Engineering* – 8e – Pearson Education (2009)

[11] Hoffer, J. A., George, J. F., Valacich, J.S., – *Modern Systems Analysis and Design* – 5e – Pearson Education (2009)

[12] As Karner G.. "*Metrics for Objectory"*. Diploma thesis, University of Linköping Sweden. No. LiTH-IDA-Ex-9344:21, December 1993

[13] *Phase Distribution of Software Development Effort*

[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, volume 3 No 4.October-2011