"Understanding of Software effort Estimation at the early Software Development of the life cycle - A literature View"

Kardile Vilas Vasantrao

Department of Computer Science.
Tulajaram Chaturchand College, Baramati
Affiliated by Pune University, Pune, Maharashtra (India)

Abstract: Software project development process is requiring accurate software cost and schedule estimation for achieve goal or success. Though a lot referred it as the "Intricate brainteaser" because of its conscience attribute which is impact by complexity and uncertainty, it is not as difficult or puzzling as people think. In fact, generating accurate estimates is straightforward—once you understand the intensity of uncertainty and module which contribute itself process. Because estimate any work and its activity is repeated incident in our daily life on the basis of the time and risk control design approach. The estimated time and risk is very according external uncertain factor and theme's condition. In our everyday life, we improve our estimation skill based on past experience in which problem solve by which method and in which condition and which opportune provide that method to produce better result .

So, Instead of unexplained treatises and inflexible modeling techniques, this will guide highlights a proven set of procedures, understandable formulas, and heuristics that individuals and complete team can apply to their projects to help achieve estimation ability with choose appropriate development approaches In this paper demonstrates myself tries to understand core factors influence and its consideration in effort estimation process, some basic issues which are focus to interior practice of effort estimation process and its pitfalls., Estimation approach and its categories implementation and the inherent limitation of effort estimation and some basic issues., Software Estimation as predictable skill?**,** Need to Estimate Uncertainties for enhance accuracy in estimation process.

Key words: software engineering methodology, early effort software estimation, software estimation method.

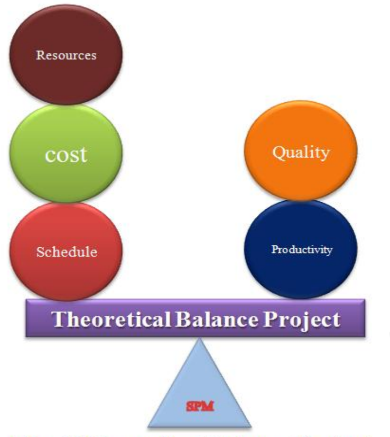
Introduction

Software estimation is essential in software engineering. Developing software project estimation does mean to plan a project and need to know how control risk. In software development project process generally development team estimate three aspect of project: effort, schedule and cost. There various method of estimation most of them stared with estimating size function point or proxy point using that effort estimation.[11] From effort developer team typically derive staffing, schedule and cost. Estimating effort is primary challenge; once development team has an effort estimate, wonderful tools are available to help you work through development schedule, staff and risk and tie them into to development project plan. The result of this thinking software effort estimation process is very straight forward but why basic instinct wasn't wrong. Software projects do fail to meet their goal with surprising regularity. They were just trying to limit their exposure. What are the causes behind it?

To find out secrete of effort estimation process, firstly consider intrinsically factors and contributory process which acts on estimation process.[1]

There are five intrinsically linked factors in estimating software product development projects:

1. The **Schedule** expected total time and effort to produce the specified product.
2. The **Productivity** ratio of Desired and actual work.
3. The **Cost**.
4. The **Resources** required for the project – their skills and availability.
5. The **Quality** for the product; the features, functionality and user acceptable experience



6. . Fig : 1 Theoretical Balance Project

In general terms, Development team is trying setting a "Goal." In project management, the general assumption is developer can set any one of the five factors as a target for a project, the other four factors to where they need to go to reach the goal.[1]

But in fact, when development team set more than one factor as fixed it generate an almost impossible pressure among the remaining factors that is strong cause to project will fail to meet its goals. It begins with superior of intentions, but with two or more factors fixed, any change in circumstances during the project creates an imbalance that cannot be corrected with the remaining factors.[1]

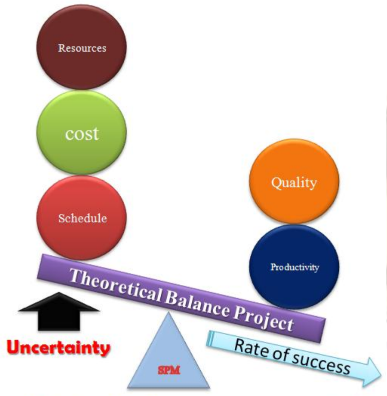


Fig : 2 Scope of Project development process practically

The question is why does this thing take place? And if such things take place and these cause to project failure, then we must think about some basic issue of effort estimation which will make estimation is critical and crucial .this motivate to think about each of the factors individually and the impact they have on the project.

Section – II Basic Issues which are focus to interior practice of effort estimation and its pitfall

In the project's life cycle there is no any facility where take past experience or track and understand where we under or overestimated so you can do better the next time? On Software Effort Estimation's various review of surveys is concluded themselves in the overruns root causes, which are might be complex, the data is not reliable always, and those responding to the surveys may have a tendency to over report causes that lie outside of their responsibility for example, customer-related causes. [2] Very often one factor become cause of project failures so it require focus on factors such as schedule, unclear or undefined or changeable requirement, change in technology, unavailability of resources, manpower and policy problem which leads to overrun the project but with that development team must focus on its control which is customer requirement though customer requirement in changeable from development team must adopt such policy which is sustain in such situation.[3]

The literature lists several common causes for overruns:

- Incomplete or unclear requirements Specification,
- In Changeable situation failing to adjust schedules
- It is failed to setting overly aggressive development schedules
- Due to Insufficient enough people or equipment.

These reasons are rather generic. In my view, the primary causes of software project overruns are:

- Due to lack of training and education.
- Undefined, Incomplete, Unclear, changing, and creeping requirements.
- Due to confusion of the desired schedule/effort target with the estimate.
- Due to confusion situation and Hope-based planning.
- User acceptable Quality surprise.

Section – III Software effort Estimation approaches and its classification with suitable opportune for implementation

Software estimation approaches are categories many ways. (Modified original By the Software development effort estimation From Wikipedia, the free encyclopedia *en.wikipedia.org/wiki/ Software_ development _effort_estimation last access 3 jan 2011)*

The top level categories are the following:

- Expert estimation: In this step estimation is formed on the basis of processes judgment.
- Formal estimation model: In this step estimation formed on the basis of mechanical processes, (the use of formula derived from historical data.)
- Combination-based estimation: In this step estimation is formed on the basis of judgmental or mechanical combination of estimates from different sources.

Below are examples of estimation approaches and its suitable opportune within each category.

Table 1. : Categories estimation process and its Implementation
(Modified Original table By the Software development effort estimation From Wikipedia, the free encyclopedia.
(*en.wikipedia.org/wiki/ Software_development_effort_estimation last access 3 jan 2011),[13,14]*

Category	Suitable opportune Estimation approach	Suitable implementation of estimation approach
Formal estimation model	Analogy-based estimation	ANGEL, Weighted Micro Function Points
	Parametric models	COCOMO, SLIM, SEER-SEM
	Size-based estimation model	Function Point Analysis, Use Case Analysis, SSU (Software Size Unit), Story points-based estimation in Agile software development
Expert estimation	Group estimation	Planning poker, Wideband Delphi
Combination-based estimation	Mechanical combination	Average of an analogy-based and a Work breakdown structure-based effort estimate
	Judgmental combination	Expert judgment based on estimates from a parametric model and group estimation

Regarding to software estimation methodology, tools and exist model have lot of documentation are available. Most of them for effort estimation and some of themes for cost estimation. Most of method utilizes analytical formula which is based project characteristics such as project size, domain, and complexity and development methodology and cost driver.

Preliminary the methodology and tools have base on data from past project experience .Development teams researcher and engineer analyze existing project scenario and formalize some formula and equation which are matched the data point. From that formula or equation no one totally accurate with past experience or not to be developers predication for the new project .so development team focus on predication of effort not to formula or equation base assumption. Effort is combination of plan and methodology of work for completion. Because development team estimates the same project using the different estimation methodologies and models. Their results vary widely, depending on their definition of the project, their assumptions, and the models used. Designer should choose complementary methods with different biases so we must consider the contributory the methods and what the strength and weakness of these methodologies.

Section – IV Available software development methodology's classification and its suitable opportune.

In the early stage of software life cycle project manager are inefficient to estimate the effort, schedule, cost estimation and its development approach .This in turn, confuses the manager to bid effectively on software project and choose incorrect development approach. That will directly effect on productivity cycle and increase level of uncertainty. Plan driven (Traditional) software development methodology is inefficient to high level of uncertainty as contrasts [4,5,7,9]

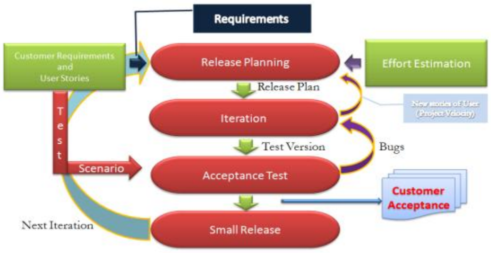


Fig 3: Process Steps of a Plan driven-project

Practice driven (Modern/Agile) software development methodology is very efficient in high level of Uncertainty but it is suitable short project. For this becomes a strong cause of project failure.[5,6,7,8,9] So to avoid such problem if we know level and sources of uncertainty in model design, It will directive the developer to design accurate software cost and schedule estimation.

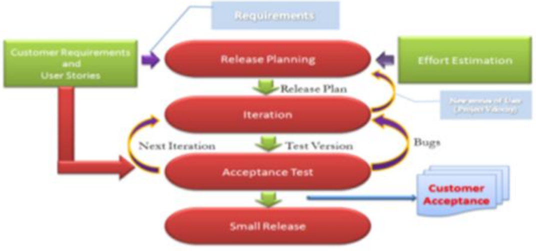


Fig 4: Process Steps of a Practice driven-project

In such situation development practitioner and Organization and company has confuse for which approach is abandoning or which adopting because of the strength and weaknesses which will force to learner for accept

"Technology never fail it will failure to produce best result due to opponent opportune."

In other hand if we provide to suitable opportune for technology when we estimate then it will give best result and trying to reduce futurity so in estimation process if estimator choose the design approach as per uncertainty.

Section –V: Software Estimation as predictable skill ?

The available methodologies and methods, if we have such a great toll in our hand then why are the estimation results so ragged? In some respects, the new question rise to Is software estimation is an science process or not. [11] To find out the answer we must consider those circumstances where estimation is not as difficult or puzzling. In fact, generating accurate estimates is straightforward—once you understand the intensity of uncertainty and module which contribute itself process. Because estimate any work and its activity is repeated incident in our daily life. We try to estimate our plan on the time and risk control design approach. The estimated time and risk is very according external uncertain factor and theme's condition. In our everyday life, we enhance our estimation based on past experience in which problem solve by which method and in which condition and which opportune provide that method to produce better result .

So, Instead of unexplained treatises and inflexible modeling techniques, this will guide highlights a proven set of procedures, understandable formulas, and heuristics that individuals and complete team can apply to their projects to help achieve estimation ability with choose appropriate development approaches. [15]

Due to incomplete and unclear requirement at the early stage of software life cycle project manager are inefficient to estimate the effort, schedule, cost estimation and its development approach .This in turn, confuses the manager to bid effectively on software project and choose incorrect development approach. That will directly effect on productivity cycle and increase level of uncertainty. This becomes a strong cause of project failure. So to avoid such problem if we know level and sources of uncertainty in model design, It will directive the developer to choose and design accurate software development methodology and then It is very straight to estimate cost and schedule estimation.

Section – VI Need to Estimate Uncertainties for enhance accuracy in estimation process

Process of estimation is probabilistic and communicates them appropriately. Every project have some uncertainties so estimates are typically the 50-percent view, means It will be probably under or over budget or may accurate. (Unfortunately, Parkinson's Law, which states that work expands to meet the time available, holds for software projects. So, this 50-percent view says we'll be on budget 50 percent of the time and over budget the other 50 percent.) [11]

Development team may estimate more accurately by accept and understand uncertainty which are inherently present in project and try to manage and change estimation accordingly.

Agile methods which are very suitable change, expect more variation. These systems estimation accuracy follows a log-normal distribution. (That is, it was underestimated far more frequently than overestimated), and this behavior and uncertainty range is nearly identical at all stages in the project lifecycle, in conflict with the cone of uncertainty.[10]
The degree of confidence is indicating contrast level of uncertainty. Mean if degree of confidence is high it indicates low level of Uncertainty. Development team estimate project with possible outcomes in unclear or not well define situation. A person is uncertain if s/he lacks confidence about the specific outcome and in this situation how s/he take a decision confidently. So if there is use of uncertainty estimation before effort estimation process it will very helpful for take decision in estimation process. and which will assist to managing the effort budget. [15]

Section – VII Discussion

Accurate software cost and schedule estimation are essential for software project success. Often it referred to as the “black art” because of its complexity and uncertainty, software estimation is not as difficult or puzzling as people think. In fact, generating accurate estimates is straightforward—once you understand the intensity of uncertainty and framework for the modeling process. Secrete of successful software estimation—distilling academic information and real-world experience into a practical guide for working software professionals. Instead of arcane treatises and rigid modeling techniques, this will guide highlights a proven set of procedures, understandable formulas, and heuristics that individuals and development teams can apply to their projects to help achieve estimation proficiency with choose appropriate development approaches

Section – VIII Conclusion

There is not much to conclude .This is early in my study, My hope is that systematic look towards enhance estimation talent, processes, and how development team communicate with estimations and try to avoid the difficulties inherent in estimation; however, development team’s ability to estimate well and have always be limited by the extent of projects uncertainty

Reference:

1) Michael Dunham The 5 Variables of Project Estimation » Haut Tech blog.sciodev.com/2010/04/06/the-5-variables-of-project-estimation/6 Apr 2010
2) Molokken, K. Jorgensen, M.. "A review of software surveys on software effort estimation". http://ieeexplore.ieee.org/xpls/abs_all.jsp?
3) Jørgensen, M.. "A Review of Studies on Expert Estimation of Software Development Effort". http://simula.no/research/engineering/publications/SE.4.Joergensen.2004.c.
4) “A Comparison between Agile and Traditional Software Development Methodologies” M. A. Awad
5) Dr. Li Liu(2007)An Economic Model of Software Development Approaches (ausweb.scu.edu.au/aw06/papers/refereed/kong/paper.html)
6) B. Boehm and R. Turner, *Balancing Agility and Discipline: A Guide for the Perplexed*,
7) Beck, K.; Boehm, B. (2003) "Agility through discipline: a debate", Computer, Volume 36, Issue 6, June 2003 Page(s):44 - 46
8) Manifesto for Agile Software Development,
9) Iansiti, M. and A. MacCormack (1997) "Developing products on Internet time", Harvard Business Review, Sep-Oct.,75(5): p108-117.
10) Todd Little published data for 120 delivered systems based on the agile methodology (T. Little, "Agility, Uncertainty, and Software Project estimation, "www.toddlittleweb.com/Papers/Agility, %20Uncertainty%0and%20Estimation.pdf)
11) Linda M. Laird the limitation of software estimation (posterous.com/getfile/files.posterous.../4._Laird-ITPro06.The_Limitatio.pdf)
12) **Software** development **effort estimation** - Wikipedia, the free (en.wikipedia.org/.../**Software**_development**_effort_estimation)**
13) Briand, L. C. and I. Wieczorek (2002). Resource estimation in software engineering. Encyclopedia of software engineering. J. J. Marcinak. New York, John Wiley & Sons: 1160-1196.
14) Morris Pam - Overview of Function Point Analysis Total Metrics - Function Point Resource Centre
15) Uncertainty in the environmental modelling process e A framework and guidance -Jens Christian Refsgaard [a*,] Jeroen P. van der Sluijs b, Anker Lajer Højberg a, Peter A. Vanrolleghem c,d J.C. Refsgaard et al. / Environmental Modelling & Software 22 (2007)

Acknowledgements

I would like to thanks to all staff member of computer sci. Dept. who inspire us for this work.