

## Understand need “Uncertainty Assessment” in the system development modeling process

**Kardile Vilas Vasantao**

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. A lot it referred to as the “Intricate brainteaser” because of its conscience attribute which is impact by complexity and uncertainty, Generally 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 module which contribute itself process. Because Design and chose approach is repeated incident in our daily life when we plan to go to our work .We estimate the time and risk need for design approach. The estimated time and risk fluctuates 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

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. 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 design accurate software cost and schedule estimation. which are require l for software project success.

This paper demonstrates need of uncertainty analysis module at the modeling process for assist to recognize modular uncertainty system development process and the role of uncertainty at different stages in the modeling processes.

**Key Words:** software development approach, Uncertainty, Uncertainty assessment

**Introduction:** Design and chose approach is repeated incident in our daily life when we plan to go to our work .We estimate the time and risk need for design approach. The estimated time and risk fluctuates according external uncertain factor and theme’s condition.

In our everyday life, we enhance our estimation based on past experience and historical data.

Design of software design approach is crucial because of today’s dynamic environment of software development firm. The World Wide Web has provided a platform for companies to communicate and transact directly with their customers and partners. Challenges arise due to fast evolving technology and increased competition as companies are under constant pressure to develop new functionalities to satisfy changing client needs and to deliver them in short cycles at low costs. (Iansiti and MacCormack 1997).

Existing methods for software development have lot of options which can be classified into two categories, plan-driven (traditional) and practice –driven (Boehm and Turner 2003; Iansiti and MacCormack 1997). During the early stage of the plan-driven approach, the user needs are identified; requirements for new functionalities are specified; technical specifications are created; development processes are defined; specific project targets are spelled out; and, acceptance criteria and tests are outlined. Many of the CMM or ISO based methods belong to this category. The focus of the project team is on development and implementation according to the plan. As a result, the success of a project using plan-driven approach hinges on the validity and reliability of the project plan. On other hand if requirement of user in changeable form then it will increases level of uncertainty the plan driven approach is less effective.

In a dynamic environment such as the World Wide Web, it is difficult to predict technological changes, clients' needs and thus the solutions to clients' problems. Very often, the client

lacks the knowledge and experience to describe their needs, not to mention requirements and specifications. There is evidence showing that effective learning by the client during the initial design phases of a project impacts positively on its success (Wastell 1999; Majchrzak et al 2005). Requirement changes at later stages, which often happen as a result of client learning occurring late in a project, could lead to not meeting client expectations, and budget or schedule overruns (Boehm 1989). Therefore, a rigid project plan becomes less effective in guiding the project team and could result in an obsolete system. Instead, the project team needs to collaborate with the client to help each other learn fast and the development process should be flexible enough to allow frequent changes.

The practice-driven (Agile) methodology refers to a family of software development methods that focus on customer needs, shorter development cycle time and adapting to changes requirements. The family includes Extreme Programming (Beck et al 2002), Scrum, DSDM and Crystal Family (Fowler 2005). Agile method is characterized by a chaotic perspective, collaborative values and principles, and barely sufficient technology (Highsmith 2002). Its core values include: individuals and interactions over processes and tools; working software over comprehensive documentation; customer collaboration over contract negotiation; responding to change over following a plan (Manifesto for Agile Software Development). Compared with the plan-driven approach, the agile methodology addresses the lack of knowledge of both the client (on the technology and development process) and the developer (on client's business needs) by encouraging closer collaboration. Usually, client representatives are collocated and work alongside the project team. Instead of working against the plan, frequent changes are embraced to address the client's changing business needs. As a result the success of project using practice-driven approach hinges on the effective communication skill and correct abstraction formalization. Of development Team.

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 fail to produce best result due to opponent opportune.”***

**So if developer get to know at early stage of estimation that which methodology is suited for which module of system then we are able to reduce failure cause of software process.**

The aim of this paper to understand modeling process and uncertainty appearance in particular module for better to estimation process. In section II introduce various factors which affect software productivity. In section III Available software development approaches comparatives. In section IV Modeling process

## Section –II:

Problem: Design a Process: In software development, Modern “lightweight” methodologies are gaining ground on more traditional “heavyweight” methodologies. Both have their advantages and disadvantages, and appropriateness where we get best result. Many project fail because of inaccurate handling design approach or decision made early some time prove to be wrong later on. The most critical and crucial part of software development approach is when planning of design development is required in the early stage of the software life cycle where problem to be solved. Estimated, requirement by user is not completely understand and problem to be solved had not yet been completely revealed. The Major issue that separates the various processes that we looked at is the amount of upfront Planning they require. We can think of this as a spectrum, which at one end has a purely Plan oriented and other end practice oriented question is then for any given situation how do find the right approach.

Alternatively You can find a situation where the approach will give best result for this causes to handle this uncertainty we must be understand Risk handle strategies of each approach. Software Risk although there has been considerable debate about proper definition for ware risk, There is general agreement that risk always involves two characterizes:

Uncertainty and Complexity (Project Risk, Technical Risk, Business risk, .etc). Which will directly affect the Software deployment Process and approach? There are four broad control factors. This factor s is interconnected .when one changes at least one other factor must also change.

- Cost- or Effort. Available money impact the amount of effort put into the system
- Schedule – A Software project is impacted as the timeline is changed.
- Requirements-The scope of the work that needs to be done can be increased or decreased to affect the project.
- Quality – Cut control by reducing quality.

To avoid such problem if we know level and sources of uncertainty in model design, initial phase of development, It will directive the developer to design accurate software cost and schedule estimation. Which are essential for software project success. However once the required efforts have estimated, little is done to recalibrate and reduce the uncertainty of the initial estimates.

### Section –III:

*This comparison Focus on* : Practice driven is sometimes characterized as being at the opposite end of the spectrum from "plan-driven" or "disciplined" methods. This distinction is misleading, as it implies that agile methods are "unplanned" or "undisciplined". A more accurate distinction is that methods exist on a continuum from "adaptive" to "predictive". Practice-driven lie on the "adaptive" side of this continuum.

Adaptive methods focus on adapting quickly to changing realities. When the needs of a project change, an adaptive team changes as well. An adaptive team will have difficulty describing exactly what will happen in the future. The further away a date is, the vaguer an adaptive method will be about what will happen on that date. An adaptive team can report exactly what tasks are being done next week, but only which features are planned for next month. When asked about a release six months from now, an adaptive team may only be able to report the mission statement for the release, or a statement of expected value vs. cost.

Predictive methods, in contrast, focus on planning the future in detail. A predictive team can report exactly what features and tasks are planned for the entire length of the development process. Predictive teams have difficulty changing direction. The plan is typically optimized for the

original destination and changing direction can cause completed work to be thrown away and done over differently. Predictive teams will often institute a change control board to ensure that only the most valuable changes are considered.

### Categories wise best practice

| Plan Driven Software Development | Practice –driven Software Development |
|----------------------------------|---------------------------------------|
| High Criticality                 | Low Criticality                       |
| Junior Developers                | Senior Developer                      |
| Requirements do not change Often | Requirement change often              |
| Large number of Development      | Small number of developer             |
| Culture that demands order       | Culture that thrives on chaos         |

### Section –IV:

**Modeling as a part of Project planning in system development** is one of the most critical activities within the project lifecycle. **Project plan development** is the main part of Project Planning Stage. The project manager takes the responsibility for creating a project plan that is a formal document showing the basis upon which to assess the performance of the project and measure its results. Let's review the steps of **project plan development** in detail.

Create the Work Breakdown Structure.

To design project plan, developer will need to settle on the *Work Breakdown Structure* (the acronym "WBS") for his/her project. Which is give detailed list of all the phases, activities and jobs required for successful project completion. In other word we can the WBS becomes the foundation for his/her project plan as S/he can use it to decide the resources required to deliver each activity or task listed. The WBS allows designer to design simple to-do lists and task lists and then assign them to members of the project team.

The benefit of WBS:

When designer try to developing a project plan WBS depicts the dependencies between tasks. Which is focus on, how each task is associated with other tasks and what (internal or external) dependencies can be set.

## 1. Define the Required Resources.

Once the plan and its activities required for deliver to project are situate, next step in creating a project plan is to identify the resources required for each task and activity. WBS showing the possibility of the project which describes resources are required and in which quantities and measures. In this step the project resource base and types of resources will be describe.

Generally project may require following resources:

- Full-time and part-time employees
- Equipment and materials

Designer of plan is calculate how many people S/he should employ to do project and Human and equipment resource.

## 2. Design a Project Schedule.

After the Work Breakdown Structure is completely design. Designer are able create a project plan in which schedule tasks and deadline for activities listed in the

To build a project schedule the following information is necessary:

- Identified tasks and activities and their dependencies (both internal and external)
- Assignments to members of the project team (who will do which task)
- Risk mitigation strategies and a contingency plan Critical milestones
- Allocated resources required for the project

Designer can design a project schedule on the basis on that information which is described at last two step of project plan development.

But if designer have knowledge of modular uncertainty in work break structure he is able to choose appropriate team for suitable module. But

unfortunately there is no such phase who will assist to designer to take proper decision while make project plan

## Section –IV:

### 4.1 Uncertainty expressions and classification

#### 4.1. A Definitions

Uncertainty and associated terms such as error, risk and ignorance are defined and interpreted differently by different authors, see Walker et al. (2003) for a review. There are different definitions available in various literature .

we adopt a subjective understanding of uncertainty in which the degree of confidence Thus according to our definition a person is uncertain if s/he lacks confidence about the specific outcomes of an event. Reasons for this lack of confidence might include decision of the information as incomplete, unclear, inaccurate, unreliable, inconclusive, or potentially false. Similarly, a person is certain if s/he is confident about the outcome of an event. It is possible that a person feels certain but has take wrong decision of the information

There are many different decision situations, with different possibilities for characterizing uncertainty. Uncertainty is also known degree of unreliability of knowledge, which translates into a state of confidence.

#### 4.1.B Classification:

Classification of unsatisfactory knowledge is represented by Brown (2004) . Which is useful to differentiate between bounded uncertainty and unbounded uncertainty

Bounded uncertainty: In which all possible outcomes are deemed 'known' and we can define only quantitative probabilities require all possible outcomes of an uncertain event and each of their individual probabilities to be known.

Unbounded uncertainty: In which some or all possible outcomes are "deemed unknown".

#### 4.2. Sources of uncertainty

Uncertainty is noticeable itself at different locations in the model-based software project management process.

In model base project management process uncertainty is noticeable at different location. Which can characterized as follows:

Context and framing,

**Context:** At the initial stage of problem phase where problem is understand the model context is defined. Which include external circumstance like the technological external economic, environmental, political, social circumstances that form the context of the problem.

Input uncertainty in terms of external driving forces (within or outside the control of the software project manager) and system data that drive the model.

**Framing:**

Model structure uncertainty is conceptual uncertainty is because of partial understanding and easy descriptions of modeled processes as compared to reality.

Parameter uncertainty is the uncertainties related to parameter values.

Model technical uncertainty is the uncertainty arising from computer implementation of the model, because of numerical approximations, resolution in space and time, and bugs in the software.

From the all above sources the total uncertainty on the model simulations and model output uncertainty, can be assessed by uncertainty propagation

#### 4.3. Nature of uncertainty

Walker et al. (2003) explain that the nature of uncertainty can be categorized into:

**Epistemic uncertainty:** the uncertainty because of imperfect knowledge and which will reduce by more study or expert advice.

**Stochastic uncertainty or ontological uncertainty:** uncertainty because of inherent variability and which is not reducible by expert advice or more study.

#### 4.4. The uncertainty matrix

The uncertainty matrix can be used as a tool to get an overview of the various sources of uncertainty in a modeling study. Which will gives 'uncertainty type' in descriptions that indicate in what terms uncertainty can best be describe and the axis identifies the location or source of uncertainty and the level and nature of uncertainty.

#### 5. Methodologies for uncertainty assessment

Many methodologies and tools suitable for supporting uncertainty assessment have been developed and reported in the scientific literature. We have selected 14 methods to represent the commonly applied types of methods and tools. In the following the 14 methods are in alphabetical order:

- Data uncertainty engine (DUE)
- Error propagation equations
- Expert elicitation
- Extended peer review (review by stakeholders)
- Inverse modeling (parameter estimation)
- Inverse modeling (predictive uncertainty)
- Monte Carlo analysis
- Multiple model simulation
- NUSAP
- Quality assurance
- Scenario analysis
- Sensitivity analysis
- Stakeholder involvement
- Uncertainty matrix

#### 6. Conclusions

##### Conclusion and Feature work.

There is not much to conclude, This is early in our study, our hope is that a systematic look towards the impact of uncertainty at module level and software development methodology, which will useful to explain, state configuration and enact software engineering process for software development process, if by that way we try to specific allocated process that will be reduce causes of software failure.

It is therefore crucial that the uncertainty is introduced in the introductory phase and tracked

throughout the model study and identification, characterization of all uncertainty sources are performed jointly by the modeler, The software project manager and stakeholders in connection with the problem framing and identification of the objectives of the modeling study, which will help to developer to choose the development approaches as per level of uncertainty.

#### References:

1. Humphrey, W. A *Discipline for Software Engineering*. Addison-Wesley, 1997.
2. "Agile Software Development The Bottom Line" Rob Keefer
3. "A Comparison between Agile and Traditional Software Development Methodologies" M. A. Awad 4. B. Boehm and R. Turner, *Balancing Agility and Discipline: A Guide for the Perplexed*,
6. B. Boehm, "Anchoring the Software Process," *IEEE Software*, July 1996, pp. 73-82.
- 7 Walker, W.E., Harremoës, P., Rotmans, J., Van der Sluijs, J.P., Van Asselt, M.B.A., Janssen, P., Kraye von Krauss, M.P., 2003. Defining uncertainty a conceptual basis for uncertainty management in model-based decision support. *Integrated Assessment* 4 (1), 5e17
9. Boehm, B.; Turner, R.(2003), "Using risk to balance agile and plan-driven methods", *Computer*, Volume 36, Issue 6, June 2003 Page(s):57 - 66
10. Klauer, B., Brown, J.D., 2004. Conceptualising imperfect knowledge in public decision making: ignorance, uncertainty, error and 'risk situations'. *Environmental Research, Engineering and Management* 27 (1), 124e128.
11. Van Loon, E., Refsgaard, J.C. (Eds.), 2005. *Guidelines for Assessing Data Uncertainty in River Basin Management Studies*. Geological Survey of Denmark and Greenland, Copenhagen, pp. 182. Available on <http://www.harmonirib.com>.
12. Uncertainty in the environmental modelling process e A framework and guidance -Jens Christian Refsgaard <sup>1,2</sup>, Jeroen P. van der Sluijs <sup>3</sup>, Anker Lajer Højberg <sup>4</sup>, Peter A. Vanrolleghem <sup>5</sup>, J.C. Refsgaard et al. / *Environmental Modelling & Software* 22 (2007)
13. Iansiti, M. and A. MacCormack (1997) "Developing products on Internet time", *Harvard Business Review*, Sep-Oct.,75(5): p108-117.
14. Fowler, M. (2005) "The new methodology",
15. Highsmith, J. (2002) *The Great Methodologies Debate: Part 2*, *Cutter IT Journal*, January 2002, Volume 15, No. 1
16. Majchrzak, A., C. M. Beath, R. Lim and W. Chin (2005) "Managing client dialogues during information systems design to facilitate client learning," *MIS Quarterly*, Vol. 29 Issue 4, p653-2672.
17. *Manifesto for Agile Software Development*, 2006
18. Wastell, D. G. (1999) "Learning Dysfunctions in Information Systems Development: Overcoming the Social Defenses with Transitional Objects," *MIS Quarterly* (23:4), pp. 581-600.

#### Acknowledgements

I would like to thanks DR.Dani A.R.. for giving us valuable guidelines and suggestion regarding this work. I also like to express our sincere thanks to Dr. Murumkar C.V., Principal, Tuljaram Chaturchand College, Baramati. I would also like to thanks to all staff member of computer sci. Dept. who inspire us for this work.