

## Imprecise Software Requirements: A Software Development Risk

Kirandeep Kaur<sup>1</sup>, Rekha Rani<sup>2</sup>, Bharti Jyoti<sup>3</sup>

<sup>1</sup>(Cse, Kcceit, Nawanshahr /Ptu, India), <sup>2</sup>(Cse, Kcceit, Nawanshahr /Ptu, India),

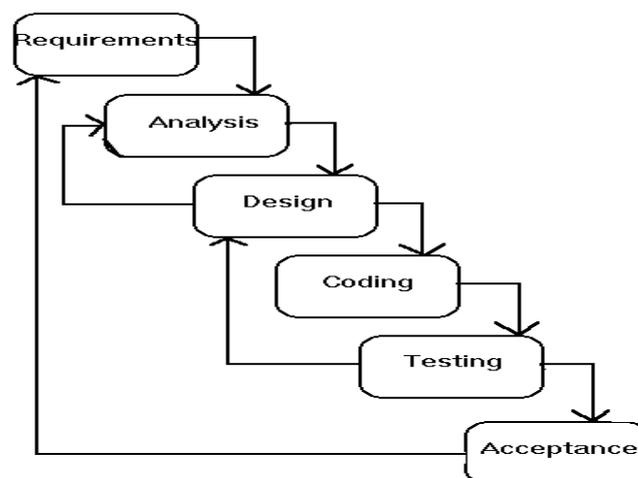
<sup>3</sup>(Cse, Kcceit, Nawanshahr /Ptu, India)

**Abstract :** Software Engineering is a profession to provide high quality software to the customers. It is a systematic approach to analysis, design, implementation, maintenance and re engineering of software. But there are many risks involved in creating high quality software like imprecise requirement gathering, poor management, gold plating, lack of proper communication within the team etc. Risks have no exact values. They are based upon uncertainties. A major contributor to project failure is the failure to spend the time at the beginning of the project to clearly define the product requirements before beginning product development i.e. gathering imprecise requirements.

**Keywords -** Software Engineering, Software Risk, Imprecise requirements.

### I. Introduction

Software Engineering defined as a systematic approach to the analysis, design, assessment, implementation, test, and maintenance and reengineering of software, that is, the application of engineering to software.



**Fig: 1 system development life cycle**

This is also known as Classic Life Cycle Model (or) Linear Sequential Model (or) Waterfall Method. This research work is based on the SDLC model's phase which is related to requirement specification and analysis. The Systems development life cycle (SDLC), or Software development process in systems engineering, information systems and software engineering, is a process of creating or altering information systems, and the models and methodologies that people use to develop these systems.

**Table 1: Software Development Phase**

Software Development Phase	Description
Feasibility study	Feasibility is the analysis of risks, costs & benefits relating to economic, technology, & user organizations.
Requirements Analysis	Defines project goals into defined functions and operation of the intended application. Analyzes end-user information needs. Important activities are: <ul style="list-style-type: none"> <li>• Completing the requirements analysis.</li> <li>• Delimiting the problem domain.</li> <li>• Roughly sketching the components of the target system.</li> <li>• Feasibility study of the planned project.</li> <li>• Creating a rough project schedule.</li> </ul>

<b>System Specification</b>	System Specification is a contract between the client & the software producer.
<b>Design</b>	Describes desired features and operations in detail, including screen layouts, business rules, process diagrams, pseudo code and other documentation.
<b>Coding</b>	The real code is written here.
<b>Testing</b>	Brings all the pieces together into a special testing environment, then checks for errors, bugs and interoperability.
<b>Acceptance</b>	The final stage of initial development, where the software is put into production and runs actual business

## II. Software Risk

Software risk involved in creating high quality software like imprecise requirement gathering, poor management, gold plating, lack of proper communication within the team etc. Risks have no exact values. They are based upon uncertainties.

**Table 2: Categories of Risks**

<b>Categories of Risks</b>	<b>Description</b>
<b>Schedule Risk</b>	Schedules often slip due to following reasons: <ul style="list-style-type: none"> <li>• Wrong time estimation</li> <li>• Resources are not tracked properly. All resources like staff, systems, skills of individuals etc.</li> <li>• Failure to identify complex functionalities and time required to develop those functionalities.</li> <li>• Unexpected project scope expansions.</li> </ul>
<b>Budget Risk</b>	Causes of Budget risks: <ul style="list-style-type: none"> <li>• Wrong budget estimation.</li> <li>• Cost overruns</li> <li>• Project scope expansion</li> </ul>
<b>Operational Risks</b>	Causes of Operational risks: <ul style="list-style-type: none"> <li>• Failure to address priority conflicts</li> <li>• Failure to resolve the responsibilities</li> <li>• Insufficient resources</li> <li>• No proper subject training</li> <li>• No resource planning</li> <li>• No communication in team</li> </ul>
<b>Technical risks</b>	Causes of technical risks are: <ul style="list-style-type: none"> <li>• Continuous changing requirements</li> <li>• No advanced technology available or the existing technology is in initial stages.</li> <li>• Product is complex to implement.</li> <li>• Difficult project modules integration.</li> </ul>
<b>Programmatic Risks</b>	These external events can be: <ul style="list-style-type: none"> <li>• Running out of fund.</li> <li>• Market development</li> <li>• Changing customer product strategy and priority</li> <li>• Government rule changes.</li> </ul>

## III. Requirement Analysis & Specification

Requirements Analysis is done in order to understand the problem for which the software system is to solve. Requirements Analysis is on identifying what is needed from the system and not how the system will achieve it goals. This task is complicated by the fact that there are often at least two parties involved in software development, a client and a developer. The developer usually does not understand the client's problem domain. The client often does not understand the issues involved in software systems. This causes a communication gap, which has to be adequately bridged during requirements Analysis.

#### IV. Imprecise Requirements

A major contributor to project failure is the failure to spend the time at the beginning of the project to clearly define the product requirements before beginning product development i.e. gathering imprecise requirements. Every requirement has to be verified before it is based lined. Requirements may contain a considerable amount of uncertainty or ambiguity that diverts the project to reach the target performance and as a result all the time and cost spent on the project go in vain. To avoid all these problems it is ideal to refine the imprecise requirements before go further on next phase of SDLC. If we will refine all imprecise requirements on the very first phase then we will be able to deliver the software to the customer which will be according to his or her requirements. So during research it has been try to find all the possible factors of imprecise requirements. For input parameters I have selected factors which lead to imprecise requirements. The factors are given:

**Table 3: Factors**

S. No.	Factors Name
1.	Unclear goals & objectives of customers
2.	Inadequate conflict management
3.	Communication gap
4.	Undetected scope creep
5.	No expert usage show up
6.	Budget constraints
7.	Lack of contextual details
8.	Inappropriate skills of analyst
9.	unrealistic expectations of the customers
10.	Poor planning

#### V. Conclusion

In Software Engineering, researchers are still working to get the more knowledge of how to gather more precise requirements from the customers. There are various factors that can arise on requirement gathering phase and they ultimately leads to gathering of imprecise requirements. There are some factors which have more adverse impact rather than others. This paper discusses list of factors associated with requirement gathering phase and their impact on software requirements is discussed in this review paper.

#### References

- [1] Shradhanand, Amarjeet Kaur, Dr. Satbir Jain, "Use of fuzzy logic in software development", Vol. 8, No. 2, pp: 238-244, 2007.
- [2] Prajakta Chandrakant Dhote, "Handling ambiguous data during requirements verification using fuzzy logic", International Journal of Computer Science and Communication, Vol. 2, No. 1, pp: 105-107, 2011.
- [3] Alexandre Bern, Satya Jaya Aparna Pasi, Uolevi Nikula, Kari Smolander., "Contextual Factors Affecting the Software Development Process – An Initial View", pp:1-8.
- [4] Abhijit Chakraborty, Mrinal Kanti Baowaly, Ashraful Arefin, Ali Newaz Bahar., " The Role of Requirement Engineering in Software Development Life Cycle", Journal of Emerging Trends in Computing and Information Sciences, ISSN : 2079-8407 , Vol. 3, No. 5, pp: 723-729, 2012.
- [5] Lachana Raimingwong, "A Review of requirements processes, Problem and models", International Journal of Engineering Science and Technology (IJEST), ISSN: 0975-5462, Vol. 4, No.06, 2012.
- [6] Subhash K.Shinde , Varunakshi Bhojane , Pranita Mahajan., "NLP based Object Oriented Analysis and Design from Requirement Specification", International Journal of Computer Applications, ISSN:0975 – 8887, Vol. 47, No.21, 2012.
- [7] Saima Amber, Narmeen Shawoo, Saira Begum., "Determination of Risk During Requirement Engineering Process", Journal of Emerging Trends in Computing and Information Sciences, ISSN: 2079-8407, Vol. 3, NO. 3, pp: 358-364, 2012.
- [8] Namrata Kapoor, Nitin Bhatia, Sangeet Kumar, "Software risk analysis using fuzzy logic", international journal of computer information system , Vol. 2, No.2, 2011.
- [9] Eric S. K. Yu and John Mylopoulos, "Understanding "Why" in Software Process Modelling, Analysis, and Design", 16th International Conference Software Engineering, pp: 1-10, 1994.