



## Spiral Model Based Object Oriented Requirement Engineering

Anandi Mahajan\*

Ph.D. Scholar

Department of Computer Science & System studies  
Mewar University, Chittorgarh (Raj.)India

Pankaj Sharma

Professor

Shri Venkateshwar Institute of Technology Indore,  
Madhya Pradesh, India

---

**Abstract** — *The fundamental principle underlying requirements engineering is the assumption that a system should be clearly specified before its design and implementation. Most requirement documents were written in ambiguous natural languages which are less formal and imprecise. Without modelling the requirement, the knowledge of the requirement is hard to be kept in a way, which can be analysed and integrated. So a model is requiring for the integration, analysing and management. Choosing suitable Requirement Engineering techniques for a particular project is a challenging and time-consuming task, requiring substantial expertise and efforts. In this paper, we propose a spiral model for the requirement engineering for the integration, analysing and manage the equipment in spiral manner because Understanding user needs is not a sequential process*

**Key words:** *ambiguous, natural languages, analysed, integrated and spiral*

---

### I. INTRODUCTION

Writing good requirements is a skill. Software requirements quality affects software product quality. For high-quality software products, software requirements must be complete. When faced with incomplete requirements, software engineers attempt to fill the requirements' gaps differently, either by getting feedback from the user or by making assumptions. Assumptions may be explicit or implicit. Explicit assumptions are preferable to implicit assumptions as explicit assumptions can be validated [1]

In order to fulfill user's requirement, it is very important to manage requirement effectively. It involves different disciplines such as psychology, languages and communications, organizational behavior and management [2]

The role of customers and other stakeholders is becoming increasingly significant during requirement engineering activities. Methods of eliciting requirements are now more co-operative. There are many techniques to obtain requirements from customers. Selecting the right techniques according to the characteristics of the project is very important. In some complex problems, combination of requirement engineering techniques should be applied for efficient and successful requirement engineering process [3].

Requirement engineering is a crucial activity, which can affect the entire life cycle of software development project. The main objective of requirement engineering is to collect requirements from different viewpoints such as business requirements, customer requirements, user requirements, constraints, security requirements etc. Information is also one of the important requirements of requirement engineering process to develop quality and updated software [4].

It is widely acknowledged that most of problems occurring in requirement engineering activities mainly results from the lack of a semantic agreement among stakeholders. It implies to find an effective mechanism to integrate existing theory, techniques and resources accumulated in requirement engineering, and furthermore to provide a favorable base to develop novel theory and techniques. This paper proposes an integrated framework for semantic requirement engineering, and points out several interesting research avenues[5].

The researchers gradually realize that the requirements engineering processes are very complex and the related technologies are in a great variety. Thus, in order to analyze and compare different types of requirements engineering processes more deeply, a simple, clear, unified description method of requirements engineering process is needed [6].

Current trends are making effective requirements practices even more important. First, the demand for high quality and reliable software is growing rapidly. Second, we see an increasing number of projects being developed with different suppliers having a multitude of stakeholders with different needs and ambitions that have to cooperate to achieve a shared objective [7].

Requirement engineering is the most effective phase of software development process. It aims to collect good requirements from stakeholders in the right way. It is important for every organization to develop quality software products that can satisfy user's needs. Therefore, it becomes necessary to apply requirement engineering practices in every phase of software development process. In requirements engineering process model to produce quality requirements for software development. Requirement management and planning phase is executed independently for an effective management of requirements. It is iterative in nature for better requirement engineering and later maintenance. The successful implementation of proposed requirement engineering process can have a good impact on the production of quality software product. [8].

## II. CLASSIC/TRADITIONAL TECHNIQUES

**Interviews:** Interviews is the common and popular method used by the requirement engineers to elicit system requirements and comprehend objectives of the system through verbal conversation with the stakeholders

**Surveys:** The survey techniques are used to get large set of requirements from a larger population that may scattered on disparate geographical locations. Surveys collect information from large number of users and it is quite economical and rapid to analyze the data through planned surveys.

**Questionnaires:** The questionnaire is a method of requirement elicitation which is simple and requires lesser time and cost. To get precise results, the questionnaire should be clear, concise and structured to obtain genuine user requirements, objective and constraints. However, this technique lacks in the mechanism to seek users' clarification on the topic.[9]

**Ethnography:** Ethnographer discerns people during their work. This technique is superior for those users work. This technique is superior for those users to-face meetings as they cannot specify their requirements verbally. The observations of ethnographer may be direct or in direct through any electronic device, like, camera.

**Focus Groups:** The main focus of this technique is to discuss the prototype of the software by different people. The needs, requirements and pros and cons of the prototype are noticed by these concerned people, having different domains and experiences in the related field.

**Task Analysis:** This technique entails constructing top-down tasks hierarchy of the system to find out the knowledge used or required in the development of the system. Using this hierarchy, the task and sub-tasks are placed at different levels in a tree structure.

**Introspection:** Introspection is a preparatory step in requirement elicitation where requirement engineers use their experience and expertise to acquire requirements of the stakeholders in terms of their expectations towards the new system. However, this technique mainly necessitates requirement analysts to have a massive experience in this area. It is very effective when analysts are well-known of the domain and goal of the system as well as experts in business processes that users ordinarily perform

## III. SPIRAL MODEL

This model takes the form of a spiral, and each round can be connected, last phase connected with the first one. The initial round is concerned with system feasibility, the second round with system requirements specification, the next round with system design, etc. The basic idea of the Spiral development process is that the developers should only define the highest priority features in each round and implement those features, then get feedback from users and customers. Based on this feedback, developers then go back to define and implement more features in the later stages of development.

To obtain high quality products along with higher productivity, it is necessary to carefully elicit, specify, analyze and manage software requirements. This not only simplifies system design and implementation but also reduces the number of defects that are identified later in the implementation stage[10].

This model considers the different phases of software requirement engineering as membranes and the various requirements are considered as objects and the requirement engineering process [11] phases are modeled as spiral model. The key process activities in the spiral model like generates requirement, feasibility of requirement, integrate requirement, analysis of requirement, problems in requirement, and requirement negotiation are considered in the model.

Most requirements are written in natural languages and represented in less structured and imprecise formats, including requirement phase. Artifacts created in phases of software life cycle are required to be modeled and integrated, so the traceability, consistency, and completeness can be ensured [12].

It is the optimization and improvement of traditional requirement engineering, getting it fit to the continuous changes of requirements [13]. In contrasts of Spiral model based Object Oriented requirement engineering and the traditional one, we can continuous changes of requirements if needed.

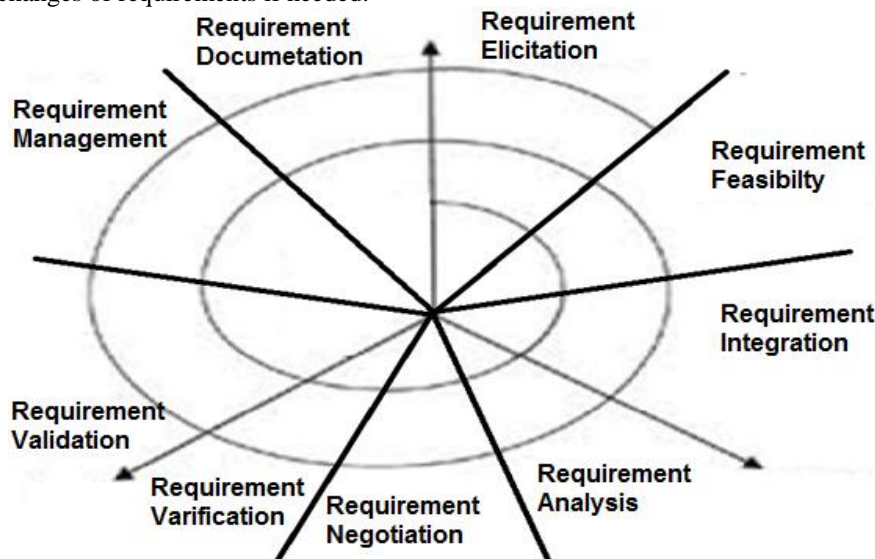


Fig.1 Spiral model of Requirement Engineering

The article concentrates on ways of expressing requirements rather than ways of generating them. As software becomes more complicated and large-scale, users' demands are becoming more varied, and the expectation level of a software product is becoming higher. Therefore, it is very important that a software engineer analyzes the user's requirements precisely and applies them effectively in the development step.

Various proposed approaches for object oriented requirements engineering framework majorly varies on the representation of the requirements objects and on the degree of deployment of object oriented paradigm

It defines the construct of a good requirement, provides attributes and characteristics of requirements, and discusses the iterative and recursive application of requirements processes

Spiral model in object oriented requirement engineering has the following phases:

The above spiral model consists nine phases:

1. Requirements elicitation. Requirements are elicited from different stakeholders
2. Requirement feasibility. To check the requirements are feasible or not.
3. Requirement integration. To integrate the requirements.
4. Requirements analysis. Requirements are analyzed.
5. Requirements negotiation. The problems identified are negotiated with the stakeholders
6. Requirements verification. Requirements are verified
7. Requirements validation. Requirements are validation to ensure that the requirements are real needs of stakeholders and are well defined.
8. Requirements documentation. Requirements are documented in appropriate notations.
9. Requirements management. Requirements management is highly related to requirements change management in this model and includes requirements traceability.

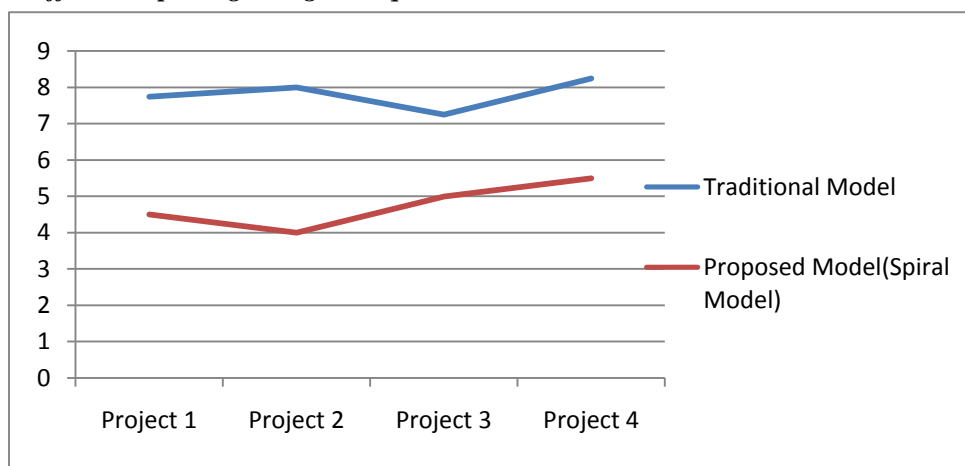
#### IV. TRADITION VS SPIRAL MODEL BASED REQUIREMENT ENGINEERING

TABLE 1 (COMPARISONS BETWEEN TRADITIONAL AND SPIRAL MODEL BASED REQUIREMENT ENGINEERING)

	<b>Traditional</b>	<b>Spiral Model</b>
<b>Planning</b>	<i>Predictive</i>	<i>gradually</i>
<b>Orientation</b>	<i>Process Oriented</i>	<i>Object(People) Oriented</i>
<b>Documentation</b>	<i>Comprehensive and constant</i>	<i>Minimal and evolving</i>
<b>Requirement Gathering</b>	<i>Waterfall</i>	<i>Spiral</i>
<b>Quality Control</b>	<i>Heave planning and strict control with late heavy testing</i>	<i>Continuous control of requirement and gathering requirement with cautiously testing</i>
<b>Risk Analysis</b>	<i>None</i>	<i>Rigorous</i>
<b>User Interaction</b>	<i>Less</i>	<i>More</i>
<b>Software failure rate</b>	<i>Higher</i>	<i>Less</i>
<b>Testing</b>	<i>Perform in single phase</i>	<i>Perform in multiple phase</i>
<b>Project Success rate</b>	<i>Low as compare to proposed model</i>	<i>Very High</i>

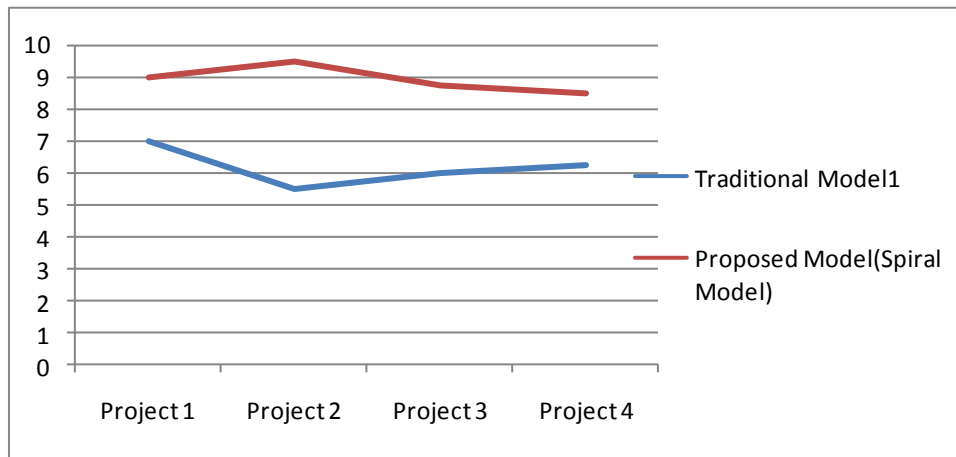
We also apply this model in four different software projects and compare the result on the following parameter:

##### 1. According to efforts in updating/change in requirements :



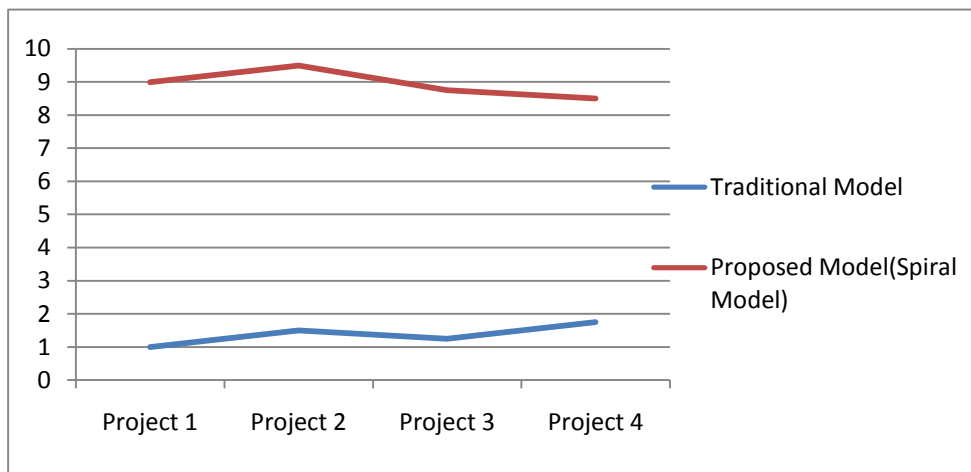
As we seen in the project 1, Project 2, Project 3 and Project 4 efforts in updating/ change requirement in tradition model are 7.75, 8, 7.25 & 8.25 respectively and in our proposed model 4.5,4,5 & 5.5 respectively. So we observe that our proposed model is need less efforts in updating/ change requirement as compare to traditional model.

2. *Reliability of requirement*



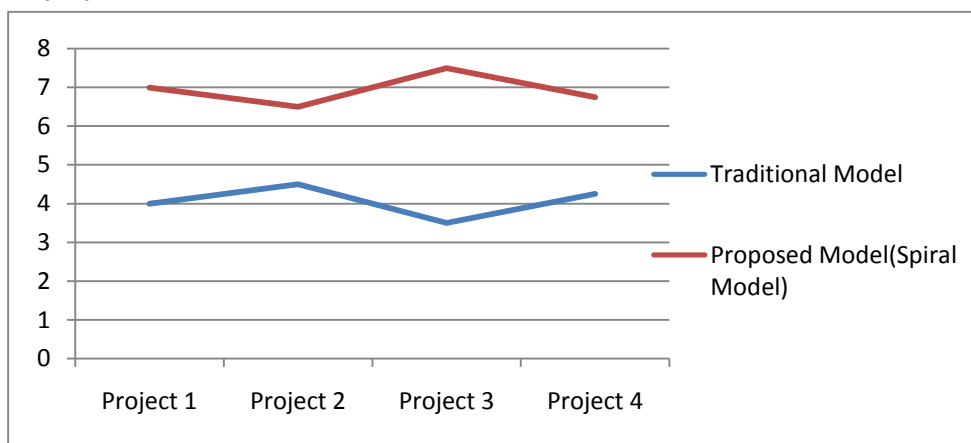
As we seen in the project 1, Project 2, Project 3 and Project 4 Reliability of requirement in tradition model are 7,5.5,6 & 6.25 respectively and in our proposed model 9,9.5,8.75 & 8.5 respectively. So we observe that our proposed model is more reliable as compare to traditional model.

3. *Extensibility of requirement*



As we seen in the project 1, Project 2, Project 3 and Project 4 as per extensible parameter in tradition model are 1, 1.5, 1.25 & 1.75 respectively and in our proposed model 9, 9.5, 8.75 & 8.5 respectively. So we observe that our proposed model is more extensible in requirement as compare to traditional model.

4. *Success rate of software:*



As we seen in the project 1, Project 2, Project 3 and Project 4 as per success rate parameter in tradition model are 4, 4.5, 3.5 & 4.25 respectively and in our proposed model 7,6.5,7.5 & 6.75 respectively. So we observe that our in proposed model success rate is very high as compare to traditional model.

## V. CONCLUSION

This model of the requirement may be useful in the development of software and quality of software too. This approach overcomes the main problem of traditional requirement models that lack a well-founded and precise method. The aim of our research is to find a suitable RE process to be applied in the organization for the development of software projects. We believe that our experience can be understood and used by other researchers.

## REFERENCES

- [1] Berry, D.M.; Lawrence, B.; , "Requirements Engineering," Software, IEEE , vol.15, no.2,pp.26-29,March-April1998
- [2] Sen, A.M.; Hemachandran, K. "Elicitation of Goals in Requirements Engineering Using Agile Methods "Computer Software and Applications Conference Workshops (COMPSACW), 2010 IEEE 34th Annual 2010 , Page(s): 263 - 268
- [3] Runde, S.; Fay, A.; Wutzke, W.-O. "Knowledge-based Requirement-Engineering of building automation systems by means of Semantic Web technologies", 2009. 7th IEEE International Conference on Topic(s): Computing & Processing (Hardware/Software) ; Power, Energy,& Industry Applications Digital Object Identifier,2009, Page(s): 267 - 272
- [4] Ranganathan, P.; Magel, K., "Understanding requirement engineering (REQ) from a software agent modeling perspective" Software Engineering and Service Sciences (ICSESS), 2010 IEEE International Conference on Topic(s): Computing & Processing Hardware/Software) Page(s): 83 - 85
- [5] Zhuozhi Chen; Pooley, R. "Rediscovering Zachman Framework Using Ontology from a Requirement Engineering Perspective" Computer Software and Applications Conference, 2009. COMPSAC '09. 33rd Annual IEEE International Volume: 2 Topic(s): Computing & Processing (Hardware/Software) 2009 , Page(s): 3 - 8
- [6] Chih-Wei Lu; Chu, W.C.; Chili-Hung Chang; Ching Huey Wang "A Model-based Object-oriented Approach to Requirement Engineering (MORE) " Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International Volume: 1 Topic(s): Communication, Networking & Broadcasting ; Components, Circuits, Devices & Systems ; Computing & Processing (Hardware/Software) ; Power, Energy, & Industry Applications ; Robotics & Control Systems ; Transportation ,2007 , Page(s): 153 - 156
- [7] Bin Ling and Yinshan Tang and Kecheng Liu and Sun, L. and Cheong, A. and Neale, K., Service Operations and Logistics, and Informatics, 2008. IEEE/SOLI 2008. IEEE International Conference on Requirements engineering for software in emerging markets. ISSN={1090-705X},2008, volume=1,pages=1148 -1152,
- [8] Mishra, D.; Mishra, A.; Yazici, A., "Successful requirement elicitation by combining requirement engineering techniques"\_Applications of Digital Information and Web Technologies, 2008. ICADIWT 2008. First International Conference on the Topic(s): Communication, Networking & Broadcasting ; Computing & Processing (Hardware/Software) 2008 , Page(s): 258 - 263
- [9] Heba Elshandidy, Sherif Mazen, "Agile and Traditional Requirements Engineering: A Survey"International Journal of Scientific & Engineering Research, Volume 4, Issue 9, September-2013 ISSN 2229-5518IJSER © 2013<http://www.ijser.org>
- [10] Emilio Infran • Gary Chastek • Patrick Donohoe • Julio Ce'sar Sampaio do Prado Leite, "Requirements engineering in software product line engineering"Published online: 6 November 2013\_ Springer-Verlag London 2013Requirements Eng (2014) 19:331–332
- [11] Jac Ky Ang; Sook Bing Leong; Chin Fei Lee; Yusof, U.K., "Requirement engineering techniques in developing expert systems" Computer & Informatics (ISCI), 2011 IEEE Symposium on Topic(s): Communication, Networking & Broadcasting ; Computing & Processing (Hardware/Software) ,2011 , Page(s): 640 - 645
- [12] Pandey, D.; Suman, U.; Ramani, A.K. , "An Effective Requirement Engineering Process Model for Software Development and Requirements Management"\_Advances in Recent Technologies in Communication and Computing (ARTCom), 2010 International Conference on Topic(s): Computing & Processing (Hardware/Software) ; General Topics for Engineers (Math, Science & Engineering) , 2010 , Page(s): 287 - 291
- [13] Haron, Azlena; Harun, Mazlan; Naz'ri Mahrim, Mohd; Sahibuddin, Shamsul; Zakaria, Nor Hawaniah; Abdul Rahman, Noorihan, "Understanding the Requirement Engineering for organization: The challenges",Computing Technology and Information Management (ICCM), 2012 8th International Conference on Volume: 2 Topic(s): Communication, Networking & Broadcasting ; Components, Circuits, Devices & Systems ; Computing & Processing (Hardware/Software) ; General Topics for Engineers (Math, Science & Engineering) , 2012 , Page(s): 561 - 567