



Framework for Requirement Engineering Process

Swarnalatha K S¹, GN Srinivasan², Lavanya N³, Namitha D⁴, Prathibha RK⁵, Raunak Kasera⁶

¹Assistant Professor, ²Professor, ^{3,4,5,6}Student

^{1,3,4,5,6}Department of Computer Science & Engineering, RVCE, Bangalore, India

²Department of Information Science & Engineering, RVCE, Bangalore, India

Abstract: A proper framework of the requirement engineering process is very important for the success of this project. All the stages in requirement engineering must be complete and they must blend into each other to form a fruitful model. This model establishes the basic structure with the help of which all the software requirement processes can be handled, irrespective of the process in particular. The proposed model stipulates a very convenient structure for the requirement engineering process, through which a better understanding of the project as well as better results are yielded at the end. Considering market trends and demand will help in the collection of the right requirements among the numerous other feasible options. Customer involvement followed by the information on company policy improves the elicitation process. StakeRare technique is used for accurate and efficient collection of requirements. Requirement traceability matrix is used as the tool for validation of requirements. The validation of the requirements assures the perfection of requirements. The specification of functional and non-functional requirements and the behavior of the system to be developed are stated. After obtaining all this information, an operational prototype of the project is developed, and the requirements are finally validated. In conclusion, all the requirements attained are managed and stored.

Keywords: Requirement engineering, requirement gathering, requirement management, traceability matrix, StakeRare, market trends.

I. Introduction

The proper framework for a requirement engineering process model is very crucial. Software systems come and go through a series of passages that account for their inception, initial development, productive operation, upkeep, and retirement from one generation to another [1]. The existing models referred to in this paper are water fall model and spiral model. The existing model provides a good framework but it can be upgraded. Inefficient collection of requirements is often the sole reason for the failure of many projects. Methods superior than conducting interviews and surveys should be used. The existing model can be amended to strengthen the requirement engineering process, in turn leading to success of project. Better requirement collection techniques should be adopted and every stage should be completely understood so as to ensure that the project is going in the correct direction. The proposed model shown in Figure 1 deals with the following phases

- Market trend
- Customer expectation and strategy
- Strategy plan and company principles
- Requirement gathering
- Filtration
- Operational prototype
- Validation
- Change requirements and Management

Understanding the market trend is the first stage. The demands in the market are first identified. The software that is being developed should be able to meet these demands. The next stage involves the customer. The requirements of the customer are taken into account; interaction with the customer is essential. Then the company policy and strategies are specified and the do-able projects of the particular organization or group are advertised. The next stage, dealing with the actual collection of requirements, involves efficient communication with the stakeholders in order to find out the appropriate requirements needed for the project. A better understanding of all the collected requirements can be achieved by this which is essential in a software process. Next an analysis is done to determine the feasibility of the project with the currently gathered requirements and these are then filtered. This is a significant stage in which the requirements are thoroughly sieved and the end products are comprised of only the efficient and necessary requirements. After this, based on the information collected in the previous stages, a baseline prototype is developed. A plan is crafted which gives the description of technical tasks to be performed, risks involved and so on. The validation stage helps in identifying the inappropriate requirements and checks the correctness of those collected so far. At the end, any unfit requirement, if present, is removed/replaced and all the details are acknowledged.

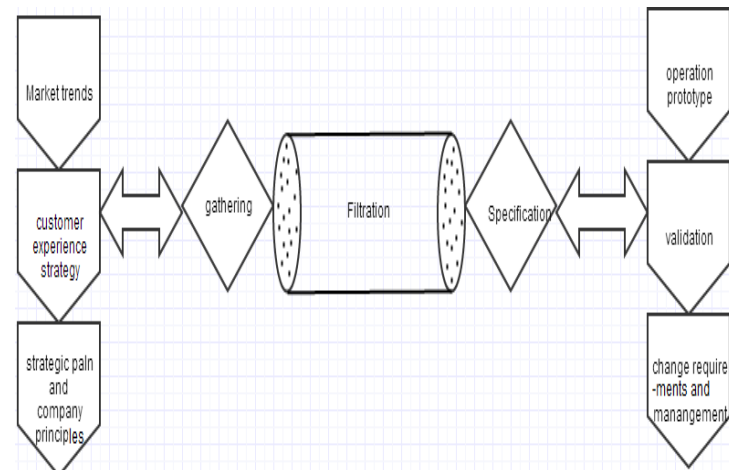


Figure 1: Framework for requirement engineering process model

II. Market trend

“The right choice makes all the difference”. Identifying market trends is a significant phase in software requirement engineering. Today the business market is evolving rapidly [21]. Hence it is essential to keep track of latest marketing demands and customer needs in order to utilize technology as a strategy advantage for one’s company. Propensity of a particular market to progress in a particular direction is called as Market trend [22]. Knowledge of these trends helps in building better tools, models and methodologies to meet the company goals. This study also deals with checking the viability of the project with the currently assembled requirements. A range of tests are conducted on an initial model of the software to test its performance under a variety of states. The performance is then evaluated and it is determined whether a requirement is viable or not.

The determining factors for the feasibility of a requirement include

- Economic viability[3]
- Current market scenario
- Maintainability
- Performance statistics
- Dynamic Nature
- User Friendly
- Robustness
- Technical feasibility

III. Customer expectation and strategy

The system must keep track of customer response to their products, and should identify what is most important to its customer, be it relevance, convenience, quality products or services [23]. Apart from that the customer may also wish for verity or selection and protection. Customer satisfaction is directly related to their expectations. Granting extra features would only have an impact on customers if these features are based on their exact needs [24]. The customer may have expectations based on opinions from friends, family, and work colleagues, or from articles, blogs and various sites about user experiences, or even their own prior experience. Customer service experience affects future services because the customer might convey his/her experience to others.

IV. Strategy plan and company principles

Company principles are the basic plan of action of the company, intended to ensure that all the requirements of the customers or users are met, along with the legal requirements, which has a favorable impact on the company-client relationship [25]. Requirements are gathered based on the company principles. Strategic planning is the means through which a company establishes its goals, and plans are formulated to achieve those goals [26]. It is analytic in nature. Strategic planning assists in requirement elicitation to assist in planning the whole process. It is a necessary step to be carried out before the requirement gathering process [27]. The organization or the company developing the software system specifies the features or the characteristics of the application which it is about to provide [10]. This phase helps the organization to attract the appropriate stakeholders with the appropriate requirements. It gives the customers and the end user the basic idea of the software system being developed. It helps the customers understand the product that is being developed.

V. Requirement gathering

Requirement gathering deals with the collection of raw materials that are essential for the development of the

software. It is the most crucial stage in requirement engineering process. We can adopt various methods for efficient requirement collection. To achieve this, interaction with the users of the software, the stakeholders as well as select technical experts is of paramount importance. Efficient communication between the software developers and the target customers is a key factor to the collection of valuable requirements [2]. Accumulating relevant data from the cautiously selected stakeholders, customers and so on, will be the deciding factor in what the software finally delivers. The requirements from the stakeholders can be drawn by the process of interviews, questionnaires or by conducting surveys [4]. Different parties have myriad sets of varying requirements and so it is essential to determine the relevant ones and to design a prioritizing scheme. Efficient communication is a must. High priority can be assigned to those requirements which are more popular among users and stakeholders alike. Now that all the requirements are gathered and prioritized, we move to the next step, which is practicality study.

VI. Filtration

This is a stage where the categorization of feasible requirements takes place. The requirements are bifurcated based on the sector in which the software is used, the constraints it satisfies, the dimension of its operation, and so on. The tool used here is called StakeRare, which incorporates social networking and collaborative study. A stakeholder is a person, group, organization, member or system that affects or can be affected by an organization's actions [6]. The first step is to identify the stakeholders. The project scope is identified and then the stake holders are categorized based on their role and influence in the project. This method then solicits the stakeholders to recommend others, and then builds a social network with these stakeholders as nodes and their recommendations as links. It prioritizes stakeholders using a variety of social network measures to determine their project influence [7]. This prioritization is carried out with the help of various social network measures. Prioritized stakeholder roles and prioritized stakeholders are recognized [8]. Next the initial requirements are identified and the rating is given to them and recommendations are made using collaborative study. The filtering of one large set of requirements is done using collaborative filtering. Finally accurate requirements can be obtained with StakeRare method.

VII. Software requirement specification

Software requirement specification (SRS) is a comprehensive description of the behaviour of the system to be developed. It also includes the use cases which describe the interaction of the users or customers with the system that is being developed [10]. It essentially describes the requirements of the customer with respect to the system. It also describes in natural language the functions and capabilities of the software system being developed. It contains functional and non-functional requirements, and not the design suggestion [11]. After feasibility study and filtering of the requirements, the specification is developed. It acts as a blueprint for the software in the latter stages of its development. The goals of an SRS are that it should provide feedback to the customers, it should decompose the problem into several component parts, it should serve as an input to the design specification, and it should act as a parent document [12]. It mainly consists of overall description, interfaces, functional and non-functional requirements and user characteristics. The purpose of the specification is to act as a baseline for evaluating the software system. The requirement specification may be written by the procurer, bidder, selected developer or the independent RE contractor. Software specification should not include product development plans, product assurance plans and designs.

VIII. Operational Prototype

Prototype is an existing process or model developed to test objectives, functionalities and development of a process. Evolutionary, throw away, incremental and extreme prototype are the different types of prototypes used. The incremental or iterative software development approaches lead to faster delivery of more useful software. But these approaches have some drawbacks like management problems, contractual problems, validation problems and maintenance problems. System prototyping plays an important role in helping software developers with implementation. Objectives of incremental development and prototyping are different. The objective of the prototype is to validate the system requirements. The objective of the incremental approach is to delivering a working system to end users, and prototyping has a short lifetime compared to the incremental approach [13]. The first phase is to identify the basic requirements. The second phase is to develop an initial prototype for the collected requirements. The customers evaluate the prototype and the developers get the feedback from customers. After getting the feedback, the prototype is enhanced and developed. This prototyping is known as throwaway prototyping. It is neither maintained by the developer nor delivered to the customer, but is mainly used in the requirement engineering process, system design process and testing process. The supporting nature of a system for work can be seen by prototyping. The weak and strong points of software and requirements can be obtained by this approach [14]. Benefits of using this approach are that usability of the system can be improved, and improved design quality and maintainability can be obtained. This helps reduce development efforts [13].

IX. Validation

A. Traceability matrix

Among the various requirements mentioned by the users, the software developer may also mention some suggestions. Keeping track of all the requirements gathered is a daunting task. Hence a standard tool for putting the requirements together in an organized way is required, and this tool is the Requirement Traceability Matrix [9]. This matrix is a link that connects all the requirements together throughout the software development lifecycle process. This tool is used for error recovery in the software development phase. In the traceability matrix, all the feasible requirements are written in the rows and the corresponding acquaintance of these requirements with the various stages of software development are specified along the columns. In detail, the columns contain requirement ID, requirement description and the test cases. Next is to map the requirements to the test cases. If there is a relation between any requirement in a particular row and the corresponding column, then the intersecting cell is marked. Each 'X' represents a point where functionality needs to be included in the system to deliver the Requirement [28]. This matrix can be created using automated tools like MS Excel. The concept of Requirements traceability could help in analyzing the impact and inconsistency among derived requirements or work products due to requirements changes. Requirements can also be mapped against objectives if required [28]. Figure 2 shows the structure the traceability matrix and figure 3 gives an example of the traceability matrix.

Requirement ID	Requirement description	TC 001	TC 002
SR-1.1	User should be able to do this	x	
SR-1.2	User should be able to do that	x	

Figure 2: Structure of Traceability matrix

	Requirements	Subsystems				Layers						Tiers				
		Administration	Publishing	Scheduling	Expenses	KPI Dashboard	Presentation	User Interface	Business Logic	Data Access	Services	Persistence	Client	Web	Application	Data
SR 1.1			x				x	x	x	x		x	x	x	x	x
SR 1.2			x				x	x	x	x		x	x	x	x	x
SR 1.3			x						x	x	x	x		x	x	x
SR 2				x			x	x	x		x		x	x	x	

Figure 3: Example of Traceability matrix

Requirement traceability matrix is used to see that all the requirements are cohesive. This is of great help in validating them, since each requirement undergoes a test. It is also used to identify the flaws in the functionality of any specific requirement, thus providing a chance for correction and improvement.

X. Change requirements and Management:

Requirements are evaluated in the validation phase. If they are found to be inappropriate, then they can be changed by this phase using requirement management tools to meet customer needs. Management is the one of the processes used to handle requirements. It involves the requirements being stored in a database and are retrieved as and when required. There are different types of software documentation. Management can be achieved using requirement citation [15]. It acts as a foundation of what the software system will do. It has different purposes. Requirements may come from customers, organization or end users. [16]. It identifies the attributes, capabilities, characteristics and quality of the software system to be developed. Software changes become more challenging, error-prone and time consuming without requirement recording [17]. The need for requirement citation is based on the type of software system developed. If the software has low life expectancy, then little recording is sufficient [18]. If the software is developed by more than one development team, citation is required for communication between the teams [19]. Requirement management tools are used to manage the documentation of the software system whose requirements are changing and are complex. It serves as an agreement between the customers and the developers on what the software system does [20]. It is necessary if the development of software system takes more than a month, if more than one person or more than one team works on it.

References

- [1] Process Models in Software Engineering: Walt Scacchi, Institute for Software Research, University of California, Irvine.
- [2] Communication Challenges in Requirements Elicitation and The Use of the Repertory Grid Technique, Christopher J. Davis, University of South Florida Saint Petersburg Saint Petersburg, Florida 33701.

- [3] Techniques for requirement elicitation, Joseph A. Goeman.
- [4] Enhancement of requirements traceability with state diagrams Soonsongtane, S. Dept. of Comput. Eng., Chulalongkorn Univ., Bangkok, Thailand.
- [5] H. Sharp, G.H. Galal, and A. Finkelstein, "Stakeholder identification in the requirement engineering process,"
- [6] Proc. Database and Expert System Application Workshop, pp.387-391, 1999.
- [7] StakeRare: Using Social Networks and Collaborative Filtering for Large-Scale Requirements Elicitation Soo Ling Lim, and Anthony Finkelstein, Member, IEEE.
- [8] International Conference on Information Systems and Computing (ICISC-2013), INDIA. Requirement Elicitation Based Collaborative Filtering Using Social Networks P. Deepika, P.S. Smitha.
- [9] Cleland-Huang J, Settini R, Duan C, Zou XC. Utilizing supporting evidence to improve dynamic requirements traceability. In: Proc. Of the 13th IEEE Int'l Requirements Engineering Conf. (RE 2005). Washington: IEEE Computer Society, 2005. 135-144.
- [10] Pressman, Roger (2010). Software Engineering: A Practitioner's Approach. Boston: McGraw Hill. p. 123. ISBN 9780073375977.
- [11] "DI-IPSC-81433A, Data Item Description Software Requirements Specification (SRS)". everyspec.com. 1999-12-15. Retrieved 2013-04-04.
- [12] Stellman, Andrew and Greene, Jennifer (2005). Applied software project management. O'Reilly Media, Inc. p. 308. ISBN 0596009488.
- [13] John Crinnion: Evolutionary Systems Development, a practical guide to the use of prototyping 9. within a structured systems methodology. Plenum Press, New York, 1991.
- [14] Hoc, J.-M. and Nguyen-Xuan, A. Language semantics, mental models and analogy. J.-M. Hoc et al., Eds. Psychology of Programming. Academic Press. London, 1990, 139-156, cited through Brad A. Myers, John F. Pane, Andy Ko, Natural programming languages and environments.
- [15] Multiple (wiki). "Code documentation". Docforge. Retrieved 2012-11-13.
- [16] Woelz, Carlos. "The KDE Documentation Primer". Retrieved 15 June 2009.
- [17] Jump up^ Microsoft. "Knowledge Base Articles for Driver Development". Retrieved 15 June 2009.
- [18] Herbsleb, James D. and Moitra, Dependra. ' In: IEEE Software, vol. 18, no. 2, pp. 16-20, Mar/Apr 2001.
- [19] Rakitin, Steven. , "Manifesto elicits cynicism." IEEE Computer, vol. 34, no. 12, p. 4, 2001.
- [20] Prause, Christian R., and Zoya Durdik. Architectural design and documentation: Waste in agile development? In: International Conference on Software and System Process (ICSSP), IEEE, 2012.
- [21] Edwards, R.; McGee, J.; Bessetti, W. H. C. (2007). Technical Analysis of Stock Trends. CRC Press. ISBN 978-0-8493-3772-7.
- [22] Preis, Tobias; Stanley, H. Eugene (2011). "Bubble trouble: Can a Law Describe Bubbles and Crashes in Financial Markets?". Physics World 24: 29-32.
- [23] Mintzberg, Henry and, Quinn, James Brian (1996). The Strategy Process: Concepts, Contexts, Cases. Prentice Hall. ISBN 978-0-132-34030-4.
- [24] Rumelt, Richard P. (2011). Good Strategy / Bad Strategy. Crown Business. ISBN 978-0-307-88623-1.
- [25] Porter, Michael E. (1980). Competitive Strategy. Free Press. ISBN 0-684-84148-7.
- [26] Drucker, Peter (1954). The Practice of Management. Harper & Row. ISBN 0-06-091316-9.
- [27] Kiechel, Walter (2010). The Lords of Strategy. Harvard Business Press. ISBN 978-1-59139-782-3.
- [28] Suzanne & James Robertson: A factor present in every successful project and absent in every unsuccessful project is sufficient attention to requirements.