



# A Survey on Software Requirement Engineering for Real Time Projects based on Customer Requirement

Swarnalatha k s<sup>1</sup>, GN Srinivasan<sup>2</sup>, Meghana Dravid<sup>3</sup>, Raunak kasera<sup>4</sup>, Kopal Sharma<sup>5</sup>

Asst Professor, Department of Computer Science & Engineering, RVCE, Bangalore, India<sup>1</sup>

Professor, Department of Computer Science & Engineering, RVCE, Bangalore, India<sup>2</sup>

Student, Department of Computer Science & Engineering, RVCE, Bangalore, India<sup>3</sup>

Student, Department of Computer Science & Engineering, RVCE, Bangalore, India<sup>4</sup>

Student, Department of Computer Science & Engineering, RVCE, Bangalore, India<sup>5</sup>

**Abstract:** Requirement engineering is the important phase of software development process. It basically aims to collect meaningful and well defined requirements from clients in the proper way . It is important develop quality software that can satisfy user's needs without errors. Requirements engineering for software development process is a complex exercise that considers software requirement demands from a number of roles and responsibilities. Therefore, it becomes mandatory to apply requirement engineering practices in every stages of software development process. In this paper, we propose a dynamic framework for requirements engineering process model to produce better requirements for any software development. Requirement management, planning and maintenance phase are executed separately for an effective management of software requirements. The successful implementation of proposed requirement engineering process can have a good impact on the production of quality and quantitative software product.

**Keywords:** Requirement engineering, requirement analysis, requirement elicitation, requirement development, requirement management.

## I. INTRODUCTION

Requirements phase are important and essential phase of which we will able to discover before building any software products based on client requirement from the formal or informal documents. A well defined requirement is software functionality that satisfies clients needs. There exists a interrelationship between human's and machines for requirement gathering that can help to produce quality and quantitative software products . software Requirements are classified as functional and non-functional [1] requirement. A functional requirement is a software requirement that specifies an particular action to be performed by a system without considering any physical constraints. Non-functional software requirement specifies system properties such as environmental condition and implementation constraints, scalability platform development dependencies, maintainability, extensibility, reliability ,documentation. [1].

Requirement engineering is commonly accepted to be the most important ,critical and complex process within the software development of real time technical systems [3, 4, 5]. In this way, it helps to describe a some of the multidisciplinary role of requirements engineering process

as well as the patterns for social interaction and behaviour . The main reason of the requirements engineering process have the high impact on the capabilities of the emerging software product. In general, software requirements engineering is a process in which most diverse set of software product that demands from the group of clients, which is already being considered by the practitioners. These two reasons can make software requirements engineering a critical system. Requirement engineering is a practical and systematic approach through which the software or system engineer collects functional or non functional requirements from different customers and design and develop them into the quality software development processes. Requirement engineering activities cover the entire development life cycle.

Requirements engineering process is an process which also specifies the requirements management is understood as an aspect of requirements engineering process phase [9, 10, 11]. Traditionally, requirements engineering is performed in the beginning of the system development lifecycle [13]. However, in critical and complex systems development, developing any accurate set of requirements from client that



would remain constant throughout the development has been analysed to be impossible in practice [14]. Therefore, requirements engineering is an incremental and iterative process, performed in parallel with other software development activities such as design, implementation, testing and documentation. Requirements engineering phase mainly contains a group of activities for discovering, analysing, documenting, verification, validating, maintaining and documenting a set of requirements from customers for a system [6]. Requirements engineering is divided into two main set of activities; namely, software requirements development phase and software requirement management and document phase. software requirement development mainly covers set of activities for discovering, analysing, documenting, verification and validating requirements where as software requirement management phase commonly includes activities related to traceability feature and dynamic change management of software requirements. software Requirements verification and validation consists of those activities that will confirm that the product of a system development process meets its technical specifications and non technical specification. Software Requirements validation consists of activities that will confirm that the behaviour aspect of the developed system meets its client requirement [7].

Requirement engineering is a significant and profound activity, which can affect the development of entire activity of software development phase. Software Requirement engineering is one of the most profound and accuracy tools for gathering requirements from customer, which is concerned with analysing, developing and documenting the software requirements [8]. We propose an dynamic framework model of requirement engineering process for software development life cycle.

## II. SOFTWARE REQUIREMENT ENGINEERING PROCESS

The main goal of software requirement engineering is to discover and gather the quality requirements from the customer that can be designed and implemented into software development. The identified requirements from the stakeholder must be clearly defined, consistent, modifiable, dynamic and traceable to produce a quality and quantitative software product that could be delivered to the clients. In this paper, we have proposed an dynamic framework requirement engineering process model. It consists of mainly five stages, namely; Software requirement elicitation, design and development, documentation of functional and non functional requirements, Software validation and verification of requirements, and Software requirement analysis and allocation and flow down of requirements described by the clients.

Documentation of functional or non functional requirements includes identification of requirements and software and system requirement of a formal or informal specification

from the clients. Software validation and verification of requirements phase is mainly concerned with conforming the documented requirements specified by the clients. The software requirement management planning phase that controls the dynamic changing requirements from the clients . The proposed requirement engineering process specifies the requirements engineering for software development systems in which requirement engineering must be a part of the software development life cycle process.

### A. Software Requirements Elicitation and Development

Software requirement elicitation and development phase mainly distinctness on examining and gathering particular requirements and objectives for the system from different points of view ie customer, users, constraints, system's operating environment, trade, marketing and standard . Software requirements elicitation phase begins with identifying clients of the system and collecting raw requirements including functional and non function requirements from various points of view. Raw requirements are requirements that have not been clearly analysed and have not yet been noted down in a well-formed software requirement notation and sequence diagram. The elicitation phase mainly focuses to collect various requirement from various point of view that are business requirements, stakeholders requirements, user requirements constraints, information and security requirements, and some standard requirement . Typically, the specification of system requirements starts with observing and interviewing people [15].

Furthermore, Customer requirements are often fail to interpret because the system analyst may fail to misunderstand the user's requirement. In addition to functional or non functional requirements gathering, standards and constraints play an crucial role in system and software development. The development of requirements may be determined, which is shown in figure 1. The software requirement engineering is a process of collecting or gathering requirements from clients and various environment in a systematic approach.

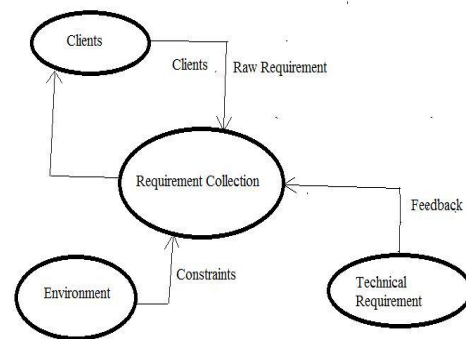


Figure 1 : Requirement Collection



The system analyst collects raw functional or non-functional requirements and then performs detailed analysis design and receives feedbacks. Thereafter, these outcomes of the project are compared with the technicality as well non technicality of the system and produce the quality and necessary requirements for software development [3, 29].

1) *Software Requirements Analysis*: The development and gathering of good quality and quantitative requirements is the important and complex activity of any organization to develop good quality software products. These software requirements are then rigorously analyzed within the particular context of user or business requirements. It is also noticed that the identified raw functional or non functional requirements may be incompatible[29]. Therefore, The analyzed requirements need to be maintained and documented to enable imparting with stakeholders and future documented of software requirements and the system. Software requirements analysis also redefines the software allocation and builds effective models of the process, data, and dynamic behavioral domains that may be treated by the software. Requirement Prioritizing the software requirements is also part of software requirements analysis phase.

2) *Dynamic Allocation and Flow-down of Requirements*: The main purpose of requirements allocation and flow-down is to achieve objectives of the project. Top-level system requirements need to be organized and defined hierarchically to the changing requirement that can assist to view , manage information and dynamic changing of requirement at different levels of abstraction. The functional or non functional requirements are clearly further decomposed down to the level at which the requirement can be designed, developed and tested. Thus, dynamic allocation and flow-down may be performed for several hierarchical levels. The level of detail increases as the work proceeds down in the changing hierarchy level. System-level requirements are general in nature, while requirements at abstraction levels in the hierarchy are very specific [17, 7]. The top-level system requirements defined in the system requirements development phase are the complex input for the dynamic requirements allocation and flow-down phase. Allocation and flow- down of requirements include:

a) *Dynamic Allocation of Requirements: Dynamic*

Allocation of requirements is an system architectural task performed in order to design the structure of the system or software and to issue the top-level functional or non functional requirements to subsystems. Architectural models provide the viewpoints for defining interaction between applications program and subsystems to meet the design requirements of the system described by the clients. The main goal of system architectural modeling is to define a

robust and reliable dynamic framework within which applications program and component subsystems may be designed and developed [15]. Each system level functional or non functional requirements is allocated to one or more elements at the next level in the hierarchy. dynamic Allocation mainly includes allocating the non-functional non functional requirements to system elements described by the clients. Each system element will need an apportionment of performance requirements [14, 18, 19, 10]. When functional and the non-functional requirements of the system have been dynamically allocated then the system / software engineer can create a model that represents the interrelationship between system elements and sets a foundation for dynamically changing requirements analysis , design , development and testing steps.

b) *Control Flow-down of Requirements: Control*

Flow-down composed of taking down requirements from the clients for the lower hierarchy level elements in response to the allocation of the resources. When a system functional requirement is allocated to a subsystem, the subsystem must have at least one functional or non functional requirement that will responds to the changing requirement allocation. The lower hierarchy level requirements either may closely having the higher level or may be very different if the system engineers recognize a capability that the lower hierarchy level element must have to meet the higher-level requirements[14]. Derived requirements are functional requirements that must be imposed on the subsystems. These functional requirements are derived from the systems decomposed phase. There are two subclasses of derived functional requirement, i.e. subsystem requirements and subsystem interface requirement. The subsystem requirements are the requirements from the stake holders that must be applied on the subsystems but do not provide a direct benefit to the end user or clients. Subsystem Interface requirements are the requirement that will arise when the subsystems need to communicate with one or another subsystem to accomplish an proper result. This outcome is needed to share data ,power ,useful computing algorithm and changing development environment. [17].In the dynamic allocation and control flow-down phase, requirements identification and completeness have to be ensured both to higher and lower level requirements as well as between requirements on the same level in the hierarchy. In theory, it produces a system in which all elements and sub components are completely balanced or optimized. In the real world scenario , complete balance is achieved only due to schedule ,fiscal constraints of the system [11, 18]. Dynamic allocation and control flow-down starts as a transdisciplinarity activity, i.e., subsystems may contain software, hardware and mechanics resources. Initially, they are considered as one subsystem of the system.



### *B. Documentation of Requirements*

A formal document is prepared after collecting functional and non functional requirements, which contains a entire representation of the external and dynamic behavior of the complex software system. Software Requirements development process is the complex and important activity which determines functionality and non functionality requirements of the system will be performed by software.

Non-functional requirements are clubbed together with functional requirements into the software requirements specification with the help of control flow-down, dynamic allocation, and derivation phase. A software requirements specification will be exists for each software subsystem, component or software configuration item, is part of this phase [11]. The documentation of requirements mainly consists of functional requirement identification phase and Software requirement specification.

1) *Functional Requirements Identification:* Functional requirements identification phase main activity to assignment of a unparalleled identifier for each requirement prescribed by the stakeholders[6]. These unparalleled identifiers are used to refer requirements during product design, development ,verification and validation ,documentation and maintained. Functional Requirements identification process consists of three basic sub activities. The basic numbering activity includes particular numbering and non-particular numbering whereas functional identification activity includes describing the requirement, construct a structure based functional identification and emblematic identification. The last technique is to support and automate the management of items, which includes dynamic and unparalleled renumbering, database record and functional identification and base lining requirements of the system [6, 22].

2) *Software Requirement Specification:* Software requirements specification document is made after the successful identification of the functional and non functional requirements. The document includes the software product to be delivered emphatic than the process of its software design and development. Software requirement specification is an powerful tool for requirement specification which gives a complete view of the changing behavior of the system or software . It consists a group of use cases diagram that indicates all the complex interactions that users will have with the software [12]. In addition to use case diagram , the SRS mainly focuses on supplementary requirements. Non-functional requirements are part of software requirements which forces the constraints on the design or development. SRS is a complete description of the purpose and environment constraints for software under development phase. The SRS completely describes what is the input to the

software and how it will be expected to produce the proper outcome. An SRS reduces the time and effort required by software developers to achieve prescribed goals and also reduces the overall development cost. A SRS specifies how an application will interact with system software ,system hardware other programs and users in a variety of real-world situations. Parameters includes operating speed of the software , response time of the software , portability of the software with other machines, maintainability cost are evaluated in SRS.

### *C. Software Requirements Verification and Validation*

When the entire functional and non functional requirements are identified and specified in the requirement specification, then different clients involved have to agree upon its nature. One should ascertain that the defined and correct requirements are validated and both the requirements are verified . Software validation and verification activities include validating the system functional and non functional requirements against raw requirements specified by the stakeholders and verifying vigorously for the correctness of system requirement documentation phase. The most common techniques for validating functional requirements are requirements reviews with the clients, and prototyping according to the requirements. Software requirements need to be validated against system hierarchy level requirements and requirement specification needs to be verified properly. Software Verification includes correctness of software developed, consistency in the software, unambiguousness and understand ability of non functional requirements. In software requirement verification and validation, Functional or non functional requirements traceability mechanism can generate an audit trail between the software requirements prescribed by the clients and tested code. The possible outcome of the software requirements development phase is a formal document includes the agreed software requirements [28].

### *D. Software Requirement Management and Planning*

Software Requirements Management and planning phase identifies and keeps track of all the changes of agreed software requirements, interrelationships between software requirements, and dependencies between the functional and non functional requirements documents and other formal documents produced during the systems and software engineering process model [9]. It is a uninterrupted and inter related-section process that begins from software requirement management planning and unbroken activities of identifying and dynamic change control during and after software requirements development process phases. Software Requirements management is a uninterrupted activity that can perform after development and maintenance phase because requirements may continue to change over the period [9, 21].





Software Requirement management is sometimes considered as the most critical and complex activity in requirements engineering process [24]. As requirement changes dynamically can have a huge impact on the developing system, which is very hard to estimate. For every change in the requirement, costs and re-development work are to be considered before approving the change in the requirement. After refinement of functional requirements, the requirement can be further organized into software development processes phases. Software Requirements management tools have been developed to manage stable and unstable requirements change and large amount of data are also collected during requirements engineering process phase [16].

### III. DISCUSSION

The proposed requirement engineering process is more effective to produce quality requirements. The other requirement engineering processes are limited to cover only few dimension of requirement engineering such as requirement elicitation, requirement specification and requirement verification and validation [10]. The proposed model introduces all important and hidden aspects of requirement engineering process. The existing requirement engineering process models are unable to communicate their phases with software development process in the right manner [10, 29]. We relate all the important aspects of requirement engineering process to software development process in order to find out good requirements from various sources that can be implemented into software development process for producing quality software products. We have also relate requirement management and planning phase to the software development phases in our model because requirements can change over the time and during software development process, this can give bad outcomes. Therefore, it is necessary to manage continuously changing requirements through requirement management and planning.

### V. FUTURE ENHANCEMENT

The proposed requirement engineering model can be used in larger software development process, where the requirements are continuously changed. It presents the new insight of requirement management and planning which can manage the changing requirement. The requirement engineering activities in the model such as elicitation, documentation of requirement and verification & validation are tightly interconnected to the software development phases. It can help to recover and modify the requirement, whenever requirements are changed in any phase of software development. Software developer can easily supply the required and modified requirements using requirement management and planning activity. Using this model, software developer can design the software product, which

will be able to avoid and manage changing requirements.

The proposed model is iterative in nature, which is useful in requirement prototyping with more user participation. Using this model, organization can allow users for any further change in requirements during software development process. The SRS can be modified accordingly and these changed requirements can be re-implemented in software development.

### VI. CONCLUSION

Requirements engineering is the initial phase of software engineering process in which user requirements are collected, understood, and specified for developing quality software products. The requirement engineering process deserves a stronger attention in the industrial practices. In this paper, we proposed an effective requirement engineering process model for software development that can be used for software development processes to produce a quality product.

### REFERENCES

- [1]. P. Jalote, An Integrated Approach to Software Engineering, 3<sup>rd</sup> edition, Narosa Publishing house, India, 2005.
- [2]. G. Ropohl, "Philosophy of Socio-technical Systems", In Society for Philosophy and Technology, 1999, pp. 59-71.
- [3]. Pandey, U. Suman, A. K. Ramani, "Social-Organizational Participation difficulties in Requirement Engineering Process-A Study", National Conference on Emerging Trends in Software Engineering and Information Technology, Gwalior Engineering College, Gwalior, 2009.
- [4]. N. Juristo, A. M. Moreno & A. Silva, "Is the European Industry Moving Toward Solving Requirements Engineering Problems?" IEEE Software, 2002, pp. 70-77.
- [5]. S. Komi-Sirvio & M. Tihinen, "Great Challenges and Opportunities of Distributed Software Development - An Industrial Survey", Fifteenth International Conference on Software Engineering and Knowledge Engineering, SEKE2003, San Francisco, 1-3 July 2003.
- [6]. J. Siddiqi, "Requirement Engineering: The Emerging Wisdom", IEEE Software, 1996, pp.15-19.
- [7]. G. Kotonya & I. Sommerville, Requirements Engineering: Process and Techniques. John Wiley & Sons, 1998.
- [8]. J. D. Saylor, System Engineering: An Introduction. IEEE System and Software Requirements Engineering, IEEE Software Computer Society Press Tutorial. IEEE Software Society Press, 1990.
- [9]. Pandey, U. Suman, A. K. Ramani, "Design and Development of Requirements Specification Documents for Making Quality Software Products" National Conference on ICIS, D.P. Vipra College, Bilaspur, 2010.
- [10]. R. H. Thayer & W. W. Royce, Software Systems Engineering, IEEE System and Software Requirements Engineering. IEEE Software Computer Society Press Tutorial. IEEE Software Society Press. Los Alamos, California, 2012.
- [11]. W. W. Royce, "Managing the Development of Large Software Systems" Proceedings of IEEE Wescon, Reprinted in Proceedings 9<sup>th</sup> International Conference Software Engineering IEEE Computer



- Society Press, Los Alamitos, California, USA,  
Nov 2012, pp. 328-338.
- [12]. M. Dorfman, System and Software Requirements Engineering  
IEEE System and Software Requirements Engineering, IEEE  
Software Computer Society Press Tutorial, IEEE  
Software Society Press, Los Alamos, California,  
1990.
- [13]. S. W. Ambler, Process Patterns. Building Large-Scale Systems  
Using Object Technology. Cambridge University Press,  
1998.
- [14]. P. Parviainen, H. Hulkko, J. Kaariainen, J. Takalo & M.  
Tihinen, "Requirements Engineering", Inventory of  
Technologies, VTT Publications, Espoo, 2003.
- [15]. Leffingwell & D. Widrig, Managing Software Requirements - A  
Unified Approach, Addison-Wesley, 2003.
- [16]. D. Nelsen, System Engineering and Requirement Allocation, IEEE  
System and Software Requirements Engineering, IEEE Software Computer  
Society Press Tutorial, IEEE Software Society Press Los Alamos,  
California, 2013.
- [17]. R. S. Pressman, Software Engineering, A Practitioner's  
Approach. Third Edition, McGraw-Hill Inc, 1992.
- [18]. S. Blanchard & W. J. Fabrycky, Systems Engineering and  
Analysis. Prentice-Hall, 1981.
- [19]. S. Lauesen, Software Requirements: Styles and Techniques,  
Addison-Wesley, 2002.
- [20]. H. Berlack, Software configuration management. John Wiley &  
Sons, 1992.
- [21]. O. Gotel, Contribution Structures for Requirements Traceability.  
Ph.D. Thesis, Imperial College of Science, Technology and Medicine,  
University of London, 2013.
- [22]. M.E.C Hull, K. Jackson & A. J. J Dick, Requirements  
Engineering. Springer-Verlag, Berlin, 2013.
- [23]. Hooks & K. Farry, Customer-Centred Products, Amacom, 2001.
- [24]. M. Lang & J. Duggan, "A Tool to Support Collaborative  
Software Requirements Management" Requirements Engineering Journal  
6(3), 2012, pp. 161-172.
- [25]. T. Gilb, Competitive Engineering. Addison-Wesley, 2003.
- [26]. Paivi Parviainen, Maarit Tihinen, Marco Lormans & Rini van  
Solingen, Requirements Engineering: dealing with the Complexity of  
Sociotechnical Systems Development, 1998.
- [27]. L. A. Macaulay, Requirements Engineering, Springer-Verlag,  
1996.
- [28]. Rong Fa Tang, Xiao Yu Huang, "The Application of  
Requirement Engineering Model in Large Software Development Process",  
Advanced Materials Research, 2012, volume 267, pp 193-198.
- [29]. Dharendra Pandey, U. Suman, "An Effective Requirement  
Engineering Process Model for Software Development and Requirements  
Management" International Conference on Advances in Recent  
Technologies in Communication and Computing, 2012, pp 287-291.