Software Engineering Requirement Model

Pardeep Kumar, Aisha khan, Abbas Ahmad, Kamran Khalid, Muhammad Wasim

Abstract The paper deals with essential inclusive coverage of the fundamentals and basics of requirements engineering model. Requirements engineering (RE) pacts with the range of requisites that must be encountered by a software system within an organization for that system to yield better results. With that description in mind, this research presents a disciplined methodology to the engineering of high-quality requirements. Helping as a helpful introduction to the fundamental concepts and ideologies of requirements engineering, this paper also deal with comprehensive review of the goal, scope, and role of requirements engineering as well as paramount practices and defects to avoid. In addition, case studies and related data are comprised that accompanied many examples provided in the paper to demonstrate how the designated method and practices are applied in applied situations.

Keywords Requirements, model, concept, engineering, spiral, developmental stages, elicitation

1. INTRODUCTION

Requirements engineering deals with the procedure of describing, specifying, detailing as well as upholding necessities to the sub-fields of systems in various disciplines of engineering, including software engineering that is concerned with this process. Requirement engineering is normally acknowledged as the most perilous and multifarious process in the evolution and progress of social and technical systems. Consequently, it alleviates to express a multidisciplinary role in Engineering of requirements route and furthermore the outlines for social interaction and welfare of mankind. [1] A requirement is a representation and description of a technique, potential or restrain. Requirements predominantly emphasize on what an organization must do, rather than how it should perform it. A requirement must be “testable”. Classes of Requirements include functional and non-functional constraints Priority of the requirement comprises of essentials, highly desirables and desirable but low priority.

2. OVERVIEW

The most significant phase in the software advancement lifecycle is the requirement model. [2] This process deals with activities that consists of gathering, analysis, validation, documentation and certification after that

preservation/maintenance of the requirements for a structure and execute them in the software development method. Requirements are the properties, i.e. attributes or capabilities that is essential be controlled by the system or system constituent that should be conveyed, to fulfill customer requirements[3].Therefore, requirements engineering has a foremost effect on the capabilities of the product resulting from this software development practice. In addition, requirements engineering analyzes relatively miscellaneous product exigency from a vast amount of different stakeholders (e.g., viewpoints, roles, responsibilities, and objectives). These objectives cause requirement engineering to be a precarious and compound process. The proposed requirements engineering distinguishes functional and non-functional requirements [2] Functional necessities identify endeavor that should be accomplished by a system whereas non-functional requirements label system potentials like performance, transferability and maintainability. The engineering of requirement model is normally accompanied at the beginning the developmental lifecycle. Evidently, one must know what capabilities a system must obtain before starting the actual development. However, the practice taught us that in large and complex system development, especially those with a long term lifecycle, requirements won't remain stable or unchanged. For this reason, we contemplate requirement engineering as continuing incremental and iterative procedure that agreement with fluctuating requirements throughout the whole development lifecycle. The requirement engineering deeds can be gathered below the requirements development and requirements management procedures. The requirements developmental process comprises accomplishments concerning the detection, analysis, documentation and validity of requirements, while requirements management pacts with the traceability and altered management of requirements.

3. OBJECTIVE OF RESEARCH

The objective of this study is to distinguish requirements which can be executed in development of software. The notorious requirements must be vibrant, well defined, consistent, adjustable and perceptible to generate a valuable product.

4. STRUCTURE OF THE PAPER

This research paper is qualitative in nature in which Section 1 of paper highlights the introduction of the requirement engineering and model to be discussed Section 2 and 3 elucidates objectives and review from other authors and literature. Section 4 illuminates in detail the whole process after it conclusions discussed.

5. WHAT IS REQUIREMENT ENGINEERING

Requirements engineering is a software engineering process traditionally executed in the initial stage of software development which includes the deeds regarding the detection or provocation, documentation and protection of the requirements for a software system. Since the 1990s, requirements engineering has originated into universal use next to the publication of [4] on system and software requirements engineering. The appropriate phases constituting the requirements engineering practice depend strongly on the variety of software or the context of the developing organization. The engineering of requirement process of software product companies, for example, varies considerably from that of customized software [5]. In addition, several categories in requirements and thus requirements engineering processes are distinguished for example the authors of the scrutinized process also worked on methodology of information requirement engineering [2] and security requirement engineering. However, phases oriented towards requirements elicitation (cf. first phase in the PDD depicted in figure 2), specification (see the SRS idea in the PDD), validation and management are regularly encountered in literature [6] and they are all – in one way or another included in the engineering of requirements process suggested by Pandey et al. (2010). The authors of the proposed requirement engineering process claim that their process is more effective in producing quality requirements (Pandey et al., 2010) compared to other processes, which have a lack of effective coherence and communication with the software development process [7] or have a lack of comprehensiveness and do not cover all the required dimensions of requirement engineering [8]

The engineering of requirement process proposed by Pandey et al. (2010) adopted some requirement engineering fundamentals described in the work of [3] which defines the requirement as a state or proficiency which should be encountered or controlled by a method to gratify a description or additional officially executed document. In addition, Jalote also classifies requirements into functional and non-functional requirements. Authors of the process do not clearly remark suitability for a definite software development context. Market compelled development for example, fluctuates suggestively from the development of modified software, hence the existence of the notion of market driven requirement engineering [5]. The process suggested by [2] does not report the potential relation with a planning procedure and in addition, stakeholders in MDRE are more intricate and diverse. The processes concerning requirements prioritization and scope change management are located within the release planning procedure area of the reference outlined by [9] emphasizes the tight integration of the engineering of requirement happenings inside the market driven settings, and indicates that the requirements process model suggested by Pandey et al. (2010) is suitable primarily for the development and progress of client specific software systems. These are some of the ins and outs why requirement engineering for market focused development require specific approach [10] However, besides the lots of differences in requirement engineering magnitudes between market driven- and customer specific development, we also identify several parallels according to Van De Weerd et al., 2006). We haven't found any publications that describe the application of the requirement engineering process by Pandey et al. (2010). However, the authors of the process elaborated a requirements modeling framework (Pandey et al., 2011a), which analyzes modern requirements modeling techniques. Several of these techniques could be assimilated into specific activities of the requirement engineering process analyzed in this paper.

6. REQUIREMENT ENGINEERING PROCESS- A SPIRAL MODEL

In this paper, operative requirement engineering model, is explained in detail. It comprises of primarily four stages, requirement elicitation as well as development, moreover documentation of the requirements, authentication in addition confirmation of requirements, as well as requirement management, execution, evaluation and planning. The requirement elicitation as well as development phase comprises credentials of analysis for requirement, provision plus flow down of the requirements. Documentations for requirement comprises other off details of the requirements. Validation as well as verification of the requirements stage is believed to be associated with conforming and imitating the standard and written records of requirements. As the requirement management, evaluation in addition to planning phase regulate and manipulate continuously fluctuating requirements. Every deed is again sub divided.

The Model of Spiral is an amalgamation of the waterfall model as well as iterative model. Hence every stage in the spiral model commences with a certain design objective then culminate with the customer appraising the final progress. Barry Boehm started the spiral model which was published in year 1986. The developmental group of Spiral-SDLC classical model begins with the minor set of the requirement and passes via each developmental stage for individuals requirements. [9] Hence development team comprises of functionality for each-increasing spirals till the solicitation is prepared and being prepared for the manufacture stage. Spiral Model is not as renowned as any other SDLC (Software Development Life Cycle) models for instance Scrum or the Kanban as well as Spiral Model that can be expensive to avail in addition doesn't work sound for the small or medium plans. It is the risk-driven model with meaning that the general success of the project greatly hinges on to the risks investigation phase. As risk investigation entails definite proficiency on each and every iteration. Therefore, to evaluate as well as analyze the ongoing project, special expertise and proficiencies are required.

At a first spectacle, it seems that model is complex and lumbering, in addition there are not any specific causes to think through this approach is considered one best options to choose. But, like the other models of SDLC, besides its drawbacks, it has its distinctive solid edges. For instance, there's opportunity to increase and augment some extra functionality in later product developmental stages. Subsequently risk checking as well as regular proficiency are essential and chief features of this methodology of RE since the overall project becomes more transparent. [1]

6.1 REQUIREMENTS ELICITATION AND PROGRESS

Requirement elicitation as well as development and collecting anticipated requirements and purposes for the method from the different lookouts and perspectives (e.g., clients, users, restraints and difficulties in system's operating setting, trade, advertising as well as standard etc.). The requirements elicitation stage initiates with the ascertaining and determining stakeholders of organization as well as assembling raw requirements as of various viewpoints. Underdone requirements are necessities that may have neither been scrutinized nor investigated before writing down in attractive requirement representation. Elicitation phase targets to collect diverse lookouts for instance professional requirements, client requirements, consumer requirements, safekeeping requirements, data requirements, and standards etc. Usually, specification of organization requirements initiates with perceiving as well as cross-examining people. [5] Moreover, consumer requirements are frequently misjudged because the system specialist may misconstrue the needs of user. Whereas requirements principles and constraints in process conduct significant part in the framework improvement. Consequently, the advance of the necessities can be circumstantial. It is additionally seen that the necessity building method is accumulating requirements from the customer as well as environment in an organized way. The system expert gathers all the rare requirements plus then it executes detailed analysis also collects responses. After it these results are compared with the adequacy of this framework and after that deliver commendable and fundamental necessities for the product development.

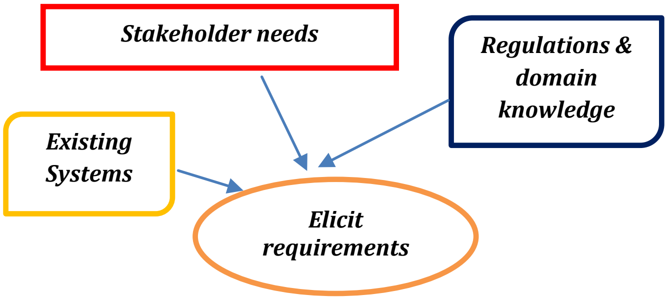

Figure 1 Eliciting Requirements Source: developed by Authors

6.2 REQUIREMENTS ANALYSIS

Development and congregation of the virtuous worthy requirements are the rudimentary needs of every organization to progress and produce quality software goods. Requirements are thoroughly viewed, identified and then evaluated within the framework of the business requirements be helpful in future and will benefit the whole company and can also perceive that the recognized requirements should not be inconsistent. For this sake arbitration, agreement, communication as well as ordering of the requirements become imperious steps of the prerequisite investigation Evaluated underdone prerequisite must be acknowledged and essential to be written to have better communication with the stakeholders, consultants as well as future conservation of the requirements of the system. Evaluation of requirements too improves the software provision as well as make mockups for whole process, data, as well as negotiating areas that must be overseen by framework. The product necessities are then prioritized according to prescribed method as it is likewise an imperative piece of programming in prerequisites investigation steps. [1]

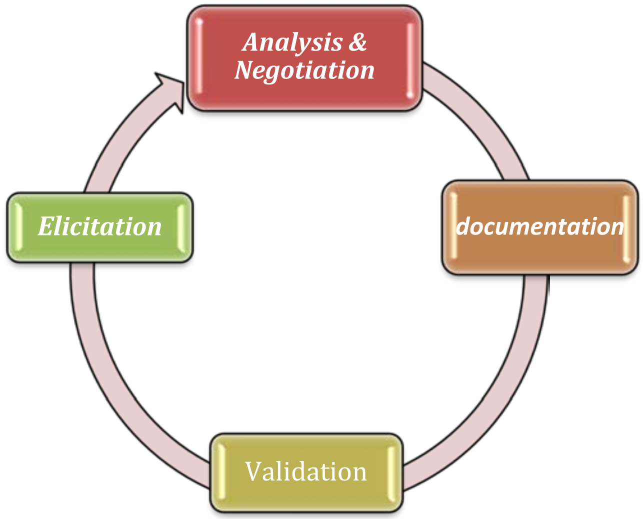

Figure 2 -- A Spiral Model Source Developed by Researchers

6.3 DISTRIBUTION OF REQUIREMENTS

Distribution is carried out to ensure that all desires of the framework are satisfied through a subsystem or via another set of subsystems that work jointly to attain purposes. The Greatest dominant system requires to be ordered hierarchically by a method that it can view as well as accomplish information at diverse phases of abstraction. These requirements are fragmented to an extent that at which they can be deliberated and investigate. Thus, flow-down can be attained for numerous ranked stages. The points of interest of each stage ascend as the work continues in the characterized arrange. Supplies in system levels have general characteristics, whereas hierarchy is precise at low levels.

6.3.1 ALLOCATION OF REQUIREMENTS

Architectural chore is the allocation of requirements which approved out to design and plan the assembling of the framework and to distribute the largest amount of framework demands to subsystems. These models convey the procedures for characterizing boundaries between systems and subsystems to encounter the requests provided by the framework. A healthy framework requires Architectural modelling inside which applications and constituent subsystems may be progress. Every framework level prerequisite is allotted to at least one section at the following level. Designation also likewise incorporates designating the non-functional prerequisite to the framework components. Each system fragment requires a division of non-functional necessities.

6.3.2 REQUIREMENT’S FLOWDOWN

Flow-down comprises of necessities for the lower level segments in answer to the dispersion. At the point when a framework prerequisite is assigned to a subsystem. The low-level prerequisites either may nearly look like to the larger amount or may be altogether differently perceive a capacity that the lower level component essentially need to encounter the high-level requirements. On the subsystems consequent requirements are requirements must be executed. From the systems putridness process these requirements are derived. These two sub-classes of resolute requirement, i.e. subsystem necessities and interface requirement. The subsystem requirements are the condition which must be enforced to the subsystems themselves though did not unavoidably give an immediate assistance to the end user. [7]

6.3.3 DOCUMENTATION OF REQUIREMENTS

After collecting need, a formal and appropriate report is generated, which contains an aggregate illumination of the outside conduct of the product framework. The action of figuring out which functionality of the framework will be proficient by programming is known as Requirements improvement technique. With the assistance of flow down, the software necessities process distribution, and induction are assimilated with functional requirements are known as Non-functional requirements. For individual software or configuration item, component a software necessities detail will be built up, and it is part of this phase. Requirement identification and requirement specification is the part of credentials of requirements. [2]

6.3.4 REQUIREMENTS IDENTIFICATION

The allocation of a distinctive identifier for each requirement is focused by requirements identification practices. Requirements throughout product development, enhancement and managing are referred by unique identifiers. Requirements identification strategy contains of three sub exercises. The essential numbering moment incorporates huge numbering as well as non-creating numbering while identification action involves of naming, framework centered recognition and authentic distinguish proof.

6.3.5 REQUIREMENT SPECIFICATIONS

After the fruitful identification of prerequisites, the need specification document is produced. Rather than the process of its development, the item to be conveyed is the describe document. A powerful instrument for prerequisite determination which is an entire depiction of the conduct of the framework or programming to be produced is known as Software requirement specification (SRS). It incorporates a set of cases that define all the associations that operators have with the system software. Notwithstanding use cases, the SRS also comprises of non-functional fundamentals. Requirements which execute constrictions on the project or application is known as non-functional prerequisite. The envisioned persistence and condition for programming below development is explained by the SRS. What the software will organize and how it will be predictable to accomplish is entirely defines by SRS.

6.3.6 REQUIREMENTS CONFIRMATION AND AUTHENTICATION

When diverse parties involve must agree upon its nature then whole requirements are pronounced and identified in the SRS. One ought to discover the right prerequisite are expressed (authentication) and these requirements are expressed correctly (confirmation). Endorsement and substantiation exercise incorporates approving the framework prerequisite in logical inconsistency of crude prerequisite and checking the precision of framework prerequisite attestation. The requirements review with the stakeholders, and prototyping are the most well-known practices for legalizing necessities. Software necessities must be validated in logical inconsistency to system level prerequisite and SRS should be properly checked. [9]

6.3.7 PREREQUISITE MANAGEMENT AND PLANNING

Planning and management stage of Requirements supervised and find out the variations of settled requirements, associations among prerequisite, and addictions among the requirement's documents and additional papers formed throughout the framework and software engineering course. That is basically a persistent and cross-sectional method which commence from formation of management and undergoes events of identification, documentation and supervised of distinctions throughout and subsequently in progression of requirements development stages. It is an unceasing action that can be achieved after development phase and in maintenance stage as requirements may change. Traceability is a piece of prerequisite administration, which implies capability to define as well as follow its lifecycle and relation with other objects in forward and backward direction equally. [7]

7. CONCLUSION

The problems configuring, a significant aspect of software requirements engineering, though lessens the project performance and output to be gained. According to some authors and researchers SR is often a delusion and biased process of decision making. Prerequisite engineering is considered as the underlying phase of software engineering in which firstly, prerequisite of users is gathered, understood, and quantified for unindustrialized eminence products. Its processes justify a more grounded thought in the engineering, industrial and manufacturing practices. Into this research artical, authors have tried to propose an operative requirement engineering advanced model for the development of software to produce valuable products.

8. REFERENCES

[1] T. Ur Rehman, M. N. A. Khan, and N. Riaz, "Analysis of Requirement Engineering Processes, Tools/Techniques and Methodologies," International Journal of Information Technology and Computer Science, vol. 5, no. 3, pp. 40–48, Feb. 2013.

[2] D. Pandey, U. Suman, and A. K. Ramani, "An effective requirement engineering process model for software development and requirements management," in Advances in recent technologies in communication and computing (artcom), 2010 international conference on, 2010, pp. 287–291.

[3] P. Jalote, “Software Requirements Analysis and Specification,” in An Integrated Approach to Software Engineering, Springer, New York, NY, 1997, pp. 73–158.

[4] M. Dorfman and R. H. Thayer, “Standards, guidelines and examples on system and software requirements engineering,” IEEE Computer Society Press Tutorial, Los Alamitos: IEEE Computer Society Press, 1990, edited by Dorfman, Merlin; Thayer, Richard H., 1990.

[5] R. H. Thayer and M. Dorfman, System and software requirements engineering. IEEE Computer Society Press, 1995.

[6] B. Regnell and S. Brinkkemper, “Market-driven requirements engineering for software products,” in Engineering and managing software requirements, Springer, 2005, pp. 287–308.

[7] C. R. Coulin, “A situational approach and intelligent tool for collaborative requirements elicitation,” PhD Thesis, Université Toulouse III-Paul Sabatier, 2007.

[8] I. Van De Weerd, S. Brinkkemper, R. Nieuwenhuis, J. Versendaal, and L. Bijlsma, “Towards a reference framework for software product management,” in Requirements Engineering, 14th IEEE International Conference, 2006, pp. 319–322.

[9] AAsa G. Dahlstedt, L. Karlsson, A. Persson, J. NattochDag, and B. Regnell, “Market-driven requirements engineering processes for software products–a report on current practices,” in International Workshop on COTS and Product Software: Why Requirements Are So Important (RECOTS), 2003.

[10] P. Wongthongtham, E. Chang, T. S. Dillon, and I. Sommerville, “Ontology-based multi-site software development methodology and tools,” Journal of Systems Architecture, vol. 52, no. 11, pp. 640–653, Nov. 2006.

About the authors

Pardeep Kumar is software engineer Email: Pardeepkumartech@gmail.com.

Aisha khan is currently pursuing master’s degree in urban planning From UET Lahore aishakhan036@gmail.com.

Abbas Ahmad is Software Engineer. Email: Itsabbasahmad@gmail.com

Kamran Khalid is expert of Telecommunication and networking. Email: Kamran.ghalzai@gmail.com

Muhammad Wasim has his Masters in GIS. Email: giswasim@gmail.com