Requirement Engineering Issues and Their Solutions

Muhammad Waqas Boota, Naseer Ahmad, Abdul Hye Masoom

***Abstract*— we know that Requirement Engineering (RE) is the first phase of the software engineering process. In this first phase user requirements are accumulated and specified. Many software failures initiate due to lack of software requirement specifications. In this paper we discuss the basic issues of requirement engineering and discuss their solutions from different point of view. We also discuss the impact of these issues and how we fix them. we discuss the issues about poor requirement quality, use case modelling, missing requirement, requirement verification, requirement validation, requirement management , requirement process, tool support and requirement engineer.**

***Index Terms*— software engineering, requirement engineering**

I. INTRODUCTION

Requirement engineering is the first phase of the software engineering process, while in the first phase user requirements are accumulated and specified.

Many software failures initiate due to lack of software requirement specifications. Most common and serious requirement issues are related with software development which is due to requirement.

Most common issues which affect our projects are:

- Lack of user input
- Inappropriate requirements and specification
- Changing requirements and specification

Hence, a correct and complete requirement is necessary for various software projects.

In this paper we discuss the various issues of requirement engineering and discuss their solutions from different point of view. We also discuss the impact of these issues and how we fix them. We discuss the issues related about poor requirement quality, use case modeling, missing requirement, etc and try to solve them. These issues are very common and try to damage our requirement. Here we suggest some best solutions for these issues which affect our project [2].

II. RELATED WORK

In 1990's, the Standish Group conduct a series of survey in order to examine the failure rate of software development projects in the US. They found that only 16% projects were completed in time and budget. This survey is very strict because many projects deliver successful project which is either little late or over budget.

In the study 53% projects were come in this category, while average cost of overrun was 189% and 222% time overrun. Finally 31% projects were cancelled before completion because the ability to manage projects according to plan was missing. The Standish Group found that American companies and government agencies spend $81 billion for the cancellation of various projects in 1995.

In 1998, there is a small improvement found that project successful rate increase from 16% to 26% and cancellation rate of projects changed at 28%. They analyzed that there was a dramatic decrease in average size of cost and overruns in these projects. The Standish Group improves such kind of projects when cost and overrun rate is high by decreasing the size of projects. Small projects are simple to handle and control the cost.

The significant part of this study was the list of success factors:

- User involvement
- Management support
- Clear statement of requirements

The list of top three failure factors were:

- Lack of user interest
- Incomplete requirements and specification
- Changing requirements and specification

Remember that these are based on the perceptions of survey participants and we deduce them carefully. However these factors are comes in requirement engineering. The quality of requirement has a great impact on the final product of the outcome.

As we know that the cost of rework of functional requirement is very high .so a project is successful when its requirement is clear, complete and there is no ambiguity in requirement engineering [1].

III. CONCEPTS AND DEFINATION OF REQUIREMENT ENGINEERING

A. *Concepts*

In RE different types of user can provide the source of various types of requirement. So, the term user may be express both direct user and other stakeholders involve in the RE process.

Here we define various terms according to persons in RE.

Manuscript received November 02, 2014.

Muhammad Waqas Boota, Department of Computer Science, Virtual University of Pakistan, Lahore, Pakistan, +923336861944.

Naseer Ahmad, Department of Computer Science, Virtual University of Pakistan, Lahore, Pakistan, +923129550327.

Abdul Hye Masoom, Department of Computer Science, Virtual University of Pakistan, Lahore, Pakistan, +923027140617.

- Client: The person who pays for the product and decides the requirements.
- Contractor: The person who construct the product for a client.
- User: The person who drive the product [2].

B. *Requirement Engineering*

Requirement engineering is a systematic process that develops the requirements through iterative process in order to analyze problem, represent results in different formats and check the precision which we understand. RE convert the business related into the information system requirement [2]. Requirements engineering consist of requirements elicitation, analysis, verification, specification and management etc.

- Requirement Elicitation is the process in which we find out, examine, document, and understand the user's needs for the system.
- Requirements analysis is the process in which we redefine the user's needs.
- Requirements specification is the process in which we documenting the user's needs clearly and accurately.
- Requirements verification is the process in which the system requirements are complete, accurate, and clear.
- Requirements management is the process in which we schedule, coordinate, and document the requirements engineering activities [4].

IV. REQUIREMENT ISSUES AND THEIR SOLUTIONS

A. *Issues of Requirement Elicitations*

The issues of requirement elicitation have three categories of issues.

- Problem of scope: In problem of scope requirements provide too little or too much information and provide unnecessary design information.
- Issues of understanding: In issues of understanding users have incomplete understanding and poor understanding of computer limitations about their needs. In this category requirements are vague and un-testable.
- Issues of volatility: In issues of volatility requirements are changing and evolve over time.

Now we proposed a set of solutions for requirements elicitation problems.

- Define the technical environment in which product is placed.
- Define one or more elicitation methods such as interview, questionnaires and team meeting etc.
- Select those people who help the specific requirements.
- Participate many people and record their point of views about the requirements.

Create use case diagram and scenario in order to help client and user for better understanding the requirements [5].

B. *Poor Requirement Quality*

It means that many requirements are ambiguous, incomplete, inconsistent, incorrect, and out of date. These requirements are not needed in our system. The problem can also arise when untrained engineer can't properly manage the requirements which are given. Poor quality requirements increase development cost and overrun.

Now we see that how these issues are removed.

- We can solve these issues when we train the client and stakeholder in a proper way for good requirements. They need to understanding and collaboration that they separate the good and bad requirements.
- We use simple tools which identify vague words in requirements. We include the Members of architecture and Team members which check the quality of requirements that it is feasible and verifiable.
- We should ensure that requirement engineer is able to collaborate with stakeholders and attains the quality of requirements. Lastly, requirement engineer rework and delete those requirements which degrade the characteristics of good requirements [3].

C. *Use Case Modeling*

At present, there is a significant focus on use case modeling because this technique can identify and analyze requirements. Use cases look like the hammer which makes every requirements problem a nail. But, use cases are best for functional requirements and other techniques are suitable for the non-functional requirements. These functional requirements are interface requirements, data requirements, quality requirements, architectural, design, implementation, and configuration constraints. Many projects develop use case diagrams instead of creating sequence diagrams which detain the ordinary and exceptional paths during the use cases. They also fail to detain use case path preconditions, triggers, steps, alternative, main sequences post conditions etc.

We can also say how the system behaves under normal situation captured, and how system behaves when it can't do. Is system behaved normally?

Now we discuss the solutions of Use Case Modeling.

- RE use all aspects of use case modeling in order to make sure that all reliable paths in the use case identified and analyzed.
- We should use the use case modeling at the same time as an identification and analysis technique rather than requirements specification technique.
- We can utilize the use cases that identify, analyze the functional requirements.
- Inspection of the use case models will also help ensure that they are adequately complete.
- RE must use suitable requirements analysis techniques for the type of requirements being engineered. We can use checklists and a robust quality model which can defines all major quality

factors consequently that no significant type of quality requirement accidentally overlooked.

D. Improper Constraints

In practice, many requirements are not compulsory. So, many of them are like architecture, design, implementation, and configuration constraints are unreasonably specified as requirements. Stakeholders and requirements engineers are incorrectly assume that a common way to implement a requirement. They confuse with the fact that the implementation with requirement and unsuitably identify that how to build the system rather than what the system can do.

This issue is generated because requirements engineers are not qualified in the problem domain and engineering areas. Different kinds of stakeholders like users, customers, marketing, operators, maintainers, etc are suitable sources of the requirements can be take in the current system and imagine that how it significantly improved by new technologies and business process re-engineering.

Solutions of these issues are following:

- Ensure that all partners in the RE process are aware of it.
- Look unacceptable specified constraints must be one of the most important items on the requirements inspection checklist.
- Lastly, architects and specialty engineers take part in the requirements evaluation process and question every requirement.

E. Requirements Not Traced

A requirement tracing is widely distinguished, mandated in contracts, which included in many RE methods and training classes. But many requirements are not properly traced in practice. The sources of requirements like higher level requirements, other documents, and stakeholders are not properly documented. Requirements are often neither assigned to architecture and design elements and not to the test sets which verify them.

Requirement tracing is very difficult manually and even with modern tool support when requirements are very large in projects. We can see that the mapping from functional requirements to architecture and design elements is something. While, one-to-one, and this mapping become difficult when the modern technologies such as object, agent, and aspect orientation, middleware and other frameworks are used. Similarly, NFRs employed with many components which scattered across architecture. As a result, functional requirements and non functional requirements are not traced at all.

Now we discuss the solutions of Requirements Not Traced:

- Ensure that requirements tracing mandated in the agreement and unambiguously specified in the RE method.
- Ensure that tracing occur early in the project through design, development, and maintenance.
- Lastly, ensure that the evaluation of requirements tracing is the document part of the requirements verification method.

F. Missing Requirements

Missing requirements are important issue in RE. Midsize systems have many requirements of large systems can end up with a number of thousand separate requirements. So, it is possibility that important requirements may be missed. If we provide iterative, incremental development cycle then these minor slips cannot cause much harm. These missing requirements later identified and added to later builds. In fact, it is possibility that many information systems have numerous features which are not used by all users and possibly that not needed at all.

While the real issue is that many architecturally significant requirements are accidentally missed. These missing requirements are usually nonfunctional requirements. Most commonly quality requirements express minimum acceptable amounts of quality, for example accessibility, interoperability, presentation, portability, reliability, robustness, safety, security, stability, and usability. This is happens when the stakeholders are the source of the requirements assumes that these requirements are clear and go without saying.

Now we discuss the solutions of missing requirements:

- Requirements engineers should elicit requirements rather than relying on stakeholders
- The requirements team collaborates with specialty engineering team of all groups of stakeholders when eliciting requirements.
- Mature methods and techniques ensure the system how to handle all credible inputs and requests in all conditions.
- Instead of drawing use case diagrams, use case modeling must include the production of sequence paths with their connected preconditions, trigger conditions, and post conditions [3].

G. Excessive Requirements Volatility including Unmanaged Scope Creep

Many systems contain long development cycles and lifecycles and their requirements may be change. Systems develop business needs to change. In past system attempts to be conventional strict waterfall development cycles, and it is impracticable to freeze requirements in practice. Requirements are changing that's why industry uses the iterative, incremental, and parallel development and life cycles.

Stakeholder's desire constantly adds a few new requirements and change one or two existing requirements there. When this happens in unrestrained manner then we get the permanent problems of excessive requirements volatility and scope creep.

Now solutions of these issues are following:

- The primary solution is not to guarantee that existing requirements are true and forbid the addition of any new requirements

- Use modern approach and permit requirements changes is a good idea. Hence changes in the requirements should be properly managed.
- the requirements should be baseline and frozen at appropriate milestones within the development cycle for every release of the system
- Baseline requirements place in configuration control like any other major work product. While the impact of changes to these requirements needs to be known before the changes authorized to take place.
- Finally, when there is change in requirements then schedule and budget should be update [8].

H. Insufficient Verification of Requirements Quality

This issue is not concerning the verifying whether the as built system employs its requirements. It is about verifying adequately early in the development process and requirements have sufficient quality to avoid the many poor requirements. Mostly requirements informally verified in small peer reviews and stakeholder reviews. Both reviews are helpful, when we cannot identify requirements defects in the systems.

Now we discuss solution of this issue:

- When practical, evaluators use inspection somewhat the less formal reviews and walk through which verify and ensure so as to all of the requirements have the appropriate characteristics like clear-cut, complete, correct, mandatory, readable, etc..
- Software Projects develop and reuse checklists of the common and harmful requirements defects.
- Requirements engineers and evaluators use simple tools in order to examine the requirements and see inherently indistinct words in the requirements.
- The requirements verification team control representatives of all main types of stakeholders.
- The requirements evaluation team holds members of the architecture and test teams to verify that the requirements are practicable and verifiable.
- Finally, the requirements team authorized and mandated to rework and delete all requirements which be deficient in the required characteristics of good requirements.

I. Inadequate Requirements Validation

Most important task of requirements engineering is that stakeholders validate their requirements and ensure that the requirements are complete and correct. But unfortunately stakeholders are not always validating requirements. Main and basic reason is that the requirements engineers often cannot access to stakeholder representatives. This problem is created in projects when there are contractual and procedural limitations in the availability of stakeholders.

Second reason is that the project's requirements engineering method cannot include requirements validation due to lack of knowledge of the tasks which comprise requirements engineering. For a while, requirements validation neglected due to lack of stakeholder time, project schedule, and project budget.

Now solutions of these issues are following:

- Ensure that requirements validation is a fundamental part of any requirements method will not drop the first time when project resources become limited.
- Ensure that requirements validation incorporated into the project's schedule and budget with the schedules and budgets of the system's stakeholders.
- Finally, eliminate all unnecessary barriers which separating the stakeholders and the requirements team.

J. Insufficient Requirements Management

Many projects cannot effectively manage their requirements, and they store their requirements in paper documents and in simple spreadsheet. Requirements can also stored separately indifferent media which is control by different teams.

For example, functional requirements are stored in requirements database, interface requirements are stored in interface control documents, data requirements are stored as in one or more data dictionaries, security requirements are stored in multiple organizational security, policy documents, and other quality requirements are stored in a supplementary requirements specification. The important point is that there is little support for access in order to control these requirements with limits on who has what kind of access. The requirements are frequently missing important metadata, for example priority, type, status, source, and rationale, etc.

Now we discuss solution of this issue:

- To Deal the large number of requirements, the invariable changes to them, and store the requirements in a database. Store all traits of requirement thus they are easy to manage and maintain.
- Ensure that the requirements warehouse hold access to control, and stop unauthorized access to sensitive requirements.

K. Inadequate Requirement Process

In many projects the requirements method used mostly undocumented and it is incomplete in terms of missing documenting important tasks. The RE method frequently based in a single technique which is unfortunately used in all types of requirements.

Another reason for inadequate RE processes is the common use of standard, generic, out of the know how to develop methods which do not get together the needs of the specific project.

Now solutions of these issues are following:

- Experienced requirements engineer and process engineer work together which ensure that the RE method is complete, include all important tasks, techniques, roles and responsibilities, and work products. The quality organization audits the RE process.

- Ensure that RE method parts are mature and effectively used in projects which were similar in size, complexity, and developed similar systems.
- Ensure that RE method parts are appropriately documented, easily understood in their target audiences, and hold the appropriate level of detail based on the training and experience of the people who will use them.

L. Inadequate Tool Support

Several requirements engineers don't have sufficient tool support when engineering their requirements. For example, mostly requirements engineers use the requirements specification document and with a simple spreadsheet or relational database table. Very Few requirements engineers utilize a real requirements management tool:

- Borland Caliber RM
- IBM/ Rationale's RequisitePro
- Telelogic's DOORS2

These tools enable them to store individual requirements with their associated attributes. In requirements identification and analysis, and many requirements engineers use simple drawing tools and while others use CASE tools to draw diagrams. The requirements, their related models and diagrams developed and stored in different two or more and incompatible tools.

Expected solution of this issue is:

- Use a powerful, user friendly, and requirements management tool which allows the storing of requirements attributes.
- Also, Use a powerful, user friendly requirements modeling tool which capture requirements diagrams and related text.
- Ensure that these tools maintain the configuration management and their models.

M. Untrained Requirement Engineers

Here is a common myth detained by certain managers when requirements are specified and using native languages such as English. This can do by literate person who talk to a few stakeholders and write down what they desire. Here the belief is that, unlike design and programming which needs explicit technical experience and training. RE is a soft discipline in which anyone can perform.

Another myth is domain experts which know the application domain. But who don't know regarding RE magically become requirements engineers overnight. These two myths are deliberately untrue and it is common to observe people Peter Principled interested in the position of requirements engineer devoid of training in RE and without any experience.

RE is often in a position that is little valued in technical people, and do not understand in engineering discipline. In fact, a good RE needs the same characteristics of a good architect. Both need to be able to communicate well with technical people and non technical people. Frequently, the position of requirements engineer is seemed to be down as not having good prospects for career advancement.

Now solutions of these issues are following

- Select right combination of peoples with training, experience, motivation, and people skills to be good requirements engineers.
- Provide training to these peoples including classes, conference tutorials, books, and journals.
- Ensure those that both management and the technical staff, and recognize the importance of the role they play in project success [9].

N. Social Issues

We should also pay attention to social values because they are going to play an important role in requirement engineering activities. We should do some research in ethnographic area of requirement engineering activity. We receive a message that up to this time, the ethnographic study has some limitations like prolonged-time, results are detailed and not straightforward, differences in culture, difficulty in abstraction, lack of skill etc. We should try to introduce some techniques to overcome these problems.

Expected solution of this issue is:

- We should introduce some techniques to study the social value in some quantitative manner. Ethnographers are the persons who study the social behaviors customers and draw results, but there are no standard and mature ways so far to communicate these results to the requirement engineers.
- UML is a tool that could be used sometimes to convey these results to the requirement engineers. Requirement engineers can themselves study the social behaviors of customers under the guidelines of ethnographers [6].

O. Quality Criteria for Requirement Document

Requirements document is a very important and critical document and has shortcomings, even if all the quality criteria defined by IEEE have been applied at a requirement engineering stage. More careful analysis and new quality criteria are required to deal with these problems in the requirements document. There are many problems in practice which are intrinsic in nature, so, they are hard to catch in requirements document even if all the known standard quality criteria have been used. Some problems are extensive, unstructured or superfluous description of details.

All these problems make the requirement hard to read and work with.
Moreover, these problems limit the space of possible solution.
Expected solution of this issue is:

- We propose two new quality criteria "Root-based refinement" and "Minimality" that can deal with the problems just described.
- Root-based refinement says that the requirement should be refined and focus towards the final goal of the system to be developed. Minimality says the no more details than necessary in the requirements [7].

Table 1: Summary of RE issues and their solutions

Sr. No	RE Issue Name	Issues	Solutions
1	**The Issues of Requirement Elicitation**	Problem of scope, Issues of understanding, Issues of volatility	Define one or more elicitation methods, Participate many people
2	**Poor Requirement Quality**	ambiguous, incomplete, inconsistent, incorrect, and out of date	train the client and stakeholder, identify vague words
3	**Use Case Modeling**	Use cases look like the hammer which makes every requirements problem a nail	RE use all aspects of use case modeling in order to make sure that all reliable paths in the use case identified and analyzed
4	**Improper Constraints**	Many requirements are not compulsory. So, many of them are like architecture, design, implementation, and configuration constraints are unreasonably specified as requirements	Ensure that all partners in the RE process are aware of it, architects and specialty engineers take part in the requirements evaluation process and question every requirement
5	**Requirements Not Traced**	Many requirements are not properly traced in practice	Ensure that requirements tracing mandated in the agreement and unambiguously specified in the RE method.
6	**Missing Requirements**	In Midsized systems have hundreds of requirements of many large systems. So, it is possibility that important requirements may be missed	Requirements engineers should elicit requirements rather than relying on stakeholders
7	**Excessive Requirements Volatility including Unmanaged Scope Creep**	Many systems contain long development cycles and lifecycles and their requirements may be change. Systems develop business needs to change	Use modern approach and permit requirements changes is a good idea. Hence changes in the requirements should be properly managed
8	**Insufficient Verification of Requirements Quality**	verifying adequately early in the development process and requirements have sufficient quality to avoid the many poor requirements	When practical, evaluators use inspection somewhat the less formal reviews and walk through which verify and ensure so as to all of the requirements have the appropriate characteristics like clear-cut, complete, correct, mandatory, readable, etc..
9	**Inadequate Requirements Validation**	Main and basic reason is that the requirements engineers often cannot access to stakeholder representatives.	Ensure that requirements validation is a fundamental part of any requirements method will not drop the first time when project resources become limited
10	**Insufficient Requirements Management**	Many projects cannot effectively manage their requirements, and they store their requirements in paper documents and in simple spreadsheet. Requirements can also stored separately indifferent media which is control by different teams	To Deal the large number of requirements, the invariable changes to them, and store the requirements in a database
11	**Inadequate Requirements Process**	In many projects the requirements method used largely undocumented and it is incomplete in terms of missing documenting important tasks, techniques, roles, and work products	Experienced requirements engineer and process engineer work together which ensure that the RE method is complete, include all important tasks, techniques, roles and responsibilities, and work products
12	**Inadequate Tool Support**	Several requirements engineers don't have sufficient tool support when engineering their requirements.	Use a powerful, user friendly, and requirements management tool which allows the storing of requirements attributes
13	**Unprepared Requirements Engineers**	Here is a common myth detained by certain managers when requirements are specified and using native languages such as English.	Select right combination of peoples with training, experience, motivation, and people skills to be good requirements engineers
14	**Social Issues**	We should also pay attention to social values because they are going to play an important role in requirement engineering activities.	We should introduce some techniques to study the social value in some quantitative manner. Ethnographers are the persons who study the social behaviors customers and draw results. UML tool is used.
15	**Quality Criteria for Requirements Documents**	Requirements document is a very important and critical document and has shortcomings, even if all the quality criteria defined by IEEE have been applied at a requirement engineering stage.	We proposes two new quality criteria "Root-based refinement" and "Minimality" that can deal with the problems just described

IV. Conclusion

In this paper we discuss the fifteen most important issues of RE and their solutions. Every issue which we discuss can create problems and we can avoid these problems and fix them. In the discussion we see that these problems are synergistically related. The bad news is that if one problem is come then it severely impacts our requirements. But the good news is that numerous solutions are also synergistically related. Apply one industry best practice can solve problem which helps to solve other problems.

Acknowledgment

Our Sincere thanks to our Respected Sir "*Dr. Ghulam Ahmad Farrukh Sb"* who have given us this opportunity of doing our research work at Virtual University of Pakistan. Our heart-felt thanks to our family, friends and colleagues who have helped us for the completion of this work. Finally we would like to direct a sincere Thank you to our family. You have always been there, always supported us with whatever means you have had available, always encouraged us, especially our wonderful parents.

References

[1] What is Requirements Engineering http:/www.cs.toronto.edu/~sme/CSC340F/readings/FoRE-chapter 01-v7.pdf

[2] Romi Satria Wahono "ANALYZING REQUIREMENTS ENGINEERING ISSUES", IECI Japan Workshop, 2003

[3] Donald Firesmith"Common Requirements Issues, Their Negative Consequences , and the Industry Best Practices to Help Solve Them, Software Engineering Institute ,USA,2007

[4] A Frame work for Software Product Line Practice, Version5. http://www.sei.cmu.edu/productlines/frame_report/req_eng.htm

[5] Requirement elicitation http://en.wikipedia.org/wiki/Requirements_elicitation

[6] Stephen Viller & Ian Sommerville"Social analysis in the requirements engineering process:from ethnography to method", Computing Department, Lancaster University, UK

[7] T. Simon, J. Streit, and M. Pizka "Practically Relevant Quality Criteria for requirements documents", itestra GmbH, Ludwigstr. 35, 86916 Kaufering Germany.

[8] Barry W. Boehm and Philip N. Papaccio, "Understanding and Controlling Software Costs," IEEE Transactions on Software Engineering, Vol. 14, No. 10, October 1988, pp. 1462-1477

[9] Donald Firesmith, "Specifying Good Requirements," Journal of Object Technology (JOT), 2(4), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, pp. 77-87, July/August 2003. http://www.jot.fm/issues/issue_2003_07/column7

[10] Karl E. Wiegers, "Inspecting Requirements," StickyMinds.com Weekly Column, 30 July 2001 http://www.stickyminds.com