

# Critical Requirements Engineering Errors Leads to Fails Software Project

Muhammed Talha

Virtual University of Pakistan Multan, Pakistan

**How to cite this paper:** Talha, M. (2018). Critical Requirements Engineering Errors Leads to Fails Software Project. *The Educational Review, USA, 2(2)*, 174-180.  
<http://dx.doi.org/10.26855/er.2018.02.003>

**Corresponding author:** Muhammed Talha, Virtual University of Pakistan Multan, Pakistan.

---

## Abstract

Requirement Engineering (RE) is one of the most important activity of Software Engineering. All the successful software relies on RE. It is noticed that there is more than one reason of software failure. All the literature are written about cost, time, budget, high risk, and software process etc. None of them exactly specify the critical RE problems and issues that fail software. In this paper, we focused on RE perspective to find causes of software failure. The place of RE in relative to software failure was discovered and studied. The most project additionally suffered from requirement errors. We specify all types of requirement engineering error that will effect to fail software project. We also introduced the errors of requirement management. We conclude recommendation for reducing the common failure factor.

## Keywords

Requirement Engineering, Software Project Failure, Failure Factors, Requirement Engineering Stages, Requirement Engineering Errors

---

## 1. Introduction

Requirement Engineering is one of the main subdivisions of software development life cycle. It is the effective methods and practices for RE stages like requirements elicitation, requirement investigation, requirement description or specification, requirement verification, requirement validation and last one requirements management. Requirement Engineering is the early stage of software engineering that is used to collect user requirements, analyze, agreed, and specified for producing quality product. This stage is mainly used to collects all types of requirements like functional, non-functional, domain and inverse requirement (Swarnalatha, Srinivasan, Dravid, Kasera & Sharma, 2014).

Requirements engineering has two core set of actions; development and management (Swarnalatha, Srinivasan, Dravid, Kasera & Sharma, 2014). Software requirement development generally focused on discovering, investigating, documenting, authentication and justification of requirements whereas software requirement management normally contains actions related modification management of software requirements (Swarnalatha, Srinivasan, Dravid, Kasera & Sharma, 2014) (Pandey & Suman, 2012).

Research states that there are numerous reason of software failures are mainly due to low cost, fewer time, not as much of resources, less budget and inadequate training of developers. There are other many reasons of project failure related to requirement engineering in which because of insufficient requirements, altering requirements, deprived requirements, and unfeasible hopes. However, the claim of organized methodology will moderate the tasks of requirement engineering and the risks of the project dying. It is also very difficult to collect precise data and material about the planned product and examine the organizational desires and practices (Hussain, Mkpojiogu & Kamal, 2015).Requirements of software

are gathered by requirements engineering whom is the method of defining requirements (Cheng & Atlee, 2009). Cheng and Atlee (Cheng & Atlee, 2009) stated that effective requirements engineering implicates the realizing of the stakeholder's requirements, understanding of that requirements contexts, show off, examining, negotiating, certifying, evaluating recognized requirements, and the supervision of requirements. Asghar and Umar (Asghar & Umar, 2010) describe in their literature that RE is known as the initially phase of software development process and it known as major key stage in software growth.

RE is a reasonably matured discipline through numerous familiar approaches and practices for recognizing, examining, requiring, managing, attesting, and confirming requirements. But if this is true so, why are so many flaws are found in requirement documents? Why are requirements engineering error quiet source of software project failures in expressions of significant charge and timetable attacks? Why project fail due to adequate quality? Do we need innovative and thoroughly enhanced requirements engineering approaches, practices, and implements? (Firesmith, 2007)

There are many reason of software failing due to poor requirements quality, inappropriate constraints, untraced requirements, missing requirements, excessive requirements, inadequate verification of requirements quality, inadequate requirements validation and management, poor requirements process, inadequate tool support and unprepared requirement engineer in requirement engineering domain (Firesmith, 2007).

The reason behind to work in this paper is to explore issues and problems that lead to failing software because of the Requirements Engineering errors. There are so many common errors and defects found in requirement engineering process when requirement engineer works on it that describe in above paragraph. In this paper, I'll categorize the all type of RE errors and also gives the idea of error of Management control in RE.

## 2. Related Work

In 2014, Shail K Dinkar (Dinkar, 2014) presented the four categories of RE error in which top three error were presented by Kurt Bittner in his article "Why requirements go bad" (Kurt Bittner, 2009) and the last one mainly research of Shail K Dinkar. There are four main categories of RE errors in his paper. Errors of conception, errors of a specification, errors of implementation and last one errors of visualization (Dinkar, 2014). All these errors categories further have more sub-types that will more explain the failure causes of a software project.

## 3. Stages of Requirements Engineering

Requirements engineering monitors a order of organized stages to collect customer requirements (Firesmith, 2007). RE emphasizes the use of systemetic and repeatable techniques that ensure the completeness, consistency, and relevance of system requirements. These organized stages are:

- 1) Requirement Elicitation and Requirement Collection – This stage consist of including of discovering, reviewing, documenting, and understanding the client's needs and constraints for the system.
- 2) Requirement Analysis or Requirement Investigation – This stage contains filtering the requirements that describe and argue this requierment may implement into system or not.
- 3) Requirement Specification or Requirement Description – This stage involves actions to store and document the analyzed requirements.
- 4) Requirement Verification – This stage ensure that all the requirements are complete, exact, reliable and perfect.
- 5) Requirement Management –This stage include activities of scheduling, coordinating, and documenting the requirements engineering activities.

All the above stages are the main activities of RE process. First of all Requirement Elicitation in this phase a Require-

ment engineer meets with stakeholders and find out all the information about system, their desires, needs, wishes and constraints. In this step user requirements is complete which mean any information about system from user or client is collected. The second step is to analyze the user requirement and make sure that is it possible to apply all requirements into implementation? Is all the functions are implementable? Is requirements are true about system requirements? If all these question find answers yes then it goes to another stage known to be Requirement Specification. In this stage clients needs and constraints are documented clearly and exactly. In Requirements Specification user requirements changes as system requirements that are more clear, exact consistant and reliable. Next steps involves the verification of requirements. In this step all the documented requirements are verify that that all the requirements are complete, exact, reliable and perfect. Verification is more important stage that will help to ensure about requirements. If the requirements are verify than it is possible to implement very easy. The last steps involve the activities of scheduling, coordinating, and documenting the requirements engineering activities.

Change management strategies must offer a proper mechanism for offering changes in the product and supporting the organized charge of how proposed changes will influence the product. In this paper I also proposed a configuration management errors that will explain when requirements changes how system will fail. How changing in requirement effect on the core assessment of system requirements. If the requirement engineer cant well understood the change management then system lead to fail.

#### **4. Errors in Requirement Engineering**

As per perceived in previous section, requirements engineering devises 5 stages so error can be found on some level of sages. Impartial Saying about requirements are vague, uncertain and unclear is encountered as a single peny of error during requirements engineering phase. Taking discovered a amount of mutual categories of faults to find how they are wrong and fail software project, Kurt Bittner (Kurt Bittner, 2009) characterized requirements errors and issues into 3 main sets and last one is proposed by Shail K Dinkar (Dinkar, 2014):

- 1) Errors of Conception
- 2) Errors of Specification
- 3) Errors of Implementation
- 4) Errors of Visulization
- 5) Errors of Requirement Management

##### **4.1. Error of Conception**

Error of conception is the root error of requirement engineering. It's main cause of misconception or misunderstanding. Errors of Conception arise after the requirement is unwell defined, i.e. when it compute the incorrect problematic that is founded on defective expectations about the result. These types of mistakes happen when solutions are under the weather assumed, demonstrating that the systems are badly facing problems. When errors are existent, the resulting requirements may be ambiguous and unclear, but nevertheless wrong. The result of these errors, system implementation will be wrong, the resulting system has little or no value. This system can't defend "requirements were satisfied" is of little consolation since the concept of the problem itself was understood incorrect. These errors generally occur when requirements were collected after incorrect basis or some stakeholder aggressively interconnected incorrect info (Dinkar, 2014).

There are many symptoms of errors of conception main cause is lack of clarity in the solution goals or the problems being solved. The second one is that unused or unneeded functions and features that are added as major functions of system. Third one is unneeded complexity. The Standish Group's Chaos Reported that 30% of the functionality of project were

not needed or idle. Client is not aware about what he really needed in his project.

Error of conception further subdivide into some of common errors that lead to fail software project (Firesmith, 2007).

- 1) Poor Requirements Quality
- 2) Missing Requirements
- 3) Un-prepared Requirement Engineer
- 4) Inappropriate constraints

#### **4.1.1. Poor Requirement Quality**

Now reality, the quality of several indicated requirements is deprived. These requirements may not show the real and acceptable requirement of the system. By poor requirements quality, I exactly mean real requirements specifications are vague and not cohesive, imperfect, unreliable, improper and old-fashioned, identified using technical terminology slightly than the terminology of the user or business/application domain. All the poor requirements must base on misconception of stakeholder or requirement engineer (Firesmith, 2007).

The poor quality must lead to design wrong specification of software then also wrong implementation of coding. After all system goes fail. All the process must be wrong if quality of requirements are poor.

#### **4.1.2. Missing Requirement**

When the system is too large or complex then it need more and complete specification of requirements to develop a good system. Large system consist of thousands of separate requirements. All's are very important to implement for perfect system development. It is notice that bits of functionality to slip through the flaws (Firesmith, 2007). The major requirement missing is related to non-functional requirements like as accessibility, interoperability, performance, movability, trustworthiness, robustness, protection, safety, strength, and usability. This naturally occurs as the stakeholders often assume that such requirements are noticeable and go without specifying. Missing requirements is the one of the best example of misconception in which stakeholder thinks that all the non-functional requirements are obvious and must be in system by company.

Missing requirements is the major cause of software failure. It's simple a reason when you have not to specify what a system does or respond then how system will successful (Hussain & Mkpjoigu, 2016).

#### **4.1.3. Inappropriate Constraints**

Like as missing requirements, constraints about systems are not specified in some projects. It is the misconception of stakeholder and requirement engineer that system constraints is not a part of requirements while system constraints are the major type of requirements types. System constraints like design, installation, configuration, architecture, execution constraints that are unnecessarily specified as requirements. They incorrectly identify how to developed the software slightly than what the system had better do (Firesmith, 2007).

When the stakeholder will not specify about system interface design, form size, text font, implementation languages, configuration method and policy then how system will developed successful. The main cause of this is misconception of stakeholder that leads to fail software.

#### **4.1.3. Unprepared Requirement Engineer**

Requirements engineering is very soft skill position. Every one can be requirement engineer overnight. Everyone can collect requirement from stakeholder but no one can elicit requirements in technical manner that shows overall function-

ality of the system. Its some thing difference like domain expert who are totally aware of business and domain. Mainly this position is related to technical people who works in technical manners and proper exhibit the system specification. The requirements engineer needs some of the same capabilities of a good architect. Stakeholder need to communicate good with non-technical people as well as technical people. Often, the place of requirements engineer is not having good views for career progression. Requirement Engineering is not well-thought-out as a fun by most technical people, who wrongly reflect the role of a requirement engineer to be like that of management than technologist (Firesmith, 2007). It is notice that in literature all the person in company thinks that they can perform as requirement engineer can collect requirements from stakeholder but it is not true if a requirement engineer is not known about his technical work then he leads to fail any of major software project.

#### **4.2. Errors of Specification**

Requirements specification is the third most stage of requirement engineering. This stage involves actions to store and document the analyzed requirements. These errors arise when requirements is going too described after analysis phase. Mostly requirement engineering error belong to errors of specification when requirement engineer can't able to describe or specify requirements that has been analyzed. Errors of specification has a vast types in it. These error can be generated by resulting from imperfect, unclear explanations, inappropriate terms, wrong techniques for descriptions; and un-needed specification leading to fail software project (Kurt Bittner, 2009).

Errors of specification further subdivide into some of comman errors that lead to fail software project (Firesmith, 2007).

- 1) Requirements Not Traced
- 2) Inadequate Verification of Requirements Quality

#### **4.3. Errors of Implementation**

Requirements validation and verification is the second last stage of requirement engineering. This stage ensure that all the requirements are complete, exact, reliable and perfect. This phase is the real implementation of raw information into real requirements. These errors get up when the idea was correct, the description was exact, but the requirement was not implemented correctly. This blame is not true that all the problems lies in project management or in testing, not in requirement engineering. But all the assumption are discrepancies are false and eventually not very valuable. There is no need to specify requirements if you are not going to implement them, the way to check requirements is to test them correctly. The project management and testing are intimately related with requirements (Kurt Bittner, 2009).

Errors of specification further subdivide into some of comman errors that lead to fail software project (Firesmith, 2007).

- 1) Inadequate Requirements Validation
- 2) Inadequate Requirements Process
- 3) Poor quality measurement

#### **4.4. Errors of Visualization**

In some projects, stakeholders want to see an urgent output of a system. What was proposed? How it'll look like? How does it work? Then Engineer proposed a prototype of that system that looks like require the system with minor functionality in agile methodology. So prototyped of a system will be developed as user requirement. An example of the real time system emulator is developed but in reality, there is a huge difference between the emulator and real system. Now there is the error of visualization when prototype can't exhibit the real needs of a system.

These errors occur when proposed system area was represented by prototypes but not drawn against a result space area.

The system best way to emulate, remains changed from a real system. This effort is to systematize the handling of the real-time system while developing a software system, the proposed system and the developed system still express in completely altered languages (Dinkar, 2014).

The symptoms of error of visualization are difficulty in using the system. When end user found that there is much difficulty to use system then there is the error of visualization occur. The second symptoms are growing requirements. When the system in under user accepting testing process there are numerous instances of the enhancement request. It means the user needs more changing in a system. Firstly he visualizes the system completely differently. The third symptoms are performance issue. (Dinkar, 2014)

1) Prototype Error

#### 4.5. Errors of Requirement Management

Requirement management is the final stage of requirement engineering that includes activities of scheduling, coordinating, and documenting the requirements engineering activities. It is the process to change and manage requirement. After requirement validation client needs to change some major requirements in his project then requirement management stages work to start. Many project managers do not effectively manage their requirements. They mostly store their requirements in simple paper sheets or in simple spreadsheets. Many types of requirements may be stored separately in different media controlled by different teams as the requirement team, the security team, the management team and mainly engineering teams (Firesmith, 2007).

For example, functional and non-functional requirements may be kept in a requirement database. Data requirement may be saves as data design definition and data dictionary, security requirement may be stored in multiple organization policy documents, and other quality requirements may be stored in a supplementary requirement specification. In requirement management there is a little support for access control to these requirements means no one can read, write, update and delete any of requirement.

The main error is that requirements stored in paper or spreadsheet rather than in requirement repository are difficult if not impossible to create, manipulate, and maintain. Dispersed requirements are hard to find, sort, query, and maintain. Lack of access control is also a big error if requirement engineer can't read the information at the right time they can't implement well and change requirements. Shortage of centralized, program management of requirements also makes it hard to capture, investigate, and account requirements metrics.

There are many ways to deal with requirement management errors. When the client has very huge requirements and the constant changes to them, then store the requirements in a verified database software or the repository of a requirement tools. Stored the requirement diagrams and models linked with original requirements. Do not disturb requirement by distributing it at many places that can't handle by a signal person. Keep them all in the same repository.

- 1) Excessive requirements instability including unmanaged scope creep
- 2) Inadequate requirements management

#### 5. Conclusion

The literature is an effort to isolate out software project failure due to critically requirement engineering errors. This paper extremely introduced all types of requirement engineering errors that lead to failing software project based on Kurt Bittner defined errors (Kurt Bittner, 2009). In this literature classification and sub classification of errors are described. A new requirement engineering issue introduced as Errors of requirement management that occur due to poor management of data recording of a project in different places. This error also so effects of software project failure. When the user

needs constant changes in his project then this error occurs. Error removal strategy also describes in this literature.

## References

- Hussain, A., Mkpojiogu, E. O. C., & Kamal, F. M. (2015). Eliciting User Satisfying Requirements for an E-health Awareness System Using Kano Model. Proceedings of the 14th WSEAS International Conference on Computer and Computational Science (ACACOS'15), Kuala Lumpur, Malaysia.
- Cheng, B. H. C., & Atlee, J. M. (Springer, 2009). Current and Future Research Directions in Requirements Engineering Design Requirements Eng.: A Ten-Year Perspective.
- Donald Firesmith, "Common Requirements Problems, Their Negative Consequences, and the Industry Best Practices to Help Solve Them" Software Engineering Institute, U.S.A. at [www.jot.fm](http://www.jot.fm). Published by ETH Zurich, Chair of Software Engineering ©JOT, 2007
- Pandey, D., & Suman, U. (2012). An Effective Requirements Engineering Process Model for Software Development & Requirements Management. Int'l Conf. on Advances in Recent Tech. in Comm. & Comput., 287-291.
- Swarnalatha, K. S., Srinivasan, G. N., Dravid, N., Kasera, R., & Sharma, K. (2014). A Survey on Software Requirements Engineering for Real Time Projects based on Customer Requirements. *Int'l J of Advanced Research in Computer and Communication Engineering*, 3 (1).
- Bittner, K. (2009). When Requirements Go Bad" at Ivar Jacobson International.
- Requirements. (2016). Towards an Understanding on Why Software Projects Fail Azham Hussain and Emmanuel O. C. Mkpojiogu. Citation: AIP Conference Proceedings 1761, 020046; doi: 10.1063/1.4960886.
- Asgar, S., & Umar, M. (2010). Requirement Engineering Challenges in Development of Software Applications and Selection of Custom-off-the-shelf (COTS) Components. *Int'l J. Soft. Eng.*, 1(1), 32-50.
- Dinkar, K. S. (2014). Requirement Engineering Errors: Errors and Ambiguities of Visualization. *International Journal of Computer Applications*, 92(12).