**Impact of Requirement Management on Failure cost in the context of software application development.**

S.P. Udhayakumar

Research Scholar

Sathyabama Institute of Science and Technology, Chennai 600119

[spuk19@gmail.com](mailto:spuk19@gmail.com)

Dr M. Sivasubramanian

Research Supervisor

Sathyabama Institute of Science and Technology, Chennai 600119

[ms2008.sb@gmail.com](mailto:ms2008.sb@gmail.com)

Abstract

*There is a high correlation between software failure cost and requirements elicitation and management. While failure cost (part of cost of quality) is the expense to fix a non-conformance or a poor quality, Requirement Elicitation & management talks of gathering & managing requirements from the customer and users. Though requirement elicitation and failure costs are two separate topics in software development, the cost of software development spikes due to high number of defects injected in the requirement phase. There are multiple reason like incomplete requirements from stakeholders, misunderstanding of stated requirements, wrong requirements, inadequate review and incorrect development of requirements why the high number of defects are injected. Fixing a requirements related defect in production phase may cost the project 100 times (refer diagram 1) the original cost of fixing/avoiding the defect in requirement phase. There is no one all-inclusive model/technique that addresses the concerns. Thus, realizing the importance of Requirements elicitation and management and its effect of cost of poor quality, there is a need for an all-inclusive, simple and scientific approach through which quality of requirements elicitation and management can be assured and related cost of poor quality can be avoided.*

This study performs a systematic review of the available data (primary data) on the topics of Requirement elicitation and management and Failure costs. Review and analysis of all found data sources, provide a method of locating, collating and evaluating the data and provide solution tenets for an all-inclusive requirement elicitation technique.

Key words: Software requirements management, requirement elicitation, Failure cost, non-conformance cost, defect fixing, cost of poor quality

## **Introduction**

Requirements elicitation is the practice of researching and discovering the requirements of a system from users, business, customers, and other stakeholders. The success of the requirements elicitation activity has a high impact on software quality, cost and application / product stability. Requirements elicitation is most difficult and a crucial first step in software development process, it involves understanding customer’s/users language, understanding the implicit requirements and should have the capability of converting the customer’s natural language into technical requirements

Requirement elicitation is an important factorfor increasing cost of poor quality (Failure cost) in a project, it is not that RE is the only responsibility for higher COQ, but as an individual factor it is significant when compared to other within the lifecycle. The greatest contribution to COQ comes from poor Requirement management and cost of fixing such problems is higher than other source of errors

Without thorough understanding of the requirements the cost of developing a product escalates multifold. Requirements elicitation is seldom well done, and an inaccurate or incomplete understanding of user or customers’ requirements will lead to increased failure cost of a software projects which eventually leads to its downfall.

**Review of Literature**

A report by Ann M. Hickey and Alan M. Davis (2003) on Elicitation Techniques selection discuss on various elicitation techniques and responses.

A study by Nilofar Mulla1 and Sheetal Girase (2012) on NEW APPROACH TO REQUIREMENT ELICITATION BASED ON STAKEHOLDER RECOMMENDATION AND COLLABORATIVE FILTERING talks about the impact on the quality of the requirements due to communication problem and recommends on close interaction between developers and end-users of the system

Sanjay Mohapatra (2015) in his paper controlling Quality At The Upstream through requirements management In Commercial Software Project Management talks about the expertise in dealing with human nature and overcoming communication difference will help result in a good starting point for a software project.

Carla Pacheco, Ivan García, Miryam Reyes (2018) in, Requirements elicitation techniques gives the various Elicitation techniques and the characteristics influencing the technique's effectiveness

Shreta Sharma, S. K. Pandey (2013) in their paper Revisiting Requirements Elicitation Techniques talks in details about various elicitation techniques like Traditional techniques, Colloborative techniques, Cognitive techniques and Innovative techniques, their advatanges and disadvantages

**Research method**

## Primary data is collected, sanitised, analysed for the root cause and used PCE (Phase containment effectiveness(refer Table 1) accounting system to find out the origin of the defects is PCE aims to identify the errors in the same phase where they were introduced and also to attribute to other phases where it got introduced in case it escaped from the previous phase. The cost is arrived using the data published by IBM System Science Institute Relative Cost of Fixing Defects.

.

**Objectives**

To prove there is huge impact of software Failure cost due to poor requirements management

Null Hypothesis: There is a significant relationship between Requirement Management and Failure cost

Alternate Hypothesis: There is no significant relationship between Requirements Management and Failure cost

## **Requirements Elicitation and Management**

A survey conducted on European companies found that more than 60% of them considered Requirement elicitation problems as very significant. If an error occurs in Requirement elicitation stage, it will percolate to subsequent development phases which will have a higher failure cost and sometimes lead to shelving the project itself.

Requirements engineering (RE) is pivotal and central to every successful software development project. There are several reasons why software projects fail; however, poorly elicited, documented, validated and managed requirements contribute grossly to software projects failure.

One of the most common reason for project failures is the inadequate attention to customer /user requirements elicitation. Approximately 50–60 % of the defects occur due to poor requirement management (refer Table 1). If the customer is not sufficiently involved in defining requirements or the analyst takes into account only the customer requirements, but the customer is unable to correctly formulate requirements meeting the business needs, then the quality of defined requirements are weak, insufficiently defined and misunderstood at the analysis stage, with each next stage of the project the amount of work needed to be done increases in order to fix the errors – around 40 % of the work at the design stage and around 70 % at the implementation stage etc. Requirements elicitation and management process is an important part of the project requirements analysis that ensures product stability and lower cost of quality. The relative costs have been shown in the below diagram.

Though variety of techniques may be used to elicit system requirements like brainstorming, questionnaire, group elicitation, document analysis, interview, prototyping etc, all of these techniques has its own limitations

## 

## *IBM System Science Institute Relative Cost of Fixing Defects*

Diagram - 1

## **Cost of Failure**

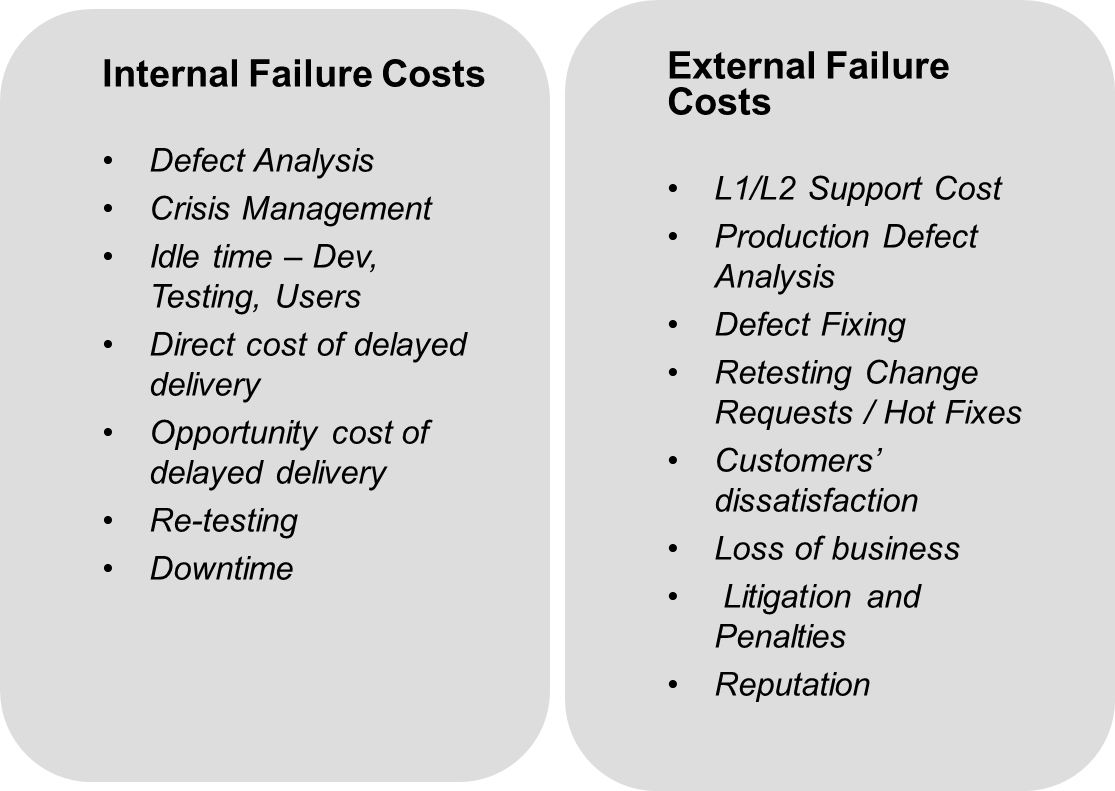
Cost of quality is a methodology that allows an organization to determine the extent to which its resources are used for activities that directly affect the quality of the organization’s products or services, and that result from failures and deficiencies. Failure cost is the cost of non- conformances, Defect fixing cost, rework, retest etc. (though Cost of Failure includes….)

* + Internal failure cost: The cost arising within an organization due to defects at any stage of the lifecycle, such as rework, retest, re-inspection and redesign. Scrap, replacement, rework and repair can be mentioned as typical examples of internal failure. •
  + External failure cost: The cost arising after delivery to a customer/user due to nonconformities or defects, which may include production defects, rework, the cost of claims against warranty etc.

There are some costs which need to spent and there are some costs which can be avoided and some which can be optimized. For e.g. prevention cost can be marginally increased, appraisal cost can be optimized, however failure cost should be avoided as much as possible.

In summary there are many reasons for software project cost jacking up; however, poorly engineered requirements process contributes immensely to the reason in increased software projects costs.

.



From the studies made by various software development communities, it is evident that most failures in software products are due to errors in the requirements and design phases – as high as 64 percent of total defect costs (Figure 3), according to *Crosstalk, the Journal of Defense Software Engineering*.

The Standish reports are used to justify a claim that more than half of overruns occur

due to poor RE (Briggs & Gruenbacher, 2002; Eberlein & Leite, 2002; Lamsweerde, 2000). This

finding is replicated in other studies in Europe (Lamsweerde, 2000) and in studies by Boehm

(2000) and Moløkken-Østvold and Jørgensen (2003).

## **Analysis and Findings – Effect of software Requirement Elicitation & management on Failure cost**

As per Boehm & Papcocio, 1998 30% to 50% of total development cost is rework cost out of which 70 to 85% are from requirement errors (leffingwell, 1997).

Many studies have shown that requirements errors are very costly. By one estimate (in an article by Donald Firesmith for the Software Engineering Institute), requirements errors cost US businesses more than $30 billion per year and often result in failed or abandoned projects and damaged careers. The common wisdom is to find and fix requirements errors early in the lifecycle of a project.

Various industry surveys indicate that half of all defects are related to requirements. In the absence of your own specific measurement, use the 50 percent estimate:

Is there one way where much of Failure cost can be avoided?? Yes. Requirement Elicitation. Most of the internal failure cost and external failure cost happens because of the defects, most of the defects are injected in Requirement phase. The reasons may be many like inadequate requirements, missing requirements, wrong understanding of requirements, conflicting requirements etc. however, a details analysis of defects taken from a set of application development projects reveals three major category of areas where the defects occur most, i.e. wrong or incomplete requirements, misunderstanding of requirements, understood the requirements correctly but wrongly coded.

Total around 59% of the defects are due to poor or ineffective requirements – Table 1

Out of 4540 defects analysed, there are about 59% of the defects got injected in Requirement Phase.

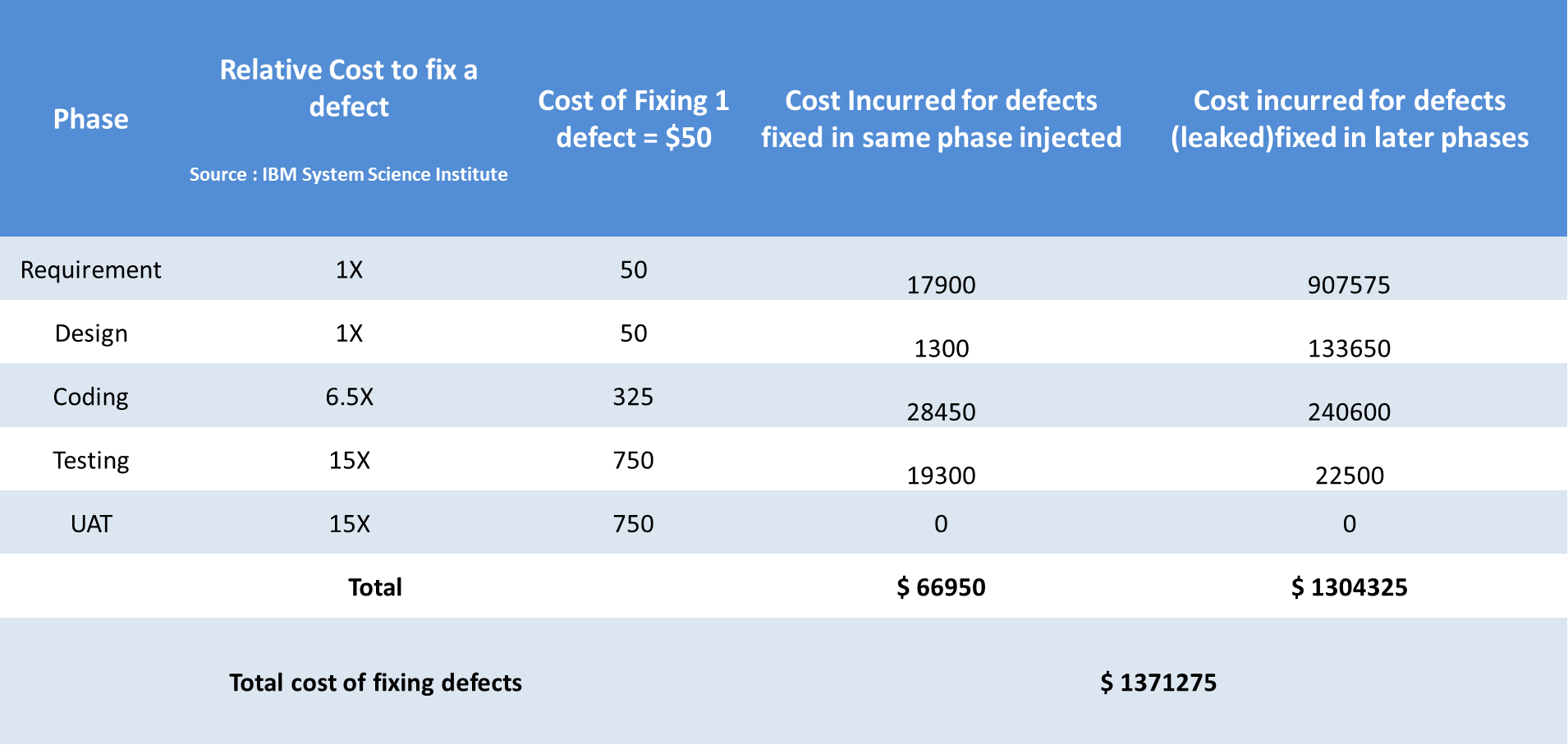
All these projects followed one or other traditional requirement elicitation technique.



it highlights the need for a scientific step by step approach for eliciting requirement.

Table - 1

Table 2



The cost to fix a software defect varies according to how far along you are in the cycle, according to authors Roger S. Pressman and Robert B. Grady. These costs are presented in a relative manner,

## **How to make Requirement elicitation effective to reduce Failure cost?**

Many researches have attempted to give a solution for this issue, but the solutions did not make a big impact as the cost of poor quality continue to soar due to poor requirement elicitation. It is because the solution should be drafted depends on each organization’s unique situational context, however here we are trying to create core components which has to be addressed which can be built through any models based on the organisation context. These components can be used for any organisation or any lifecycle model.

Three aspects to it (1) All Stakeholders (2) The Process (3) Scientific method

* The solution should involve all the stakeholders in the elicitation
* A traditional requirement analysis may take couple of hours or a day with the Business Analyst on understanding the requirements from the customer, while involving all the stakeholder in requirements elicitation exercise may take 3 to 4 days. The cost of this exercise will be high but it will outweigh the cost that is saved from avoiding rework.
* Create a method to track the understanding of all the stakeholders
* Derive at a method using statistical techniques on the parameters of the requirements to be concentrated (based on lessons learnt)e.g. ambiguous requirement
* Try use case diagrams while doing the elicitation
* The technique should be lifecycle agnostic as both waterfall and Agile methods can follow the technique. (in waterfall model, as part of requirement analysis and in Agile projects as part of Product backlog refinement ) as this challenge is there in all types of projects
* Involve the Business/Customer/user to understand the development and architectural challenges as they will understand what can be developed and what are the alternative way it there is a challenges.
* Collaboration with the stakeholders and subject matter experts during elicitation will help the successors in the lifecycle to correct the mistakes done by the predecessors. For e.g. While developing if there is mistake committed by the designer, it can be corrected by the developer as both are part of elicitation process and know what they are supposed to develop (internal review mechanism is already inbuilt)
* Customers need to elaborate the use cases during the elicitation sessions by typing more detail, adding sub processes that are automatically linked
* Create parameters on which the requirement elicitation and analysis can be done. Parameters can be derived based on the root cause analysis of the previous project defects
* Request comments from other stakeholders on each of the requirement. Your requirements automatically are stored on a collaboration server, and now you can gather comments from other stakeholders who can see the requirements and diagrams.
* It is totally the responsibility of the development team to get all the requirements clear for their respective areas
* Have a weighing mechanism so that a toll gate can be established. For e.g. Greater than 95% score will make the requirements qualifies the requirement to be taken up for development
* Track the rate of change of requirements through a mechanism, so that those areas can be covered in the next projects RE processes.
* Those areas where there is more clarity required but not immediately available should be tracked through a proper risk management process.
* Produce the final requirements specification document or product backlog document

**Conclusion**

You now have a simple, practical way to prevent the defects upfront in the requirement elicitation and analysis stage itself. If a robust methodology is developed around the solution tenets discussed in the paper, organisations may well be able to improve quality of requirements, achieve a steep reduction in the cost of finding and fixing requirements errors in downstream phases and as a result achieve better project outcomes.  People can use this method for any upcoming software development project and avoid / reduce the defects. The parameters can be customized in this model which make this model more flexible for any type of projects irrespective of domain and lifecycle.

**Reference**

* *Requirements Elicitation Problems: A Literature Analysis, Bill Davey, RMIT University, Jan 2015*
* *The Cost of Poor Quality Software in the US: A 2018 Report, Herb Krasner Member, Advisory Board Consortium for IT Software Quality (CISQ), September 26, 2018*
* *What Is the Cost of a Requirement Error?,*[*Joe Marasco*](https://www.stickyminds.com/users/joe-marasco)*- June 26, 2007*
* *Impact of Requirements Elicitation Processes on Success of Information System Development Projects, Līga Bormane1 , Jūlija Gržibovska2 , Solvita Bērziša3 , Jānis Grabis4 1–4 Riga Technical University, Dec 2016*
* *The Role of Requirements in the Success or Failure of Software Projects, April 2016, Azham Hussain1 \*, Emmanuel O. C. Mkpojiogu2 , Fazillah Mohmad Kamal, 1 School of Computing, Universiti Utara Malaysia, Sintok 06010, Malaysia, 2 School of Computing, Universiti Utara Malaysia, Sintok 06010, Malaysia, 3 School of Quantitative Sciences, Universiti Utara Malaysia, Sintok 06010, Malaysia*
* *Requirements elicitation techniques: a systematic literature review based on thematurity of the techniques, Carla Pacheco1, Ivan García1 , Miryam Reyes1 (2018)*
* *Revisiting Requirements Elicitation Techniques, Shreta Sharma, S. K. Pandey (2013)*