

# An approach of Requirements Tracing by using a multilevel Framework

Anurag Yadav<sup>1</sup>, Yashpal Singh<sup>2</sup>

<sup>1</sup>Computer Science & Engineering ,Faculty of Engineering & Technology Agra College,Agra.  
anuragyadav13@rediffmail.com

<sup>2</sup>Computer Science & Engineering,Bundelkhand Institute Engineering & Technology, Jhansi.  
Yash\_biet@yahoo.co.in

**Abstract:** This paper presents a multilevel framework for tracing the requirements. Before presenting the framework, a review in systematic fashion was conducted and based on the results of that we tried to find out the gaps between the actually implemented requirements traceability practices and the industrial needs. After that, two companies have been interviewed to find out about their requirements traceability practices. Based on these interviews these companies can be classified in to two categories: high end traceability users and low end traceability users. This categorization is based upon their requirements traceability practices that they are currently using. Then based on the results of the systematic review and industrial interviews a multilevel model for tracing the requirements was presented. This model was able to trace functional as well as non functional requirements. In that model there are three levels named as : level 0,level 1 and level 2. The level 0 is specifically focusing on pre RS requirements traceability, but level 1 and level 2 are concentrated on post RS traceability. The traceability of functional requirements are being discussed in level 0 and level 1 while level 2 discusses the traceability of non functional requirements. Finally the framework is validated by both the concerned companies. The framework is equally useful for requirements traceability practices of both the different types of companies. The validation of the framework ensured that it can be used in real industrial

environment for tracing the functional and non-functional requirements. The framework is equally useful for requirements traceability practices of both the different types of companies. The validation of the framework ensured that it can be used in real industrial environment for tracing the functional and non-functional requirements.

## Introduction:

Requirement Engineering can be defined as a collection of activities that can be used for discovering, documenting and maintaining a collection of requirements for a given system. Requirement Engineering practices can be categorized as :

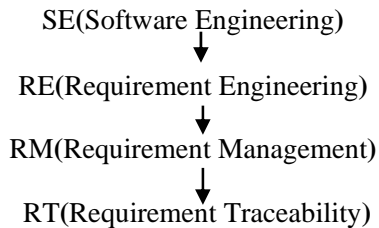
**Elicitation of requirements-** It is the process by which the customers and engineers discover, review and verify the requirements of users. Different techniques such as interviews, surveys and questionnaire are used for elicitation.

**Analysis-** In this different inconsistencies and conflicts are detected and resolved.

**Specification-** It is the process by which all the discussed and agreed requirements are provided a proper documentation. This documentation is named as Software Requirements Specification(SRS).

**Validation** – It ensures the completeness and consistency of the SRS.

**Requirement Management-** It helps in managing the evaluation of requirements. It is also involved in requirement changes. One of the main tasks of requirement management is the assurance of requirements traceability.



Tracing the requirements can also be defined in terms of describing and following the life of a requirement in backward and forward direction.

**Backward Traceability-** Links the requirements to its origin(a document or a group of persons).

**Forward Traceability** – Links the requirements to the different artifacts of the SDLC.

**Pre RS traceability-** Pre RS traceability relates the requirements to their origin and other requirements. The origin can be stakeholders. Business rules or previous rules.

**Post RS traceability-** It relates the requirements to test cases and ensures that different components satisfy them.

Traceability can be classified into the following types :

**Backward from traceability** – Relates requirements to their sources.

**Forward from traceability-** Links requirements to the design and implementation.

**Backward to traceability** – Relates design and implementation back to the requirements.

**Forward to traceability-** Relates other documents(such as operational manuals, description of system functionality) to the concerned requirements.

### **Conducting a review and its results-**

Before conducting the industrial interviews a review of all the existing materials, papers and articles was conducted. This review was also aimed to find the different problems faced by industry and academia and also to know about the gaps between their needs. Different traceability tools like RETRO, Rational Requisite Pro, DOORS and traceability techniques like VBRT(Value Based Requirement Traceability), FORT(Feature Oriented Requirement Traceability) are also reviewed and their different advantages, shortcomings and implementation environment were studied.

### **Interviewing the industry personals-**

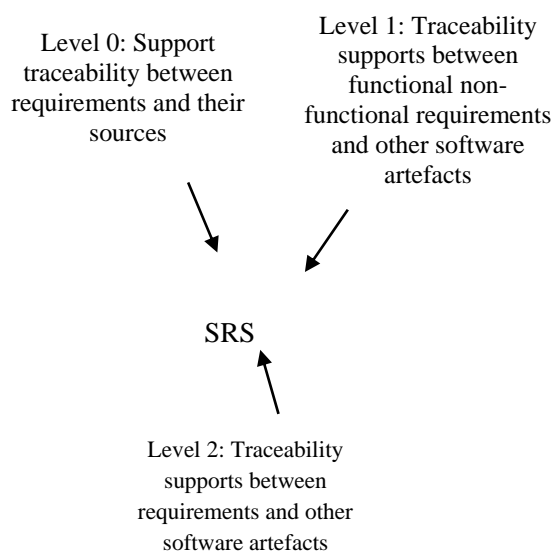
Two companies like A and B were studied based on the interviews of the requirements management experts from both. These companies have been categorized into two groups:- low end traceability users and high end traceability users. These companies are already using some traceability tools but some of their requirements were not being fulfilled by these tools. The main problem reported about company A was that the high maintenance cost of requirement management tool that they are using. Company B was facing the problems :

- (i) They had to conduct the manual traceability due to the lack of that much automated tool that was required.
- (ii) They were lacking in traceability between design and coding sections. There were also

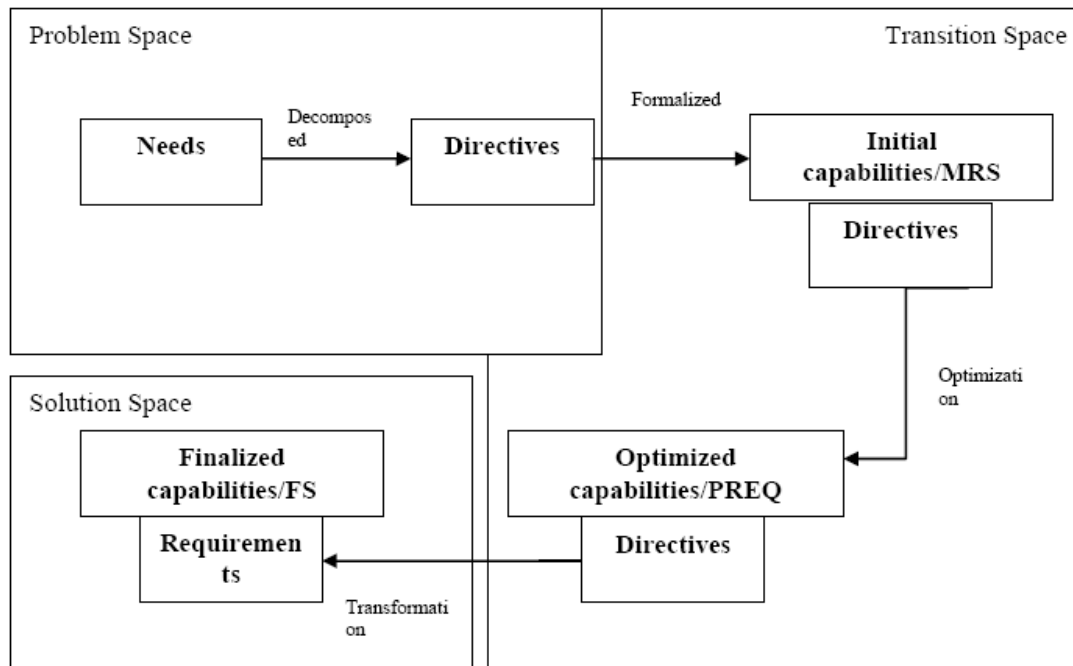
some common problems that both companies were facing they are given below :

- (a) High cost and more efforts required to manage the traceability tools and to trace the different requirements.
- (b) Should have an efficient technique to handle the different requirement change requests.
- (c) There should be some mechanism to trace the different non-functional requirements.
- (d) Requirements could not be traced back to their sources.
- (e) There should be traceability support between all phases of SDLC.

**Multi level framework-** After conducting the review and industrial interviews, based on their results a framework having three levels was being presented, that was designed to suit the high end as well as low end traceability users. That model was consisted of the three levels : level 0, 1 and 2. The level 0 was designed to handle the Pre RS traceability but level 1 and level 2 were aimed to the post RS traceability. Functional requirements were dealt by level 1 and non-functional requirements by level 2.



**Level 0 –** This level is specifically for Pre-RS traceability. For implementing the pre-RS traceability the process “Capability Engineering” is used. In this mechanism, the procedure is defined in terms of capabilities and change tolerant systems. Here the entire system is decomposed into three segments that are named as: problem segment, transition segment and solution segment. User needs are decomposed in to the different directives using the functional decomposition graphs. Directives are further decomposed into as requirements with detailed information of the system. Directives serve two purposes : (a) Help in capturing the domain information. (b) Links problem and transition segments. The formulation activity is applied in the transition segment by which initial capabilities are identified from directives. For both the companies, user requirements are converted into Main Requirement Specification(MRS) which are abstract requirements and are classified as initial capabilities. Now these initial capabilities are formulated as optimized capabilities that depends upon schedule and technology constraints, and links the transition and solution segments. The directives contain domain knowledge that must be preserved in the transition segment. In the solution segment finalized capabilities(or requirements) are obtained from optimized capabilities. For both companies functional specification behaves as the finalized capabilities. The entire mechanism is an example of forward-to-traceability. In the same way, finalized capabilities can be traced back to optimized capabilities, then initial capabilities, directives and needs and supports backward-from traceability. So new requirements can be traced back to their sources.



**Level 1** – This level is specifically for post-RS traceability with functional requirements. For tracing the functional requirements the technique 'Value Based Requirements Traceability(VBRT)' is used. VBRT is consisted of the following phases :

**Requirements Definition-** In this step functional requirements are fetched out from the SRS.

**Requirements prioritization-** In this phase the fetched requirements have to be prioritized on the basis of three parameters: value, risk and efforts. For representing the importance of any given requirement the value parameter is used. Normally this importance is shown to the stakeholders. More important requirements are traced in more detail than the less important requirements. Risk parameter represents the changing nature of requirement. More risky requirements may require more changes and have many cycles of adjustments. The third parameter effort is usually used to show the estimated time and human efforts required to

fulfill a requirement. Based on these parameters each requirement is prioritized as low, medium or high.

**Requirements packaging-** Based on the priority levels of the requirements they are adjusted in clusters. By adjusting them in clusters based on priority level architecture development can be facilitated. Sometimes for some cases this phase can be optional.

**Requirements linking-** In this phase some traceability links can be created and adjusted between requirements and some other software artefacts requirements(like test case, design documents and source code). A plan specifically for traceability can also be developed that is helpful for requirements tracing at a particular level. For this purpose requirement traceability matrix(RTM) can be used. By using RTM forward and backward both traceability types can be supported.

**Evaluation-** In that phase by analyzing the plan created in the previous phase various decisions can be taken, that reflect the overall progress of tracing the functional

requirements. So the total evaluation is performed here for example we can observe the impact of change for a specific requirement.

Various issues that are solved by level by level 1 can be stated as :-

- (i) By prioritizing requirements the associated cost and efforts with requirements tracing can be reduced significantly.
- (ii) Requirements change can be effectively managed.

**Level 2-** This level is for tracing the non-functional requirements and on post RS Traceability. The systematic review was not able to trace non-functional requirements such as security, usability etc. For tracing the NFR the mechanism of GCT (Goal Centric Traceability) is implied. GCT is consisted of the following phases:

**Goal modeling-** In this phase, the softgoal interdependency graph is used for representing the NFR by goals that can also be further divided into subgoals. Subgoals usually represent the desired qualities of the system. For achieving a goal operationalization is used. An operationalization can constitute negatively or positively to the parent goal or sometimes may not have any impact on the parent goal.

**Impact detection-** On performing some change in the NFR, there can be some components that are affected by that, they are studied in this phase. Between the components and the functional model some new links for traceability are established that can be in the form of UML diagrams or code. Stakeholders can evaluate these links and can discard the incorrect links.

**Goal analysis-** In this phase, the propagation of impact of change to all the affected zones in SIG is done. Then the role of impact of change to the satisfaction of parent's goals is reevaluated in terms of operationalization. Finally an impact analysis report is obtained that shows the effect of change on the NFR.

**Decision making-** In this step, the impact analysis report is examined by the stakeholders. Based on the result of that examination, the stakeholders decide to continue with that change or discard it. In case of continuation, the impact of change is evaluated by the developers. The main advantage of level 2 is that NFR can be traced in that level. Validation of the framework- The validation of that framework was done by both the involved companies and based on the validation the following conclusions can be drawn:-

1. It is a customizable framework.
2. All the levels of the framework are compulsory and can be implemented in the form of a tool.
3. By prioritization, the cost and efforts associated with the requirements traceability can be reduced.
4. Based on three parameters value, risk and effort requirements can be prioritized by the person owns the budget.
5. For tracing the NFR in level 2, the person should be perfectly aware of the SIG.

## 6. References

- [1] I. Sommerville, P. Sawyer, *Requirement Engineering: A Good Practice Guide*, John Wiley & Sons.
- [2] G. Kotonya, I. Sommerville, *Requirements Engineering*, John Wiley & Sons, New York.
- [3] F. Blaauboer, K. Sikkil, M.N. Aydin, "Deciding to Adopt Requirements Traceability in Practice", Springer Lecture Notes in Computer Science.

- [4] C.W. Dawson, *The Essence of Computing Projects – A Student's Guide*, Prentice Hall, Harlow UK.
- [5] A. Aurum, C. Wohlin, *Engineering and Managing Software Requirements*, Springer, Berlin Heidelberg.
- [6] C.Wohlin, P. Runeson, M. Host, M.C. Ohlsson, B. Regnell and A. Wesslen, *Experimentation in Software Engineering An Introduction*, Kluwer Academic Publishers, Dordrecht, the Netherlands