International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

# Change Impact Analysis – Integrating Traceability to IT Organizations Using Trace Requirement Artifacts Log Model (TRALM)

**Smita Tank[1], Savitha Shetty[2]**

[1]Department of Information Science & Engineering

[2]M. S. Ramaiah Institute of Technology, Bangalore, India

**Abstract:** ***During development of any software product, requirement is variant; it changes at any point of time of development or after releasing product also. For any product development customer satisfaction is needed. Customer may come at any point of time with a new requirement or modification in requirement. In an organization for any product, requirements are dependent on each other. Change at one place should be reflected to its dependent places for avoiding bugs and inviting risks. Thus discipline of requirement management – Traceability can be used for managing requirements and changes. This paper describes how traceability can be integrated and identified during development of any software through representation of a model called Trace Requirement Artifacts Log Model (TRALM). TRALM can be used effectively for Change Impact Analysis.***

**Keywords:** Requirement Management, Rationale Requirement Engineering, Traceability, bug, Change Impact Analysis, Software Development Life Cycle (SDLC), Requirement Traceability Matrix (RTM)

## 1. Introduction

In the world of modern, complex, efficient and large software products and applications, development of such applications and products require maintenance. For any successful product or application development three things should be taken care: Customer satisfaction, Verification and dependency. Whatever requirements are given by customer should be satisfied which is major success criteria of product development. To satisfy customer all the requirements should be verified. During development of product/application or after releasing a product, following possibilities can be occurred:

- Addition of new features
- Modifications in existing features
- Fixing bugs
- Adapting new technology

According to Brooks, changeability is one of the essential difficulties in software development [4]. Applying such changes to only one module or feature or requirement of software product may cause side effects to other dependent one. If developer does not apply change to corresponding dependent module or feature or requirement of the product then it may result in new bugs and loss of some information. Changes should be taken care to satisfy customer by fulfilling his/her requirements [1]. But at the same time by agreeing too many changes may hurt deadline and cost for the project. Thus before applying change, developers should have prediction of how many dependent places need to be changed. How to know which requirements are dependent on others in case of thousands of requirements for a project to apply a change?

One way to solve this is to integrate traceability concept to business framework. For such cases, in most organizations source code artifacts are taken into account, along with various types of program dependency like (such as control and data dependency [2, 3], or functional dependency or from Call dependency graph. But with this approach, still there may be chances of bugs and risks. There can be the possibility that applying change is required to change design document also, but here only source code is being traced to see what part of source code is dependent on others rather than seeing from starting phase of SDLC. This can lead to failure at later stage of life cycle which is costlier, time consuming and complex to solve it. There may be chances that instead of applying change directly to source code, if some design requirement is changed then less number of changes are required to apply. As applying too many changes is also not a good practice to follow.

In this paper all artifacts at each phase of SDLC and types of traceability such as horizontal traceability, vertical traceability, Forward traceability and backward traceability is taken into account as good practice which should be integrated in business framework of the organization.

In this paper, how traceability can be integrated to a business framework is explained by introducing a model called TRALM (Trace Requirement Artifacts Log Model). There are many benefits of adapting this model in business framework of organization which will be explained in **section 3.**

## 2. Approach

### A. Artifacts flow chain and traceability and Types of Traceability:

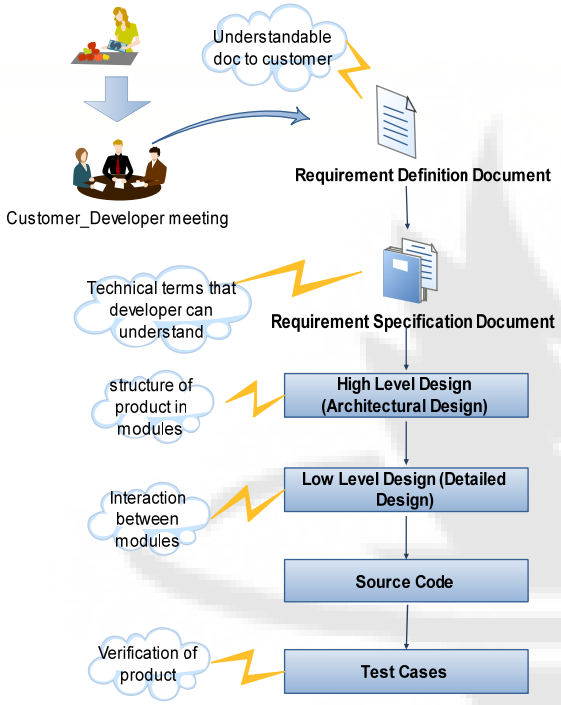


**Figure 1 :** Artifacts in SDLC

Traceability in real world can be said as "Food from farm to spoon". In technical terms, Requirement Traceability can be defined as "Requirements traceability refers to the ability to describe and follow the life of a requirement, from requirement phase to release stage (forward) and from release stage back to requirement stage (backward)" [5]. To exactly understand traceability following figure 1 can be used:

As shown in Figure 1, for developing any software product customer comes with requirements and discuss with developers, designers and other stakeholders. At this point developer will make Requirement Definition Document (RDD) in which requirements are stated in formal language which customer can understand. For each requirement of RDD, developer generates Requirement Specification Document (RSD) in which technical language is been used to state the requirements as RSD will be passed to technical employees who are going to implement it. To downgrade complexity of project, requirements are broken down according to functionalities to modules in High Level Design (HLD) or architectural design document. Low Level Design document indicates interaction between modules. Finally these requirements are implemented in source code and verification and validation of the product is performed.

Traceability is to trace all the requirements which are covered from requirement phase to release phase of product lifecycle. From Figure 1, traceability can be described as "tracing all documents at each level either from RDD to test cases (Top - Down Traceability) or from test cases to RDDs (Bottom – up Traceability)". Tracing can be performed between different modules or components of the product which can be termed as "Horizontal Traceability". While vertical traceability is tracing requirement artifacts of different phases within one module.

Once main requirement document is found for applying a change, and then easily it's all dependent documents design, code all can be located. In this way maintainability is easy to achieve while using traceability. Traceability can help in reusing the components which are already implemented. And main advantage is change impact analysis which is already explained and in this paper it is focused.

For integrating traceability to business framework one approach is to use "traceability matrix". Requirement Traceability matrix (RTM) captures all requirements during product development from starting to release phase and documents the traceability at the end of life cycle. It can be used to find links between requirements and thus to trace. But this approach doesn't work well if count of link is large. And that is why size of traceability matrix will be huge which is hard to maintain on a single document. It is used to capture small amount of requirement information [7]. Thus, integrating Trace Requirement Artifacts Log Model (TRALM) to the business framework and make a particular organization specific tool is successful approach which will result in developing quality software products.

### B. Trace Requirement Artifacts Log Model (TRALM)

Most of the software developing organizations do not follow documenting requirements throughout all stages of Software Development Life cycle (SDLC).They only concentrate directly on implementation. As explained different artifacts can be documented at different stages in Figure 1, following Figure 2 shows what standard structure of those documents can be.

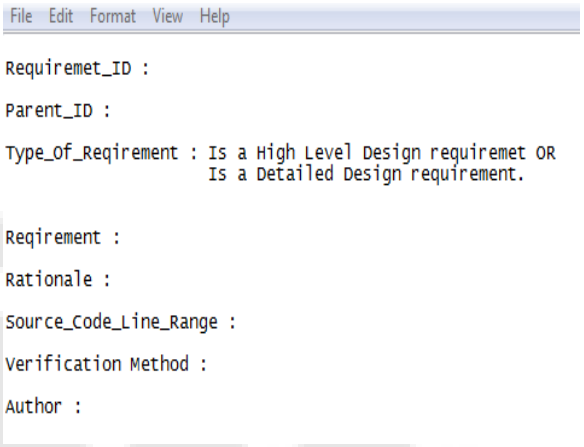


**Figure 2:** Artifact Standard Structure

One line description of above fields are as following:

1. *Requirement_ID:* It is representative ID for particular requirement.
2. *Parent_ID:* Contains dependent Requirement_IDs for particular one Requirement_ID.
3. *Type_of_Requirement:* It specifies whether it is high level or low level requirement.
4. *Rationale:* It specifies reason for existence of requirement.
5. *Source_Code_Line_Range:* It specifies range of lines of code for implementation of a requirement.

6. *Verification Method:* It specifies which verification method is used.
7. *Author:* It specifies name of the author who has developed a requirement.

**Scenario:**

Consider Railway Reservation System (RRS) to understand how traceability can be integrated and identified to the system by using TRALM.

A user will loin to system and then FROM and To stations, day of journey and type of ticket are entered and then user will request for showing available trains between requested 2 stations with schedule. After that user can book a ticket according to his convenience of time and travelling hours it takes. In this system travelling hours are calculated for one request between two stations at run time. A user will get ticket by SMS and Email.
Small part of requirements for developing RRS is taken into consideration as in following documents:

```
File  Edit  Format  View  Help

Requiremet_ID : Rail_Reserv_SRDD_05_01

Parent_ID : Rail_Reserv_SRS_05

Type_Of_Reqirement : Is a High Level Design requiremet

Reqirement : Design database to retrive data and display to user.

Rationale : For retriving schedule between specified stations with
            appropriate details of each train.

Verification Method : Inspection

Author : Nicholas
```

```
File  Edit  Format  View  Help

Requiremet_ID : Rail_Reserv_SRS_05

Reqirement : Display train schedule between specified stations by user.

Rationale : To display available trains to user. So ticket reservation can be
            done according to user convenience ( time, date, no of travelling hours,
            type of train).

Source_Code_Line_Range : NA

Author : James
```

```
File  Edit  Format  View  Help

Requiremet_ID : Rail_Reserv_SRDDLL_05_01_001

Parent_ID : Rail_Reserv_SRDDLL_05_01, Rail_Reserv_SC_05_01_002

Type_Of_Reqirement : Is a Detailed Design requirement.

Requirement : Create DB1 "List of trains" having fields like
              Train_No, Train_Name,Stop_Stations, HaltTime_PerStan,
              TotalTimeBetwn_Two_Stans.

Rationale : This DB can be used to supply data to user.

Verification Method : test_case

Author : Jesica


Requiremet_ID : Rail_Reserv_SC_05_01_002

Parent_ID : Rail_Reserv_SRDDLL_05_01_001

Requirement : Calculate toltal time taken between two stations requested by user,
              from "List of trains" DB1 and update HaltTime_PerStan field in same DB1,

Rationale : This DB can be used to supply data to user.

Verification Method : test_case

Author : Jack
```

**Figure 3:** Sample Artifacts and requirement Dependency

From above documents, by extracting Requirement_ID and Parent_ID from the document we can say that for particular one requirement is derived from or is dependent on requirement mentioned in Parent_ID field easily. For example in case of Rail_Reserv_SRDDLL_ 05_01_001, dependent requirements are Rail_Reserv_SRDDLL_05_01 and Rail_Reserv_SC_05_01_002.

These documents are written and updated by employee of the organization. Organizations can develop their own tool to extract these information, to link documents and can generate *"TraceLog"* as shown in figure 4. Available tools don't complete all needs to identify traceability. Some of the tools provide unidirectional traceability; some tools are working for some specific documents only.

Organizations can go for other traceability tools available online but scope of them will be limited compare to individual organization's requirement. Thus, according to requirement of organization, developing a Traceability tool to generate TraceLog is efficient.

TraceLog - Notepad

File Edit Format View Help

| Requirement_ID | Parent_ID | Verification_Method | Author |
|---|---|---|---|
| Rail_Reserv_SRS_05 | Empty | Empty | James |
| Rail_Reserv_SRDD_05_01 | Rail_Reserv_SRS_05 | Inspection | Nicholas |
| Rail_Reserv_SRDDLL_05_01_001 | Rail_Reserv_SRDDLL_05_01 | test_case | Jesica |
| Rail_Reserv_SRDDLL_05_01_001 | Rail_Reserv_SC_05_01_002 | test_case | Jesica |
| Rail_Reserv_SC_05_01_002 | Rail_Reserv_SRDDLL_05_01_001 | test_case | Jack |

**Figure 4**: TraceLog.TXT

Main key advantage of adapting TRALM is *change impact analysis*. From TraceLog file it can easily be done. For example in above RRS scenario at what places change must be applied can be triggered using TraceLog, is shown in next section.

### C. Change Impact Scenario and Analysis

In above scenario, suppose some train's stops are being added for stations where it was not available. Now if stations

(stops) are being added then at what all places it (CHANGE) can be impacted that can be found from TraceLog. Rail_Reserv_SRDDLL_05_01_001 should be updated for applying this change as number of stops is being increased. From Figure 4, we can find its links which should be taken into account for applying change. In this way, places where change will be impacted can be efficiently found.

This small scenario is taken just for understanding how TRALM can be integrated to any application. But effect of adapting TRALM can be analyzed when there is a large project having thousands of documents.

Before applying a change stakeholders can find impact of applying a change. If one change is creating many places to be changed then if it is hurting deadline or it costs more to apply change then it can be refused.

## 3. Benefits and Future Trends

- *Change Impact Analysis* is key advantage for adapting TRALM.
- During development of software product, many developers and designers discuss about rationale for why particular design has been designed and developed, but at later stage in case of large and complex project or after few years of releasing, it is difficult to know by whom it was developed and for what reason it was designed and developed. Thus, TRALM can result in better *understanding of existence of requirement.*
- If there are any *risks* to apply a change then it can be detected before applying a change.
- *Management decisions* can be taken if applying a change is hurting cost and deadline.
- Finding impacted places manually directly in code can be missed. Thus TRALM can help in *avoiding human errors.*
- *Customer satisfaction* is very important while developing any product. Only looking from TraceLog, customers can verify whether all requirements are implemented or not.

A tool which works for TRALM can be developed. So some organizations can use it if they follow same document structure as it is structured in TRALM.

## 4. Conclusion

By adapting Trace Requirement Artifacts Log Model (TRALM), the major problem of software engineering *"Change Impact"* can be solved efficiently. In addition, TraceLog can be used for future purpose; it can clear understanding of customer or developer or designer by rationale for existence of any requirement at any point of time; Management decisions become easier by integrating TRALM to the business framework of the organization. Cost and time saving can be achieved through TRALM.

## References

[1] S. Bohner and R. Arnold, "Impact analysis - towards a framework for comparison," Proceeding of the IEEE International Conference on Software Maintenance, Sep. 1993, pp. 292-301.
[2] J. Ferrante, K. J. Ottenstein, and J.D. Warren, 'The Program Dependence Graph and its use in optimization', *ACM Transactions on Programming Languages and Systems*, 9 (3), July 1987, pp. 319-349
[3] Podurski and L.A. Clarke, 'A Formal Model of Program Dependences and its Implications for Software Testing,Debugging and Maintenance', *IEEE Trans. on Software Engineering*, 16 (9), Sept. 1990, pp. 965-979
[4] F. P. Brooks, "No silver bullet'" IEEE Computer, vol. 25, no. 1, pp.91-94, February 2008
[5] Requirement traceability: http://en.wikipedia.org/wiki/Requirements_traceability
[6] Documenting Requirements Traceability Information:A Case Study : Helsinki University of Technology, Department of Computer Science and Engineering, Virve Leino
[7] Requirement Traceability Matrix (RTM): http://www.tutorialspoint.com/software_testing_dictionary/requirements_traceability_matrix.htm

## Author Profile

**Smita Tank** received B.E degree from Veer Narmad South Gujarat University in 2011. Then she joined ACE academy for GATE preparation in Hyderabad in during 2011-2012. She cleared GATE exam in 2012 and pursuing M.Tech in M.S. Ramaiah Institute of Technology for which degree will be received to her in 2014. During M.Tech took industry training for 9 months 2013 to 2014. And now she is an employee in the same industry.