

A Survey on Importance of Requirement Traceability in Software Engineering

Falak Khursheed, Mohammad Suaib

CSE Department, Integral University, Lucknow, India

Abstract— In this paper the main topic of discussion is about requirement traceability, problems associated with it and the appropriate solution to those problems. The solution given is not exactly up to the mark because requirement traceability itself is the most challenging task. In this paper many research papers are analyzed on traceability and with much research investigation some techniques are collected which are beneficial to trace it. Some comparison regarding techniques used in different papers are also discussed with their benefits. At the end this paper is summarized by providing the result or outcome that according to all the research made is the solution to the traceability problem.

Index Terms—Elicitation, Forward and Backward Traceability, Traceability, Traceability Technique.

I. INTRODUCTION

The most important goal of all the software development organization is to meet the requirement that is envisioned. Their goals may be differ depending on their measured assessment of situation of the environment in which they work and the task they wish to complete. In this paper overview of requirement has been discussed. Traceability related terms are also discussed. This paper also consist of review study on problems of traceability, the techniques used in tracing requirements and comparative study of all the techniques used in different research papers for requirement traceability. Each of the proposed solution has its own shortcomings. Although with so many researches, requirement traceability still remains the most problematic area. At last a brief study about future scope of requirement traceability has been talked about.

A. Concept of requirements

Requirement can be described as the need or necessity of user according to the prevailing condition. Requirement can said to be a continuous phenomenon that does not stop at point of time else it keeps on changing according to the situation. In the engineering approach requirement can be described as inputs to the design stages of product development for verification process. If we are talking about the nature of requirement, it is not static but dynamic that keeps on changing and evolving throughout the software development [1].

We can classify requirement into two categories:

- Resource Requirement that describe the **property of resource**.
- Requirement of **methodology** used by the developing organization.

B. Requirement Elicitation

Requirement elicitation is an iterative facility. It is a process of seeking, acquiring, collecting, and uncovering, elaborating the requirement. It is about emergence and development of requirement. Requirement can be spread out to many sources such as stakeholders, users, customers, developer's etc. Some techniques like interviews, questionnaires, card sorting etc are used to gather requirement.

C. Requirement Traceability

Traceability can be defined as the degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor or master-subordinate relationship to one another [4]. Requirement traceability can be defined as the method of finding the origin of each requirement both in forward and backward direction and to analyze changes that were made to this requirement. Traceability is important as it understands relationship between the desired requirement and the delivered system. It lowers the risk and maintains consistency.

II. TRACEABILITY RELATED TERMS

- **Forward Traceability** means that requirement that can occur in future can be analyzed after the changes are made [3]. If the requirements are traced in this manner it will help the designers to convert the requirements into blocks that target a particular module of the system based on the block requirement.
- **Backward Traceability** means that requirement can be analyzed from the origin or source of the development phase [3]. If backward tracing starts from low level requirement that it can help unit testers to see what part of their module plays in the big picture of the system based on requirement. By finding the defect in low level requirement changes can be done accordingly in high level requirements.
- **Inter-requirements traceability** describes the relationships between requirements [5]. Inter-requirements traceability is important for requirements analysis and to deal with requirements change and evolution.
- **Extra-requirements traceability** The links between requirements and other artefacts are captured by extra-requirements traceability.

III. NEED OF TRACEABILITY

- The effect of changing requirement can be estimated.
- In future the development system will require less time and effort as past implementation decision can be utilized.
- The product quality can be evaluated with respect to user requirement.
- It will be easier for designer to verify the design requirement.

IV. TRACEABILITY TECHNIQUE

In this section all the manual traceability techniques are mentioned such as cross reference, document and structured centered. Traceability is very hard nut to crack since the requirements keep on changing throughout the lifecycle of software. It is very difficult task for testing team to trace the dynamic requirement without hindering the ongoing process. So here some of the basic techniques to handle this process is mentioned with their small description in the table given below.

Table 1

Technique	Description
1. Cross reference centered	In this technique documents that hold online forms are supported automatically by linking ends of cross reference hyper textually. Examples of this scheme are those based on some form of explicit requirements tagging, numbering, or indexing [6].
2. Document centered	In Document centered techniques requirement can be traced by covering either all or part of content of the project documentation [6].
3. Structure centered	This technique helps in achieving the requirement traceability by restructuring the document in the form of graph or network [6].

V. ADVANTAGE OF TECHNIQUES

- The above mentioned techniques provide early feedback from the customers by which the analyst team gets more time to respond to the changes in requirement. Another important benefit is that verification process becomes easier to implement.
- Sometimes it happens that some requirements gets missed which can be identified by cross reference.
- The graph or network view of particular project or document will be easier to trace requirement as graphical representation is easy to implement.

VI. COMPARATIVE STUDY

Table 2

Paper	Technique	Advantage
I. A new model for requirement to code traceability .	Requirement to method, classes, packages.	In this approach functional requirement has been traced. It is helpful in achieving automated requirements to code traceability [22]. Another important benefit of this approach is that it establish traceability link between forward to backward and backward to forward direction.
II. Traceability technique- A critical study.	Parameters such as scope, coverage, validation, levels of details, Domain specific have been used to trace requirement.	The techniques used are close to each other so they can be used in combination to achieve the desired level of traceability [7, 8, 9, 10]. These techniques are less risky, less expensive and are used to reduce delay.
III. A systematic mapping on supporting approaches for	Traceability matrix, vector space model, Latent semantic	This technique is most satisfactory to view broad area of traceability and it involves graph implementation for

<p>requirement traceability in the context of software projects.</p>	<p>information.</p>	<p>better visualization [11, 12, 13, 14, 15, 16, 17, 18]. This technique keeps track of all the business requirement. It checks the consistency among requirement document. If manual tools are used it can also be used with automation projects.</p>
<p>IV. Requirement traceability and the effect on the system development lifecycle.</p>	<p>Use case methodology.</p>	<p>Use case are very useful as they provide traceability throughout the lifecycle [21]. All the functional requirements are covered.</p>

While most of the ideas mentioned followed the manual approach but it was felt that some realistic approach can provide some better solution and it can be further enhanced. It would be interesting to see the outcome with companies who have no RT processes in place, and others that do.

X. CONCLUSION

This paper is all about the need of requirement. How to trace the origin or source of requirement is the main goal of this paper. Some of the techniques used for capturing traceability with advantage of each one is also defined. In particular, the main topic highlighted is the comparative study of all the techniques used by different researchers and comparison is done in terms of time and effort. This paper also draws some attention towards requirement traceability problem, the main cause of those problems and solutions to those problems. At last it can be concluded that all the related terms, techniques, problems associated to traceability is briefly discussed in this paper. And by reading so many research works on this topic it can be said that it needs some realistic approach to reach at the solution.

REFERENCES

- [1] O. Gotel and A. Finkelstein, "An Analysis of the Requirements Traceability Problem", Proceedings of the 1st International Conference on Requirements Engineering, 1994, pp. 94-101.
- [2] <https://en.wikipedia.org/wiki/Requirement>.
- [3] Jarke, Matthias (1998, December). Requirements tracing. Commun. ACM 41, 12, 32-36.
- [4] IEEE, IEEE Standard Glossary of Software Engineering Terminology, Std. 610.12-1990.
- [5] Alford M.W. A requirements engineering methodology for real-time processing requirements. IEEE Transactions on Software Engineering, 1977 January; 3(1):60-69.
- [6] Gotel, O. 1995. Thesis: Contribution Structures for Requirements Traceability. London. Imperial College of Science, Technology and Medicine, University of London. 354 p.
- [7] B. Ramesh and M. Jarke, "Towards Reference Models for Requirements Traceability". IEEE Transactions on Software Engineering, 27(1), January 2001.
- [8] A. Egyed, "A Scenario-Driven Approach to Trace Dependency Analysis", IEEE Transactions on Software Engineering, Vol. 29, No. 2, February 2003.
- [9] K. Pohl, "PRO- ART: Enabling Requirements Pre-Traceability", Proceedings of ICRE '96, IEEE 1996.
- [10] Gotel & A. Finkelstein, "Extended Requirements Traceability: Results of an Industrial Case Study", IEEE 1997.
- [11] Naslavsky, L.; Ziv, H.; Richardson, D. J. Towards traceability of model based testing artifacts, Proceedings of the 3rd international workshop on Advances in model-based

VII. REQUIREMENT TRACEABILITY PROBLEM

The first and foremost problem of traceability is that it is misunderstood and it does not have single modeling method. The manual method used in capturing traceability for large software projects need lots of time and effort. These manual methods also cause errors which are very difficult to catch. From some investigations and reports it was found that main cause of requirement traceability problem is information that are hidden, the complexity in information, Lack of commitment by all parties, immature integration technology [23,24,25,26]. Problem is not uniform. It is influenced by the different user, project, task and information requirement.

VIII. SOLUTION OF TRACEABILITY PROBLEM

The traceability problem can be solved if more realistic technique like information retrieval is used. Requirement should be checked using traceability matrix as it checks both the forward and backward requirement as the changes are done in the system. If ones the origin or source of requirement is gathered it is easier to trace the remaining requirement [27, 28].

IX. FUTURE WORK

The idea of tracing requirement presented in this paper has many opportunities for further expansion and research.

- testing, p.105-114. London, United Kingdom. July 09-12, 2007.
- [12] Dekhtyar, A.; Hayes, J. H.; Smith, M. Towards a model of analyst effort for traceability research. In Proceedings of the 6th International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE '11). ACM, New York, NY, USA, 58-62. 2011.
- [13] Schwarz, H.; Ebert, J.; winter, A. Graph-based traceability: a comprehensive approach. *Softw. Syst. Model.* 9, 4 (September 2010), 473-492. 2010.
- [14] Guo, Y.; Yang, M.; Wang, J.; Yang, P.; Li, F. Ontology Based Improved Software Requirement Traceability Matrix. In Proceedings of the 2009 Second International Symposium on Knowledge Acquisition and Modeling - Volume 01 (KAM '09), Vol. 1. IEEE Computer Society, Washington, DC, USA, 160-163. 2009.
- [15] Omoronyia, I.; Sindre, G.; Stålhane, T. Exploring a Bayesian and linear approach to requirements traceability, *Information and Software Technology*, v.53 n.8, p.851-871, August, 2011.
- [16] Hussain, T.; Eschbach, R. Automated fault tree generation and risk based testing of networked automation systems. *Emerging Technologies and Factory Automation (ETFA)*, 2010 IEEE Conference on, vol., no. pp.1,8, 13-16 Sept. 2010.
- [17] Jiang, H.; Nguyen, T. N.; Chen, I.; Jaygarl, H.; Chang, C. K. Incremental Latent Semantic Indexing for 2008. 23rd IEEE/ACM International Conference on, vol., no., pp.59, 68, 15-19 Sept. 2008.
- [18] De Lucia, A.; Oliveto, R.; Tortora, G. Adams re-trace: traceability link recovery via latent semantic indexing. In Proceedings of the 30th international conference on Software engineering (ICSE '08). ACM, New York, NY, USA, 839-842. 2008.
- [19] Sultanov, H.; Hayes, J. H. Application of reinforcement learning to requirements engineering: requirements tracing. *Requirements Engineering Conference (RE)*, 2013 21st IEEE International, vol., no. pp.52,61, 15-19 July 2013.
- [20] Eyal-Salman, H.; Seriai, A.-D.; Dony, C. Feature-to-code traceability in a collection of software variants: Combining formal concept analysis and information retrieval. *Information Reuse and Integration (IRI)*, 2013 IEEE 14th International Conference on, vol., no. pp.209, 216, 14-16 Aug.2013.
- [21] Kulak, Daryk & Guiney, Eamonn (2000). *Use Cases: Requirements in Context*. New York, New York: ACM Press.
- [22] IBM (2013). Requirement management for system and advanced IT applications. Retrieved May 27,2013, from <http://www.ibm.com/software/andtools/doors/>.
- [23] Wright, S. (1991). Requirements Traceability - What? Why? and How?, Tools and Techniques for Maintaining Traceability During Design, IEE Colloquium, Computing and Control Division, Professional Group C1 (Software Engineering), Digest Number: 1991/180, December 2, pp. 1/1-1/2.
- [24] Robinson, D. (1991). CASE Support for Large Systems, in Van Lamsweerde, A. and Fugetta, A. (Eds.), *ESEC '91: 3rd European Software Engineering Conference*, Milan, Italy, October 21-24, Springer-Verlag, pp. 504-508.
- [25] Hoffman, D. (1990). On Criteria for Module Interfaces, *IEEE Transactions on Software Engineering*, Volume 16, Number 5, May, pp. 537-542.
- [26] Brown, A. W., Earl, A. N. and McDermid, J. A. (1992). *Software Engineering Environments: Automated Support for Software Engineering*, McGraw-Hill.
- [27] G. Anoniol, G. Canfora, G. Casazza, A. De Lucia, "Information Retrieval Models for Recovering Traceability Links between Code and Documentation", Proceedings of the International Conference on Software Maintenance, 2000, pp. 40-49.
- [28] [Anoniol 02] G. Anoniol, G. Canfora, G. Casazza, A. De Lucia, E. Merlo "Recovering Traceability Links between Code and Documentation", *IEEE Transaction on Software Engineering*, Vol. 28, No. 10, pp. 970-983, October, 2002.

AUTHOR'S PROFILE

Falak khursheed has completed BTECH in CS from Dr.M.C Saxena College of Engineering and Technology, Lucknow. Now pursuing MTECH in CS from INTEGRAL UNIVERSITY, Lucknow.

Mohammad Suaib is pursuing PHD from Integral University and working as Assistant Professor in Integral University.