

An Experimental Analysis of Software Requirement Traceability and Enhancement of Quality of Traceability Process using Quality Assurance Interface

Ankur Aggarwal, Rajwinder Kaur

Abstract: *In this paper, we discuss the importance of requirement traceability in software development process. In order to produce quality requirements, they prerequisite is that it should be quantified and traced throughout the project lifecycle. A new technique is also proposed using Quality Assurance Interface to enhance the quality of requirement traceability process. This proposed model will ensure that the requirements mentioned in the database are appropriately met by creating flags against each requirement with the help of interface. It will also mention the requirements that need be redone again.*

Keywords: QA, Traceability

I. INTRODUCTION

All the software development organizations aim to achieve all the requirements that are desired and proposed by the stakeholders. The degree to which a relationship can be dependable between two or more outcomes of the development process is called as Traceability [1]. Relationships among these outcomes may be successor-predecessor or master-subroutine. Requirement traceability can be also be defined as the process of inventing the source of all the requirements which can be derived from both forward and backward direction. Since requirements keep evolving, it's another important task is to search for the changes that are applied to these requirements. Traceability plays a very important role as it comprehends and maintain the relationship between the desired requirements and the finally delivered system by maintaining the consistency and minimizing the risks involved during the development process. Traceability is carried out to ensure that the requirements are quantified and managed appropriately throughout the lifecycle. It plays a very important role in business analysis as the requirements keep on evolving and changing throughout the development process and are never static [2].

Traceability offers a proficient technique to identify software errors which occur due to a erroneous state or behavior of the system and remains static throughout the development process. It helps to identify the code which is directly

connected to the proposed requirement and can avoid any kind inconsistency or discrepancy that may occur during the development. This helps the system to avoid errors and failures that may be very expensive to dispense in the later stages.

Any change in the high level requirements generates the need to make changes in the low level requirements which substantiates the need for requirements traceability. Requirements traceability is most advantageous and apparent when the system changes and evolves. It has a great impact on the software testing and quality as it benefits the system by identifying which high level requirement will not be achieved if a low level requirement fails during the testing process. Moreover, if an error occurs, all those fragments can be identified, documented and reviewed that might be influenced due to traceability process. Therefore, Requirements traceability is identified as a concern in an increasing number of standards and guidelines for Requirement Engineering [3]. This concern has been emulated by all the systems that have been developed or are going through the development process [4].

Traceability allows the documentation of dependencies between the various requirements and establishes an imaginary line from the requirements to related items such as test cases.

II. REQUIREMENT TRACEABILITY WATCHWORDS

A. Requirement Traceability Matrix

Requirement Traceability Matrix (RTM) is the most efficient and common way of ensuring the existence of requirement traceability to the full extent. The traceability matrix explicitly verifies that all the requirements that are either stated or derived are linked with the corresponding project design elements, components, modules and other deliverables. This method is known as forward trace. This matrix is also used to backward tracking in which the matrix will verify and document the original source of the requirements. It may also question the stakeholder regarding the reference of certain features which may not be required in the system. This is known as backward trace.

Revised Manuscript Received on May 17, 2019.

Dr.(Ms.) Ankur Aggarwal, Assiatant Professor, Computer Science Department, DAV Centenary College, Faridabad, India.

Ms. Rajwinder Kaur, Assiatant Professor, Computer Science Department, DAV Centenary College, Faridabad, India.



B. Forward Traceability

The relationship between the functional requirements and system requirements and the different objects created during the design, development and testing of the system play an important role in forward traceability. It signifies that the analysis of the newly generated requirements is possible after implementing the changes to it [5]. Designers can easily translate the requirements into blocks that target a module based on the requirement if the forward trace is followed.

C. Backward Traceability

Since diverse stakeholders will have diverse views of the requirements, a requirement traceability matrix must be created to trace each developed feature back to the group or entity that requested it during the requirement gathering. Backward trace denotes the need to specify the origin of all the requirements defined in Requirement Specification (SRS). This process signifies that the requirements can be analyzed from the origin or source of the development phase [5]. By identifying the defect in low level requirements using the backward trace, unit testers can foresee which part of their module participates in the bigger module of the system based on requirements and accordingly the changes can be done in high level requirements.

D. Inter-requirements traceability

Requirement transformation and evolution are important for requirement analysis. Inter-requirements traceability plays a crucial role in describing the relationships between the requirements using an inter-requirements matrix [6]. This matrix will display the complex relationships among the requirements to be analyzed.

E. Pre - Requirement Traceability:

Using traceability, a stated process can be traced back to the person who had given the requirement for it. This can help to assign the priority to the requirement and identify to what extent the requirements is important. This arises from the fact that requirements come from different people with different requirements of the product.

F. Post Requirement Traceability:

A requirement relationship with all the products associated with it exists such as models, test cases, test results, analysis results etc. which can be traced using requirement traceability. This proves to be a great help in business analysis.

III. NEED AND IMPORTANCE OF REQUIREMENTS TRACEABILITY

During the software development, the requirements are elicited according to the demand of the customers. The requirements need to be changed several times through the development process to meet the needs of the stakeholder. Therefore, establishing the technical requirements is a crucial part during the design, development and testing of a new software system. During the implementation process, the impact brought by the requirement change has to be estimated well in time. The resources associated with the product also

need to be assigned appropriately for the smooth implementation process. Requirements traceability supports to control the total time involved in implementation, a project's cost which might increase at a high rate if the risks are not subjected timely, and the quality of the final product.

A. Recognition of Need

A considerate amount of knowledge about the users and the most important principles for the requirements must be understood by the developer before the process of software design begins. The developer must be able to trace a requirement back to its need that is an essential component of the proposed project which can be achieved by backward trace. By inspecting each need, any of the missing, incorrect or extra requirements can be identified early in the design or implementation process.

B. Prioritizing the requirement

Prioritizing requirement analysis is an important task which might depend on the customer's claim. Traceability helps us to effectively prioritize the requirement by evaluating the value of each requirement in accordance with the appropriate business objective.

C. Antedating Modifications

Any modifications or changes implemented before a system has been completely redesigned results in a change in the system. Requirements traceability allows to keep track of all the possible changes in the requirements with resulting changed system. Tracing requirements through the different phases of development process shows whether a requirement needs to illustrate the related business objectives and a requirement has been successfully addressed or not. Effective traceability at the right time and right place helps to gain a better understanding of what type of changes are needed and the developer and the project manager must be able to antedate what is required to adjust to these changes made in the system.

D. Rationalizing the Testing Phase

Testing usually entails a specific amount of risk of occurrence of a problem or failure and its impact on the organization. Testing each and every requirement from the repository is a time-consuming and expensive process. Therefore, traceability assists the project manager and the development team to determine in what areas the requirements needs to be tested.

E. Assuring Project Accomplishment

The prime objective of requirements traceability is to achieve the right product outcome by avoiding the implementation of



unnecessary requirements. It also helps in assuring the project completion, controlling the cost which may arise otherwise and preventing any delays due to temporary failures. Throughout a project's development process, traceability guarantees the availability of adequate resources such as time, manpower and money. This will enable the system to code, test and verify the project requirements effectively. The better available resources, the better will be the project outcome resulting in an efficient system.

F. Evaluation of prospective changes

Traceability helps in quick and easy assessment of any change implemented in the system during its development. It also supports in identifying the requirements associated with a failed test case. This helps in enhancing the process of resolving the problems.

G. Monitor and control across the lifecycle of requirements

The requirements also go through a complete lifecycle during the development of a process. Requirements may be validated, few might be pending, few rejected etc. during this process. A traceability matrix can be used to help manage the requirements which can be further stored in the repository accordingly for easy access.

IV. FRAMEWORK TO ENHANCE THE QUALITY OF TRACEABILITY PROCESSING QUALITY ASSURANCE INTERFACE

Requirement traceability is a very important phase that needs to be acknowledged in the early phases of a project life cycle. Few important points are required to be mentioned before the traceability begins:

- To specify the size of project, its complexity and the risks involved.
- Estimated Cost and available resources.
- Time constraints defining the strict delivery dates.
- Knowledge of complex system and tools with business analysis experience of the project and organization.
- Knowledge about the advanced tools. Complex traceability requirements require advanced tools and greater knowledge.

Many techniques have been practiced for Requirement Traceability including: key phrase dependencies [8], cross referencing schemes [9], matrices [10], templates [11], matrix sequences [12], integration documents [13] and many more. A new technique is proposed which uses Quality Assurance Interface to enhance the quality of Requirement Traceability. To start with the traceability process, the business analyst must use a specialized tool to add the requirements into a spreadsheet which are imported into the database table. The interface is called Quality Assurance Interface (QAI) which a quality assurance specialist may use to verify that all the requirements are met efficiently for the desired code which is

being checked. The proposed framework is designed to address the concern of verifying and validating that the desired requirements are actually met. Through this framework, the software engineer can either create a flag or associate a tag to the requirement which is met. QAI accomplishes two tasks: one is to ensure and tag/flag the requirements which are met. Flags are used in QAI to indicate the status of the requirement and code verification. CVS flag states that a table is generated which will have all the file names and line numbers which satisfy the requirements whenever the source code is checked into a version control system. Another flag NCF signifies that the requirement is not been verified by the QAI and does not have any corresponding code in the database. Second task performed by QAI is to report the requirements or the source code which do not match. When a source code is checked, all the file names and line numbers which satisfy certain requirements are stored in the table. QAI parses this table to ensure that the requirements are achieved accordingly as specified in the database. Initially, QAI will identify the requirements from the database by reading its description that has been linked with the requirements of the database. In the next step, the QAI will identify the file and the line from file name/line number combination from the match stored in requirement traceability repository and then evaluate the source code according to the match to identify if the match with the file/line code is found or not and the requirement is met or not. If the match is checked, next step will be to create a flag which will signify whether the requirement has been accomplished and has been sufficiently met. If the requirements are not met accordingly, a message is sent to the developer to redo the source code again.

An important task of QAI is to handle the flags for software requirements. There are nine different flags that can be assigned to the requirement.

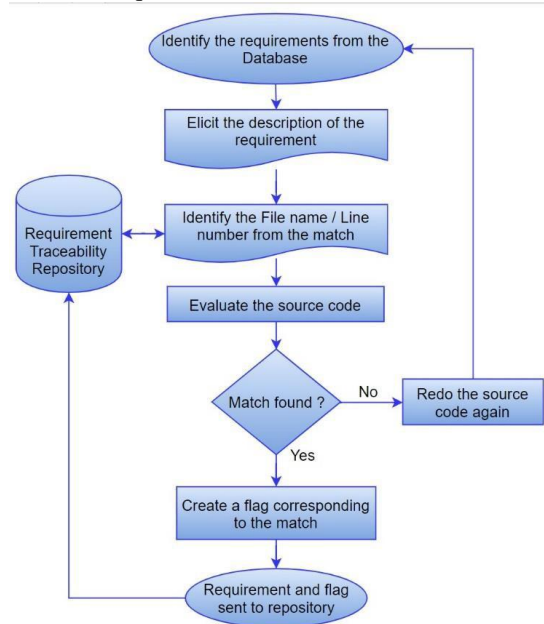


FIG.1: PROPOSED FRAMEWORK FOR ENHANCING THE QUALITY OF TRACEABILITY PROCESS



The QAI table is reset to Requirement Changes Flag (RCHF) to display the changes made to the requirement by QAI. Also, all the requirements listed in the related requirements field will have QAI flag reset as Related Requirement Changed Flag (RRCF). Only the source code that relate to the requirements that are essential to be changed will be reviewed with these two flags to guarantee that the new requirements are achieved.

V. CONCLUSION

Since achieving software quality has been a challenge since the inception of software, software requirements gathering, analysis, and specification are assessed as the primary cause of many of the software multifarious problems by many researchers. Requirements traceability is considered as one of the most significant and challenging job in achieving flawless and succinct requirements. In order to produce quality requirements, they need to be indicated and traced all through the software life cycle. We have illustrated the multifaceted nature of Requirement Traceability. In conclusion, to achieve improvement in the quality of requirement traceability process, there is a need to refocus on the enhancement techniques to facilitate the matching of the desired requirements and the actual requirements met. The proposed framework provides a platform to match the requirements in a sequential manner using QAI which will enable flags to describe each requirement being met or pending.

REFERENCES

1. IEEE, IEEE Standard Glossary of Software Engineering Terminology, Std. 610.12-1990.
2. 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.
3. Dorfman, M. & Thayer, R.H. (1990). Standards, Guidelines, and Examples on System and Software Requirements Engineering, IEEE Computer Society Press Tutorial.
4. IEE. (1991). Tools and Techniques for Maintaining Traceability During Design, IEE Colloquium, Computing and Control Division, Professional Group C1, Digest No.: 1991/180.
5. Jarke, Matthias (1998, December). Requirements tracing. Commun. ACM 41, 12, 32-36.
6. Alford M.W. A requirements engineering methodology for real-time processing requirements. IEEE Transactions on Software Engineering, 1977 January; 3(1):60-69.
7. Gotel and A. Finkelstein. —An Analysis of the Requirements Traceability Problem, Proceedings of the First International Conference on Requirements Engineering, Colorado Springs, April 1994
8. Jackson, J. (1991). A Keyphrase Based Traceability Scheme, in [25], pp. 2/1-2/4.9.
9. Evans, M.W. (1989). The Software Factory, John Wiley and Sons
10. Davis, C.G. & Vick, C.R. (1977). The Software Development System, IEEE Transactions on Software Engineering, Vol. 3, No. 1, pp. 69-84
11. Interactive Development Environments. (1991). Software Through Pictures: Products and Services Overview, IDE, Inc.
12. Brown, P.G. (1991). QFD: Echoing the Voice of the Customer, AT&T Technical Journal, March/April, pp. 21-31.
13. Lefering, M. (1993). An Incremental Integration Tool Between Requirements Engineering and Programming in the Large, Proceedings of the IEEE International Symposium on Requirements Engineering, San Diego, California, Jan. 4-6, pp. 82-89.

AUTHORS PROFILE



Quality Assurance in Software Engineering and Big data.

Dr. (Ms.) Ankur Aggarwal has done Master degree in Computer Science and has done Doctorate in the field of Software Engineering. She is Assistant Professor in DAV Centenary College, Faridabad having teaching experience of more than 12 years. She has more than 13 publications in National and International conferences and Journals. Her research area in Computer Science is



Ms. Rajwinder Kaur, has done Master degree in Computer Science and has done M.Phil also. She has also qualified UGC NET. She is Assistant Professor in DAV Centenary College, Faridabad having teaching experience of more than 9 years. Her research area in Computer Science is Software Engineering.