A Discussion on Traceability Matrix and Challenges

Sachhidanand
Asst. Professor, Department of Computer Science, Govt. First Grade College Aurad(B), INDIA

ABSTRACT
Requirements traceability is a critical component of any rigorous software development process. Requirements traceability also helps to control and manage evolution of a software system. The challenges of implementing successful and cost-effective traceability have created a compelling research agenda that has addressed a broad range of traceability related issues. This paper attempts to address the need by studying the factors that make traceability important and discusses the challenges facing traceability practices in industry.

Keywords---- Requirements Traceability, Requirements Traceability Tool, Test case, Defects.

I. INTRODUCTION

A requirements traceability matrix is a document that traces and maps user requirements with the test cases. The purpose is to make sure that all the requirements are covered in test cases so that while testing no functionality can be missed. Performing a requirements traceability analysis is an important part of the software engineering process as it ensures that all of the requirements have been adequately considered during each phase of the project. To make sure that the developed system does not have any missed requirements. Traceability matrices can be established using a variety of tools including requirements management tool, databases, any spreadsheets or even with tables or hyperlinks in a word processor.

In order to discuss how using requirements traceability is more beneficial than not tracing requirements and need to discuss the traditional software gathering methodology of the Software Requirements Specification (SRS). The SRS is generally a document-centric approach to requirements gathering where the deliverable is a main requirements document.[2]

II. TYPES OF TRACEABILITY MATRICES AND TECHNIQUES

- Forward Traceability: This kind of Traceability matrix is used to Mapping of requirements to test cases.
- Backward Traceability: Its used to Mapping the test cases to the requirements.
- Bi-Directional Traceability: It is used in software testing, the references from test cases to basis documentation and vice versa.

\[\text{Fig 1: Bidirectional (Forward & Backward) Traceability[1]}\]

The traceability can be achieved by using one or more of the following techniques.

- **Cross referencing**, involves embedding phrases like "see section x" throughout the project documentation. For example, tagging, numbering, or indexing of requirements, and specialized tables or matrices that track the cross references.

- **Specialized templates and integration or transformation documents.** Used to store links between documents created in different phases of development. These documents are useful in the all the phases of development.

- **Restructuring** documentation is restructured in terms of an underlying network or graph to keep track of requirements changes. For example, assumption based truth maintenance networks, then chaining system, constraint networks and propagation.
III. BENEFITS OF USING TRACEABILITY MATRIX

a) Always make obvious to the client that the software is being developed as per the requirements.
b) Ensuring that the developers are not creating features that no one has requested.
c) Making it easy to identify missing functionalities.
d) It makes easy to find out which test cases needed updating if there are change requests.
e) The TM may contains module, component, sub-system Ids. It becomes simpler to identify the impacted modules whenever a requirement changes.
f) Easy to update when requirement changed
g) Displaying the overall defect and execution status with focus on business requirements.

IV. REQUIREMENTS TRACEABILITY MATRIX

Traceability is the ability to trace requirements in a specification to their origin from higher level to lower level requirements in a set of documented links. It provides information which helps in determining whether all relationship and dependencies are addressed.[3]. Traceability makes project management easier by simplifying project estimates. Although, the importance of traceability seem to be well accepted in the software engineering industry.

<table>
<thead>
<tr>
<th>FSD_Section ID</th>
<th>Test Scenario Test ID</th>
<th>Test Case ID</th>
<th>Status</th>
<th>Defects</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSD_1.1-NewCustomer</td>
<td>Test_S_Loan_00 1- User validates the &quot;Apply Loan&quot; feature as a NewCustomer</td>
<td>TC_NewCustomer_001</td>
<td>Passed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TC_NewCustomer_002</td>
<td>Failed</td>
<td>Defect_01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TC_NewCustomer_003</td>
<td>Blocked</td>
<td>Defect_02</td>
<td></td>
</tr>
<tr>
<td>Test_S_Loan_00 2- User validates the &quot;Apply Loan&quot; feature as a NewCustomer</td>
<td>TC_NewCustomer_004</td>
<td>Passed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TC_NewCustomer_005</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TC_NewCustomer_006</td>
<td>Failed</td>
<td>Defect_03, Defect_04</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TC_NewCustomer_007</td>
<td>Passed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| FSD_1.2-ExistingCustomer | TS_Loan_00 5- Login to the loan portal as a already a customer with a loan and check the information displayed | TC_ExistUser_01 | Passed |          |
|                        | TS_Loan_008-Check the Loan whose status is "Reviewed and deleted" | TC_ExistUser_02 | Failed | Defect_42 |
|                        | TS_Loan_008-Check the Loan whose status is "Reviewed and deleted" | TC_ExistUser_03 | Failed | Defect_43 |
|                        | TS_Loan_008-Check the Loan whose status is "Reviewed and deleted" | TC_ExistUser_04 | Passed | Defect_44 |

V. CHALLENGES

The challenges can be identified under the areas of cost in terms of time & effort and the difficulty of maintaining traceability through change. The high costs of traceability can be approached with the attitude that the costs incurred will save much greater costs further along in the development process due to the benefits that traceability offers to software project.

Studies have shown that change can be expected throughout the lifecycle of nearly every software project. When the changes occur, it’s necessary to update the traceability data to reflect these changes. This requires discipline on the part of those making the change to update the traceability data. Dealing with change and its impact on traceability is a difficult prospect. Some commercial off the shelf tools offer assistance with identifying the impact of change on existing traceability data. A contributing factor to poor support for traceability may be the fact that many different viewpoints regarding traceability exist and even among different stakeholders on a project.

Maintaining traceability through changes to the system is another. Because the standards requiring traceability are vague organizations have a lot of leeway to set their own procedures in place for implementing traceability. Which can reduce the amount of confusion about traceability and leads to more consistent viewpoints among the stakeholders involved.

A factor contributing to poor support for traceability are the differing viewpoints regarding traceability.

Organizational problems provide a significant challenge to the implementation of traceability. Individuals feel that traceability data will be used against them in performance reviews or as a threat to their job.

VI. STRENGTHEN THE TRACEABILITY
Traceability will help your business to maintain a high level of quality and ensure compliance with industry regulations. Stakeholders may believe that strengthening traceability will mean more hours or days lost to manually creating a traceability matrix and stakeholders may be more amenable to adopting the best practices, new strategy and software tools. The following practices will strengthen the traceability in your product development lifecycle.

Find out the Key Assets: Break the documents down to more manageable core elements. Instead of treating the entire requirements document as a single asset just focus on specific requirements.

Naming convention and Terminology: Get all participants to agree on one naming convention per artifact. Establish a simple meaning for each artifact, and write the meaning on a sticky note. Some of the original words can be used. A general definition of the different types of artifacts can be written.

Analyze the Impact of Change: You now need to determine how change affects other related artifacts. Analyze the impact on a test case when a requirement changes and also analyze the impact on the test cases when test data has been changed.

Trace Relationships: Define the types of relationships that need to occur if change happens. Try to define one key relationship per group at a minimum.

Make Sense of Data: Is it possible to easily pull the appropriate data for the report. Ensure, is it possible to analyze key data quickly.

Tools and Current Processes: A good process will also reduce risk and identify the areas that should be controlled to ensure success and check whether our methods and practices are outdated.

VII. TOOLS

There are many software tools that are available to allow users to trace the requirements throughout the system development lifecycle. Few tools are listed below.

Rational RequisitePro: It helps project teams to manage their requirements then to write good use cases, improve traceability, to strengthen collaboration, reduce project rework and to increase quality. Then, it also supports requirements traceability and impact analysis and currently in use in industry.

Doors: DOORS is a requirements management tool developed initially by Telelogic. Later, it has been acquired by IBM Rational in 2008.

CaliberRM: CaliberRM is the requirements management tool by Borland.

DevCpmplete: It is developed by SmartBear software. It provides full traceability among requirements tracking, the project tasks and defects to improve team agility.

SLATE: System Level Automation Tool for Engineers, is computer-aided engineering tool for system designers. It uses MATLAB to evaluate all aspects of each design.

VIII. CONCLUSION

Many organizations find it difficult to dedicate the time or resources necessary to develop traceability strategies and best practices. This paper presented an introduction to the benefit offered by traceability and the challenges faced by the practice of traceability in software projects today. The lack of quality commercial off the shelf traceability tools is a significant challenge facing the implementation of traceability in the software engineering industry. These challenges lead many organizations to implement only as much traceability as is required by their customers. Automated Requirements Tool, Dynamic Object Oriented Requirements Systems, Requirements Use Case Tool and IBM Rational Rose are few automated requirements tools used for collecting, viewing and changing requirements and these tools manage, changes and provide traceability metrics. The use of automated tools enhances requirements management but before using any requirements tool and its function, its uses and limitations must be understood. Each tool has some strong and weak points usually. But none of them supports automated link detection clearly Poor tool support for traceability remains an exception, this is an area that is still an open problem in the software industry.

REFERENCES