

# **1. Requirements Traceability Matrix**

## **1.1 Definition**

Requirements traceability provides System Integrators with the capability to describe and follow the life of a requirement, both forward and backward directions, from its origins, through its development and specification, to its subsequent deployment and use.

Traceability matrices can be established using a variety of tools, including requirements management software, databases, spreadsheets, or even with tables or hyperlinks in a word processor. They are created by associating requirements with the work products that satisfy them. Tests are associated with the requirements on which they are based and the product tested to meet the requirement.

## **1.2 Reasons for Requirements Traceability**

The traceability needs of different stakeholders – project sponsors, project managers, analysts, designers, maintainers, and end users of a system – differ due to differences in their goals and priorities (Please refer to the Program Management Plan for different roles and responsibilities). In the requirements engineering phase, it is important that the rationales and sources of the requirements are captured in order to understand requirements evolution and verification.

Modifications during design appear if the requirements evolve, or if the system is developed incrementally. Modifications after the delivery of the system will happen due to various reasons (e.g. to correct faults or to adapt the system to a changing environment). During design phase, requirements traceability allows the capability of continuously tracking what happens when a change request is implemented before a system is re-designed. Traceability can also give information about the justifications, important decisions and assumptions behind requirements.

Test procedures can be traced to requirements or design. This kind of traceability helps to design and modify test procedures. These kinds of modifications are called system evolution. Empirical studies show that even experienced software professionals predict incomplete sets of change impacts. With complete traceability, more accurate costs and schedules of changes can be determined, rather than depending on the engineer or programmer to know all the areas affected by these changes.

## **1.3 Traceability Structure**

The RTM traces the requirements to those identified in the RMT FRD. A quality management tool, Mercury Quality Center, is being used for requirements management. The Mercury Quality Center, a web-based automated software quality testing and requirements management tool, will collect and maintain the RMT system requirements.

The Requirements Management sub-module links the software requirements in Test Director to ensure traceability of requirements through to test cases and to defects.

The tool outlines the requirements. Next, a test plan is developed, along with test cases that link to the requirements through the creation of test sets, which are groups of test cases that target a particular testing goal. The test sets are executed and defects are identified and tracked during execution.

## **1.4 Description of Matrix Fields**

The RTM contains the following matrix fields (Please refer to Appendix A: RTM Table for the table showing these matrix fields):

1. FRD Reference -indicates what FRD section refers to a requirement.
2. System Modules – the RMT system module that a requirement is covering.
3. Requirement ID - a unique identification number (ID) generated by the Mercury Quality Center tool. It is assigned in ascending order (e.g., 1.0; 1.1; 1.2).
4. Derivative ID – defines the parent-child relationship between each requirement statement.
5. Name – the name for each requirement
6. Description – a description of each requirement
7. SCR Number – the system change request (SCR) associated with the requirement
8. External/Interface Document – identifies the document that captures all the interface requirements related to the specific requirement
9. Technical Design Reference – identifies the design document that captures the comprehensive software design, consisting of distinct but interrelated activities: data design, architectural design and procedural design.
10. Design Review Reference – document is a review of the design by other developers.
11. Code Review Reference – identifies the code review document that finds and fixes mistakes overlooked in the initial development phase, improving the overall quality of the software.
12. Direct Cover Status – provides whether the requirement has a test case associated with it, the status of the test case – whether it has been executed - aborted, passed or failed.
13. Test Plan Type – the type of test plan (i.e., manual)
14. Test Plan Name – name of the test plan
15. Release Number – system release number associated with approved requirement

## **1.5 Management and Uses of Requirements Traceability Matrix**

The Requirements Traceability Matrix is the controlling document for the software engineering from design specifications to integration testing and Independent Verification and Validation (IV&V). It is the key document when developing requirements, regardless of development practices (e.g., OO, Extreme Programming, Agile or Prototyping). It defines “what is required and the function that it must satisfy.”

All subsequent Engineering activity shall be managed by direct reference to the Requirements listed and described in the Requirements Traceability Matrix. Documents and files ID numbers and names (naming and ID convention) must incorporate the same ID number and names used Matrix for establishing the logical and physical relationship between the requirements and the other artifacts, files and control processes.