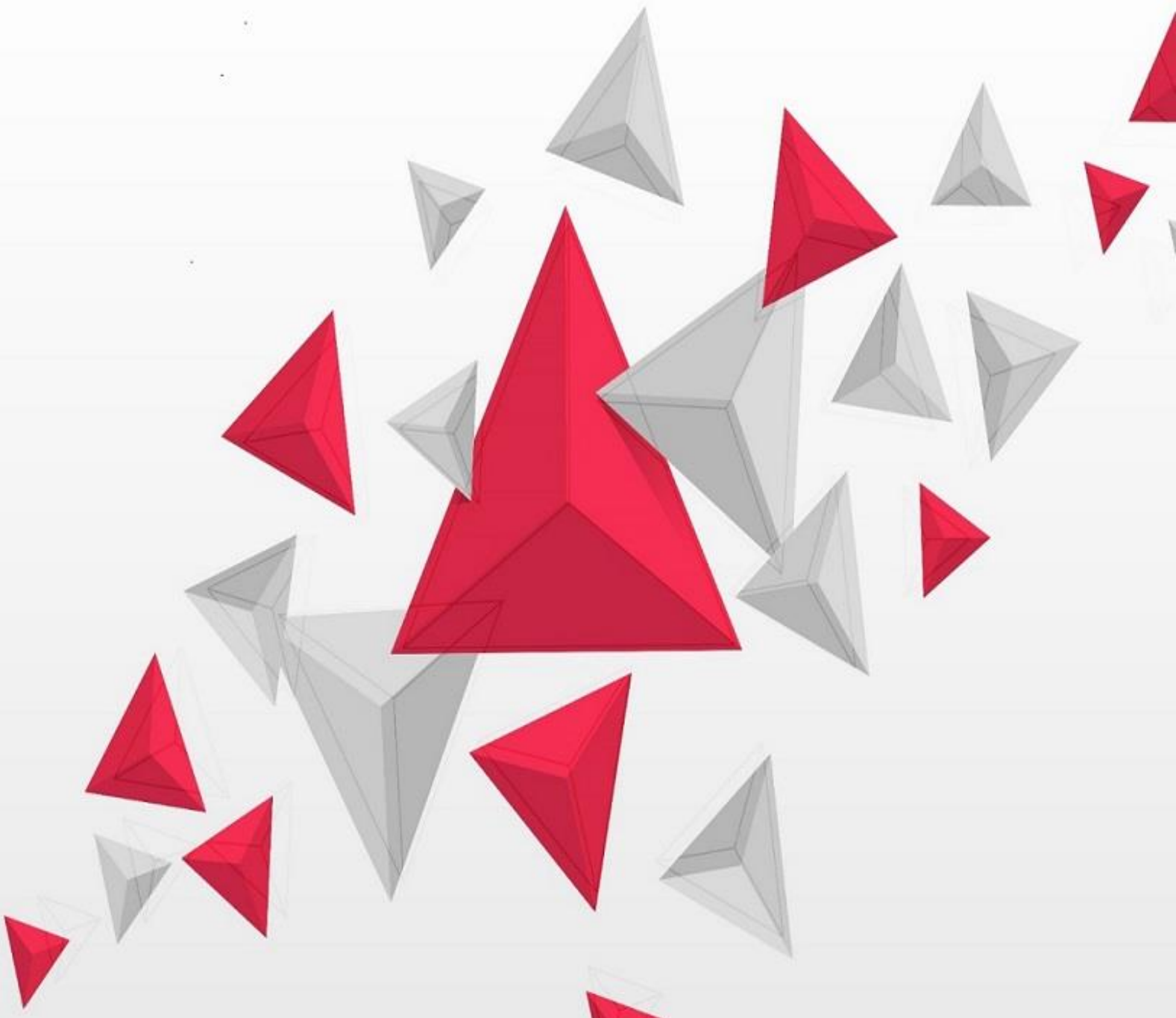


Requirements Traceability Matrix

Guide Book





REQUIREMENTS TRACEABILITY MATRIX – INTRODUCTION

Although each project starts with a fixed set of requirements, they change over time, and so does their priority. Now, with so many changes happening, if nobody is tracking, there is no way you could save a project from failing.

So, here comes the savior, the Requirements Traceability Matrix.

In my experience as a business analyst and a project manager, I would consider that an RTM is by far the most essential document in the project for tracking and monitoring the vitals of the project. By definition:

A Requirement traceability matrix records and tracks the relationship of the project requirements to the design, documentation, development, testing, and release of the project/product.

Thus, in one single document, you will be able to find the details of a requirement, to which module it belongs, in which use case document is it described, what are the test cases against it, where its code located, and what's the release the functionality is available in.

Now that's a one-stop shop!

Creating an RTM is the sole responsibility of a BA, and he must be skilled in doing so. BAs should note that **RTM assists in both forward as well as backward tracing of a project**. Forward tracing when we ought to know the requirement, the module it belongs to, its development, associated test cases, and releases. Backward tracing when, in case of a new change request, we wanted to check whether this functionality fits into achieving the business need or not.

A personal story:

Once I was newly induced in a major investment bank as a senior BA. After an initial project walkthrough, I was handed over the project documentation, which contained the Functionality Matrix, use cases, clarifications documents, and design documents. To my utter surprise, I couldn't find an RTM there, and when I asked for the latest documents, I was told that *they were* the latest documents. There had been a lot of changes to the projects, but none of them were chronicled in an RTM; they were either in the change requests documents or were in emails. It took me close to 2 months to gather all that information from numerous sources and then create an RTM!



ASPECTS OF A REQUIREMENTS TRACEABILITY MATRIX

To reinforce the importance of RTM, let's take a quick look at all the benefits a project has by creating an RTM.

1. Controls and validates changes to project scope

RTM helps control changes by *tracing* whether they are needed and reduces the chances of incorporating uncontrolled modifications to the project - This is one of the most significant advantages of an RTM and is highly applicable in big, module-driven projects.

2. Helps in impact analysis

In case of changes being introduced to the project, the RTM doubles as a document that helps assess the change's impact on all the related functionalities and modules.

3. Helps attain transparency and visibility for the project

The project that maintains an updated and thorough RTM always has an advantage as the complete project essentials are carefully laid down, and everybody has access to them.

Such activity increases the project's transparency and visibility as an ideal project in the eyes of the project team and delivery managers.

4. Ensures structured and thorough test coverage

The details of every test case are captured in an RTM, and when functionality is to be released, the testers know what test cases are to be executed. Thus an RTM ensures proper and complete test coverage before every release and improves the overall quality of the application being built.

5. Increase productivity and minimize management time

The RTM maintains the requirements, its associated development code, and the related test cases – all in one place. Any new resource, be it a BA, Tester, or developer, can easily pick up and grasp the project with all the information they require being handy in the RTM. Such a practice increases the productivity of the complete team as they don't need to spend time providing this information. Moreover, PMs can track and manage things pretty effectively as they have a consolidated view.



AUDIENCE OF A REQUIREMENTS TRACEABILITY MATRIX

Let's talk about different stakeholders who will be referencing the RTM:

1. The first comes the *Project Manager* will make sure that the overall execution of the project is in-line with the project management plan, and the project is headed in a direction that fulfills its business objective. Additionally, the PM will use the RTM for tracking and planning the releases and overall roadmap of the project.
2. Next, the *Quality Team* will use the RTM to ensure a percent test coverage for all the listed functionalities and features.
3. The *Developers* will use the FM to map and catalog their codes vis-à-vis the functionalities
4. The *Business Analysts*, the RTM owner, will either create the RTM independently or take ownership of getting it completed by the relevant team members. They will make sure that all the requirements are present and the document is up to date, validate the alignments of requirements to the respective use and test case documents and unearth any inconsistencies against the same.
Also, he will continue to refer to the RTM for impact analysis when there are changes/modifications to the requirements.



HOW TO CREATE A REQUIREMENTS TRACEABILITY MATRIX

Okay, let's now get our feet wet, learning how to create an RTM.

As a first step, the BA should have the already approved functionality matrix with him for reference. Next, using the FM, the BA should list all the requirements that exist in the project in the 'Requirement Details' column of the RTM spreadsheet-based template. Since the FM being used is already reviewed and approved for correctness and coverage, it acts as a base for creating a comprehensive RTM.

Below are the different sections that comprise the RTM and the details that go into each one of them:

1. Functionality Matrix Reference

When all the requirements are entered in the RTM, it needs to be ensured that the requirements identifier is the same across all the documents. Thus, the same identifier that was entered into the functionality matrix needs to be entered here.

2. Application Requirements Definition/Use cases

We have learned that requirements are entered in some form of requirement definition document, preferably a use case document. The name of that specific use case document needs to be entered under this section. Such a practice ensures that all the functional requirements are being discussed, elaborated, and captured in a use case document.

3. Module

Modules are different subparts/sections of the overall product or application being built. We need to enter the module's name to which the requirements belong in the 'Module' section to categorize the requirements better.

4. Source Code

This column contains the actual name of the code file, which includes this functionality, and since the code files are kept at a strategic code repository (like SVN or GIT), a link to that location is also given here. In the case of nested functionalities, many functionalities could have the same source code files.

5. Unit/Module Test Cases IDs

The details, i.e., the name and location of the test cases where the requirements respective test scenarios are written, go under this column. These test cases are either unit test cases, i.e., test cases written for that specific functionality, or module test cases, i.e., test cases written for the complete module of which the functionality is only a part. With the test case ID given, even testers can determine whether their test cases cover all the requirements or something is yet to be covered.

6. Release details

Each functionality should be released into UAT first for acceptance and then into the production environment. Having a release date alongside each requirement aids in better tracking and release planning.

7. Version History

Finally, your RTM should have a version history that should be updated whenever you modify, add, or delete something from the RTM. Sometimes, in a hurry, we change something in our document, and then, later on, we forget the change done and the reason behind that change. Such situations will never occur if proper version history is maintained. This tab contains details like Version No., Date revised, Change Description, A/M/D (Add/Modify/Deleted), Prepared By, Approved By.

When it comes to creating an RTM, every primary office suite has a spreadsheet that can be configured to support requirements tracing.

We have included a sample RTM template for your reference using Microsoft excel. Download this template, and feel free to use this matrix in your next analysis endeavor.

One last point before we move onto our next section – As a BA, you should note that an RTM is a living document, meaning an RTM is modified and maintained throughout the lifecycle of the various releases in a project, and it's a vital document to track project scope, requirements and changes in any project.



REQUIREMENTS TRACEABILITY MATRIX - BEST PRACTICES

1. To leverage the best out of an RTM, it's **imperative to have your Functionality Matrix up to date at all times**. It should contain not only the initial requirements but even those functionalities that were added as change requests or enhancements.

Also, each of these requirements should have a unique identifier (traceable to the functionality matrix), which remains unchanged throughout the life cycle of the requirement.

2. An RTM **should not only be used for forwarding but backward traceability as well** – something most analysts and managers forgets in the thick of things. Many times, activities like code reviews are added as requirements when they are not. RTM helps identify such mistakes by identifying elements that cannot be traced back to its 'Origins'.

3. Another way to help maintain a traceability matrix is by **keeping it in a centralized location** where the complete team has access to it. This way, when many members eyeball the same document, then any irregularity will be pointed out at once. Also, it helps the BA maintain an up-to-date version as he has a moral obligation to furnish the latest information to the team.

4. The RTM **should be subjected to at least one review every month by the PM** – even when it's not asked for. Not only it's a good project management habit, but it speaks a lot about the pro-activeness of the BA handling the RTM.

5. If a BA finds it very hard to maintain a comprehensive RTM due to the project's number of requirements, he should at least manage an RTM for the primary set of requirements. This way, they still have a preliminary document that could always be updated later or can even be delegated to other team members.

6. An analyst should be creative in making use of an RTM by going beyond the defined template and adding his own columns or color-coding the existing rows. An RTM could be extended to double as a defect tracking or code review tracking or execution status tracking sheet as well – such types of innovative modifications help a busy BA save a lot of time and re-work.