📘 Chapter 6: Requirement Traceability Matrix (RTM)

✅ What is RTM?

A **Requirement Traceability Matrix (RTM)** is a document that **maps and traces user requirements** with the corresponding **test cases**, ensuring that each requirement is **verified and validated** during the software testing lifecycle.

It helps teams answer:

- “Have we tested everything we were supposed to?”
- “Do we know which test cases validate each requirement?”
- “What will be impacted if this requirement changes?”

📌 Why RTM is Crucial in QA

Benefit	Description
✅ **Ensures Coverage**	Confirms that all requirements are tested
✅ **Reduces Missed Tests**	Highlights untested or incomplete areas
✅ **Helps in Impact Analysis**	Shows what tests are affected by changing or deleted requirements
✅ **Supports UAT and Audits**	Easy to show proof that requirements were validated

✅ **Improves Communication**	Aligns QA, dev, and business teams on expectations and results

"RTM is your **map from 'What should be built' to 'What was tested'**."

📊 RTM Structure – Sample Template

Req ID	Requirement Description	Test Case ID(s)	Status	Comments
REQ-001	User can register via email	TC-001, TC-002	Pass	Covered in Sprint 1
REQ-002	Reset password via OTP	TC-003	Fail	Issue with OTP mismatch
REQ-003	Dashboard loads in < 3 secs	TC-004	Pass	Performance Test Passed

0
3

📄 Columns You Can Include (Optional Based on Project Needs)

Column	Purpose
Requirement Priority	Helps prioritize critical requirements
Functional Area	Categorize based on module or screen
Test Scenario ID	Link to broader scenario if test cases are granular
Defect ID / Status	Track defects linked to that requirement
Sprint/Release	Track when it was tested or deployed

🛠️ Creating an RTM – Step-by-Step

🧩 Step 1: List All Requirements

- Gather business and functional requirements from the **BRD, FRD, or user stories**

- Assign **unique Req IDs** (e.g., REQ-001, REQ-002)

🧩 Step 2: Design Test Cases

- For each requirement, create **one or more test cases**
- Make sure each test case has a unique ID (e.g., TC-001, TC-002)

🧩 Step 3: Map Test Cases to Requirements

- For every requirement, identify the test cases that validate it
- Use a table or Excel format to link them (many-to-one or one-to-many)

🧩 Step 4: Track Execution Status

- After testing begins, update the status of each requirement (Pass/Fail/Blocked)
- Optionally, link to defects raised or screenshots

🧩 Step 5: Maintain and Review

- **Update RTM** after every sprint, release, or test cycle
- Perform **impact analysis** if requirements change

🔁 Forward vs Backward Traceability

Type	Description
Forward Traceability	From **requirement** → **test case** (Did we test this?)
Backward Traceability	From **test case** → **requirement** (Why are we testing this?)

✅ A good RTM supports **bidirectional traceability**.

🔍 Real-World Example

Let's say the requirement is:

> **REQ-004**: "The user shall be able to add up to 5 items to the wishlist."

You should have:

- **TC-010**: Add 1 item → Success
- **TC-011**: Add 5 items → Success
- **TC-012**: Add 6th item → Error displayed

All mapped to **REQ-004** in the RTM.

📌 Tips for Maintaining RTM

- Use **Excel**, **TestRail**, **qTest**, or **JIRA plugins** to manage RTMs
- Review with BA and stakeholders after each release
- Automate RTM generation if using integrated test management tools

🧾 Summary

Element	Purpose
RTM	Ensures every requirement is tested
Test Case Mapping	Links requirement to validation
Status Tracking	Shows real-time testing progress
Bidirectional Traceability	Tracks both directions for completeness

---

“If a requirement is not in your RTM, it might never get tested. If a test case is not linked to a requirement, ask why you're running it.”