CO23115
Software Testing- Unit 4

Test Planning, Documentation and Bug Reporting (8M)

Test Planning, Documentation and Bug Reporting

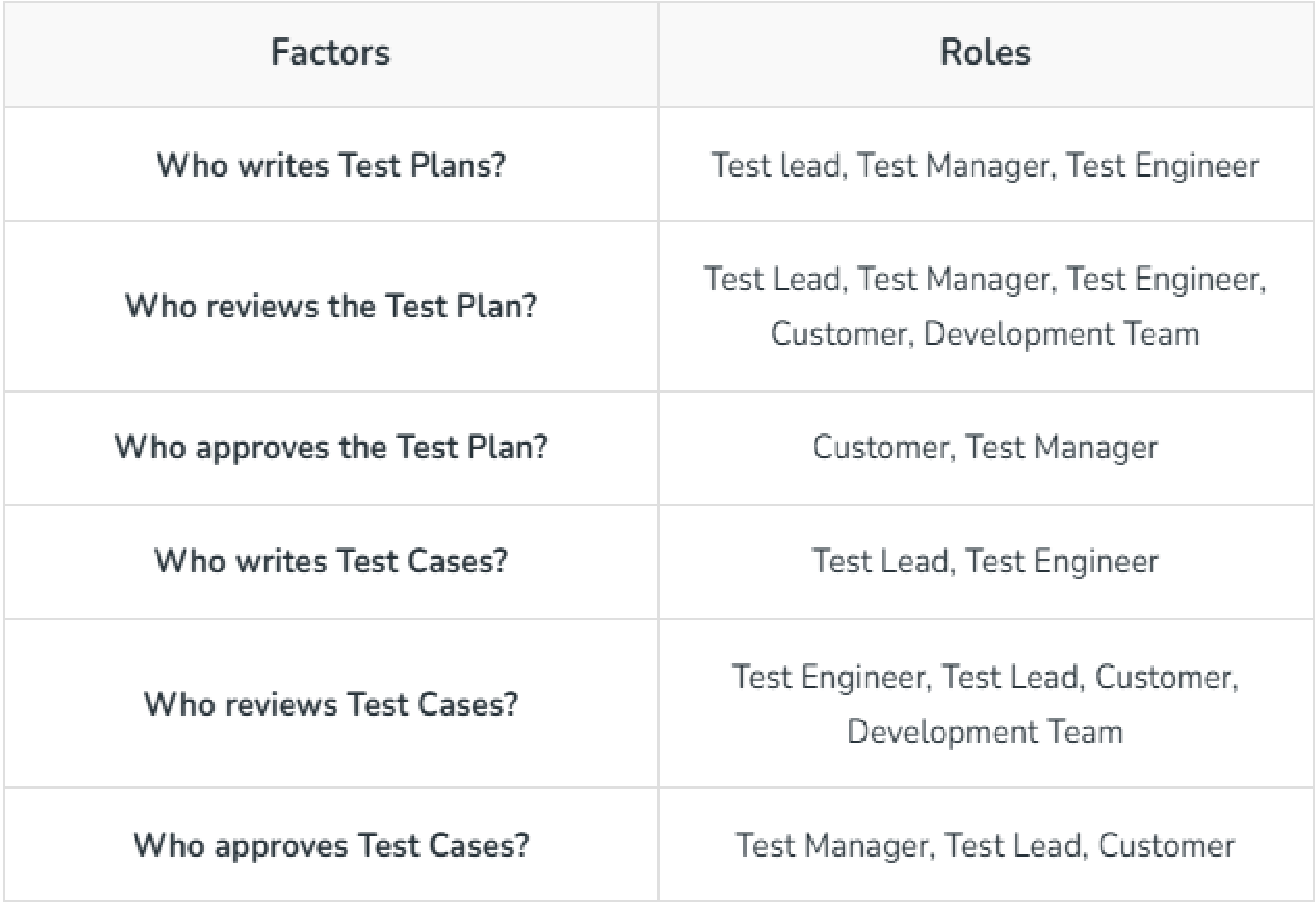
- A clear and detailed test plan is the foundation of successful testing, guiding everything from creating test cases to fixing issues.

Test Plan

- A test plan is a document that consists of all future testing-related activities.
- It is prepared at the project level and in general, it defines work products to be tested, how they will be tested, and test type distribution among the testers.
- Before starting testing there will be a test manager who will be preparing a test plan.

Test Plan

- In any company whenever a new project is taken up before the tester is involved in the testing the test manager of the team would prepare a test Plan.
- The test plan serves as the blueprint that changes according to the progressions in the project and stays current at all times.
- It serves as a base for conducting testing activities and coordinating activities among a QA team.
- It is shared with Business Analysts, Project Managers, and anyone associated with the project.
- A test plan is a comprehensive document outlining all testing-related activities for a project.
- It details what will be tested, how, and by whom, serving as a blueprint for testing and coordinating among the QA team.



Factors	Roles
Who writes Test Plans?	Test lead, Test Manager, Test Engineer
Who reviews the Test Plan?	Test Lead, Test Manager, Test Engineer, Customer, Development Team
Who approves the Test Plan?	Customer, Test Manager
Who writes Test Cases?	Test Lead, Test Engineer
Who reviews Test Cases?	Test Engineer, Test Lead, Customer, Development Team
Who approves Test Cases?	Test Manager, Test Lead, Customer

Test Plan: Goal of the Test Planning

- **Resource requirements**
- Resource requirement is a detailed summary of all types of resources required to complete project task.
- Resource could be human, equipment and materials needed to complete a project.
- The resource requirement and planning is important factor of the test planning because helps in determining the number of resources (employee, equipment…) to be used for the project.
- Therefore, the Test Manager can make the correct schedule & estimation for the project.

Resource requirements

- Planning the resource requirements is the process of deciding what’s necessary to accomplish the testing strategy.
- Everything that could possibly be used for testing over the course of the project needs to be considered.
- For example:
- • **People**. How many, what experience, what expertise? Should they be full-time, part-time, contract, students?
- • **Equipment.** Computers, test hardware, printers, tools. • Office and lab space. Where will they be located? How big will they be? How will they be arranged?
- • **Software**. Word processors, databases, custom tools. What will be purchased, what needs to be written?

Resource requirements

- • **Outsource companies**. Will they be used? What criteria will be used for choosing them? How much will they cost?
- • **Miscellaneous supplies**. Disks, phones, reference books, training material. What else might be necessary over the course of the project?
- The specific resource requirements are very project-, team-, and company-dependent, so the test plan effort will need to carefully evaluate what will be needed to test the software.
- It’s often difficult or even impossible to obtain resources late in the project that weren’t budgeted for at the beginning, so it’s imperative to be thorough when creating the list.

Tester Assignments

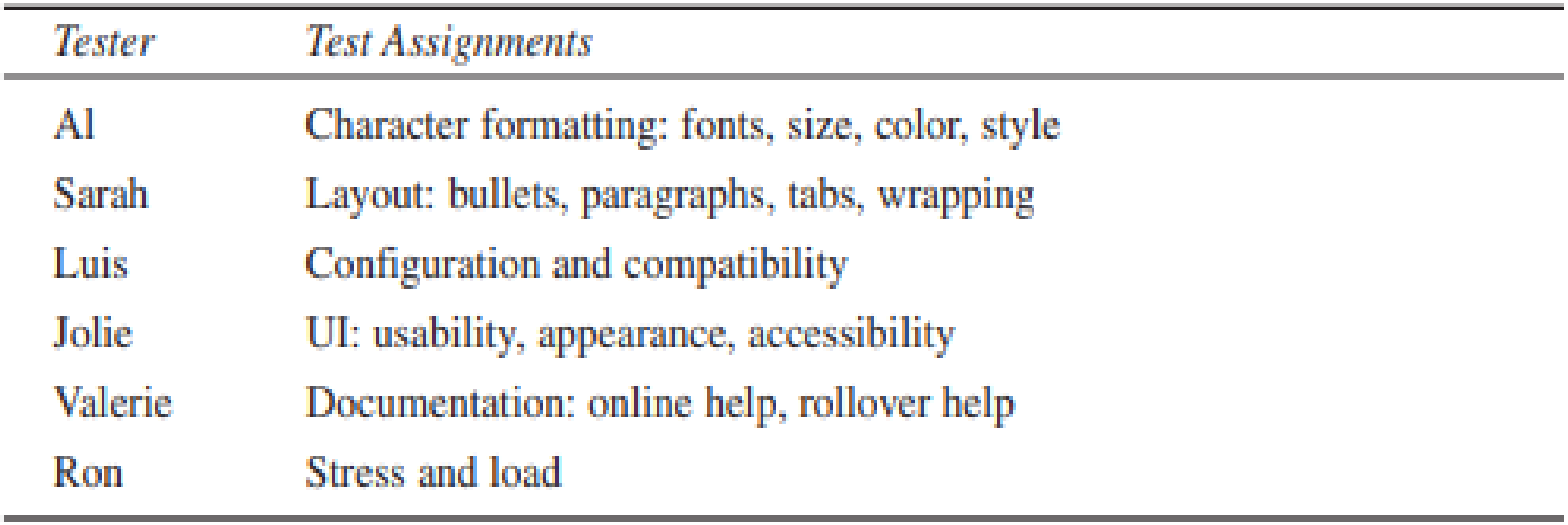
- • Once the test phases, test strategy, and resource requirements are defined, that information can be used with the product spec to break out the individual tester assignments.
- Planning the tester assignments identifies the testers (this means you) responsible for each area of the software and for each testable feature.

Tester Assignments

- • Once the test phases, test strategy, and resource requirements are defined, that information can be used with the product spec to break out the individual tester assignments.
- Planning the tester assignments identifies the testers (this means you) responsible for each area of the software and for each testable feature.

Tester Assignments

- Table shows a greatly simplified example of a tester assignments table for Windows WordPad.
- Table: High-Level Tester Assignments for WordPad



Tester	*Test Assignments*
Al	Character formatting: fonts, size, color, style
Sarah	Layout: bullets, paragraphs, tabs, wrapping
Luis	Configuration and compatibility
Jolie	UI: usability, appearance, accessibility
Valerie	Documentation: online help, rollover help
Ron	Stress and load

- A real-world responsibilities table would go into much more detail to assure that every part of the software has someone assigned to test it.
- Each tester would know exactly what they were responsible for and have enough information to go off and start designing test cases.

Test Schedule

- The test schedule takes all the information presented so far and maps it into the overall project schedule.
- This stage is often critical in the test planning effort because a few highly desired features that were thought to be easy to design and code may turn out to be very time consuming to test.
- Completing a test schedule as part of test planning will provide the product team and project manager with the information needed to better schedule the overall project.
- They may even decide, based on the testing schedule, to cut certain features from the product or postpone them to a later release.

Test Schedule

- The effect of this gradual increase is that the test schedule is increasingly influenced by what happens earlier in the project.
- If some part of the project is delivered to the test group two weeks late and only three weeks were scheduled for testing, what happens?
- Does the three weeks of testing now have to occur in only one week or does the project get delayed two weeks?
- This problem is known as schedule crunch.

Test Schedule

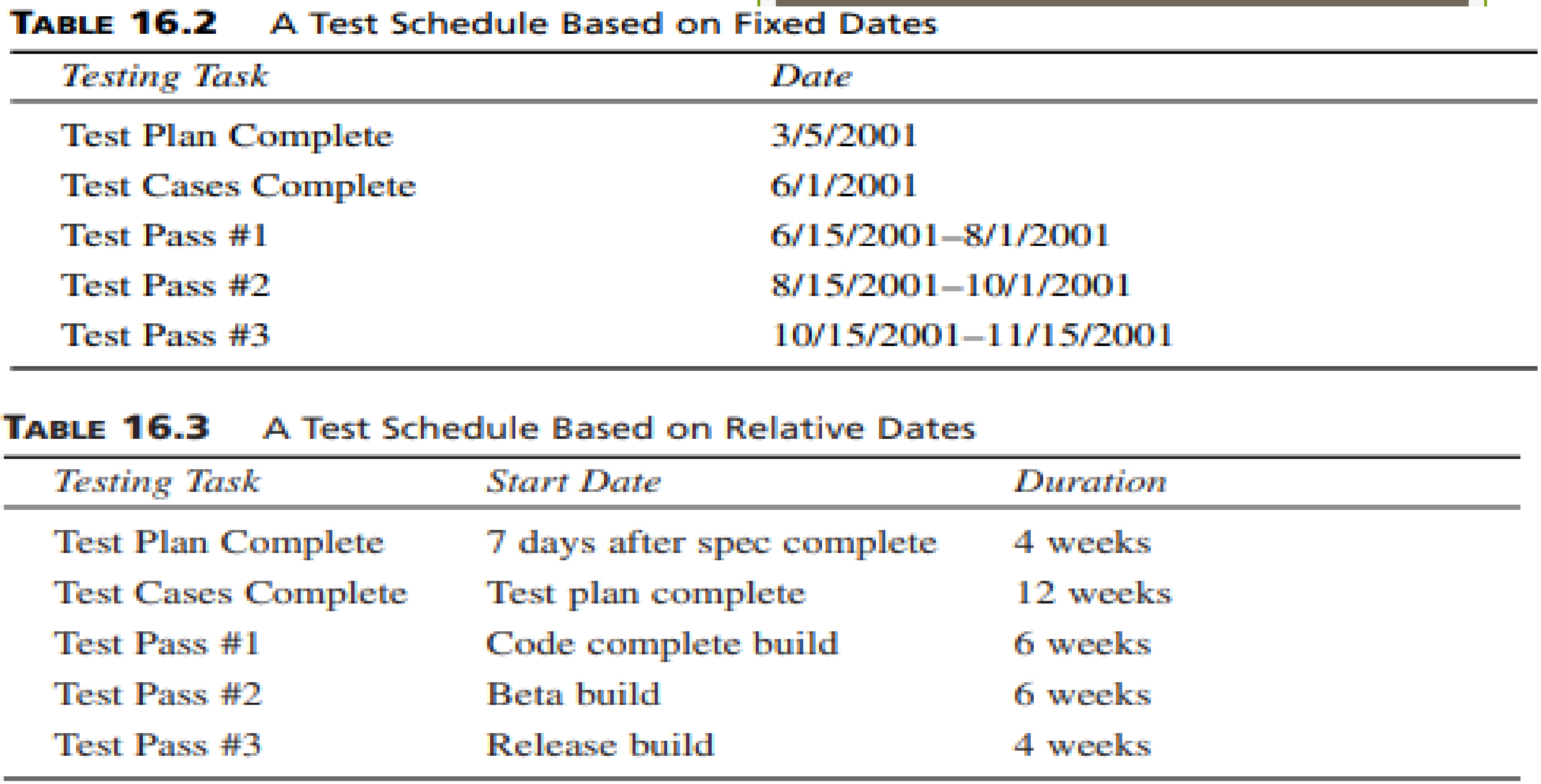


TABLE 16.2 A Test Schedule Based on Fixed Dates

Testing Task	*Date*
Test Plan Complete	3/5/2001
Test Cases Complete	6/1/2001
Test Pass #1	6/15/2001–8/1/2001
Test Pass #2	8/15/2001–10/1/2001
Test Pass #3	10/15/2001–11/15/2001

TABLE 16.3 A Test Schedule Based on Relative Dates

Testing Task	*Start Date*	*Duration*
Test Plan Complete	7 days after spec complete	4 weeks
Test Cases Complete	Test plan complete	12 weeks
Test Pass #1	Code complete build	6 weeks
Test Pass #2	Beta build	6 weeks
Test Pass #3	Release build	4 weeks

Your project manager or test manager is ultimately responsible for the schedule.

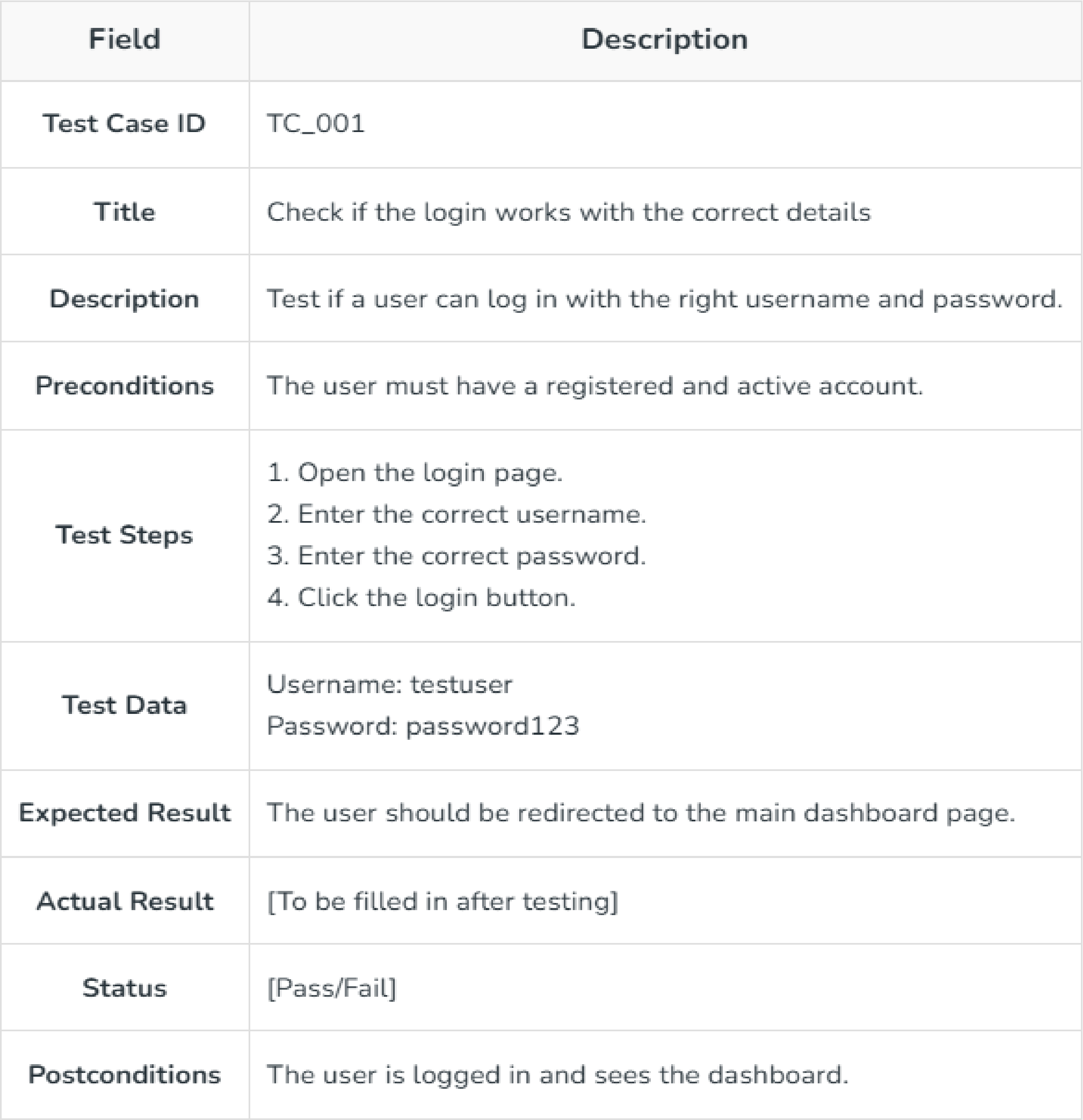
Test Case

- A Test Case is a step-by-step guide used to check if a specific part of a software application works correctly as expected. It defines the actions to be taken, the required data, and the expected result to validate that a feature functions as per the requirements.

- **Components:**

- **Test Case ID**: A unique name or number to identify the test case.
- **Title/Description:** What the test case is meant to check.
- **Preconditions:** What needs to be set up before running the test?
- **Test Steps:** The actions to perform during the test.
- **Test Data:** The specific information needed for the test.
- **Expected Result:** What should happen if the software is working right?
- **Actual Result:** What happens when the test is run.
- **Status:** Whether the test passed or failed.
- **Post-conditions:** What should be true after the test.

Test Case Format



Field	Description
Test Case ID	TC_001
Title	Check if the login works with the correct details
Description	Test if a user can log in with the right username and password.
Preconditions	The user must have a registered and active account.
Test Steps	1. Open the login page. 2. Enter the correct username. 3. Enter the correct password. 4. Click the login button.
Test Data	Username: testuser Password: password123
Expected Result	The user should be redirected to the main dashboard page.
Actual Result	[To be filled in after testing]
Status	[Pass/Fail]
Postconditions	The user is logged in and sees the dashboard.

RTM

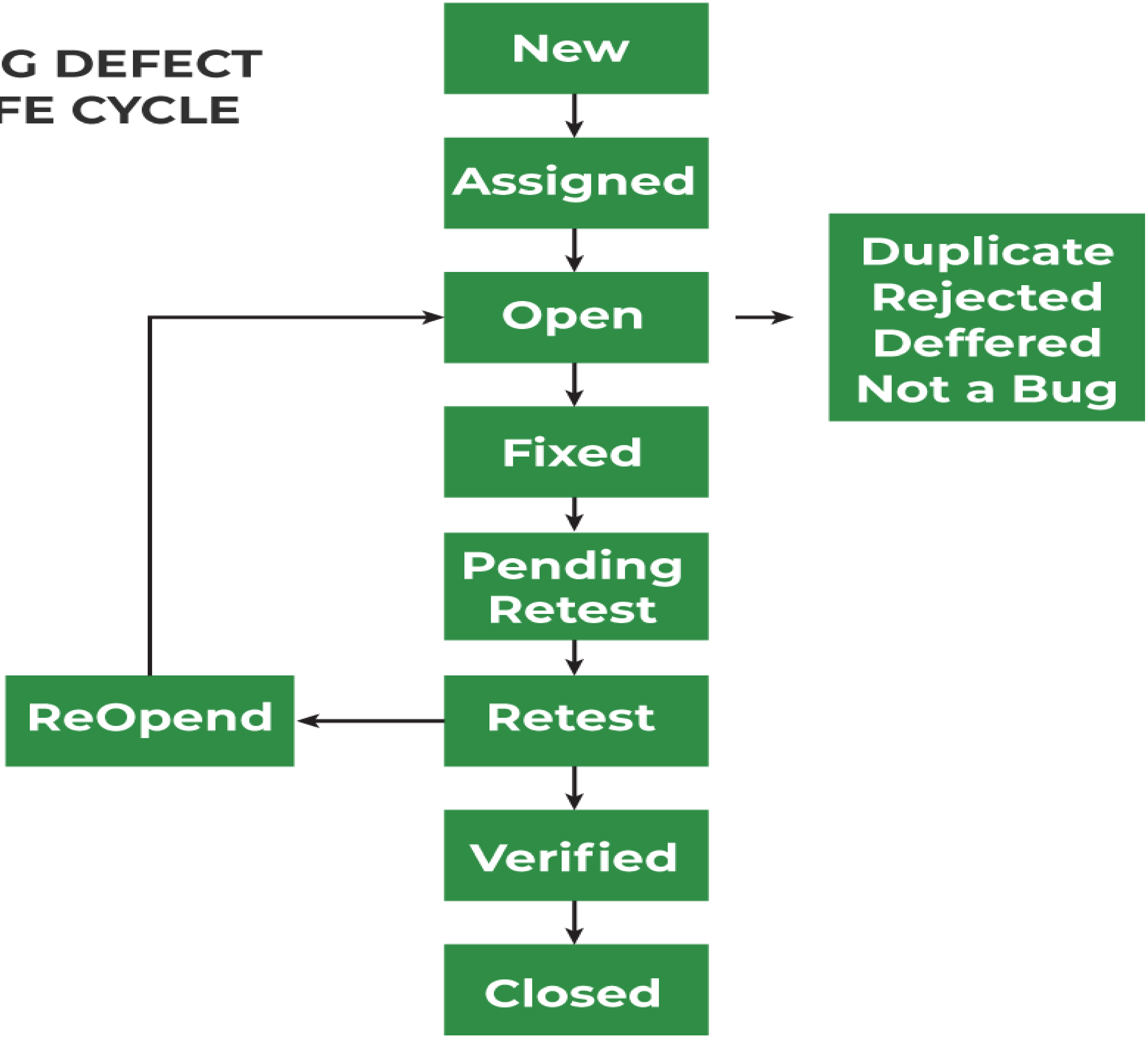
- RTM stands for Requirement Traceability matrix. RTM maps all the requirements with the test cases.
- By using this document one can verify test cases cover all functionality of the application as per the requirements of the customer.

- **Requirements:** Requirements of a particular project from the client.
- **Traceability:** The ability to trace the tests.
- **Matrix:** The data which can be stored in rows and columns form.
- The main purpose of the requirement traceability matrix is to verify that the all requirements of clients are covered in the test cases designed by the testers.

RTM

- A Traceability Matrix is a document that maps or rather traces the relationship between two baseline documents.
- Here, one of the documents has the requirement specifications, whereas the other one has test cases.
- The Traceability Matrix is an essential document used during the software development lifecycle of a product, and it ensures completeness and transparency of the underlying product.
- It is also called Cross Reference Matrix (CRM) or Requirement Traceability Matrix (RTM).

BUG DEFECT LIFE CYCLE



New

Assigned

Open

Duplicate
Rejected
Deffered
Not a Bug

Fixed

Pending Retest

Retest

ReOpend

Verified

Closed

Bug life cycle

- Mainly bug life cycle refers to its entire state, starting from a new defect detected to the closing off of that defect by the tester. Alternatively, it is also called a Bug Life Cycle.
- **1. New**
- When any new defect is identified by the tester, it falls in the 'New’ state. It is the first state of the Bug Life Cycle. The tester provides a proper Defect document to the Development team so that the development team can refer to Defect Document and can fix the bug accordingly.
- **2. Assigned**
- Defects that are in the status of 'New' will be approved and that newly identified defect is assigned to the development team for working on the defect and to resolve that. When the defect is assigned to the developer team the status of the bug changes to the 'Assigned' state.
- **3. Open**
- In this 'Open' state the defect is being addressed by the developer team and the developer team works on the defect for fixing the bug. Based on some specific reason if the developer team feels that the defect is not appropriate then it is transferred to either the 'Rejected' or 'Deferred' state.

Bug life cycle

- **4. Fixed**
- After necessary changes of codes or after fixing identified bug developer team marks the state as 'Fixed'.
- **5. Pending Retest**
- During the fixing of the defect is completed, the developer team passes the new code to the testing team for retesting. And the code/application is pending for retesting on the Tester side so the status is assigned as 'Pending Retest'.
- **6. Retest**
- At this stage, the tester starts work of retesting the defect to check whether the defect is fixed by the developer or not, and the status is marked as 'Retesting'.

Bug life cycle

- **7. Reopen**
- After 'Retesting' if the tester team found that the bug continues like previously even after the developer team has fixed the bug, then the status of the bug is again changed to 'Reopened'. Once again bug goes to the 'Open' state and goes through the life cycle again. This means it goes for Re-fixing by the developer team.

- **8. Verified**
- The tester re-tests the bug after it got fixed by the developer team and if the tester does not find any kind of defect/bug then the bug is fixed and the status assigned is 'Verified'.

- **9. Closed**
- It is the final state of the Defect Cycle, after fixing the defect by the developer team when testing found that the bug has been resolved and it does not persist then they mark the defect as a 'Closed' state.

Bug Tracking System

- A bug tracking system is software that keeps track of bugs that the user encountered in any software development or in any project.
- Bug tracking is the process of logging and monitoring bugs or errors during software testing. It is also referred to as defect tracking or issue tracking. Large systems could have hundreds or thousands of defects. Each needs to be evaluated, monitored and prioritized for debugging.
- **The three main functionalities of the Bug Tracking system is:**
- Creating a new text file and writing the details entered by the user into the text file.
- Option to change the status of the bug.
- Report of specific bug file.

Bug Tracking System

- ## Test Incident Report
- A test incident report is a formal document detailing a deviation from the expected behavior of software during testing.
- **Purpose**: To communicate issues, log defects, and monitor their status throughout the development lifecycle.
- **Content:** Includes test case information, steps taken, expected and actual outcomes, environment details, and potentially screenshots or logs.
- **Categorization:** Bugs are often categorized by severity (impact on the system) and priority (urgency of the fix)

Bug Tracking System

- **Manual Bug Reporting and Tracking.**
- This refers to the human-driven process of identifying, documenting, and monitoring software defects.
- **Reporting:** When a tester encounters a bug, they manually create an incident report, often using a template that includes fields for the bug's summary, detailed description, steps to reproduce, severity, and environment.
- **Tracking:** After a bug is reported, it's tracked from its initial discovery through its resolution. Testers often assign a unique ID to the report for tracking purposes and monitor its status (e.g., open, in progress, fixed, closed).
- **Importance:** It provides developers and other stakeholders with comprehensive insights into observed defects, facilitating effective problem-solving and ensuring all issues are addressed before release.