Understanding Test Cases, Test Plans, and Test Scenarios

Rashmi Ruwanthika Athukorala 4 min read · 3 hours ago

In the world of **software testing**, ensuring product quality is not just important it’s essential. To manage this process effectively, QA (Quality Assurance) teams use structured documents that define **what to test**, **how to test**, and **when to test**.

The three most commonly used documents in testing are:

- **Test Plan**

- **Test Scenario**
- **Test Case**

These three terms may sound similar, but they represent **different levels of detail** and serve **different purposes** in the testing process. Think of them like a pyramid the **Test Plan** sits at the top (broadest), followed by **Test Scenarios** (medium-level), and then **Test Cases** at the bottom (most detailed).

Let’s understand each one step by step 👇

🗺️ 1. The Test Plan: The Big Picture Strategy

A **Test Plan** is the highest-level document in the testing process. It acts like a **master guide** or **blueprint** for all testing activities in a project.

It answers big questions like:

- What features will be tested?
- How will the testing be done?
- Who will perform the testing?
- What tools and environments are needed?
- When will each testing activity take place?

The **Test Manager** or **QA Lead** usually creates the Test Plan at the beginning of the project. It serves as a **reference document** for the entire software development lifecycle (SDLC).

🧩 Key Components of a Test Plan:

1. **Scope:** Defines what is included in testing and what is excluded (In-Scope and Out-of-Scope).
Example: Login and registration features are in scope; payment gateway testing is out of scope.

2. **Objectives:** The goals of testing (e.g., "Ensure 99% of transactions complete successfully without errors").

3. **Testing Strategy:** Describes the overall approach — manual, automated, or a hybrid method.

4. **Resources:** Lists the QA team members, tools, and hardware/software required.

5. **Schedule:** Timelines for testing phases such as test design, execution, and reporting.

6. **Test Environment:** Defines the hardware, operating systems, browsers, and databases to be used.

7. **Entry and Exit Criteria:**

- **Entry Criteria:** Conditions that must be met before testing can start (e.g., development completed, environment ready).

- **Exit Criteria:** Conditions that must be met to stop testing (e.g., all critical bugs fixed, 95% test cases passed).

💡 Example:
A Test Plan is like the **blueprint for building a house**. It defines the design, the materials, the workers, and the deadlines — everything you need before construction begins.

📝 2. The Test Scenario: The High-Level Testing Goal

A **Test Scenario** is a **high-level description** of a feature or functionality that needs to be tested. It's not about detailed steps yet — it's about identifying *what areas* of the software should be tested.

It answers the question:
👉 **"What needs to be tested?"**

Test Scenarios help ensure that **all features and business flows** of the software are covered by testing. They are simple, short statements that describe the functionality to be verified.

🧩 Examples of Test Scenarios:

- Verify that a user can successfully log in.
- Test the process of adding an item to the shopping cart.
- Check the functionality of the password recovery feature.
- Validate that the user can update their profile information.

Each Test Scenario can later be expanded into **multiple Test Cases**, covering different paths or conditions.

💡 Example:
If the **Test Plan** is the blueprint of the house, the **Test Scenario** is like a room inside it — for example, “Check the kitchen plumbing.” It gives you the idea of what part of the project to test.

3. The Test Case: The Step-by-Step Testing Instruction

A **Test Case** is the most detailed and specific document. It describes exactly **how to test a particular feature** and what the expected result should be.

It answers the question:
👉 **“How should it be tested?”**

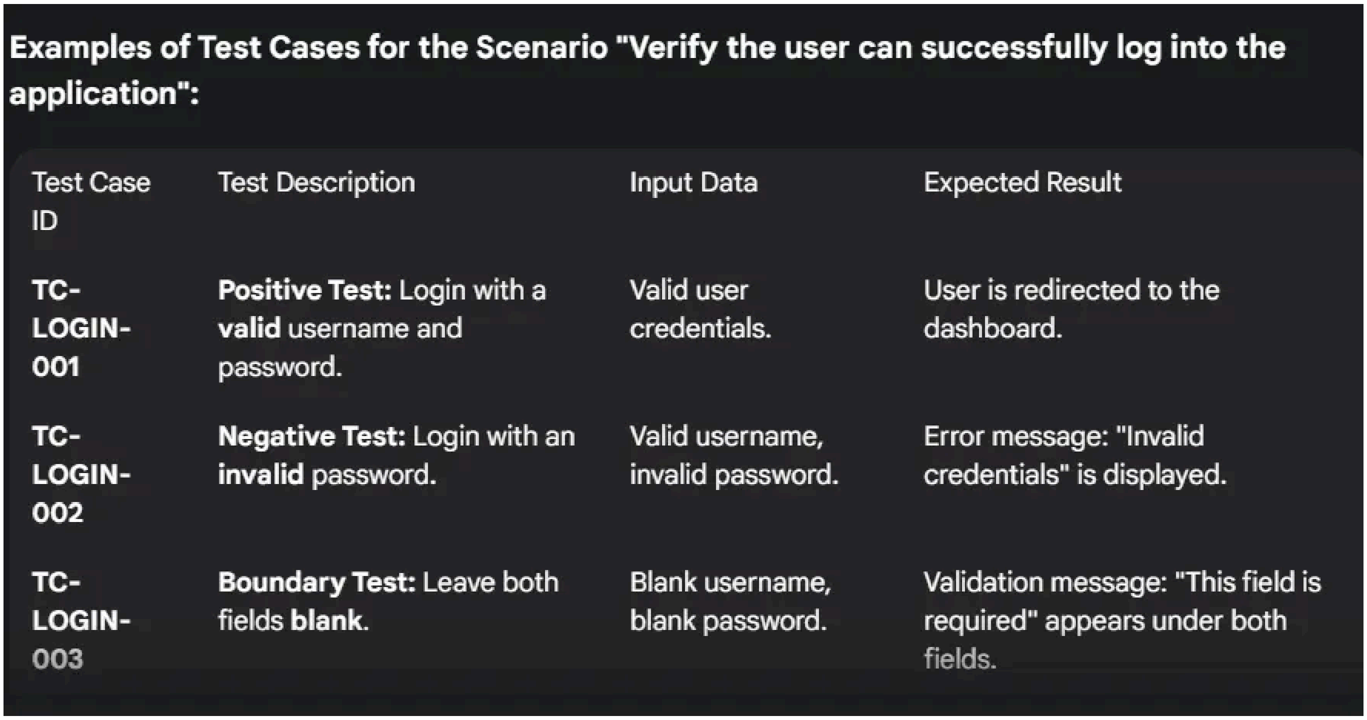
Each Test Case provides a **clear, repeatable set of steps** that a tester must follow to verify whether a feature works correctly.

Usually, one Test Scenario is covered by **several Test Cases** — each testing a different condition (positive, negative, or boundary).

🧩 Key Components of a Test Case:

1. **Test Case ID:** A unique identifier (e.g., TC-LOGIN-001).
2. **Description/Objectives:** What the test case is checking (e.g., “Verify login with valid credentials”).
3. **Pre-conditions:** What must be true before testing (e.g., “User must have a valid registered account”).
4. **Test Steps:** Step-by-step actions the tester should perform.
5. **Expected Result:** What should happen if the software works correctly.
6. **Actual Result:** What actually happened during the test.

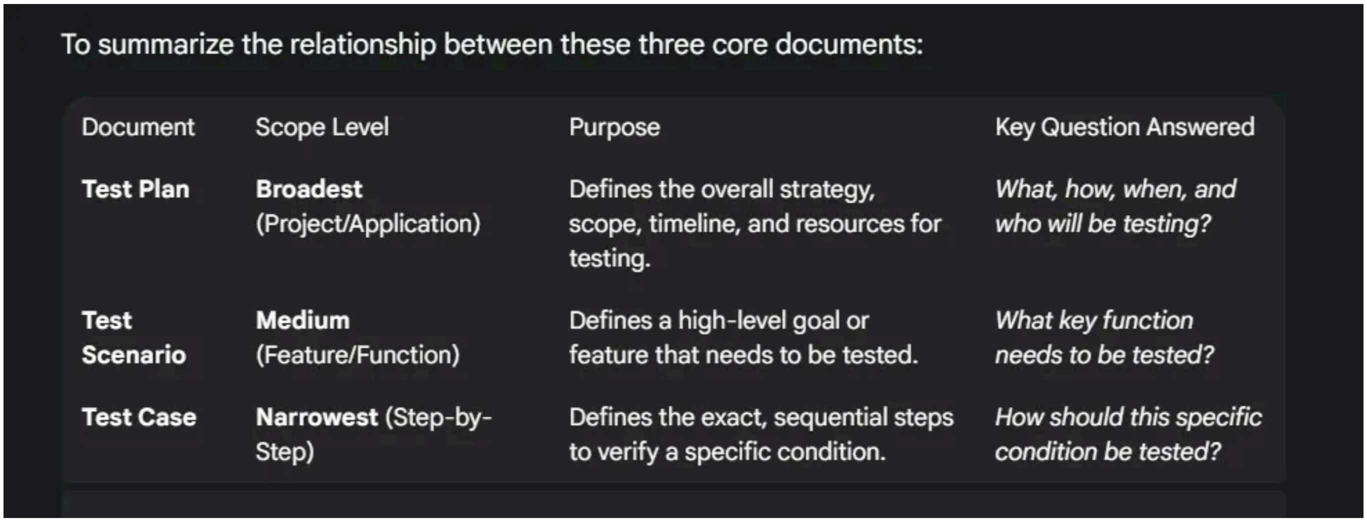
7. **Status:** Pass or Fail



Examples of Test Cases for the Scenario "Verify the user can successfully log into the application":

Test Case ID	Test Description	Input Data	Expected Result
TC-LOGIN-001	**Positive Test:** Login with a **valid** username and password.	Valid user credentials.	User is redirected to the dashboard.
TC-LOGIN-002	**Negative Test:** Login with an **invalid** password.	Valid username, invalid password.	Error message: "Invalid credentials" is displayed.
TC-LOGIN-003	**Boundary Test:** Leave both fields **blank**.	Blank username, blank password.	Validation message: "This field is required" appears under both fields.

💡 Example:
If the **Test Scenario** is "Check the kitchen plumbing," then the **Test Case** is "Turn on the hot water tap and check if the water flows properly." It's specific, measurable, and has a clear expected result.



To summarize the relationship between these three core documents:

Document	Scope Level	Purpose	Key Question Answered
Test Plan	**Broadest** (Project/Application)	Defines the overall strategy, scope, timeline, and resources for testing.	*What, how, when, and who will be testing?*
Test Scenario	**Medium** (Feature/Function)	Defines a high-level goal or feature that needs to be tested.	*What key function needs to be tested?*
Test Case	**Narrowest** (Step-by-Step)	Defines the exact, sequential steps to verify a specific condition.	*How should this specific condition be tested?*

To deliver high-quality software, QA teams must be **structured, detailed, and consistent.**

- The **Test Plan** gives direction.
- The **Test Scenarios** define the coverage.
- The **Test Cases** ensure accurate and repeatable testing.

When these three are used together, they help create a **well-organized QA process** that leads to reliable, bug-free software and happier users

Written by Rashmi Ruwanthika Athukorala

Edit profile

0 followers · 1 following

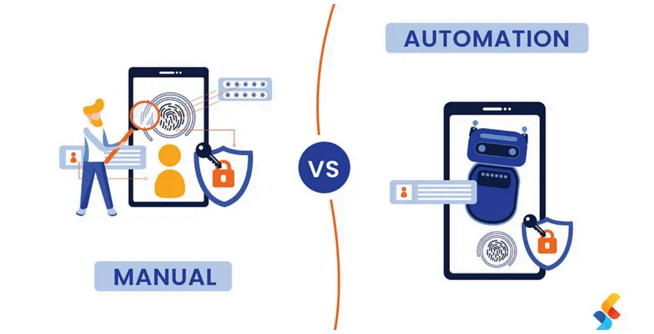
BSc Hons Computer Science and Technology | Undergraduate at Uva Wellassa University, Sri Lanka | QA Enthusiast | Software Tester

No responses yet

Rashmi Ruwanthika Athukorala

What are your thoughts?

More from Rashmi Ruwanthika Athukorala

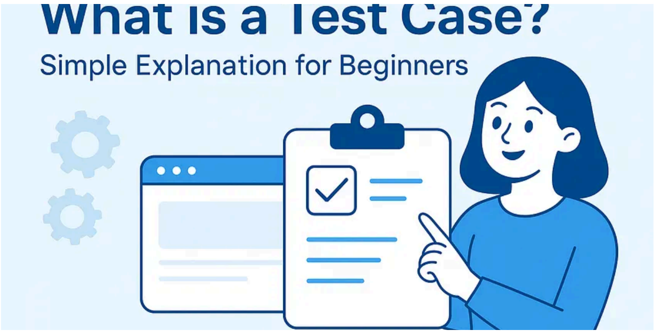


Rashmi Ruwanthika Athukorala

🤖 Manual vs. Automated Testing: Which Should You Choose for You…

Choosing the right testing approach is crucial for delivering high quality software. Should…

22h ago



Rashmi Ruwanthika Athukorala

What Is a Test Case? A Simple and Complete Guide for Beginners

Software today is everywhere in our phones, laptops, cars, banks, hospitals, and even in…

2d ago