

# Core Features

Unlock the full potential of your testing process with Appvibe's Core Features. This chapter delves into key functionalities such as test case management, test cycle execution, and linking test cases to your requirements. Whether you're creating detailed test cases or organizing test cycles for efficient execution, this chapter will guide you through every step to make sure your tests are accurate, traceable, and repeatable.

- [Test Case Management](#)
- [Test Steps and Action Items](#)
- [Test Case Traceability](#)
- [Test Cycles](#)

# Test Case Management

**Test Case Management** is one of the core components of **Appvibe Test Management**. It allows you to create, organize, and track test cases efficiently, ensuring all your product requirements are fully tested.

## What is a Test Case?

A test case is a set of conditions and steps used to validate a particular function or feature of your product. Each test case includes:

- A unique **Test Case ID** for easy identification.
- **Test Steps** detailing the actions a tester must perform.
- **Expected Results** that define what should happen when the test case is executed.
- **Actual Results** captured during execution to determine whether the test passed or failed.

## Creating a Test Case in Appvibe:

1. **Navigate to Your Test Management Board** on monday.com.
2. **Click on the 'Add Test Case' button**, which will prompt you to fill in the following details:
  - **Test Case Name/ID:** A brief identifier for the test case.
  - **Description:** A detailed description of what this test case will cover.
  - **Preconditions:** Any setup required before executing the test (e.g., the user is logged in, the environment is pre-configured).
  - **Test Steps:** The detailed actions required to perform the test.
  - **Expected Results:** What should happen if the system is functioning correctly.
  - **Priority:** Set the priority level for this test case (e.g., High, Medium, Low).
3. **Save the Test Case** to store it in your test case library, where it can be executed as part of a test cycle.

## Test Case Versions:

Appvibe allows you to manage **Test Case Versions**, ensuring that as your product evolves, your test cases are updated accordingly. Each version tracks changes made to the test case, allowing you to review past iterations if necessary.

# Test Steps and Action Items

Test steps are the core instructions in a test case, detailing the actions the tester must perform to verify a particular feature of the system. **Appvibe Test Management** ensures that every test case is structured with clear, actionable steps to ensure thorough testing.

## Creating Test Steps:

1. **Open a Test Case** from your test management board.
2. In the test case editor, add a list of **Test Steps**. Each step should describe:
  - **What the tester must do:** This could be clicking a button, entering text, or making an API call.
  - **Expected Outcome:** What should happen after performing the step (e.g., the system saves the data, the screen updates, etc.).

## Test Action Items:

Within each step, **Action Items** refer to specific tasks or actions the tester must perform. For example, in a test step titled "Enter user credentials," the action item would be "Enter username 'testuser' and password 'password123'." Each action item is designed to ensure clarity for the tester, minimizing errors during execution.

## Best Practices for Test Steps and Action Items:

- **Keep test steps clear and concise:** Avoid ambiguous language that could confuse testers.
- **Ensure test steps are sequential:** The order of execution should be logical and flow naturally.
- **Link each test step to an expected outcome:** This ensures testers know exactly what to look for after performing each action.

Once the test case is completed and the steps are defined, you can execute the test case as part of a test cycle.

# Test Case Traceability

**Test Case Traceability** is essential for ensuring that all requirements are covered by test cases. In Appvibe, traceability links test cases to their respective product requirements, ensuring that every feature or functionality has been properly validated through testing.

## What is Traceability?

Traceability is the ability to map a test case back to a specific requirement. This ensures:

- Every requirement is linked to at least one test case.
- You can easily track whether each requirement has been tested.
- Changes to requirements can be monitored for impact on existing test cases.

## Setting Up Traceability in Appvibe:

### 1. Link Requirements to Test Cases:

- When creating a test case, you can select the related product requirement from your monday.com board.
- This creates a clear link between the requirement and the test case, ensuring full coverage.

### 2. View Traceability from the Requirement Side:

- Open a product requirement on your monday.com board, and you'll be able to see which test cases are linked to that requirement.
- This makes it easy to see which features or requirements still need test cases.

## Traceability Reports:

Appvibe automatically generates **Traceability Reports**, providing a visual representation of which requirements are linked to test cases and which are not. This report helps you identify gaps in your testing and ensure that all critical areas of your product are covered.

## Benefits of Traceability:

- **Ensures complete coverage:** You can confirm that every requirement is thoroughly tested.
- **Facilitates change management:** When a requirement changes, you can easily see which test cases need to be updated.
- **Improves quality control:** By ensuring that all requirements are tested, you reduce the risk of releasing a product with untested features.

# Test Cycles

Test cycles are a core component of the **Appvibe Test Management for monday.com**, designed to help teams organize and execute groups of test cases around specific releases, sprints, or milestones. A **Test Cycle** is essentially a collection of related test cases that need to be executed together to validate the functionality or quality of your system. Test cycles streamline the testing process by grouping test cases for efficient execution and reporting.

## What is a Test Cycle?

A **Test Cycle** is a group of test cases, often linked to a specific release, sprint, or milestone in your project. It allows you to track the execution of multiple test cases together, record their outcomes, and assess overall system quality at specific points in the development process.

Test cycles make it easier to:

- Manage large-scale test executions by organizing test cases into logical groups.
- Track progress across multiple test cases.
- Ensure that critical test cases are executed in a systematic and organized manner.

## Creating a Test Cycle:

1. **Navigate to Your Test Management Board** in monday.com.
2. **Click 'Create Test Cycle'**: This option is available on your test management board, allowing you to start a new cycle.
3. **Name Your Test Cycle**: Provide a meaningful name (e.g., "Sprint 10 Functional Test Cycle" or "Release 1.2 Regression Test").
4. **Select Test Cases**: Add test cases to the cycle by selecting them from your library. You can choose test cases related to specific features or requirements based on your test strategy.
5. **Set Priorities**: Define the priority level for each test case in the cycle (e.g., High, Medium, Low) to ensure that critical tests are run first.
6. **Assign Testers**: Assign individual testers or groups responsible for executing the test cases in the cycle.
7. **Define Deadlines**: Set deadlines for when the test cycle should be completed, helping teams stay on track and meet project timelines.

## Executing a Test Cycle:

Once your test cycle is created, it's time to execute the tests and track the results.

1. **Launch the Test Cycle:** When you're ready to execute the cycle, click the "Start Cycle" button.
2. **Execute Test Cases:** Testers can begin running the individual test cases assigned to them. As they complete each test case, they will mark it as **Passed**, **Failed**, or **Blocked**, depending on the outcome.
  - **Recording Actual Results:** Testers should record the actual results of each test case, comparing them to the expected outcomes defined in the test case.
  - **Reporting Defects:** If a test case fails or encounters a bug, testers can report defects directly within the monday.com board, linking them to the failed test case.
3. **Monitor Progress:** Use the **Test Cycle Dashboard** to track the status of all test cases within the cycle. You can see how many test cases have been executed, how many have passed, failed, or are blocked, and how much of the cycle remains incomplete.

## Re-running Test Cycles:

If defects are identified during the execution of a test cycle, or if new features are added, you may need to re-run certain test cases. Appvibe allows you to:

- **Clone Test Cycles:** Quickly create a copy of a previous test cycle for re-execution.
- **Add or Remove Test Cases:** Modify test cases in a cloned cycle to include newly added functionality or exclude obsolete tests.

## Test Cycle Reporting:

After a test cycle has been completed, you can generate reports to analyze the results. Test cycle reports provide insights into the pass/fail rate, defects identified, and overall quality metrics. These reports can help you:

- Assess whether the product is ready for release or further testing is needed.
- Identify problematic areas that require attention.
- Understand test coverage and traceability.

## Benefits of Test Cycles:

- **Organized Testing:** Grouping test cases into cycles helps manage complex projects more effectively, ensuring nothing is missed.
  - **Tracking and Accountability:** Test cycles provide a clear picture of test execution progress, allowing stakeholders to track the quality of the product throughout development.
  - **Improved Focus:** By prioritizing test cases within a cycle, critical areas of the product are tested first, reducing the risk of releasing buggy or incomplete features.
  - **Efficient Reporting:** Test cycle reports make it easy to see overall quality, helping teams make informed go/no-go decisions.
-

# Best Practices for Test Cycles:

1. **Define Clear Objectives:** Before creating a test cycle, make sure you understand what the cycle is meant to validate (e.g., a specific feature set, regression tests, etc.).
  2. **Organize by Sprint or Release:** Align your test cycles with your development sprints or product releases to ensure all new or updated features are tested before going live.
  3. **Review Test Cycle Reports:** After each test cycle, review the reports to identify gaps in coverage or recurring issues that require attention.
  4. **Automate Where Possible:** Consider automating certain test cases in your test cycles to increase efficiency and reduce manual effort.
- 

By using test cycles in **Appvibe Test Management**, teams can systematically validate their products and track progress toward ensuring the highest quality before release. This feature helps to ensure that all critical aspects of the product are tested thoroughly and efficiently.