

QMetry Test Management - Agile Implementation Guide

Being Agile means adapting, collaborating, and delivering value continuously.

The Agile Implementation Guide helps streamline test management by aligning QMetry with Agile workflows. This guide provides step-by-step instructions for integrating test planning, execution, and reporting within an Agile environment.

Let's understand this with an example:

Background

FitTracker is a Tech company that makes Smart watches. Fit Tracker Program Management manages multiple projects, which are independent but with critical integration points which cannot be ignored.

Process & People

FitTracker uses Agile Scrum methodology to manage all of their projects and leverage JIRA as their project management tool.

FitTracker have minor releases every 2 sprints and major release every 6 sprints. Each sprint consists of 2 weeks.

Each agile team is responsible for managing independent deliverable which contributes to FitTracker ecosystem. Few Examples of these projects in JIRA:

1. **FitTracker Watch** - Smart watch hardware design and development.
2. **FitTracker Firmware** - Smart watch firmware development & testing
3. **FitTracker Web Application** - Web Portal.
4. **FitTracker iOS App** - iOS app
5. **FitTracker Android App** - Android App

Agile Implementation

Steps for Implementing QMetry in Agile

Integrating QMetry into your Agile framework enhances testing efficiency and project transparency. By following strategic steps, you can streamline your testing process, ensuring all team members are aligned and moving towards shared goals.



Step 1: Setup People

Integrate LDAP authentication or Single Sign On authentication with QMetry and enable all your users to log in QMetry. Alternatively, you can also use QMetry's Authentication if you do not have LDAP or SSO.

This is also good time to define roles. Typical roles recommended are:

- QA Professional – Create & Maintain Test Cases, Test Suites and Defects.
- Developer – Create & Maintain requirements & Defects, while view Test cases & Test Suites.
- QA Manager – Can do what QA Professional & Developer can do, for given project
- Program Manager – Can do what QA Manager can do, for multiple projects.
- Admin – Has rights to create projects and perform other administrative tasks that could potentially impact whole QMetry Instance. Refer help documentation for further details.

Resources

- To set up user go to Customization >> User >> Add. Read more [here](#).
- To set up roles go to Customization >> Roles. Read more [here](#).
- To edit a role, go to Customization >> Roles >> Edit Role. Read more [here](#).

Step 2: Setup Projects

In case of FitTracker, each Agile team is responsible for exactly one project, to keep things simple. However, you can have multiple teams managing multiple projects. In that case defining roles and permissions could be more complex but definitely possible.

i If you are using JIRA and plan to integrate JIRA with QMetry, we strongly recommend to use same project structure as JIRA. This way syncing of issues with JIRA will be extremely simple.

In our example, we will setup same projects as JIRA:

1. FitTracker Watch - Smart watch hardware design and development.
2. FitTracker Firmware - Smart watch firmware development & testing
3. FitTracker Web Application - Web Portal.
4. FitTracker iOS App - iOS app
5. FitTracker Android App - Android App

Resources

- To set up project/release/cycle go to Project Management >> Projects. Read more [here](#).

Step 3: Create Tests & Link to Stories

Go to Test Case module and start writing test cases. Document Test Steps, Test Data, and expected outcome. Attach screen shots or other material to support the test case.

Link Test Case to Stories. In our example, we have integrated QMetry with JIRA. QMetry will automatically sync requirements from JIRA to QMetry and keep them up to date.

i Linking test cases to Stories is a good practice to determine test coverage and produce more meaningful reports.

Resources

- To create test case, go to Test Case Module. Read more [here](#).
- To link Requirement to Test case, go to Test case detail page >> Requirement tab. Read more [here](#).

Step 4: Estimate Your Test Cases

To estimate, how much can you really test in each sprint, it's important to estimate each test case. Estimation is optional, if done, could be helpful in reporting.

QMetry currently supports estimation in Hours. However, you can use point based or any other estimation technique as far as its numeric to match with your estimation technique in JIRA.

Resources

- To estimate test cases, Create Test Case >> Enter the - Estimated Time. Read more [here](#).

Step 5: Plan Release and Cycles

In each project, you can define Release and Cycles (a.k.a. sprints). Releases and Cycles can have start and end dates. Each release can have multiple Cycles/Sprints.

Resources

- To create Release and Cycle, go to Project Management >>Release/Cycle. Read more [here](#).

Step 6: Link Releases/Cycles to Test Cases

Depending on scope of your stories in a specific sprint, you can move corresponding Test Cases to same release/cycle(sprint). Linking of Test Cases to Sprint can be done from multiple places including Test Cases or Test Suites.

Resources

- To link Release/Cycle to Test cases, open Test Case >> Associate Release & Cycle. Read more [here](#).

Step 7: Manage Test cases under folder hierarchy

Manage all your Test cases under folder hierarchy. Create folders as name of your Release and Cycle. Group subfolders with the name of components/features.

Example:

Release Name is Avengers, Sprints are Avenger 1, Avenger 2 etc., Components are Walk, Run, Yoga etc. Folder structure should be defined as.

- Avengers >> Avenger 1 >> Walk
- Avengers >> Avenger 1 >> Run
- Avenger >> Avenger 1 >> Yoga
- Avengers >> Avenger 2 >> Walk
- Avengers >> Avenger 2 >> Run
- Avengers >> Avenger 2 >> Yoga

Create build number as per your development cycle, assign test suite executions to a build and execute them. The build reports will provide build summary of the executions done for it.

Resources

- To create test case folder, go to Test Case >> Create Folder. Read more [here](#).
- To create builds, go to Projects >> Build. Read more [here](#).

Step 8: Execute Tests and Track Progress

Create Test Suites for all test Cases that are part of current sprint. Depending on platforms, invite Testers to execute tests and fill results. Log new defects as discovered during test execution.

If your team does automation, this also good time to setup Jenkins/Bamboo integration to bring automation results inside QMetry. This way we can get comprehensive view of manual and automated testing.

Discuss progress in daily standup meeting, and re-assign test executions as required. Read more [here](#).

Step 9: View Requirement Coverage Report.

Requirement Coverage report gives you 360-degree view of manual and automation testing. For the given scope of sprint, coverage report determines what is covered, how many tests are executed, and analyze passed vs failed ratio.

Coverage report also shows defects identified during testing, their priority and current status. This makes it really simple for executives to make informed go-no-go decision after reviewing manual and automated test coverage, test results along with defect status and severity.

To view executive coverage report, go to Reports >> Requirement Report. Read more [here](#).

Step 10: Complete the sprint and hold retrospective.

As you complete the sprint, discuss what worked and what did not work. Use top defects report to identify areas where maximum defects came from. Use Traceability report to identify patterns and relationship between defects, requirements and test cases.

Once you close the sprint, your stories may evolve for next sprint. Let's say for "Login" story, in next sprint we will support "Login via mobile phones". In this case you can create a new version of Login Test Case without impacting older version or test results. And link new version of login test case with enhanced login story.