Tips for scheduling executions

Scheduling executions of Tests can be made by creating Test Executions containing the Tests that should be run.

In fact, a Test Execution contains Test Runs (i.e., instances of the Tests in the context of that Test Execution). It contains a copy of the Test specification along with the respective result.

- Executing Tests in multiple environments
- How to manage assignment
  - Test Executions per each test executor
  - Test Executions assigned to QA manager
- Recommendations
- FAQ
  - Do I need to create a Test Execution every time I need to run the same Tests? Can't I update an existing one?
  - How many Test Executions should I create? And when?
  - Do I have to use Sub-Test Executions?
  - Why should I use Sub-Test Executions?

Executing Tests in multiple environments

Sometimes, you're testing in different target systems, browsers, devices, and database providers.

If the Tests are exactly the same, but you want to track the results on those different environments, then you should create Test Executions for each environment and assign the proper Test Environment value.

More info about Test Environments can be found in Working with Test Environments.

How to manage assignment

The Test Execution provides two levels of assignments:

- at the Test Execution issue level (since it's an issue, it has an assignee)
- at the individual Test (run) level

Test Executions per each test executor

By default, the individual Test Runs are assigned to the same user as the Test Execution assignee. This is the simplest scenario.

Therefore, you can create different Test Executions for different persons, each one will contain only the Tests aimed to be run by that person. In this case, the Test Execution assignee would be the same as the assignee of the individual runs.

Test Executions assigned to QA manager

Some teams have a QA manager/lead that is responsible for managing the lifecyle of the Test Execution, including

- ensuring the execution progress is on track
- reviewing the results

Test Executions may contain many Tests to be run. In this case, the Test Execution assignee would be the QA manager while the individual Test Runs would be assigned to one or more different persons.

Recommendations

- Avoid making cumulative testing assumptions because it may not be that reliable. You may have a bunch of Tests that you executed once (some passed and some failed), then you create Test Executions just for the failing tests, assuming that the other tests will still have the same result. As you may know, changes you make related to a faulty/incomplete requirement may affect other requirements; therefore, changes you make due to failed tests may implicitly affect other tests that supposedly were already OK.
- Automate as much as possible, including your regression testing.
- Take advantage of the fact that the Test Execution is an issue type. That means you can use workflows in order to track the progress of the Test Execution. Xray provides some workflow possibilties for Test Executions as mentioned in Miscellaneous

FAQ

Do I need to create a Test Execution every time I need to run the same Tests? Can't I update an existing one?

For historical reasons, you should create a new Test Execution with the Tests that you want to run in that version/revision of the system. That way you'll be able to see the results you obtained in that specific iteration, i.e., in that Test Execution that ran in a specific revision of the system.

You can also update the same Test Execution but you'll loose the benefit of tracking how your test results evolve through time.

How many Test Executions should I create? And when?

Each project is different so you should create as many as needed in order to ensure the quality of the product you're working on. The quantity and the timing depends on the process and methodology you have implemented. A rule of thumb is you should start testing as soon as possible and avoid testing only at the end of your development cycle.

If you want to validate some components or subsets of the system, you may create specific Test Executions for it. The how and when depends on your approach.

Do I have to use Sub-Test Executions?

No, you don't. They're optional. If you don't want to use them, you can just remove the issue type from your project.

Why should I use Sub-Test Executions?

Sub-Test Executions provide an easy way to schedule an execution containing all the Tests that validate a given requirement.

They are handled as sub-tasks of the parent requirement. This means that you gain some out-of-the-box features such as the:

- ability to track the progress of the Test Execution in the Agile board, by seeing it in-context with the parent requirement
- ability to count the time estimates of the Sub-Test Executions within the overall time estimate at the parent requirement
- ability to optionally restrict workflow transition on the parent requirement based on the workflow status of the related sub-tasks (e.g., allow transition of the requirement to "closed" only if the Sub-Test Executions are also "closed")