## How to Align Testing with an Agile Delivery Process

Once you determine which testing methodology is right for your organization, you’re not quite done yet. You still need to align testing with the development and delivery. To achieve this goal we recommend a three-pronged approach:

### **1) Get involved in the development process as early as possible**

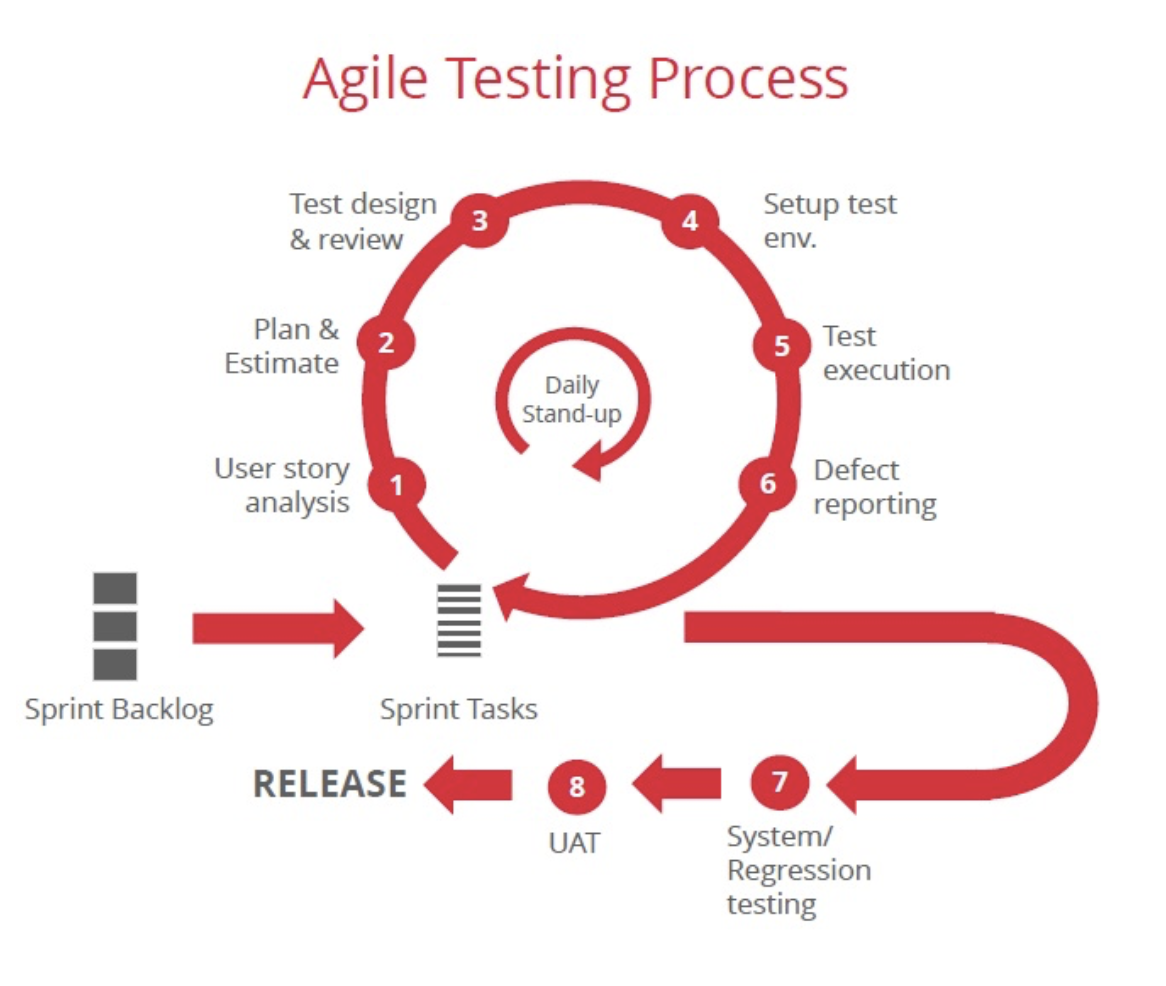
The sooner testers can get involved, the better. Ideally, testers should be present from day one. That’s because QA should and must be a part of process from the beginnings and will provides a higher level of insight into requirements and goals, encourages collaboration and helps to conduct frequent (if not continuous) testing.

### **2) Test frequently, but thoughtfully**

As more and more teams adopt Agile methodologies, efficiency is everything. User story validation and continuous integration as well in order to keep things moving, and that requires testing more frequently. But in the midst of an efficiency and frequency focused shakeup, testers need to remain thoughtful so as not to create more overhead and run unnecessary tests that actually slow down the process.

### **3) Hit the ground running with test creation**

Keeping in mind the need for speed in today’s Agile, DevOps driven world, testers need to hit the ground running when it comes to getting tests created. Specifically, the more testers can reduce the time from requirements gathering to test creation, the better. Having a seat at the table for all conversations from the very beginning should help in this regard.



## What’s Next for Agile Testing?

Going forward, more widespread adoption and greater maturity of Agile methodologies will require testers to go beyond test creation and execution and begin to focus on code delivery and integration too. At the same time, testers will need to hone their automation skills, become more involved in the entire software development process and continue to develop a collaborative relationship with developers.

In the future, three key tenets will become particularly important for testers working in Agile environments:

### **1) Communication**

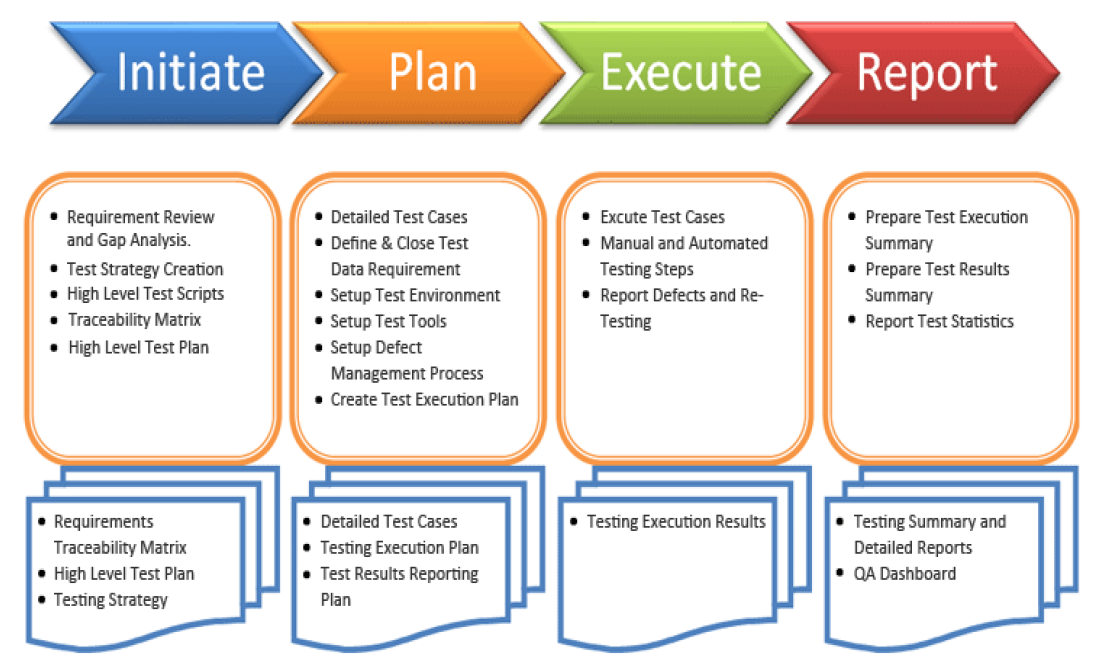
Agile requires a tight collaboration between testers and developers, and that collaboration makes communication a top priority for testers. Additionally, in a world where quality becomes everyone’s responsibility, testers will become “quality champions” that serve as internal experts, which will put their ability to clearly communicate testing needs and reasoning under the spotlight.

### **2) Skill Diversity**

In an Agile environment, everything can change on a dime, and that requires testers to be adaptable. Part of this adaptability is having a diverse skillset so that testers can change course as needed. For instance, functional testers need to expand their skills beyond manual scripted execution. This diverse skillset will be a must as different sprints require different types of testing to be executed in a short timeframe.

### **3) Business Mindset**

Finally, Agile takes on a very customer-centric approach in order to ensure customers receive as much value as possible as quickly and as early on as possible. Testers have a big role to play in delivering this value, but it requires them to take on a business mindset so that they can understand customer expectations, desires and concerns and develop their testing strategies accordingly.



# Test Matrix for Test Run- ( sample doc in wiki)

* + QA team will create a test matrix with test cases required to be run for feature validation
  + QA will provide a status report of feature validation during milestone and risk analysis. in case testing process may not be completed by assigned milestone.

# Test Scenarios are ideated, generated and provisionally reviewed and approved.

Review by other QA Tester or Developer?

Of course having a “Requirements-Usability” person reviewing test cases adds a lot of value to the validity of test cases.

(Here, we are talking about developing system test cases or user acceptance test cases).

The problem is that this process is often overlooked due to lack of resources and organizing reviews with the potential users or user representatives, instead test case reviews would progress only with the developer involved on the same project and functionality. As we know very well, testers and developers have different mindsets and look at testing with different views. Although the developer can give useful information to the tester designing test cases, such as which areas of the software are more complex or which parts have been directly affected by a code change, they should never guide testers in writing test cases or tell the testers how a particular function should be tested, that is the job of the tester alone! Therefore, a review by the developer would find more technical issues whereas a review by the user or user representative would find more functional / usability issues.

The more pragmatic approach would be to have the test cases reviewed by a peer, possibly more experience and senior tester before sending the test case document for review to any other role (developer or other stakeholders).

Benefits of having test case reviews

* Defect prevention while reviewing Requirements: SQA tester could gain knowledge of the application plus spot any ambiguity / unclear statements in requirement document before any code being developed.
* Conceptual and Technical Coverage: Requirements – Usability ensures the coverage from the Concept point of view and Developer ensures the coverage from the Technical Point of view. The traceability coverage track is assumed by traceability tools (Internal Tool to manage user stories, issues, completion, report etc..)
* Defect prevention while reviewing test cases: If the developer has the opportunity to check the test cases while implementing code, it is possible that this will help him to realize codes that may cause potential defects.

After having the test cases reviewed, the QA team receives all the feedback and decides, based on experience and knowledge on Software Testing and QA, and also on the functionality if comments have been applied or not. When not applied, the reason should be explained and discussed with the developer since there should be a final agreement with all stakeholders on the requirements and consequently on the test cases written.

# Test Scenarios are converted into Test Cases

* + Test scenarios: test instructions or step by step instructions of how to perform the test ( Specific areas to cover, special steps etc)
  + Test case: include test specification/test header (what need to test) and test instruction/test scenario (how to test) with different method or approaches as possible.
  + Step by step with "positive and negative" test cases, error , alerts, notification validation
  + Test cases must include special pre-conditions for specific test scenario (for example: test cases related to IOS version 9.0.2 only etc)

# Test Cases, inclusive of Pre-Conditions, Test Steps, etc. are authored

* + All required steps and format referred to best practice template: Best Practices for Test Case Creation

# Test Cases are reviewed and approved.

* + QA Team a major part of the scram process , QA involved in discussion and analytics of time and resources to finish validation from day one.
  + QA engineering resources will be assign to user story from day one
  + QA Engineer start evaluation user story and analyst test approach to validate feature / new functionality
  + QA will provide number of sprint-point to cover testing part in sprint ( if max point over a sprint max numbers it will require 2 separate test story or one test story for 2 sprint)
  + QA Engineer will be involved in discussion with developer to set expectation of testing scenario and testing steps to validate feature developer will start develop
  + QA will provide a test cases/ test scenario for review for all required stakeholders( PM, Developer. etc)

# To succeed, testers need to be flexible and able to adapt to moving targets.

## **\*\*\*\*\* For valuable tips and information:**

From numbers of years of Agile testing experience ( how much time I'll get upfront for any unpredicted requests and emergency support, dev request for help etc... that time I will have to use to create and run all best possible test cases to cover as much as possible for such time frame)

* + Never fully utilize QA team in the projects, keep buffers for side tasks. ( As usually it may comes emergency request from support, dev team or PM etc, and it will affect our delivery schedule)
  + Best practice is to take TC writing / review separately from execution test run in sprint ( as usual it may have no time for review by other eng resources, and need some time for change etc..)
  + If feature required to be tested on multiply devices or environment, always to add extra time for pre-set, preconditions setup, lab use, etc etc..
  + if resources available 5 hours a day, it does not mean it will stay and work for all 5 hours, need to make sure we need extra time for brakes, meetings, consulting etc...( actually it may take 3 hours)
  + If possible to use last day of the sprint as a buffer zone in case of any unpredictable..( deployment failed, bug with P-1 , environment is not ready etc..)
  + If User story verification failed with issue it may pass and close development story, but bug need to be fixed before close sprint ( unless if not critical and not stoped from other validation ( example. UI , performance issue not affect other test cases..)