

# Development - QA Process Overview

12/29/2023 4:25 pm EST

## QA Strategy and Tactics

"A vision without a strategy remains an illusion."  
-- Lee Bolman

## Contents

1. Overall Strategy
2. QA Tactics
3. Bugs
4. QA Workflow

## Overall Strategy

"The best way to get a project done faster is to start sooner."  
-- Jim Highsmith

The team uses a variety of methods and best practices to improve the overall product quality. At a high-level view, our goals are to:

- Find bugs early in the sprint lifecycle to reduce the cost of the quality and accelerate delivery.
- Use checks and balances to reduce dependence on individual achievement.

To achieve these goals, we:

1. Use several QA tactics (Exploratory testing and Regression testing) to increase the probability of finding defects.
2. Track metrics such as quality escapes and hotfixes to drive a strategy of continuous improvement.
3. Document all test cases and test results. This is an extremely important part of our strategy (see below).
4. Start testing early in the development process! Don't wait for an entire feature to be completed before ever testing it.

## QA Tactics

"Strategy without tactics is the slowest route to victory. Tactics without strategy is the noise before defeat."

To ensure that the team catches all defects, we rely on the following tactics to meet our QA goals:

- **Test Cases** - To build repeatedly into our QA process, we document test cases and test results used to validate various features in the product. We record the result of each test case each time we execute it. This ensures that the quality of testing does not depend on the ability of the person running the test. Instead, anyone can run tests because the steps are clearly documented.
  - **Feature Test Cases** - The QA Engineer writes these test cases against the acceptance criteria developed in the planning phase. The feature is not complete until all test cases pass.
  - **Bug Test Cases** - When a bug is found on a feature, the QA Engineer will create a bug and associate it to the feature and create a test case on the feature that covers the bug. When an external bug is found, the QA Engineer will create a bug with an "External" tag and create a test case on the Bug.
- **Unit Test** - A software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit tests are run as part of our continuous integration.
- **Regression Test** - When a feature is complete, the team creates a deployment package containing the verified changes for each of the stories of the sprint. We call this package a "release candidate." A member of the QA team identifies the areas of the product that have changed since the last release and selects a set of appropriate test cases to create a regression test suite. The goal of the regression test suite is to verify that the changes introduced by the release candidate do not break any working features of the product. The QA engineers execute the tests in the regression test suite. Fixes for bugs found during regression testing are applied to create a new release candidate.
- **Design Reviews** - Before developing new features or fixing issues, developers often conduct formal or informal design reviews with their peers, including QA team members. This practice leads to improved design and increases the testability of the product.
- **Feature Testing** - We test each user story of code change to verify that it meets the requirements. There are no exceptions--all code changes are tested before we deploy the change to production. We use the following types of tests:
  - **Manual Tests:** A QA engineer creates a set of test cases in DevOps to verify the code change. This allows anyone on the team to execute the manual test. We record the result of each test case each time we execute it.
  - **Automated Unit Tests:** A software developer writes code that verifies a module or component of the system. The build runs these tests automatically; if a test fails, the build fails.
  - **Automated Integration Tests:** A QA engineer or software developer writes an automated test that exercises a feature from the UI. The tests inject user actions into the browser and fail if the behavior does not match the expected result.

- **Branching/Pull Requests** - We use branching to test new features without disrupting the work of others. Once testing is complete, we create a pull request and then merge the branch into the master branch for deployment. For a pull request to be completed, it must be code reviewed by another developer and approved by a QA Engineer.
  - **Exploratory Testing** - A QA Engineer will pull the branch and run the test cases against the feature/bug. Most of the time Test Cases will give a good starting point for testing but fail to cover 100% of potential issues. When the QA engineer pulls the branch, they will also spend time doing "Exploratory Testing," or, testing things that are not document as a specific "Test Case." This is a vital step in the testing process as many, if not most, bug are found during this time.
  - **Code Reviews** - A different team member than the one who did the development reviews code changes before completion of the pull request. This gate detects defects early and ensures quality.
- **Beta Testing** - Before we enable new features for all clients, we first enable the feature for a subset of Beta clients using the feature flags. Feature flags allow us to enable/disable the feature on a client-by-client basis. The clients using the Beta release have typically expressed interest in the new feature and have offered their help in testing. The Beta test is a controlled exercise and the Scrum Master and/or Product Owner follow up with Beta clients to gather feedback about the new feature. We may find that a new feature does not meet the needs of some clients and this limited release allows us to modify the feature before rolling it out to all clients.

## Bugs

"Before software can be reusable it first has to be usable."  
-- Jeremy S. Anderson

Process Definition:

1. **Communicate Bug:** Any team members or customers can find a bug and communicate it to a member of the support team.
2. **Review Bug:** The support team member will review the bug and attempt to reproduce it to ensure it is valid. If the bug is valid, it is sent to the QA engineer.
3. **Add bug to TFS:** The QA Engineer reviews the bug and adds it to DevOps and creates a test case if it's valid.
4. **Fix Bug:** The QA Engineer then assigns the bug to a sprint and commits to fixing it.

Priority

A subjective rating of the bug as it relates to the business or customer requirements. Priority indicates the order in which code defects should be fixed. You can specify the following values:

- 1: Product cannot ship without the successful resolution of the work item, and it should be addressed as soon as possible.
- 2: Product cannot ship without the successful resolution of the work item, but it does not need to be addressed

immediately.

3: Resolution of the work item is optional based on resources, time, and risk.

### Severity

A subjective rating of the impact of a bug on the project or software system. For example: If clicking a remote link (a rare event) causes an application or web page to crash (a severe customer experience), you might specify Severity = 2 - High and Priority = 3. Allowed values and suggested guidelines are:

1 - **Critical:** Must fix. A defect that causes termination of one or more system components or the complete system, or causes extensive data corruption. And, there are no acceptable alternative methods to achieve required results.

2 - **High:** Consider fix. A defect that causes termination of one or more system components or the complete system, or causes extensive data corruption. However, an acceptable alternative method exists to achieve required results.

3 - **Medium:** (Default) A defect that causes the system to produce incorrect, incomplete or inconsistent results.

4 - **Low:** A minor or cosmetic defect that has acceptable workarounds to achieve required results.

## QA Workflow

"Quality is never an accident; it is always the result of intelligent effort."

-- John Ruskin.

- A Feature is added in DevOps
  - A QA Engineer will read the requirements of the Feature and all the PBIs underneath that feature, talk to the developer about how the Feature will work, and fully understand the Functionality.
  - A QA Engineer will then write every test case that s/he can think of for this Feature/
- Pull Requests
  - A pull request will be created, the QA engineer will pull the branch down to his/her local machine and run all of the already written Test Cases.
  - Spend some extra time doing Exploratory testing on the branch. This will require knowledge of the application and an understanding of what areas could potentially have been broken by the new code.
  - If a Pull Request has an issue, QA will create a new Task to the Work Item and tag it with "Failed QA".
- Regression Testing
  - Once a sprint is complete, we will create a Release Candidate. The release Candidate is the cumulative of all the new features that have been code reviewed and tested over the course of the sprint. A QA Engineer will take the release candidate and run our full suite of Automated Regression Tests and review the results.