# Applied Software Project Management

## Software Testing

# Quality

- Quality means “conformance to requirements”
    - The best testers can only catch defects that are contrary to specification.
    - Testing does not make the software perfect.
    - If an organization does not have good requirements engineering practices then it will be very hard to deliver software that fills the users’ needs, because the product team does not really know what those needs are.

# Test Plans

- The goal of test planning is to establish the list of tasks which, if performed, will identify all of the requirements that have not been met in the software. The main work product is the *test plan*.
    - The test plan documents the overall approach to the test. In many ways, the test plan serves as a summary of the test activities that will be performed.
    - It shows how the tests will be organized, and outlines all of the testers' needs which must be met in order to properly carry out the test.
    - The test plan should be inspected by members of the engineering team and senior managers.

# Test Cases

- A *test case* is a description of a specific interaction that a tester will have in order to test a single behavior of the software. Test cases are very similar to use cases, in that they are step-by-step narratives which define a specific interaction between the user and the software.
  - A typical test case is laid out in a table, and includes:
    - A unique *name* and *number*
    - A *requirement* which this test case is exercising
    - *Preconditions* which describe the state of the software before the test case (which is often a previous test case that must always be run before the current test case)
    - *Steps* that describe the specific steps which make up the interaction
    - *Expected Results* which describe the expected state of the software after the test case is executed
- Test cases must be repeatable.
  - Good test cases are data-specific, and describe each interaction necessary to repeat the test exactly.

# Test Execution

- The software testers begin executing the test plan after the programmers deliver the *alpha build*, or a build that they feel is feature complete.
    - The alpha should be of high quality—the programmers should feel that it is ready for release, and as good as they can get it.
- There are typically several iterations of test execution.
    - The first iteration focuses on new functionality that has been added since the last round of testing.
    - A *regression test* is a test designed to make sure that a change to one area of the software has not caused any other part of the software which had previously passed its tests to stop working.
    - Regression testing usually involves executing all test cases which have previously been executed.
    - There are typically at least two regression tests for any software project.

# Defect Tracking

- The *defect tracking system* is a program that testers use to record and track defects. It routes each defect between testers, developers, the project manager and others, following a workflow designed to ensure that the defect is verified and repaired.
  - Every defect encountered in the test run is recorded and entered into a defect tracking system so that it can be prioritized.
  - The defect workflow should track the interaction between the testers who find the defect and the programmers who fix it. It should ensure that every defect can be properly prioritized and reviewed by all of the stakeholders to determine whether or not it should be repaired. This process of review and prioritization referred to as *triage.*

# Smoke Tests

- A *smoke test* is a subset of the test cases that is typically representative of the overall test plan.
    - Smoke tests are good for verifying proper deployment or other non invasive changes.
    - They are also useful for verifying a build is ready to send to test.
    - Smoke tests are <u>not</u> substitute for actual functional testing.

# Test Automation

- *Test automation* is a practice in which testers employ a software tool to reduce or eliminate repetitive tasks.
    - Testers either write scripts or use record-and-playback to capture user interactions with the software being tested.
    - This can save the testers a lot of time if many iterations of testing will be required.
    - It costs a lot to develop and maintain automated test suites, so it is generally not worth developing them for tests that will executed only a few times.

# Postmortem Reports

- The *postmortem report* is an overall account of the team’s experience in building the software, and of the experience of the users and stakeholders in working with the team.
    - The report should contain an honest assessment of how the team members, users, and stakeholders perceived the end product, and assessed the decisions made throughout the project.
    - The purpose of the post-mortem report is to highlight the team’s successes and identify any problems which should be fixed in future releases.