Introduction to QA (Glossary)

Omid Vahdaty

Glossary

- **Acceptance test**: quick test to determine if software is ready for release.
- **Ad Hoc testing** : a fly by - try to test the product freely to find bugs.
- **Alpha testing**: testing done before a release. Can be done by an outside company
- **Beta testing** - testing done after a release for a narrow audience.
- **Compatibility testing**: confirm software is compatible with hardware
- **Component Testing**: a specific component testing.
- **Data integrity**: Correctness, completeness, wholeness, soundness, and compliance with the intention of the creators of the data
- **Functional testing** : Testing with the purpose of finding bugs in functionalities of the software.
- **Integration Testing**: Functional testing of the interaction between two or more integrated components.

Glossary

- **Negative Testing**: Testing that checks situations involving:
  - User error
  - System failure
  - Post Mortem: analysis after failure/critical bug.
- **Regression Testing (RT)**
  - Checking to determine if legacy features are broken because of a particular change in the software; e.g.:
  - 1. It’s often extremely difficult for a programmer to figure out how a change in one part of the software will echo in other parts of the software
  - 2. Programmers sometimes change software without even trying to figure out if their changes might break something else
  - regression testing comes after new feature testing.

Glossary

- **Test case**: an idea to be verified. I.e Scenario + Expected Result.
- **Test Coverage**: Depending on the context, this means one of the following:
  - 1. The coverage of possible scenarios (in product)
  - 2. Test case execution coverage (in code)
- **Test plan**: a master document containing information about activities regarding the testing of the certain features (or other possible subjects of testing – e.g., how the system handles load).
- **UI testing**
- **Usability Testing:** The evaluation of the user's experience when he or she uses our software.
- **Use Case**:
  - How a software will be used

Build/Version/RC/Release

- **Build** − executable or a library created by compiling source code.

- **Version** − a software build. New version is a different build.

- **Release Candidate** (RC) is the build **released** internally to check if any critical problems have gone undetected into the code during the previous development period. **Release candidates** are NOT for production deployment, but they are for testing purposes only.

- **Release** − (public release) a version intended for use by general population.

Approaches

- **Black box testing:**
  - – The tester usually has no idea about the internals of the back end
  - – Ideas for testing come from expected patterns of user behavior
- **Grey Box testing**
  - Testing that combines the elements of Black and White Box Testing:
  - On the one hand, the tester uses black box methodology to come up with test scenarios.

Approaches

- **White Box Testing**
  - The number of testing techniques that require a comprehensive understanding of the software code
  - A programmer can perform white box testing by comparing The requirements from the spec

Aliases

- End to end testing = System tests.
- Exploratory testing = Ad hoc testing.
- Load testing = Performance testing.
- Sanity testing = smoke/confidence testing

Perspective

- User Perspective - any scenario that end user is affected.
- System Perspective - any scenario where the overall user experience is not compromised, for instance - high redundancy testing.