

We now begin our discussion of software testing. Testing is part of the broader field of Quality Assurance (QA). Design and Testing are related subjects. The basis of a high-quality product is a high-quality design. Designs can also incorporate features that make the implemented system easier to test.

Software Development Products

In order to talk about QA and testing, we need a brief overview of the products created by a software development project:

- 1) The first step is to understand what the customer really wants (often times they don't even know). Work with the customer to learn **what** (not **how**) the software must accomplish. Document the customer requirements in a **requirements specification (or Product Requirements Document, or PRD)**. Usually created by a product manager or analyst that knows the customer's needs.
 - a. Example: Employees must be able to submit problem tickets to the IT department from work or home.
- 2) From the requirements specification we can create a **functional specification** that precisely defines the external behavior of the product we intend to build. It defines **what** we are going to build.
 - a. This is what your professor typically provides for a class programming project
- 3) From the functional spec we can create high-level design documents that define the architecture of the implementation (i.e., **how**).
- 4) From the high-level design we can create low-level designs for each subsystem.
- 5) From low-level designs we can create code
- 6) From the PRD and functional spec we can also create test plans.
- 7) From Functional Specification and the actual software we can write end-user documentation.

Validation and Verification

To avoid creating software customers don't like, we must constantly focus on two important questions:

- 1) Are we building the right product?
- 2) Are we building the product right?

1) Validation

- a. Are we building the right product?
- b. Does the product we're building meet the customer's needs?
- c. Validation activities
 - i. Accurately and completely capturing customer requirements
 - ii. Prototyping
 - iii. Delivering frequent builds that users can test drive
 - iv. Usability testing
 - v. System testing
 - vi. Alpha/Beta testing

2) Verification

- a. Are we building the product right? (If we're building the right product, and we also build it right, then we succeed.)
- b. Validation ensures that customer requirements are met. Verification ensures that the system is built according to spec.
- c. Verify that the product of each development phase is complete and consistent with respect to the product of the previous phase. (This can be done even if you're building the wrong product.)
 - i. Does the product defined in the functional spec satisfy all requirements in the PRD?
 - ii. Does the design cover everything in the functional spec?
 - iii. Do our unit tests cover everything in the design?
 - iv. Does the test plan test every feature described in the functional spec?
 - v. Etc.
- d. Every aspect of downstream work products should be **traceable** back to the customer requirements. Every requirement should be **traceable** to the downstream work products that realize it.
 - i. Does the functional spec contain features the customer didn't ask for?
 - ii. Are developers secretly adding cool features the customer didn't ask for?
 - iii. Are we testing things that don't matter to the customer?

Slides