13 Testing

13.1 Introduction

Different people have come up with various definitions for Software Testing, but generally, the goal with testing is:

- To ensure that the software meets the agreed requirements and design
- The application works as expected
- The application doesn't contain serious bugs
- Meets its intended use as per user expectations

Testing can be performed on different levels and by different persons. Testing is a very important part of software development. About 50% of the software development is about testing your software.

Since modern software has become very complex, testing has become a very important part of software development (see Figure 13-1).

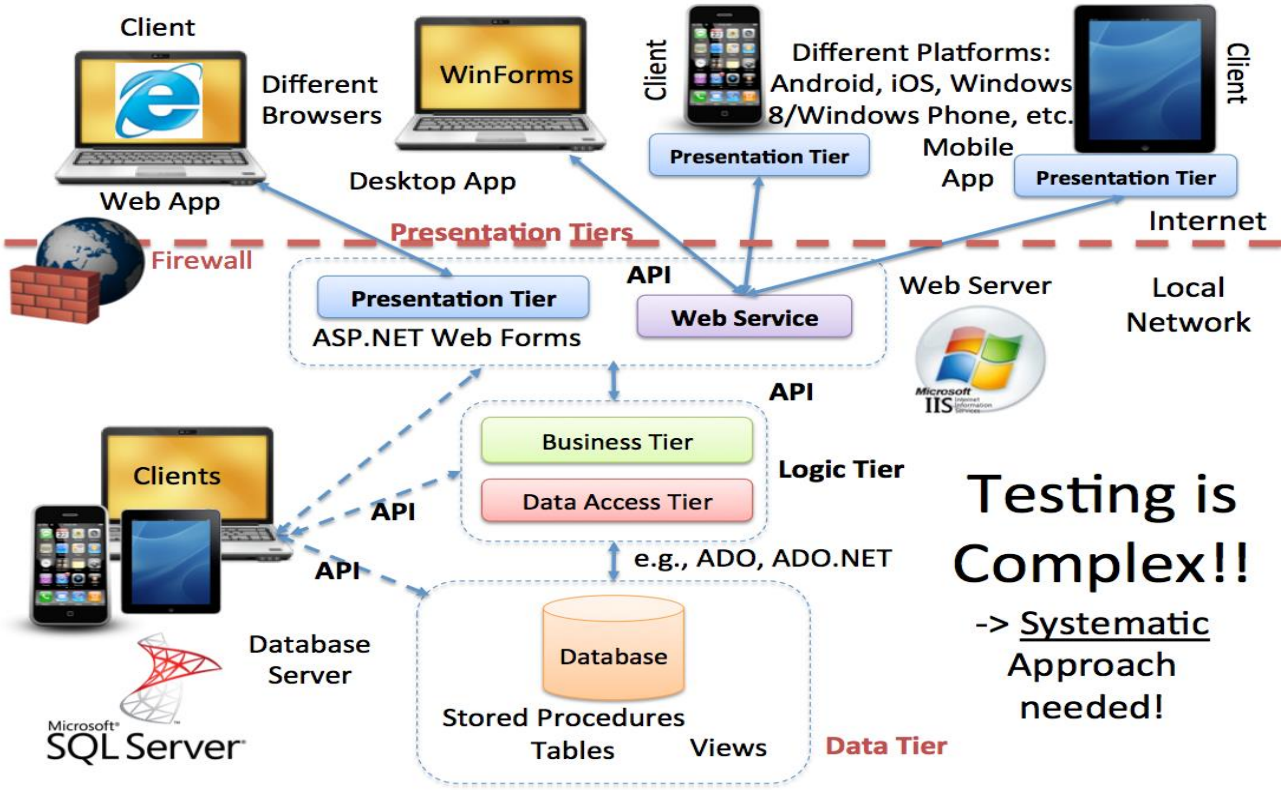

Figure 13-1: Modern Software Testing has become very complex

Since testing of advanced software systems is quite complex, we need a systematic approach to testing that involves different levels of testing (see Figure 13-2).

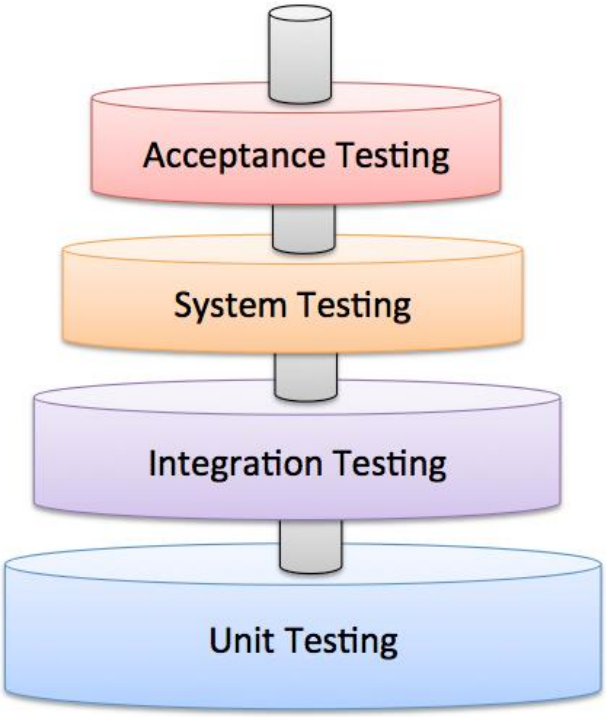

Figure 13-2: Systematic Testing

Since Software Development today involves different platforms, different devices, network, servers and clients, etc., it has become very complex to test it. Today we have not only ordinary Desktop Apps, we have Web Apps, Mobile Apps, Apps for TVs, etc.

The software we create is a layer between the user of the software and the hardware and the operating system (Figure 13-3).

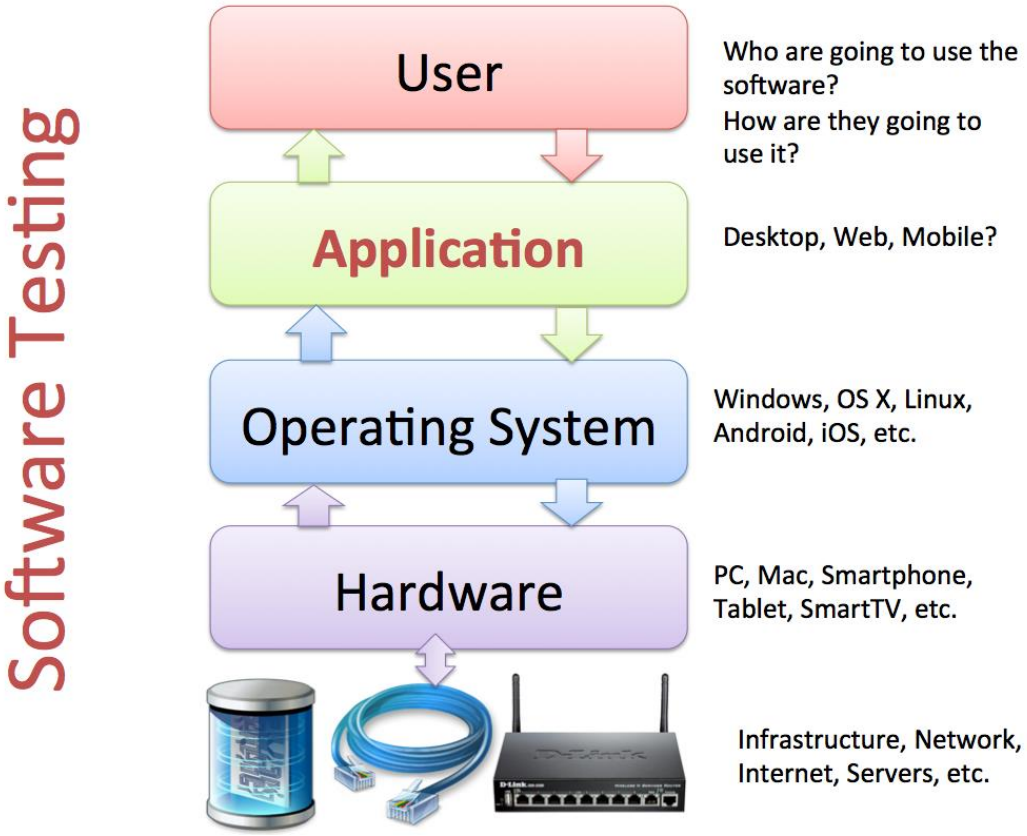

Figure 13-3: Components involved in Software Development & Testing

If we find bugs at the earlier stage, the cost to fix this will be less and thus it will reduce the overall cost of the application (Figure 13-4).

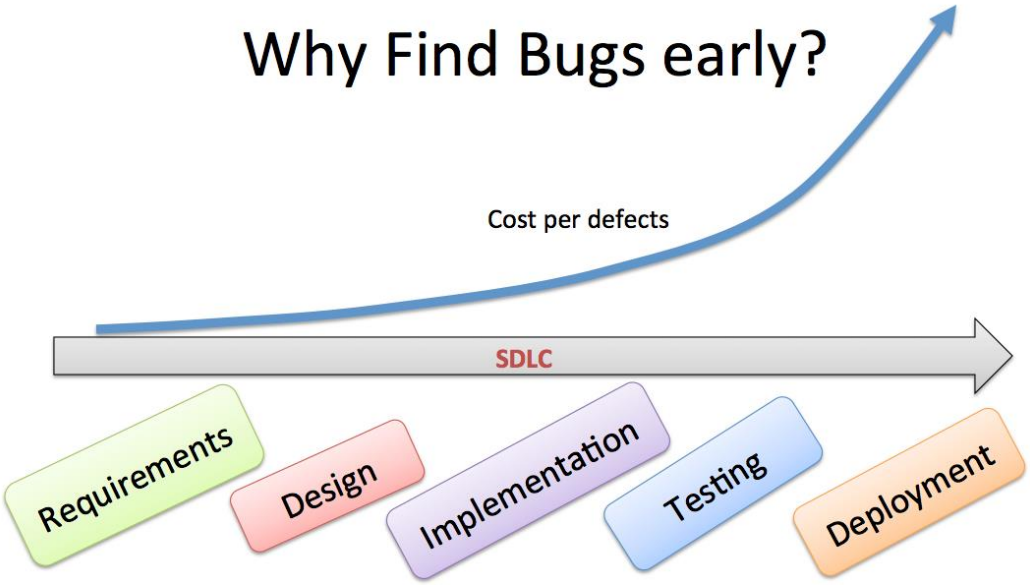

Figure 13-4: Find Bugs at an early stage

Figure 13-5 illustrates the necessary steps involved in testing.

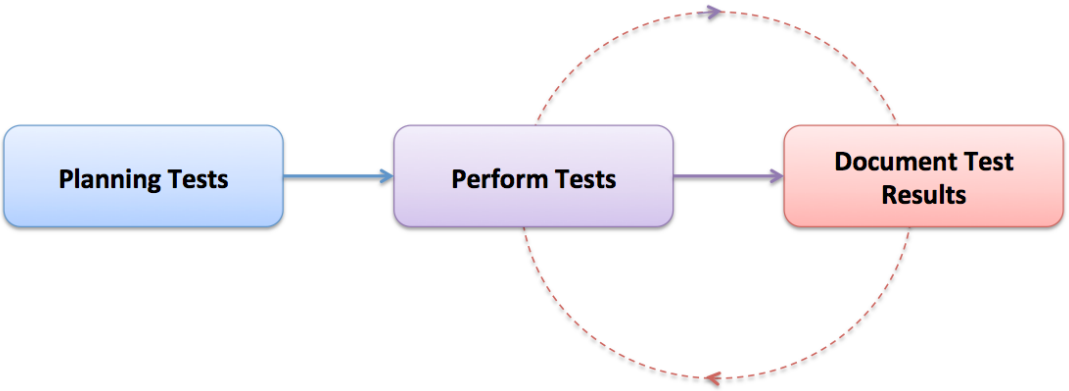

Figure 13-5: Software Testing

Testing is intended to show that a program does what it is intended to do and to discover program defects before it is put into use. When you test software, you execute a program using artificial data. You check for the presence of errors NOT their absence.

Testing is part of a more general verification and validation process, which also includes static validation techniques.

What is the purpose with Testing?

The main purpose with testing is as follows:

- To demonstrate to the developer and the customer that the software meets its requirements.
    - For custom software, this means that there should be at least one test for every requirement in the requirements document.
    - For generic software products, it means that there should be tests for all of the system features, plus combinations of these features, that will be incorporated in the product release.
- To discover situations in which the behavior of the software is incorrect, undesirable or does not conform to its specification.
    - This means undesirable system behavior such as system crashes, unwanted interactions with other systems, incorrect computations and data corruption.

A primary purpose of testing is to detect software failures so that defects may be discovered and corrected.

If we summarize why we do Testing:

- Finding Bugs in the Software before it is released to the Customer
- Finding unwanted system behaviors
- Verify/Validate that the Software works as expected (according to the Specifications)

- Find bugs as soon as possible!

It is commonly believed that the earlier a defect is found the cheaper it is to fix it.

There are different steps involved in the software testing process.

The steps are as follows:

- Design Test Cases
- Prepare Test Data
- Run the Software with the necessary Test Data
- Compare the results with the Test Cases

The final output of this process is a Test Report.

Basically, we do the following: Planning the Test, then we execute the Tests, finally we document the Test results.

Documents used in testing and created in the test process:

- **SRS** – Software Requirements Specifications: A document stating what at application must accomplish. The documents is the basic for the test plan, etc.
- **SDD** – Software Design Document: A document describing the design of a software application. The documents is the basic for the test plan, etc.
- **STP** - Software Test Plan: Documentation stating what parts of an application will be tested, and the schedule of when the testing is to be performed
- **STD** - Software Test Documentation: Introduction, Test Plan, Test Design, Test Cases, Test procedures, Test Log, …, Summary

We have the following stages in testing:

1. **Development testing**, where the system is tested during development to discover bugs and defects. Development testing includes all testing activities that are carried out by the team developing the system.
2. **Release testing**, where a separate testing team test a complete version of the system before it is released to users.
3. **User testing**, where users or potential users of a system test the system in their own environment.

Development testing: Development testing is the responsibility of the software development team. A separate team should be responsible for testing a system before it is released to customers.

Release testing: Release testing is the process of testing a release of a system that is intended for use outside of the development team.

The primary goal of the release testing process is to convince the supplier of the system that it is good enough for use. Release testing is usually a black-box testing process where tests are only derived from the system specification.

User testing: We have different types of user testing:

- **Alpha** testing
  - Users of the software work with the development team to test the software at the developer's site.
- **Beta** testing
  - A release of the software is made available to users to allow them to experiment and to raise problems that they discover with the system developers.
- **Acceptance** testing
  - Customers test a system to decide whether it is ready to be accepted from the system developers and deployed in the customer environment. Primarily for custom systems.

13.1.1 Test Levels

In Figure 13-6 we see different test levels.

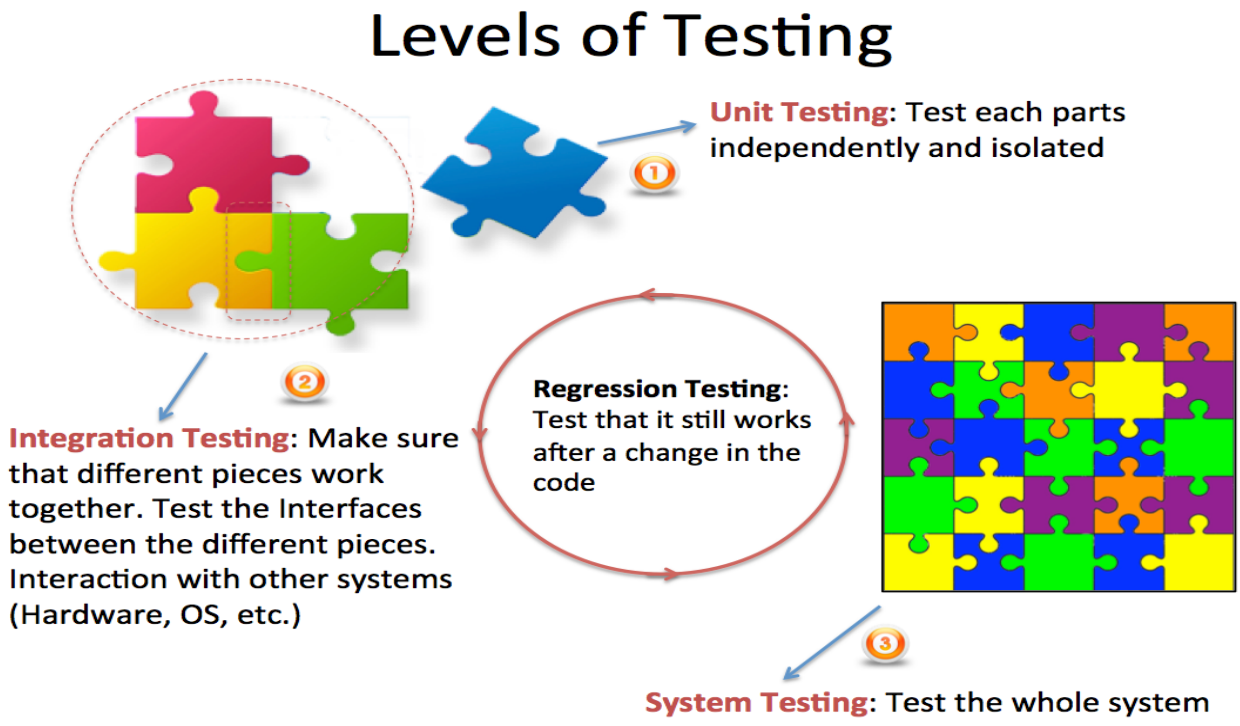

Figure 13-6: Levels of Testing

Short overview of the different Test levels in Figure 13-6 (more details later):

- Unit Tests are written by the Developers as part of the Programming. Each part is developed and Unit tested separately (Every Class and Method in the code)
- Regression testing is testing the system to check that changes have not “broken” previously working code. Both Manually & Automatically (Re-run Unit Tests)
- Integration testing means the system is put together and tested to make sure everything works together.
- System or validation testing is Black-box Tests that validate the entire system against its requirements, i.e., checking that a software system meets the specifications
- Acceptance Testing: The Customer needs to test and approve the software before he can take it into use. We have 2 types: FAT (Factory Acceptance Testing) and SAT (Site Acceptance Testing).

13.1.2 Bug Tracking

A software bug is an error, flaw, failure, or fault in a computer program or system that produces an incorrect or unexpected result, or causes it to behave in unintended ways

They found a bug (a moth) inside a computer in 1947 that made the program not behaving as expected. This was the “first” real bug.

13.1.3 Software versioning

Software versioning is used to separate different version of the same software, both before it has been released and for subsequent releases. See example in Figure 13-7.

Before the software is released:

- Alpha Release(s)
- Beta Release(s)
- RC - Release Candidate(s)
- RTM – Release To Manufacturing

Maintenance (after the software is released):

- Patches (small fixes)
- SP - Service Packs
  (lots of small fixe and patches bundle together)
- …
- Start Planning next release

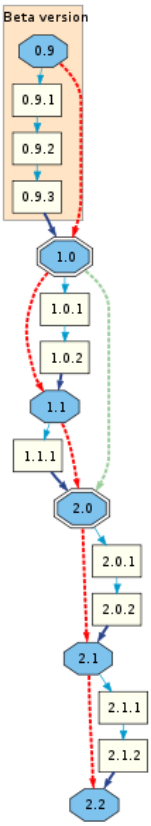

Figure 13-7: Software versioning

Software testing should be performed during the whole Software Development Life Cycle (SDLC) as shown in Figure 13-8.

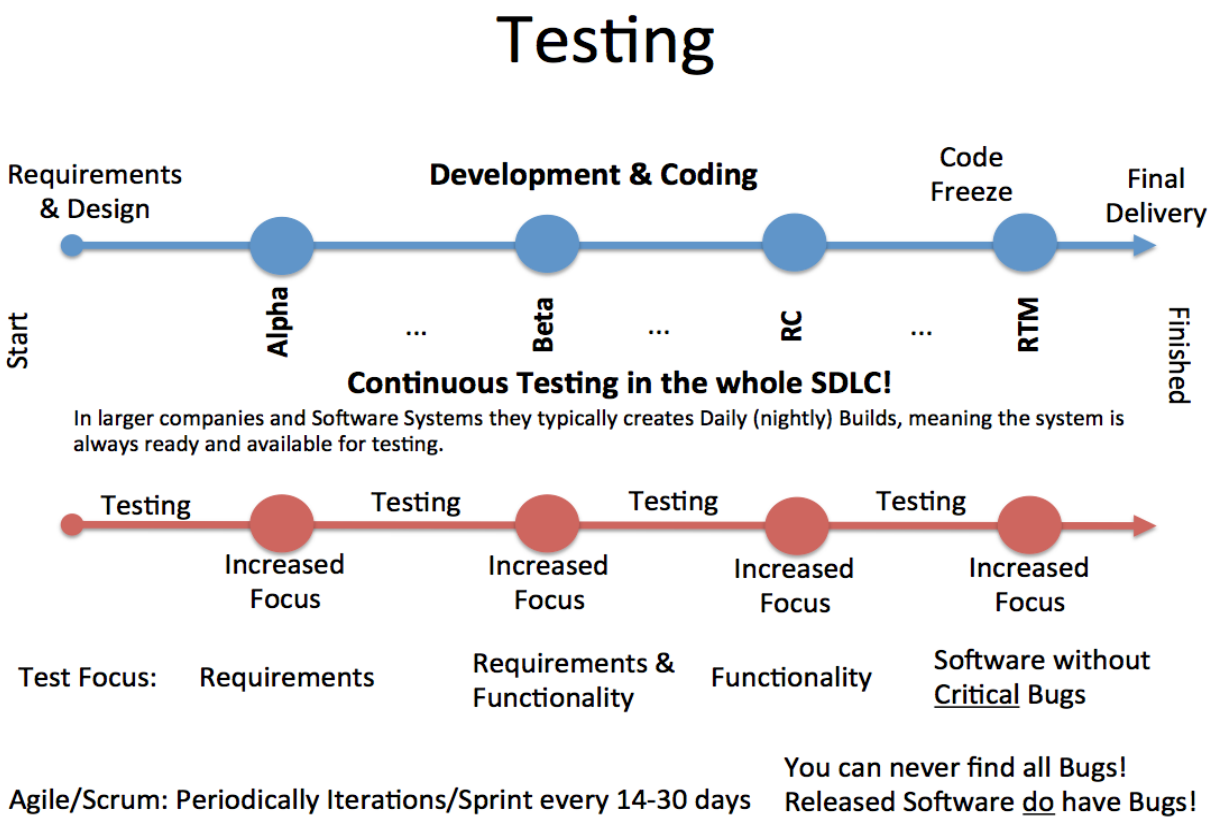

Figure 13-8: Testing during the Software Development Life Cycle (SDLC)

Sooner or later you have to say enough is enough and release version 1.0 (see Figure 13-9).

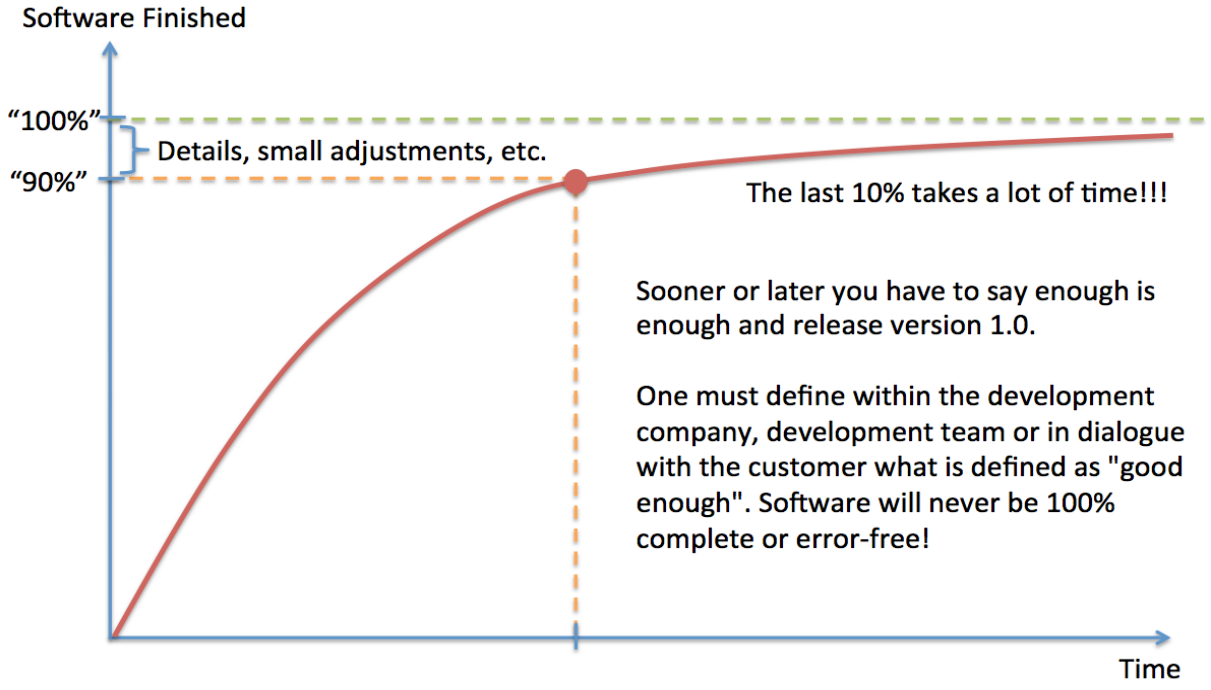

Figure 13-9: When is the Software Finished?

One must define within the development company, development team or in dialogue with the customer what is defined as "good enough".

Software will never be 100% complete or error-free (see Figure 13-10)!

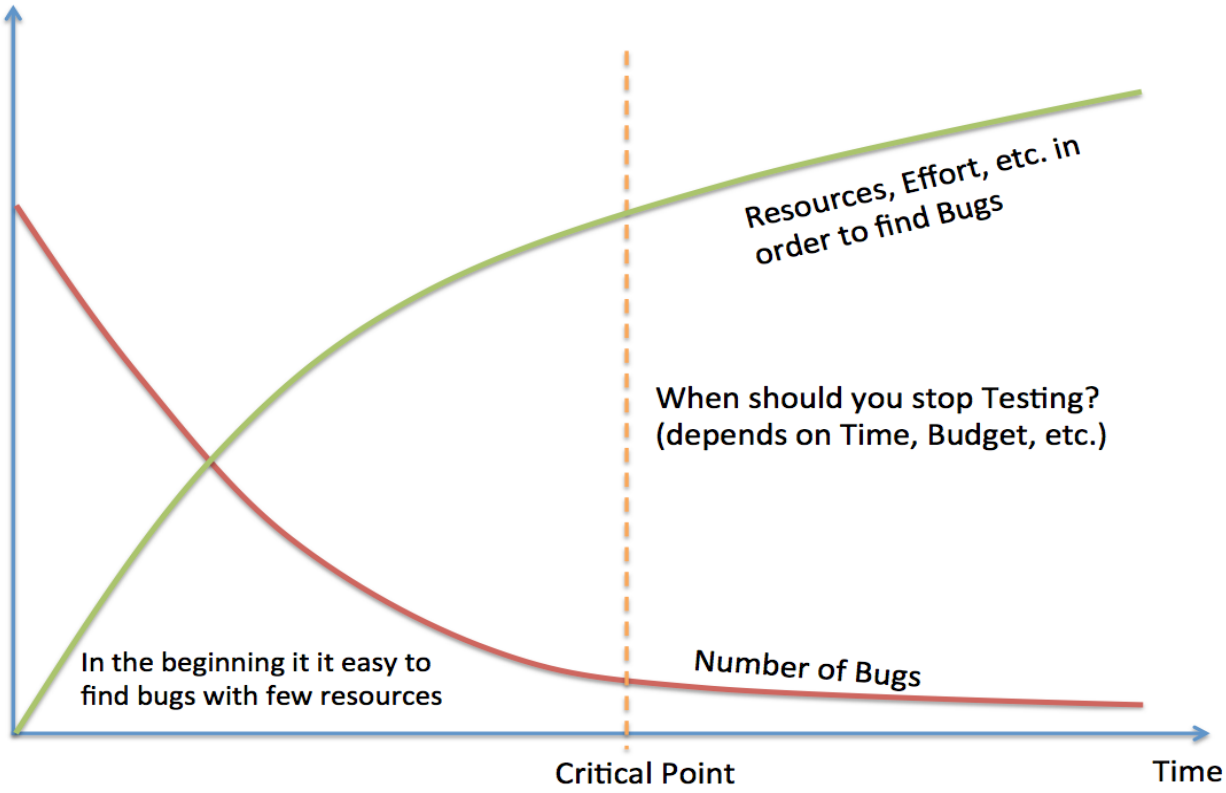

Figure 13-10: When are you Finished with Testing?

13.2 Test Categories

We can divide testing into 2 different categories, which is:

- Black-box
- White-box Testing

13.2.1 Black-box Testing

Black-box testing is a method of software testing that examines the functionality of an application (what the software does) without going inside the internal structure (White-box Testing).

You need no knowledge of how the system is created. Black-box testing can be done by a person who only know what the software is supposed to do. You may compare to driving a car – you don't need to know how it is built to test it.

13.2.2 White-box Testing

In White-box Testing you need to have knowledge of how (Design and Implementation) the system is built. White-box Testing is also called “Glass-box testing”.

In Figure 13-11 we see how White-box testing is working.

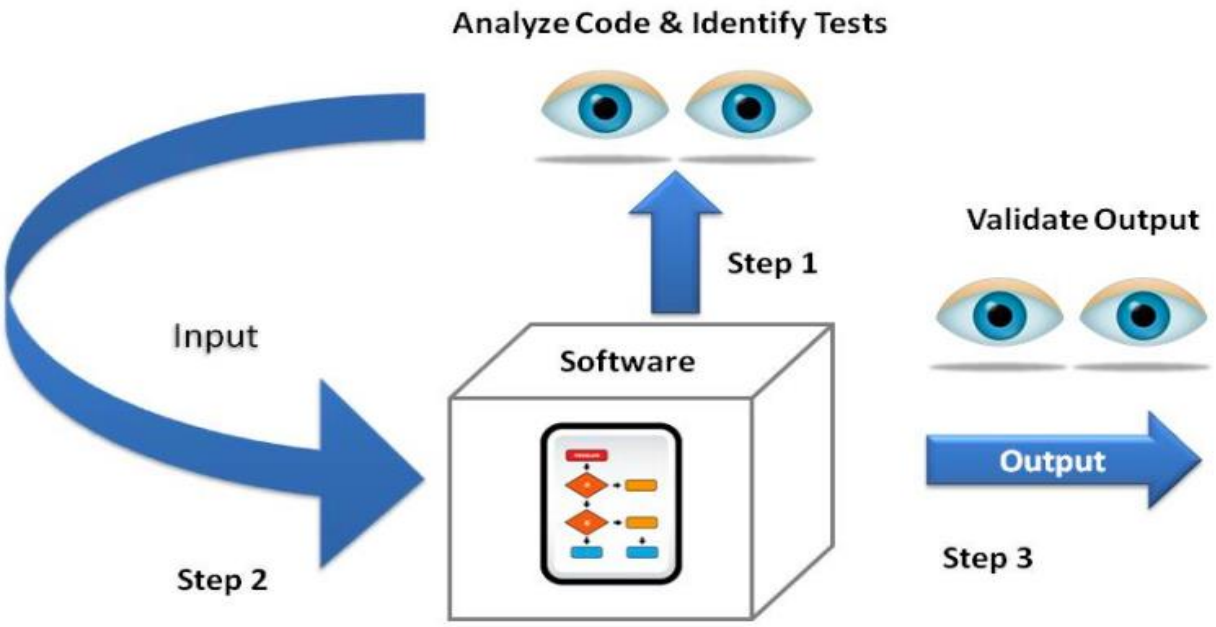

Figure 13-11: White-box Testing

13.3 Test Levels

As mention earlier we have different Levels of Testing (see Figure 13-12).

- Unit Testing
- Regression Testing
- Integration Testing
- System Testing
- Acceptance Testing

These are explained more in detail below.