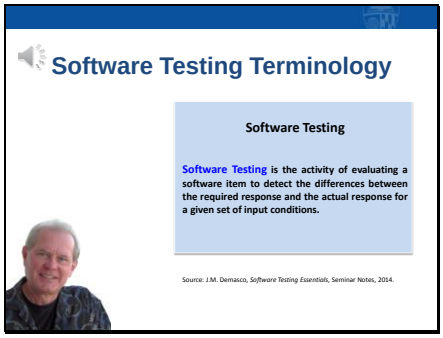
1

In this lecture we'll introduce some terminology associated with software testing.

2

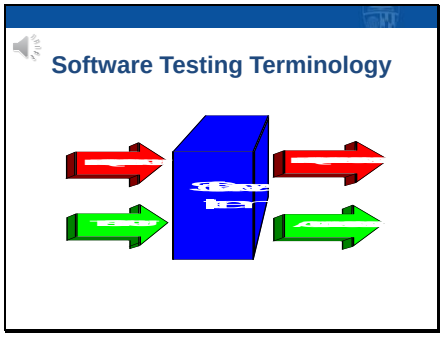


So...what is software testing anyway?

Simply put, software testing is the activity of evaluating a software item to detect the differences between the item's required response and its actual response for a given set of input conditions.

Now, there's more to software testing from a project perspective than is covered in this definition. The overall process of software testing consists of numerous activities such as test planning, test design, test execution, and so forth, that we'll talk more in detail about later.

3

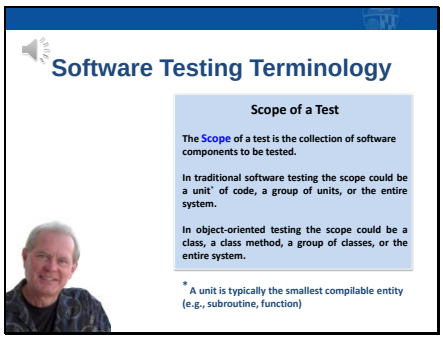


I like this definition of testing because it describes what it means to test into more of an engineering framework, which I illustrate here. And...it also maps nicely into what needs to be documented as part of each test applied to the item.

The blue box is the software item being tested. The item can be an entire system, a subsystem, or a very atomic component like a function or subroutine. The item under test will take certain inputs, and, if the item has been implemented correctly and completely it should provide certain outputs or responses to the inputs.

When we test the item, we create a set of test inputs, apply them to the item...usually by executing it...and observe an actual response. The actual response is then compared to the required or expected response. If the responses match, that's good. If the responses don't match, that's an indication that something may be wrong with the item.

4

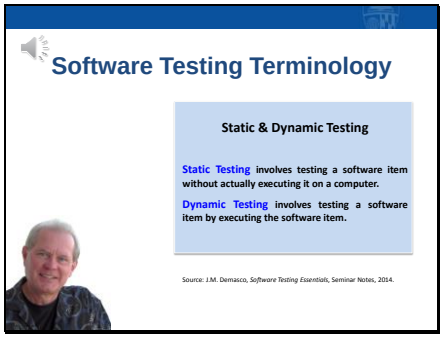


Sometimes testers talk about the scope of a test. The scope of a test is the collection of underlying software components that will be tested.

In traditional software testing, the scope could be a unit of code tested in isolation, a group of units, or the entire software product. When object-oriented software is tested, the scope could be a class, one or more methods in a class, a group of classes, or the entire system.

The level of testing that's being performed, and who is doing the testing, typically determines the scope.

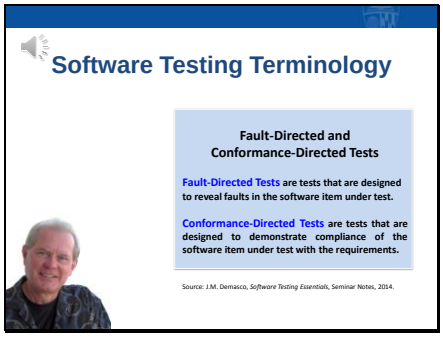
5



When most software engineers think about the activity of testing they usually think in terms of executing the software, as was illustrated in my earlier diagram. That's the most common type of testing activity...and we refer to it as dynamic testing.

But there's also another type of testing activity, called static testing, that is often performed both on software components and certain work products. This form of testing doesn't execute any code. The most common form of static testing is some time of review...like a code walkthrough or code inspection.

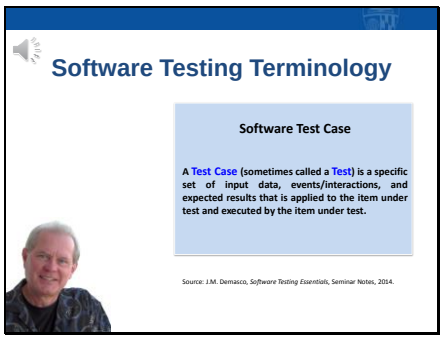
6



When I review client test efforts, I like to categorize tests into those that are fault-directed and those that are conformance-directed. Fault-directed tests are designed to try and expose faults in the item under test. For example, a test may include invalid data types or ranges. Conformance directed tests are designed to demonstrate that the item under test satisfies its requirements. For example, a test of a web-based shopping cart product might involve selecting several items for purchase to verify that the total purchase amount and any applicable discounts are correctly calculated.

A good test effort will consist of both categories of tests. A poor test effort might focus mostly on conformance-directed tests.

7

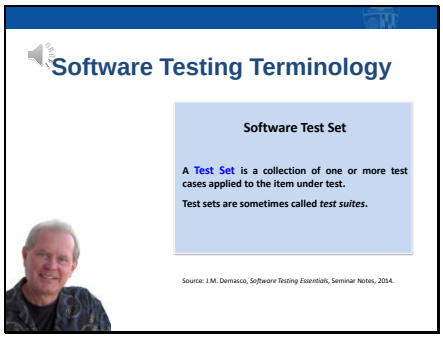


I've been using the term test all through this lecture, but I've yet to define it. A test is a specific set of input data, events and interactions, and associated expected results, that is applied to the item under test.

The data associated with a test can be a single set of data associated with a simple transaction...like entering three integers that are supposed to represent the sides of a triangle...or...it could represent a a set of transaction data, like a month's worth of debit and credit data for an accounting system.

I generally prefer to use the term test case rather than test...because the term implies something more specific. For example, when testing an application that determines triangle types...I may have one test case to test for each type of triangle...and the data for each test case would vary. When I test an accounting system, I might have a month's worth of debit and credit data, and use that same data to test different accounting functions. The test of each different function would consist of at least one test case.

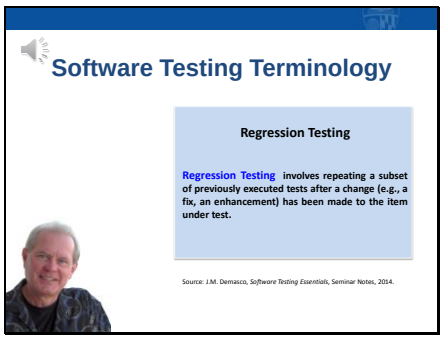
8



A test set is a collection of one or more test cases applied to the item under test. Test sets normally group test cases that are testing for similar types of responses...for example, I may have a test set that has five test cases used to test error-handling in the item under test, and a test set of three test cases to test performance issues, and so forth.

You'll see several examples of test sets in this course module.

9



A type of testing that is very important to consider is called regression testing. Regression testing involves repeating a subset, and sometimes a complete set, of previously-executed tests after a change has been made to the item under test.