



SWE 4603

Software Testing and Quality Assurance

Lecture 6

Prepared By Maliha Noushin Raida, Lecturer, CSE
Islamic University of Technology

Lesson Outcome

- Smoke Testing
- Sanity Testing
- Regression Testing

Week 7 On:

- Chapter 8: Regression Testing

Testing Categories & Techniques

Testing Category	Techniques
Dynamic testing: Black-Box	Boundary value analysis, Equivalence class partitioning, State table-based testing, Decision table-based testing, Cause-effect graphing technique, Error guessing.
Dynamic testing: White-Box	Basis path testing, Graph matrices, Loop testing, Data flow testing, Mutation testing.
Static testing	Inspection, Walkthrough, Reviews.
Validation testing	Unit testing, Integration testing, Function testing, System testing, Acceptance testing, Installation testing.
Regression testing	Selective retest technique, Test prioritization.

Smoke Testing

Smoke Testing

Assume you order an item from **Daraz**.

What will you do after getting the parcel from the Daraz delivery man?

1. check that the parcel is addressed to you
2. and then make sure parcel is intact and not torn.
3. Next, you open the parcel and see the item is what you ordered and also make sure it is new, not old.

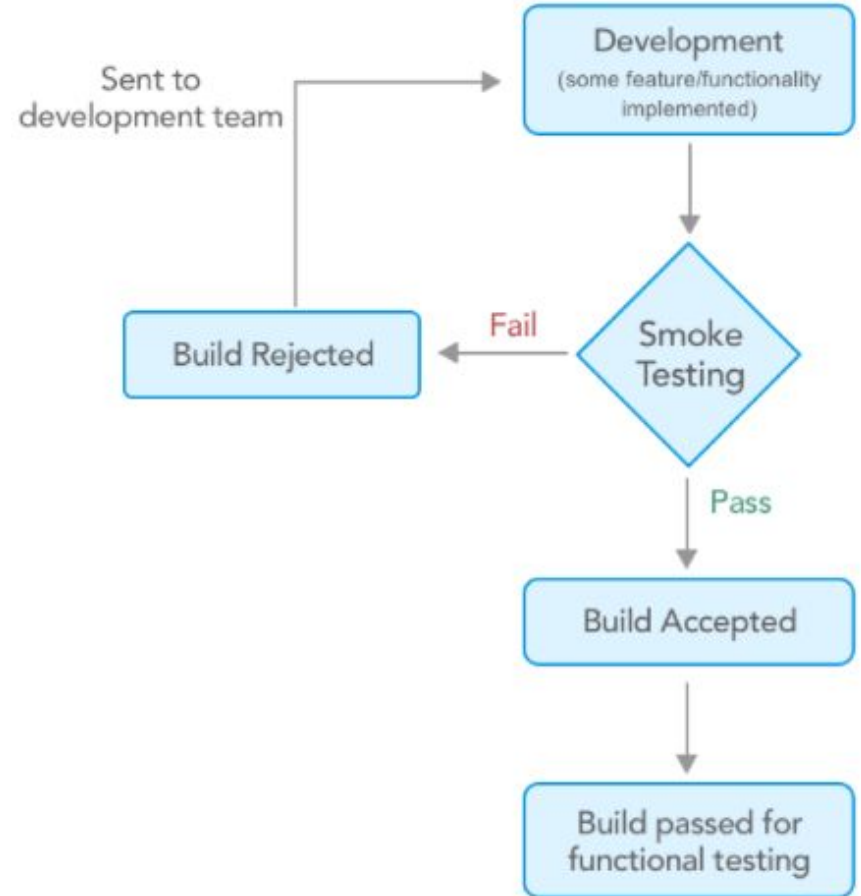
Well, you just completed your smoke testing on the parcel.

In case of SW testing, Smoke testing is performed on the 'new' build given by developers to QA team to verify if the basic functionalities are working or not.

It is the first test to be done on any new build.

Smoke Testing

- ❖ In smoke testing, the test cases chosen cover the most important functionality or component of the system.
- ❖ The objective is not to perform exhaustive testing, but to verify that the critical functionality of the system is working fine.
- ❖ Smoke testing is a confirmation for QA team to proceed with further software testing.



Smoke Testing

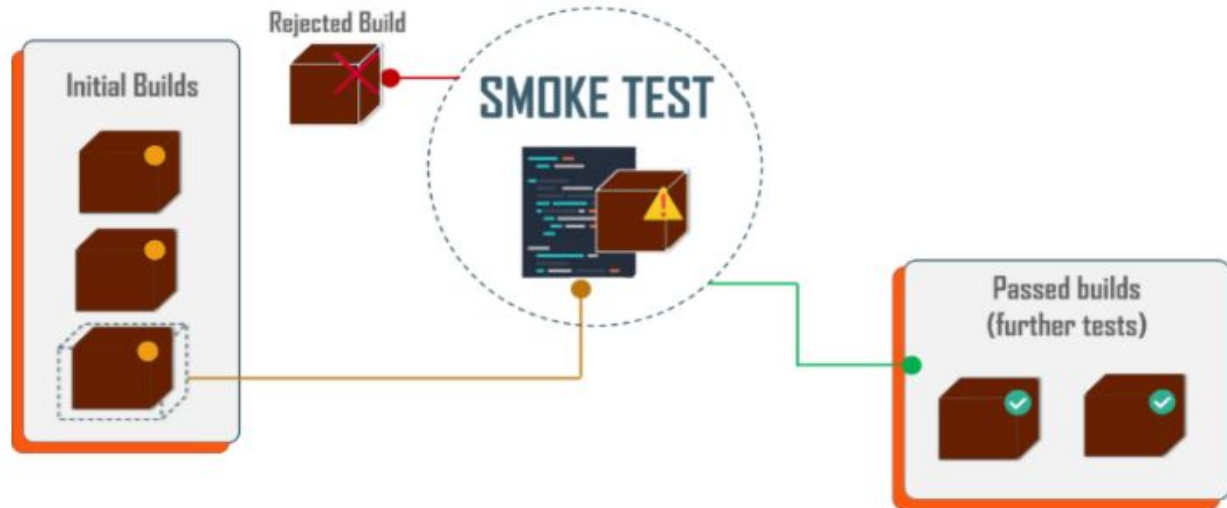
Why do we need Smoke Testing

- ❖ Just imagine a situation where you have a testing team consisting of 10 members.
- ❖ all the 10 testers start to test the application and raise the defects for failures they find.
- ❖ Now, at the end of the day, the development team may come back say, sorry this is not the right build or the QA team may stop the testing saying there are too many issues.
- ❖ But again 10 people have already wasted their 8 hours for this which means 80 hours of productivity is lost.
- ❖ This is the reason why we need to have a smoke test done, before jumping into a full-fledged testing cycle.

Smoke

When and How Often do We Need Smoke Testing?

- ❖ Smoke Testing normally takes **a maximum of 60 minutes** and should be done for every new build.
- ❖ Once the product is stable, you can even think about automating the smoke tests.
- ❖ a smoke test is very critical, because it will prevent an unstable or broken build from being pushed into production.



Smoke Testing

What are Scenarios that need to be included in a Smoke Test?

- ❖ **Build Verification:** The first and foremost step in a smoke test is to verify the build, the build number, and environment availability. .
- ❖ **Account Creation:** If your application involves a user creation, then you should try to create a new user and check if the system is successfully allowing you to do that
- ❖ **Log in Logout:** If applicable in your SUT (System Under Test), as part of the smoke test you should try to successfully login with old and newly created credentials. also test logout
- ❖ **Business Critical Features:** This is very important. For all the major or business-critical features, we should a simple test to ensure that the most commonly used functionalities are not broken.

Smoke Testing

- ❖ **Integration Scenarios:** This is the most important part of a smoke test. The effectiveness of this part depends upon the understanding of the system integrations by the tester. For example, if the tester knows that there is some data that flows from system A to system B, then he must make it a point to check that as part of the smoke test (any 1 value).
- ❖ **Add/Edit/Delete:** Data is always saved in a database. The three basic operations in a database are added the record, edit record and delete a record.

Smoke Testing

Who will Perform the Smoke Test?

- ❖ **Usually QA lead** is the one who performs smoke testing. Once the major build of software has been done, it will be tested to find out it's working well or not.
- ❖ The entire QA team sits together and discusses the main features of the software and the smoke test will be done to find out its condition..

Smoke Testing

Advantages:

- ❖ It helps to find faults earlier in the product lifecycle.
- ❖ It saves the testers time by avoiding testing an unstable or wrong build
- ❖ It provides confidence to the tester to proceed with testing
- ❖ It helps to find integration issues faster
- ❖ Major severity defects can be found out
- ❖ Detection and rectification will be an easy process
- ❖ Since execution happens quickly, there will be a faster feedback

Sanity Testing

Sanity Testing

- ❖ When a new build is received with minor modifications, instead of running a thorough regression test suite we perform a sanity test.
- ❖ It determines that the modifications have actually fixed the issues and no further issues have been introduced by the fixes.
- ❖ Sanity testing is generally a subset of regression testing and a group of test cases executed that are related to the changes made to the product.

Many testers get confused between sanity testing and smoke testing

Initial Builds when
the software is
relatively unstable

BUILD 1
BUILD 2
BUILD 3

Verifies critical functionalities like
Application starts successfully ?

Smoke Testing

•
•
•

Relatively Stable Builds
after multiple rounds
of regression tests.

BUILD 30
BUILD 31
BUILD 32

Sanity Testing

Verifies new functionality,
bug fixes in the build

•
•

Test
Pass ?

YES

System and /or
Regression
Testing

Regression Testing

Regression Testing

It is a type of software testing to confirm that a recent program or code change has not adversely affected existing features.

It is nothing but a **full or partial** selection of **already executed** test cases which are **re-executed** to ensure existing functionalities work fine.

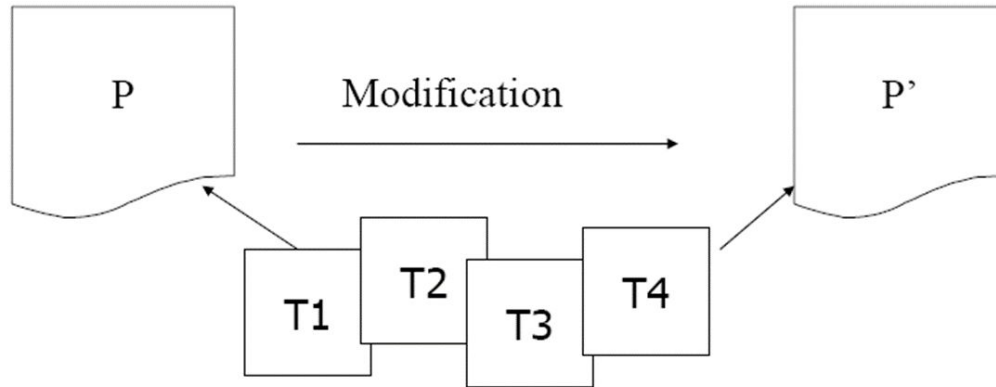
Regression test cases - **carefully selected** to cover as much of the system as possible that can **in any way have been affected** by a maintenance change or any kind of change.

Regression Testing

Regression testing is the execution of a set of test cases on a program in order to ensure that its **revision** does not produce **unintended faults**, does not "regress" - that is, become less effective than it has been in the past

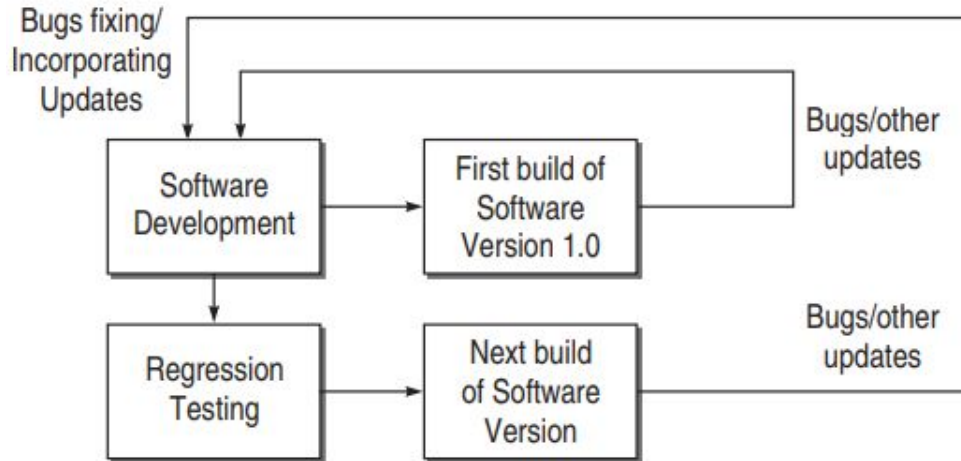
In a more formal way-

Given program **P**, its modified version **P'**, and a test set **T** that was used to previously test **P**, find a way to utilize **T** to gain sufficient confidence in the correctness of **P'**.



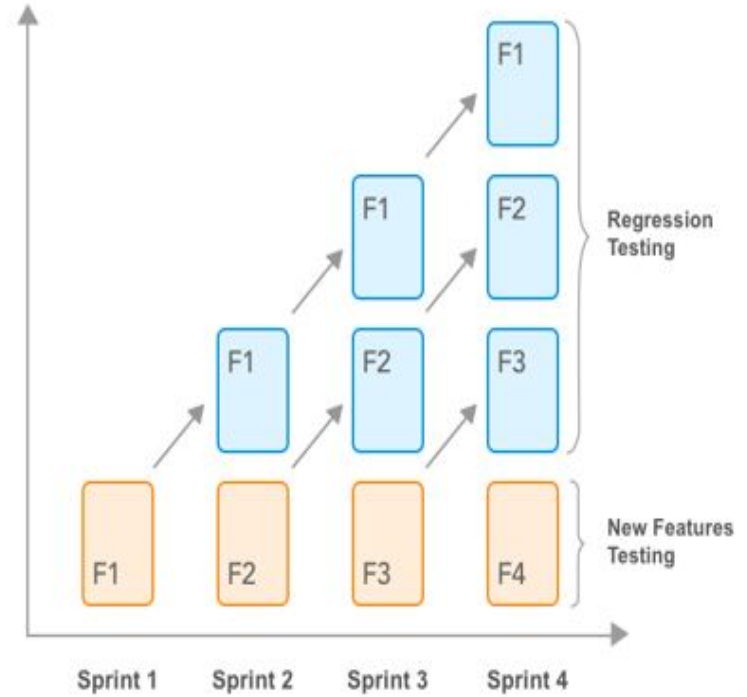
Regression Testing

- ❖ Regression testing is crucial during maintenance.
- ❖ It is a good idea to automate regression testing so that all test cases are run after each modification to the software
- ❖ When you find a bug in your program you should write a test case that exhibits the bug
- ❖ Then using regression testing you can make sure that the old bugs do not reappear



Regression Testing

- ❖ Its purpose is to find bugs that may have been accidentally introduced into the existing build and to ensure that previously removed bugs continue to stay dead.
- ❖ regression testing is the process of re-testing the modified parts of the software and ensuring that no new errors have been introduced into previously tested source code due to these modifications.



Regression Testing

When and How Often do We Need Regression Testing?

- ❖ We typically think of regression testing as a software maintenance activity
- ❖ however, we also perform regression testing during the later stage of software development.
- ❖ During this stage of development, we fine-tune the source code and correct errors in it, hence these activities resemble maintenance activities.

Few cases when we need to perform regression test:

- Addition of new functionalities
- Change requirements
- After bug fix
- Performance issue
- Environment change

Regression Testing

There are **three different techniques** for regression testing.

- ❖ **Regression test selection technique**

- to reduce the time required to retest a modified program by selecting some subset of the existing test suite..

- ❖ **Test case prioritization technique**

- to reorder a regression to **execute high priority test cases first.**

- **General Test Case Prioritization** For a given program **P** and test suite **T**, we prioritize the test cases in **T** that will be useful over a succession of subsequent modified versions of **P**, without any knowledge of the modified version.
- **Version-Specific Test Case Prioritization** We prioritize the test cases in **T**, when **P** is modified to **P'**, with the knowledge of the changes made in **P**.

- ❖ **Test suite reduction technique**

- It reduces testing costs by **permanently eliminating redundant test cases** from test suites in terms of codes or functionalities exercised.

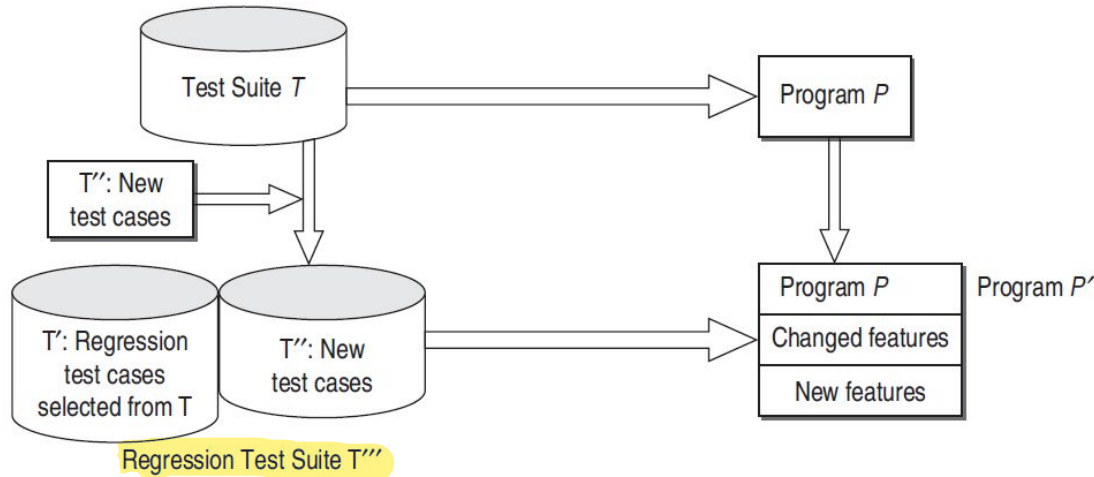
Regression Testing

Selective Retesting Technique

- ❖ Selective retest techniques attempt to reduce the cost of testing by identifying the portions of P' that must be exercised by the regression test suite.
- ❖ objective of selective retest technique is **cost reduction**.
- ❖ It is the process of selecting **a subset of the regression test suite** that tests the changes.
- ❖ Important things to know about selective retest technique:
 - It minimizes the resources required to regression test a new version.
 - It is achieved by minimizing the number of test cases applied to the new version.
 - It is needed because a regression test suite grows with each version, resulting in broken, obsolete, uncontrollable, redundant test cases.
 - It uses the information about changes to select test cases.

Regression Testing

Selective Retesting Technique



- ❖ 1. Select T' subset of T , a set of test cases to execute on P' .
- ❖ 2. Test P' with T' , establishing correctness of P' with respect to T' .
- ❖ 3. If necessary, create T'' , a set of new functional or structural test cases for P' .
- ❖ 4. Test P' with T'' , establishing correctness of P' with respect to T'' .
- ❖ 5. Create T''' , a new test suite and test execution profile for P' , from T , T' , and T'' .

Regression Testing

Regression Test Selection Techniques:

- ❖ **Minimization techniques:** attempt to select minimal sets of test cases from T that yield coverage of modified or affected portions of P.
- ❖ **Dataflow techniques:** select test cases that exercise data interactions that have been affected by modifications.
- ❖ **Safe techniques:** safe regression test selection techniques guarantee that the selected subset T' contains all test cases in the original test suite T that can reveal faults in P'.
- ❖ **Ad hoc/Random techniques:** When time constraints prohibit the use of a retest all approach, but no test selection tool is available, developers often select test cases based on 'intuitions' or loose associations of test cases with functionality. Another simple approach is to randomly select a predetermined number of test cases from T.
- ❖ **Retest-all technique:** The retest-all technique simply reuses all existing test cases. To test P', the technique effectively selects all test cases in T.

Regression Testing

Regression Test Prioritization:

1. Select T' from T , a set of test to execute on P' .
2. Produce T'_p , a permutation of T' , such that T'_p will have a better rate of fault detection than T' .
3. Test P' with T'_p in order to establish the correctness of P' with respect to T'_p .
4. If necessary, create T'' , a set of new functional or structural tests for P' .
5. Test P' with T'' in order to establish the correctness of P' with respect to T'' .
6. Create T''' , a new test suite for P' , from T' , T'_p , and T'' .

Regression Testing

Is Regression testing a problem?

- ❖ Large systems can take a long time to retest.
- ❖ It can be difficult and time-consuming to create the tests.
- ❖ It can be difficult and time-consuming to evaluate the tests. Sometimes, it requires a person in the loop to create and evaluate the results.
- ❖ Cost of testing can reduce resources available for software improvements.