SM-515
Software Verification & Validation

Fall 2002 Semester

Lecture 9: Software Testing
09/12/2002

Dr. Kivanç DINÇER

TÜBITAK-UEKAE/ILTAREN
http://www.uao.tubitak.gov.tr/

Testing

- Testing is the process of executing programs with the intention of finding errors.
- Testing can show the presence of bugs but never their absence.

(Common misconceptions about testing, difficulty of testing)

Difficulty of Testing

- Let's illustrate how difficult it is to develop a set of through test cases for even a trivial program.
  - "A program accepts as input three integer values. The three values represent the three sides of a triangle. Based on the three values, the program is to determine whether the triangle is isosceles, scalene, or equilateral."
  - Try writing down the set of test cases that you think would adequately test this simple program.
    - See Appendix I.

Feasibility for Testing

- Consider another trivial program that analyzes strings of alphabetic characters, 10 at a time.
  - There are 26^{10} possible combinations of inputs that this program could expect to see.
  - Would it be feasible to test all possible combinations of inputs?
- To test all possible inputs, a minimum of 26^{10} (~141 trillion) tests would be required.
  - If it takes one microsecond to execute each test, it would take about 4.5 years to execute all of the 141 trillion tests once.
- For most software products where the size of the input space is many orders of magnitude larger than this trivial example, the time required to develop and execute such large numbers of tests cannot be economically justified, even if it were feasible.

Selective Testing

- If we cannot test all possible combinations of inputs, our objective then becomes to select a relatively small number of tests that have a high probability of finding defects.
  - How do we write tests that can do this?

Good Testing Principles

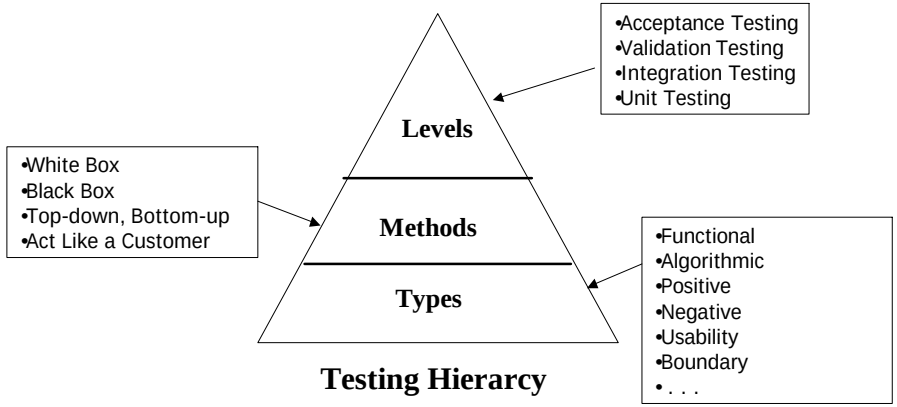
- A good test case is one that has a high probability of detecting an undiscovered defect, not one that shows that the program works correctly.
- It is impossible to test your own program.
- A necessary part of every test case is a description of the expected result.
- Avoid nonreproducible or on-the-fly testing.
- Write test cases for valid as well as invalid input conditions.
- Thoroughly inspect the results of each test.
- As the number of detected defects in a piece of software increases, the probability of the existence of more undetected defects also increases.
- Assign your best people to testing.
- Ensure that testability is a key objective in your software design.
- Never alter the program to make testing easier.
- Testing must start with objectives.

Good Testing Principles

- A good test case is one that has a high probability of detecting an undiscovered defect, not one that shows that the program works correctly.
- It is impossible to test your own program.
- A necessary part of every test case is a description of the expected result.
- Avoid nonreproducible or on-the-fly testing.
- Write test cases for valid as well as invalid input conditions.
- Thoroughly inspect the results of each test.
- As the number of detected defects in a piece of software increases, the probability of the existence of more undetected defects also increases.
- Assign your best people to testing.
- Ensure that testability is a key objective in your software design.
- Never alter the program to make testing easier.
- Testing must start with objectives.

Levels, Methods and Types of Tests

- Testing can be viewed as a hierarchy composed of different levels, methods, and types.



Test Levels

- Each level of testing has specific objectives and limitations.
  - Unit or Module Testing
  - Integration Testing
  - Validation or System Testing
  - Regression testing
  - Acceptance Testing – the customer is actively involved.

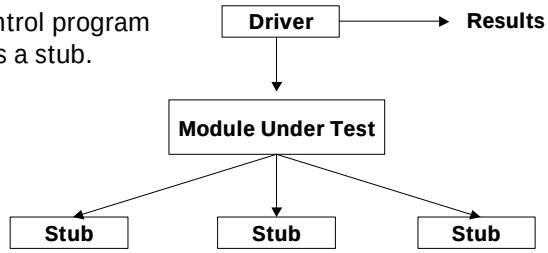
Validation: The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements.

Alpha and Beta Testing: Customer evaluates the prerelease software. (However customers are frequently reluctant to participate in such activities.)

Unit Testing

- The objective is to find bugs in individual modules by testing them in an isolated environment.
- It is usually considered part of the coding process and typically requires a significant investment in "scaffolding."

Driver: A control program developed as a stub.



Unit Testing Environment

Unit Testing

- Unit testing is often viewed more as a debugging activity than as a testing activity.
- In many organizations, unit testing is performed by software engineers on their own modules with little or no test documentation.
  - To increase ROI, use a buddy system.
  - Why?

Debugging vs. Unit Testing

•Debugging is defined as the process of detecting, locating and correcting faults in a computer program.

•Testing is defined as the process of operating a system or component under specified conditions, observing or recording the results, and making the evaluation of some aspect of the system or component.

Unit Test Cases

- The following questions can be used as a checklist when developing unit test cases:
  - Algorithms and logic
  - Data structures (global and local)
  - Interfaces
  - Independent paths
  - Boundary conditions
  - Error handling
- IEEE Std.1008-1987 provides additional info on unit testing.

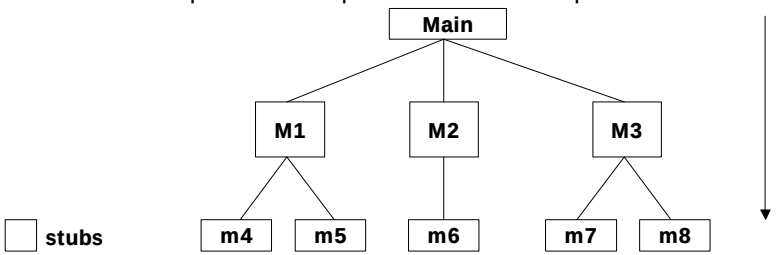
Integration Testing

- The objective of integration testing is to find bugs related to interfaces between modules as they are integrated together.
- Question: If all modules are unit tested, why is integration testing necessary?

- Answer:
  - One module can have an adverse effect on another.
  - Subfunctions, when combined, may not produce the major function.
  - Individually acceptable imprecision in calculations may be magnified to unacceptable levels.
  - Interfacing errors not detected in unit testing may appear.
  - Timing problems (especially in real-time systems) are not detectable by unit testing.
  - Resource contention problems are not detectable by unit testing.

Incremental Integration

- Integration testing covers a broad range of activities, beginning with the testing of few modules and culminating with the testing of the complete system.
- Incremental Integration:
  - the product is constructed and tested in small chunks so that errors are easy to observe, isolate, and correct.
  - can be performed top-down or bottom-up.



Top-Down Integration Testing Environment

Top-Down Integration

1. The main module is used as a driver, and stubs are substituted for all modules directly subordinate to the main module.
2. Depending on the integration approach selected (depth or breadth first,) subordinate stubs are replaced by modules one at a time.
3. Tests are run as each individual module is integrated.
4. On the successful completion of a set of tests another stub is replaced with a real module.
5. Regression testing is performed to ensure that errors have not developed as a result of integrating new modules.
6. If not done, Go to step 2

Problems with Top-Down Integration

- Many times, calculations are performed in the modules at the bottom of the hierarchy.
- Stubs typically do not pass data up to the higher modules.
- Delaying testing until lower-level modules are ready usually results in integrating many modules at the same time rather than one at a time.
- Developing stubs that can pass data up is almost as much work as developing the actual module.

Bottom-Up Integration

- Integration begins with the lowest-level modules, which are combined into clusters, or builds, that perform a specific software subfunction.
- Drivers are written to coordinate test case input and output.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program structure.

Problems with Bottom-Up Integration

- The whole program does not exist until the last module is integrated.
- Timing and resource contention problems are not found until late in the process.

General Problems

- In many organizations, software developers are responsible for performing some form of integration testing.
  - Developers frequently use the Big Bang approach: integrate all modules at once and start testing!
- The distinction between unit testing and integration testing is often fuzzy.
  - Often, unit tests are repeated as modules are integrated together.

Validation Testing

- The objective of validation testing is to determine if the software meets all its requirements as defined in the SRS.
- Frequently, organizations perform validation testing without the benefit of written requirements.
  - Can validation testing be effective without written requirements?
- As part of validation testing, regression testing is performed to determine if the software still meets all of its requirements in light of changes and modifications made to the software.
  - Regression testing involves selectively repeating existing validation tests, not developing new ones.

- Answer: No!
- You need one of the following:
  - Write them down before the tests are developed.
  - People who write tests will need to have domain knowledge, that is, knowledge of the product and how customers use the product in their environment.
- Otherwise, testing will likely be superficial and of little value to you or your customers.

Alpha and Beta Testing

- The objectives of alpha and beta testing are often vague.
  - If you are going to invest time and resources in this activity, there should be clear objectives to maximize ROI.
- Beware of the following,
  - Provide the customers with an outline of the things that you would like to focus on and specific test scenarios for them to execute. (effectiveness)
  - Provide the customers with a commitment to fix defects that they find. (motivation)

Acceptance Testing

- Similar to validation testing, except that customers are present or directly involved.
  - Acceptance testing can be a subset of the same tests used for validation testing or can employ tests developed entirely by customers.
  - In the latter case, as your customer for those tests in advance and run as many of them as possible as part of the validation testing.