Chapter 8

Testing

Principles of Object-Oriented Testing
- Object-oriented systems are built out of two or more interrelated objects
- Determining the correctness of O-O systems requires testing the methods that change or communicate the state of an object
- Testing methods in an object-oriented system is similar to testing subprograms in process-oriented systems

Definitions: Error and Fault
- Error - refers to any discrepancy between an actual, measured value and a theoretical, predicted value. Error also refers to some human action that results in some sort of failure or fault in the software
- Fault - is a condition that causes the software to malfunction or fail

Definitions: Failure
- Failure - is the inability of a piece of software to perform according to its specifications. Failures are caused by faults, but not all faults cause failures. A piece of software has failed if its actual behavior differs in any way from its expected behavior

Test Plan
- A test plan specifies how we will demonstrate that the software is free of faults and behaves according to the requirements specification
- A test plan breaks the testing process into specific tests, addressing specific data items and values
- Each test has a test specification that documents the purpose of the test

Test Plan
- If a test is to be accomplished by a series of smaller tests, the test specification describes the relationship between the smaller and the larger tests
- The test specification must describe the conditions that indicate when the test is complete and a means for evaluating the results
Test Oracle

- A test oracle is the set of predicted results for a set of tests, and is used to determine the success of testing.
- Test oracles are extremely difficult to create and are ideally created from the requirements specification.

Test Cases

- A test case is a set of inputs to the system.
- Successfully testing a system hinges on selecting representative test cases.
- Poorly chosen test cases may fail to illuminate the faults in a system.
- In most systems, exhaustive testing is impossible, so a white box or black box testing strategy is typically selected.

White Box Testing

- The tester uses knowledge of the programming constructs to determine the test cases to use.
- If one or more loops exist in a method, the tester would wish to test the execution of this loop for 0, 1, max, and max + 1, where max represents a possible maximum number of iterations.
- Similarly, conditions would be tested for true and false.

Black Box Testing

- The tester knows nothing about the internal structure of the code.
- Test cases are formulated based on expected output of methods.
- Tester generates test cases to represent all possible situations in order to ensure that the observed and expected behavior is the same.

Unit Testing

- The units comprising a system are individually tested.
- The code is examined for faults in algorithms, data and syntax.
- A set of test cases is formulated and input and the results are evaluated.
- The module being tested should be reviewed in context of the requirements specification.

Integration Testing

- The goal is to ensure that groups of components work together as specified in the requirements document.
- Four kinds of integration tests exist:
  - Structure tests
  - Functional tests
  - Stress tests
  - Performance tests.
System Testing
- The goal is to ensure that the system actually does what the customer expects it to do
- Testing is carried out by customers mimicking real world activities
- Customers should also intentionally enter erroneous values to determine the system behavior in those instances

Testing Steps
- Determine what the test is supposed to measure
- Decide how to carry out the tests
- Develop the test cases
- Determine the expected results of each test (test oracle)
- Execute the tests
- Compare results to the test oracle

Analysis of Test Results
- The test analysis report documents testing and provides information that allows a failure to be duplicated, found, and fixed
- The test analysis report mentions the sections of the requirements specification, the implementation plan, the test plan, and connects these to each test

Special Issues for Testing Object-Oriented Systems
- Because object interaction is essential to O-O systems, integration testing must be more extensive
- Inheritance makes testing more difficult by requiring more contexts (all sub classes) for testing an inherited module

Configuration Management
- Software systems often have multiple versions or releases
- Configuration management is the process of controlling development that produces multiple software systems
- An evolutionary development approach often results in multiple versions of the system
- Regression testing is the process of retesting elements of the system that were tested in a previous version or release

Suggested Group Activity
- Divide into your software engineering teams
- Create a test plan: TestPlan.doc
Suggested Individual/Class Activity

- Discuss the following system failures: TestingEX.doc