Chapter 6- Coding and Testing

Shilpa Das
Assistant Professor
School of Computer Engineering, KIIT DU

Reference Books:
1) Fundamentals of Software Engineering, Rajib Mall , PHI, Latest edition
2) Software Engineering, I. Sommerville, Pearson Education, Asia.

Topics covered

- Coding basics
- Testing Overview

Coding

- Transform the design of a system into code in a high-level language, and perform unit testing.
- The integration and system testing phase is undertaken after completion of testing all individual modules.
- Coding standard and coding guidelines

Code review

- Cost-effective strategy for eliminating coding errors
- Performed after removing all syntax errors
- Locating the error is done directly
- Types of code review techniques
    - Code Inspection
    - Code Walkthrough

Code Walkthrough

- Finds the algorithmic and logical errors
- Informal analysis performed by three to seven members
- Each member selects some test cases and traces the execution through different statements and functions of the code manually
- A walkthrough meeting is held
- Discussions should focus on discovery of errors and avoid deliberations on how to fix the discovered errors

Code Inspection

- Check for the presence of some common types of errors and whether the coding standards have been adhered to
- A checklist of common mistakes are kept to find the errors
- Some examples
  - Incompatible assignments
  - Non-terminating loops
  - Array indices out of bounds
  - Use of uninitialized variables

Documentation

- Internal Documentation
  - the code comprehension features provided in the source code itself
  - Examples – Comments embedded in the source code, **use of meaningful variable names**, module and function headers, code indentation etc.
- External Documentation
  - provided through various types of supporting documents such as users’ manual, software requirements specification document, design document, test document
  - Should be up-to-date with the software changes, understandable for the reader

Testing

Testing

- Input data to the program and observe the output
- If the program behaves not as expected
  - The test conditions are noted in **Test report**
  - Debug and correct the error once the test report is finished
- Takes 50% development effort but 10% development time
- “Monkey testing ”

Terminologies

- A **mistake** is essentially any programmer action that later shows up as an incorrect result during program execution
- An **error/fault/defect/bug** is the consequence/result of a mistake committed by a developer in any of the development activities.
- A **failure** of a program essentially denotes an incorrect behavior exhibited by the program during its execution.
- Every failure is caused by some bugs present in the program but every program error may or may not cause any failure.

Terminologies

- **Testability** of a requirement denotes the extent to which it is possible to determine whether an implementation of the requirement conforms to it in both functionality and performance.
- A **test case** is a triplet [I , S, R], where I is the data input to the program under test, S is the state of the program at which the data is to be input, and R is the result expected to be produced by the program.
- A **test scenario** is an abstract test case in the sense that it only identifies the aspects of the program that are to be tested without identifying the input, state, or output.

Terminologies

- A **test script** is an encoding of a test case as a short program.
- A **test suite** is the set of all test that have been designed by a tester to test a given program.
- A **positive test case** is designed to test whether the software correctly performs a required functionality. A **negative test case** is designed to test whether the software carries out something, that is not required of the system.

Verification vs Validation

1. Verification is the process of determining whether the output of one phase of software development conforms to that of its previous phase;

Validation is the process of determining whether a fully developed software conforms to its requirements specification.

Verification vs Validation

2. The primary techniques used for verification include review, simulation, formal verification, and (unit) testing;

Validation techniques are primarily based on product (system) testing.

3. Verification is carried out during the development process to check if the development activities are proceeding alright;

Validation is carried out to check if the customer requirement has been correctly developed.

Verification vs Validation

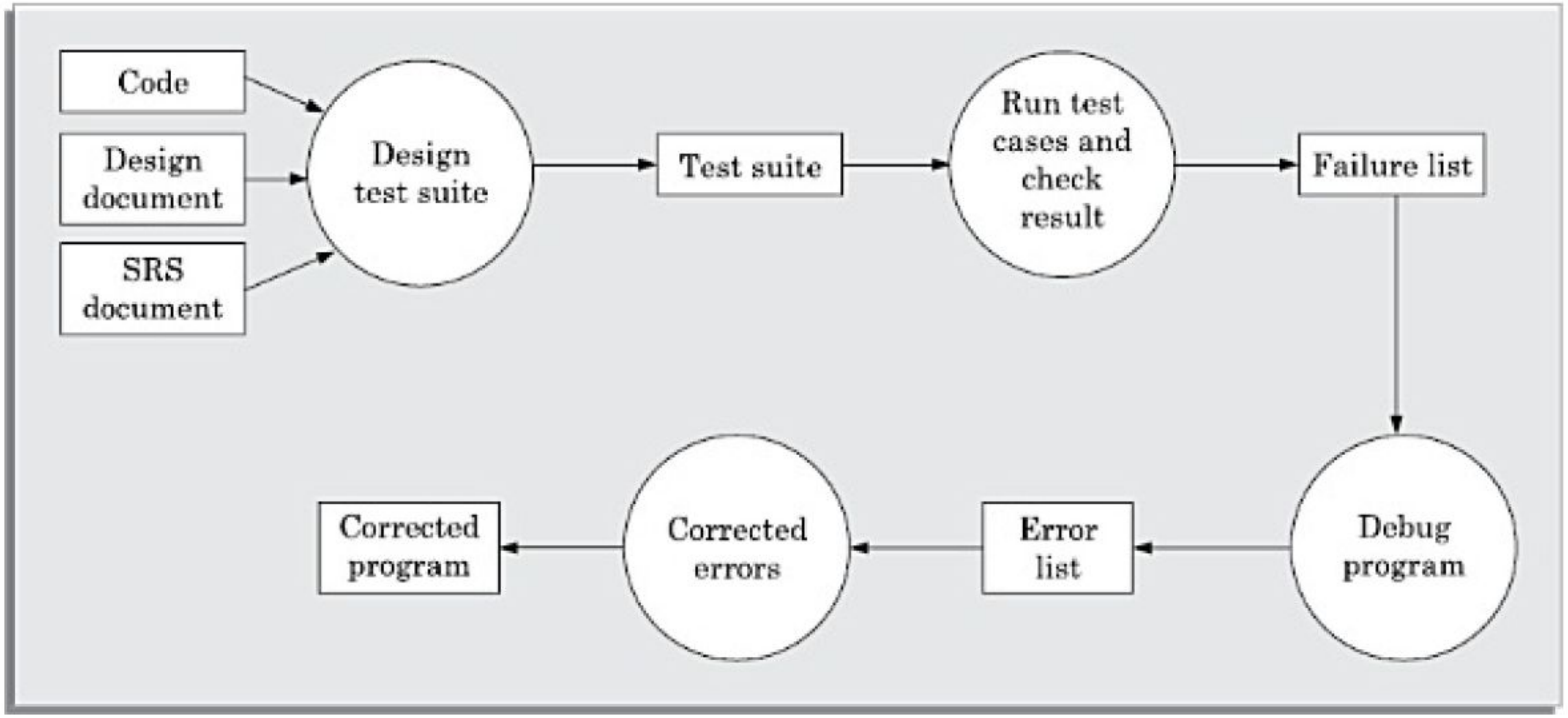
4. Verification does not require execution of the software;

Validation requires execution of the software.

5. Verification is concerned with phase containment of errors;

The aim of validation is to check whether the deliverable software is error free.

Testing Activities



Testing Activities

- **Test suite design:** The set of test cases using which a program is to be tested is designed possibly using several test case design techniques.
- **Running test cases and checking the results to detect failures:** Each test case is run and the results are compared with the expected results. A mismatch between the actual result and expected results indicates a failure. The test cases for which the system fails are noted down for later debugging

Testing Activities

- **Debugging to locate error**: In this activity, the failure symptoms are analysed to locate the errors. For each failure observed during the previous activity, the statements that are in error are identified.
- **Error correction:** After the error is located during debugging, the code is appropriately changed to correct the error.

Next Session

- Software Testing