

Information Systems (තොරතුරු පද්ධති)

Part 6

(System Development and Testing)



Objectives – Learning Outcomes

- The objectives of this section are to get knowledge on system development and testing
- When you have followed this section you will:
 - Understand to describe program coding
 - Understand to identify test cases
 - Understand white box testing
 - Understand black box testing
 - Understand unit testing
 - Understand integrated testing
 - Understand system testing
 - Understand acceptance testing



Outline

- Program Coding / System Development
- Testing
- Test cases
 - White box testing /Black box testing
 - Unit testing/ Integrated testing
 - System testing/Acceptance testing



System Development (Coding)

- What is System Development /Program coding?
 - Once the design is completed, most of the major decisions about the system have been made.
 - The goal of the coding phase is to translate the design of the system into code in a given programming language.
 - For a given design, the aim of this phase is to implement the design in the best possible manner.
 - The coding phase affects both testing and maintenance profoundly.



System Development (Coding) Cont.

- What is System Development /Program coding?
 - A well written code reduces the testing and maintenance effort.
 - Since the testing and maintenance cost of software are much higher than the coding cost, the goal of coding should be to reduce the testing and maintenance effort.
 - Hence, during coding the focus should be on developing programs that are easy to write.
 - Simplicity and clarity should be strived for, during the coding phase.



System Development (Coding) Cont.

➤ Programming languages - examples

Language	Main application area	Compiled/interpreted
ADA	Real-time	Compiled language
BASIC	Programming for educational purposes	Interpreted language
C	System programming	Compiled language
C++	System object programming	Compiled language
Cobol	Management	Compiled language
Fortran	Calculation	Compiled language
Java	Internet oriented programming	Intermediary language
MATLAB	Mathematical calculations	Interpreted language
Mathematica	Mathematical calculations	Interpreted language
LISP	Artificial intelligence	Intermediary language
Pascal	Education	Compiled language
PHP	Dynamic website development	Interpreted language
Prolog	Artificial intelligence	Interpreted language
Perl	Processing character strings	Interpreted language

System Development (Coding) Cont.

➤ Code - example

```
import java.util.Scanner;

class basicalc{
    public static void main(String args[]){
        Scanner user_input = new Scanner(System.in);
        double fnum,snum,ans;
        int ch;

        System.out.println("1. Addition");
        System.out.println("2. Subtraction");
        System.out.println("3. Multiplication");
        System.out.println("4. Division");
        System.out.print("Choose you operator:");
        ch = user_input.nextInt();
        System.out.println(" ");

        switch (ch){
            case 1:
                System.out.println("You choose Addtion");
                System.out.print("Enter first num: ");
                fnum = user_input.nextDouble();

                System.out.print("Enter second num: ");
                snum = user_input.nextDouble();
                ans = fnum + snum;
                System.out.println("Answer is: " + ans);
                break;

            case 2:
                System.out.println("You choose Subtraction");
                System.out.print("Enter first num: ");
                fnum = user_input.nextDouble();

                System.out.print("Enter second num: ");
                snum = user_input.nextDouble();
```

System Testing

- What is System Testing?
 - Testing is the major quality control measure employed during software development.
 - Its basic function is to detect errors in the software. During requirement analysis and design, the output is a document that is usually textual and non-executable. .



System Testing Cont.

➤ What is System Testing?

- After the coding phase, computer programs are available that can be executed for testing phases. This implies that testing not only has to uncover errors introduced during coding, but also errors introduced during the previous phases. Thus, the goal of testing is to uncover requirement, design or coding errors in the programs.



System Testing Cont.

➤ What is Test Case?

- A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly.



System Testing Cont.

TEST CASE TEMPLATE >>

Test Suite ID	The ID of the test suite to which this test case belongs.
Test Case ID	The ID of the test case.
Test Case Summary	The summary / objective of the test case.
Related Requirement	The ID of the requirement this test case relates/traces to.
Prerequisites	Any prerequisites or preconditions that must be fulfilled prior to executing the test.
Test Procedure	Step-by-step procedure to execute the test.
Test Data	The test data, or links to the test data, that are to be used while conducting the test.
Expected Result	The expected result of the test.
Actual Result	The actual result of the test; to be filled after executing the test.
Status	Pass or Fail. Other statuses can be 'Not Executed' if testing is not performed and 'Blocked' if testing is blocked.
Remarks	Any comments on the test case or test execution.
Created By	The name of the author of the test case.
Date of Creation	The date of creation of the test case.
Executed By	The name of the person who executed the test.
Date of Execution	The date of execution of the test.
Test Environment	The environment (Hardware/Software/Network) in which the test was executed.



System Testing Cont.

TEST CASE Example >>

Test Suite ID	TS001
Test Case ID	TC001
Test Case Summary	To verify that clicking the Generate Coin button generates coins.
Related Requirement	RS001
Prerequisites	<ol style="list-style-type: none">1. User is authorized.2. Coin balance is available.
Test Procedure	<ol style="list-style-type: none">1. Select the coin denomination in the Denomination field.2. Enter the number of coins in the Quantity field.3. Click Generate Coin.
Test Data	<ol style="list-style-type: none">1. Denominations: 0.05, 0.10, 0.25, 0.50, 1, 2, 52. Quantities: 0, 1, 5, 10, 20
Expected Result	<ol style="list-style-type: none">1. Coin of the specified denomination should be produced if the specified Quantity is valid (1, 5)2. A message 'Please enter a valid quantity between 1 and 10' should be displayed if the specified quantity is invalid.
Actual Result	<ol style="list-style-type: none">1. If the specified quantity is valid, the result is as expected.2. If the specified quantity is invalid, nothing happens; the expected message is not displayed
Status	Fail
Remarks	This is a sample test case.
Created By	John Doe
Date of Creation	01/14/2020
Executed By	Jane Roe
Date of Execution	02/16/2020
Test Environment	<ul style="list-style-type: none">• OS: Windows Y• Browser: Chrome N



System Testing Cont.

➤ System/Software Testing Types

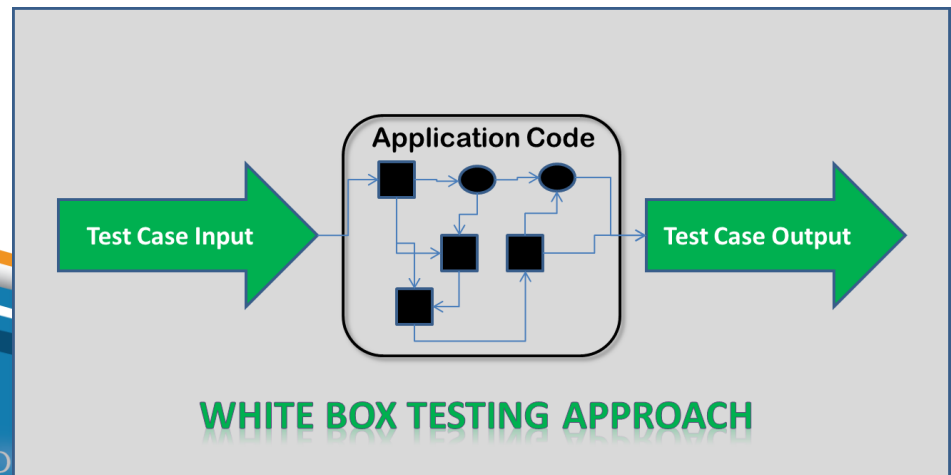
- White Box Testing
- Black Box Testing
- Unit Testing
- Integrated Testing
- System Testing
- Acceptance Testing



System Testing Cont.

➤ White Box Testing

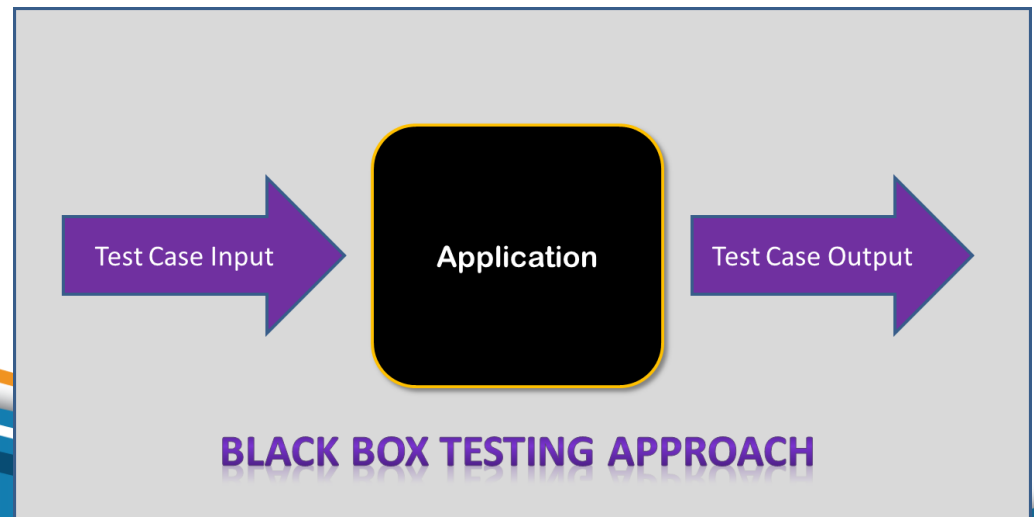
- This testing is based on knowledge of the internal logic of an application's code. Also known as Glass box Testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.



System Testing Cont.

➤ Black Box Testing

- Internal system design is not considered in this type of testing. Tests are based on requirements and functionality.



System Testing Cont.

➤ Unit Testing

- Testing of individual software components or modules.
- Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code.
- May require developing test driver modules or test harnesses.



System Testing Cont.

➤ Integrated Testing

Integration testing is defined as the testing of combined parts of an application to determine if they function correctly.

- Integration testing can be done in two ways:
 - Bottom-up integration testing
 - Top-down integration testing

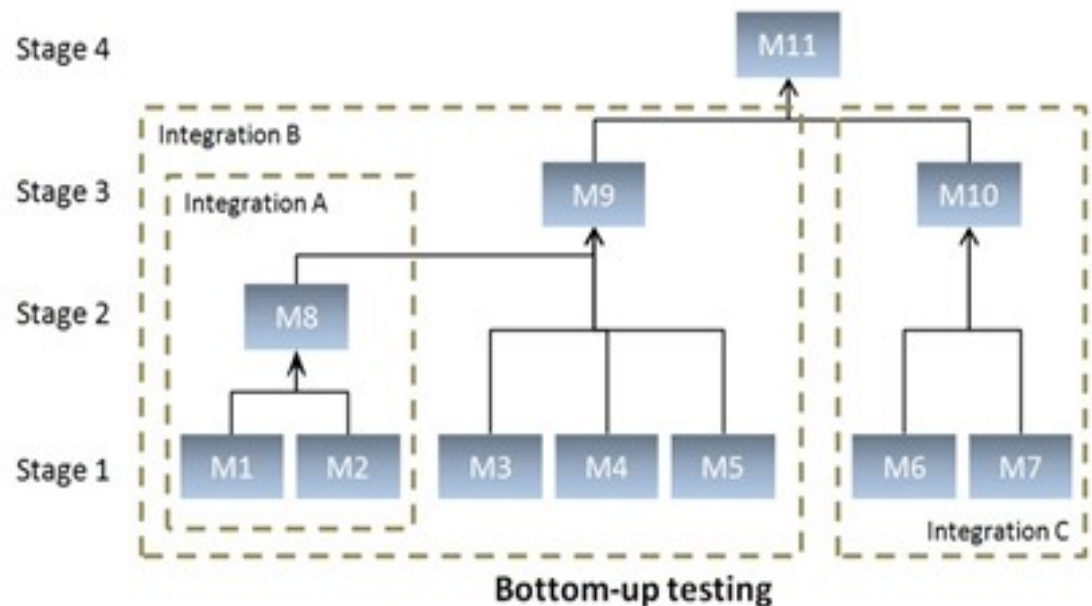


System Testing Cont.

➤ Integrated Testing

Bottom-up integration

This testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds.

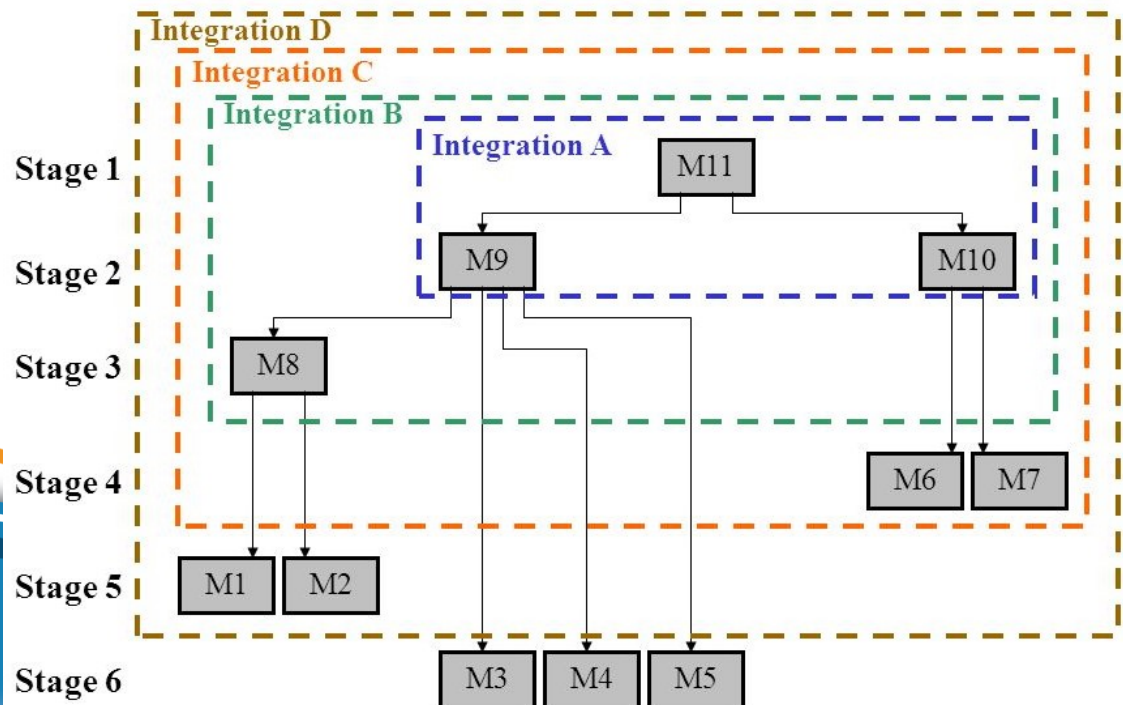


System Testing Cont.

➤ Integrated Testing

Top-down integration

In this testing, the highest-level modules are tested first and progressively, lower-level modules are tested thereafter.



System Testing Cont.

➤ System Testing

- System testing tests the system as a whole.
- Once all the components are integrated, the application as a whole is tested rigorously to see that it meets the specified Quality Standards.
- This type of testing is performed by a specialized testing team.



System Testing Cont.

➤ Acceptance Testing

- This is arguably the most important type of testing, as it is conducted by the Quality Assurance Team who will gauge whether the application meets the intended specifications and satisfies the client's requirement.
- The QA team will have a set of pre-written scenarios and test cases that will be used to test the application.



Summery

In this section you have given an idea about the entire process of System Development and System Testing.

Also this section covered about different testing types that you need to perform when system development stage.



References

- SOFTWARE ENGINEERING, Ninth Edition, Ian Sommerville
- <https://www.tutorialspoint.com>

Next >>>

**Lesson 07 : System Implementation and
Maintenance**

