CP317 Software Engineering

week 6-2 - Testing

Shaun Gao, Ph.D., P.Eng.

Agenda

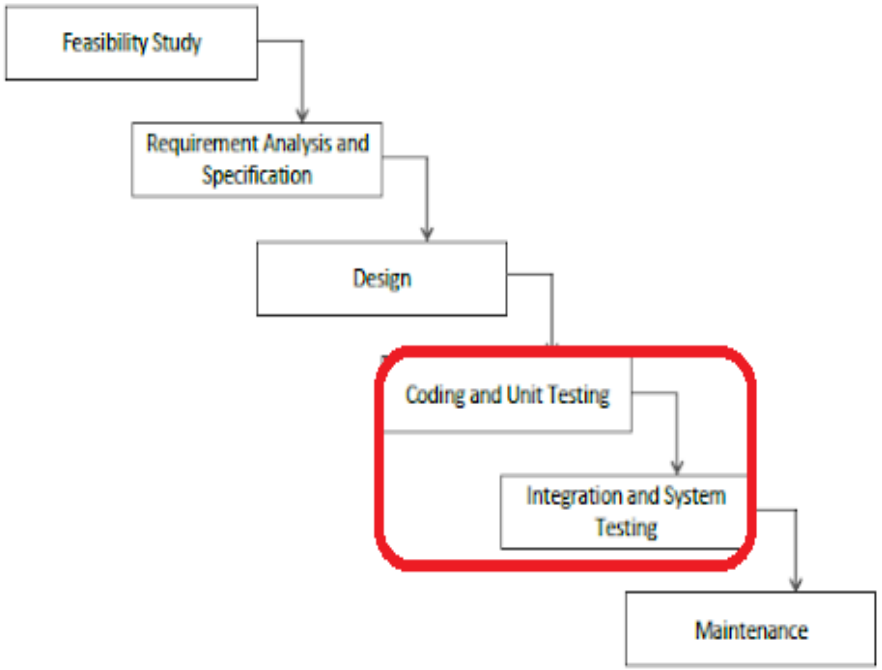
- Review week 6-1 implementation
- Software testing
  - Concept
- Software testing plan
- Levels of software testing
  - Unit testing
  - Integration testing: top down, bottom up
  - System testing
  - Acceptance testing
- Testing techniques
  - Black-box testing
  - White-box testing
- Test driven development (TDD)
- Summary

Review Implementation

- Introduction to implementation
  - Good programmer? Computer security
- Coding standard
  - Concept, benefits of using coding standard
- Source code control
- Understanding API
- Recommendations
  - Avoid side effect in programming
  - Offensive programming
  - Exception handling
- Tips for developers
  - Memory managements
  - Problems with functions
  - Safety consideration
  - Constructor/destructor
  - Keep destructor focus on releasing resources only

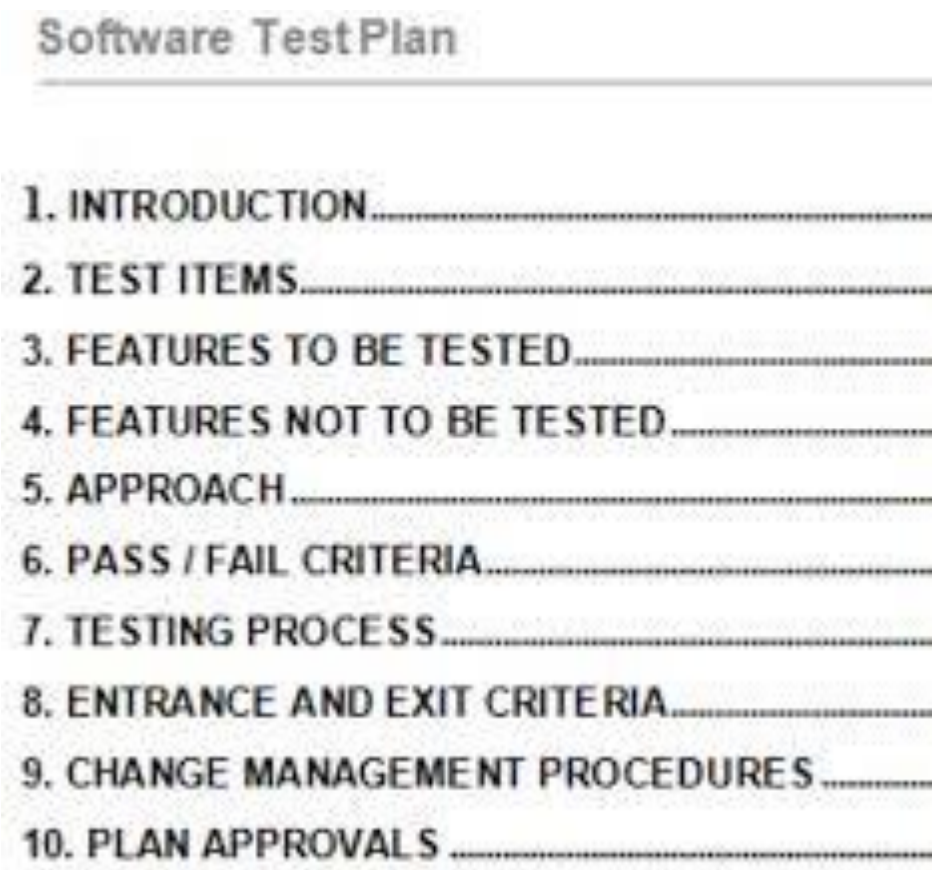
Introduction to testing

- Software testing
  - Software testing is to evaluate the software against requirements gathered from users and system specifications.
- Types of testing
  - Unit testing
  - Integration testing
  - Regression testing
  - System testing
- Test time duration: approximate = coding



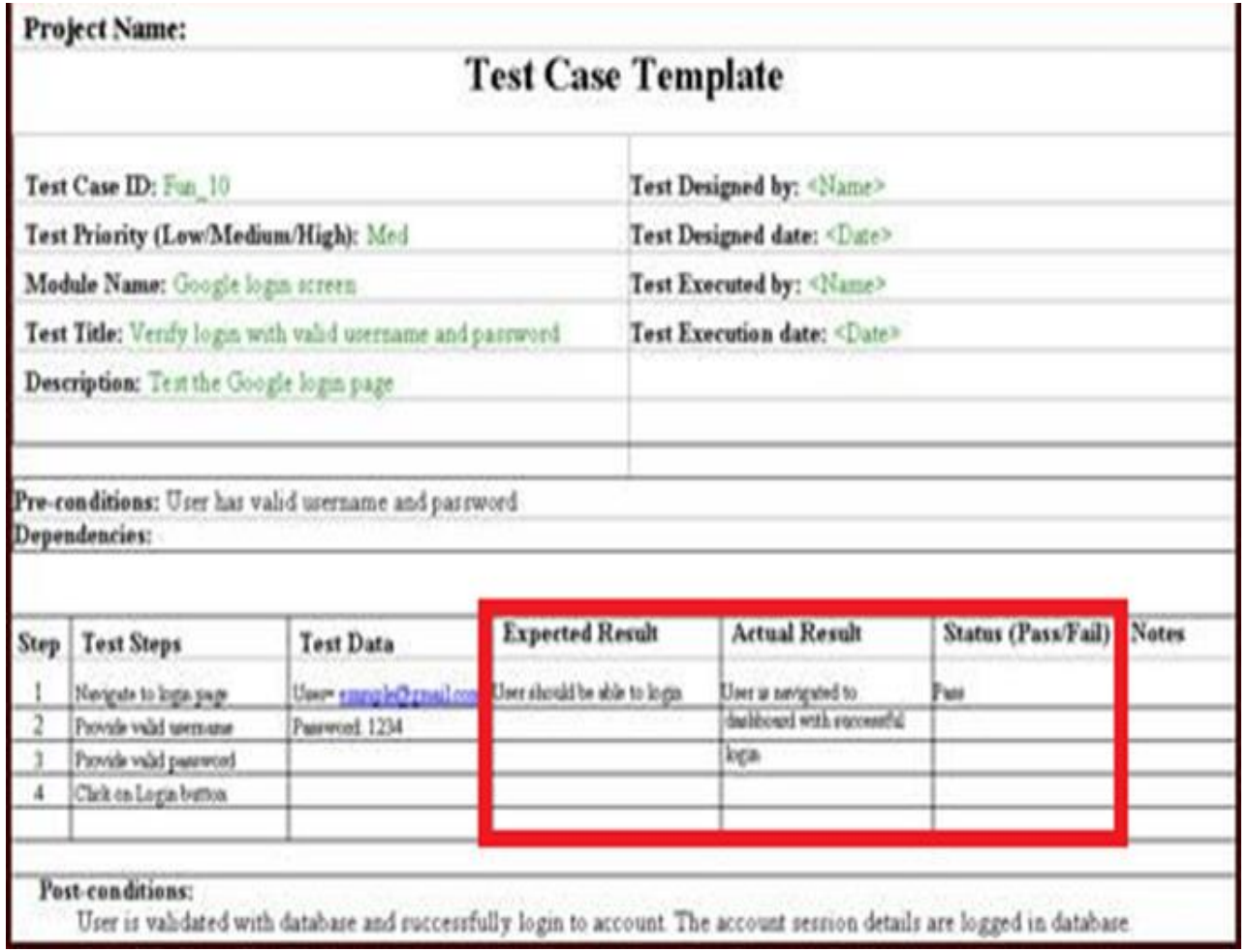
Software test plan

- A software test plan is a document describing software testing scope and activities.
- Test case procedures
  - Procedures
  - Inputs
  - Expected outputs/results
- Examples:
  - Unit testing plan
  - Integration testing plan
  - System testing plan



Software test plan – cont.

- Test case
  - A test case is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular software testing objective, such as to exercise a particular program path or to verify compliance with a specific requirement.



Project Name:

Test Case Template

Test Case ID: Fun_10	**Test Designed by:** <Name>
Test Priority (Low/Medium/High): Med	**Test Designed date:** <Date>
Module Name: Google login screen	**Test Executed by:** <Name>
Test Title: Verify login with valid username and password	**Test Execution date:** <Date>
Description: Test the Google login page	

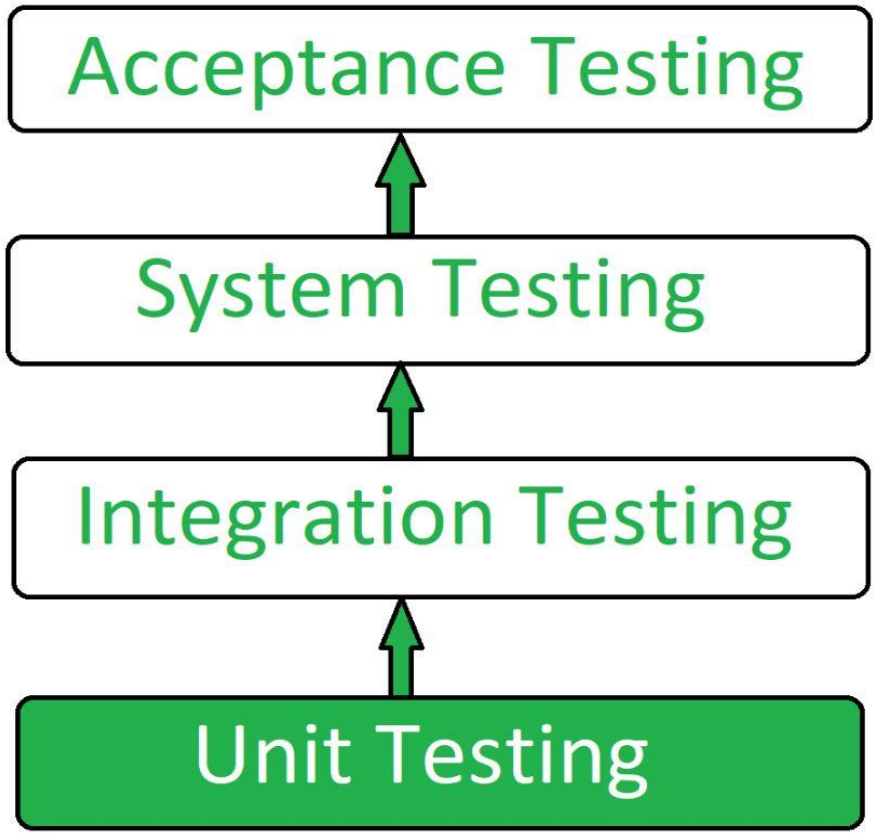
Pre-conditions: User has valid username and password
Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to login page	User: [redacted]	User should be able to login	User is navigated to	Pass	
2	Provide valid username	Password: 1234		dashboard with successful		
3	Provide valid password			login		
4	Click on Login button					

Post-conditions:
User is validated with database and successfully login to account. The account session details are logged in database

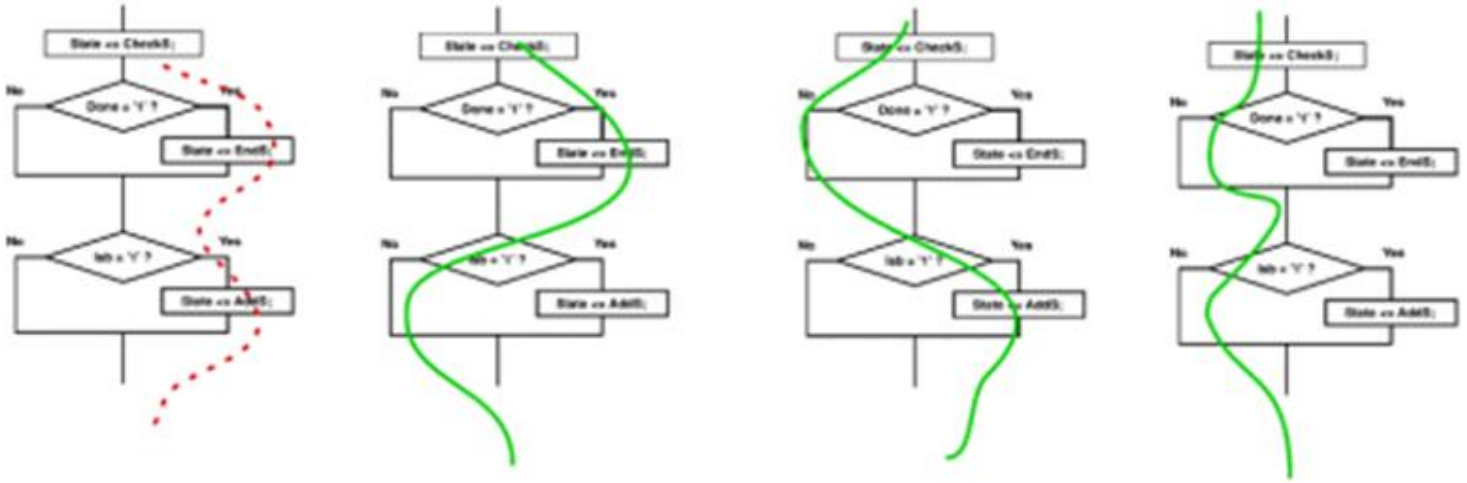
Levels of software testing

- Unit testing
- Integration testing
- Regression testing
- System testing
- Acceptance testing



Unit testing

- Unit test is a method by which an individual unit (method or function) of source code is tested to determine if it behaves correctly.
- Focus on (functions or modules) functional correctness and completeness of individual program units
- Usually unit testing is performed by software developers
- Test all paths



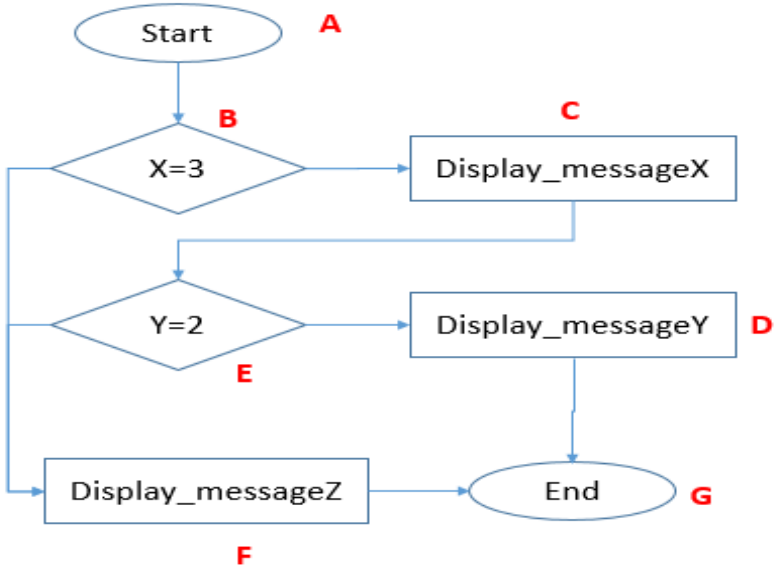
Unit testing – cont.

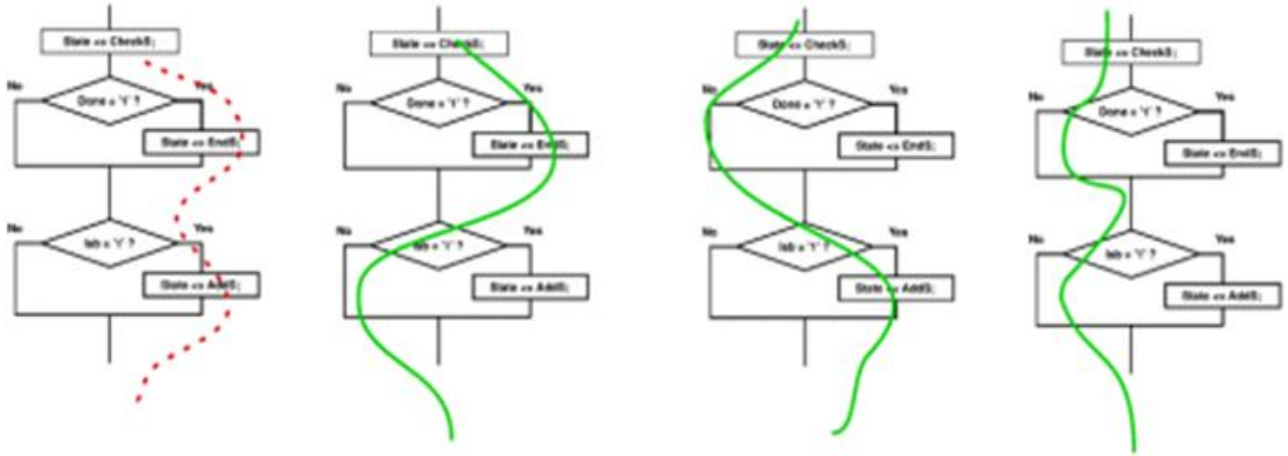
- Test Coverage
- Test coverage is a measure used to describe the degree to which the source code of a program is executed when a particular test suite runs.

  ABCEDG,

  ABFG,

  ABCEFG
- The higher coverage, the better

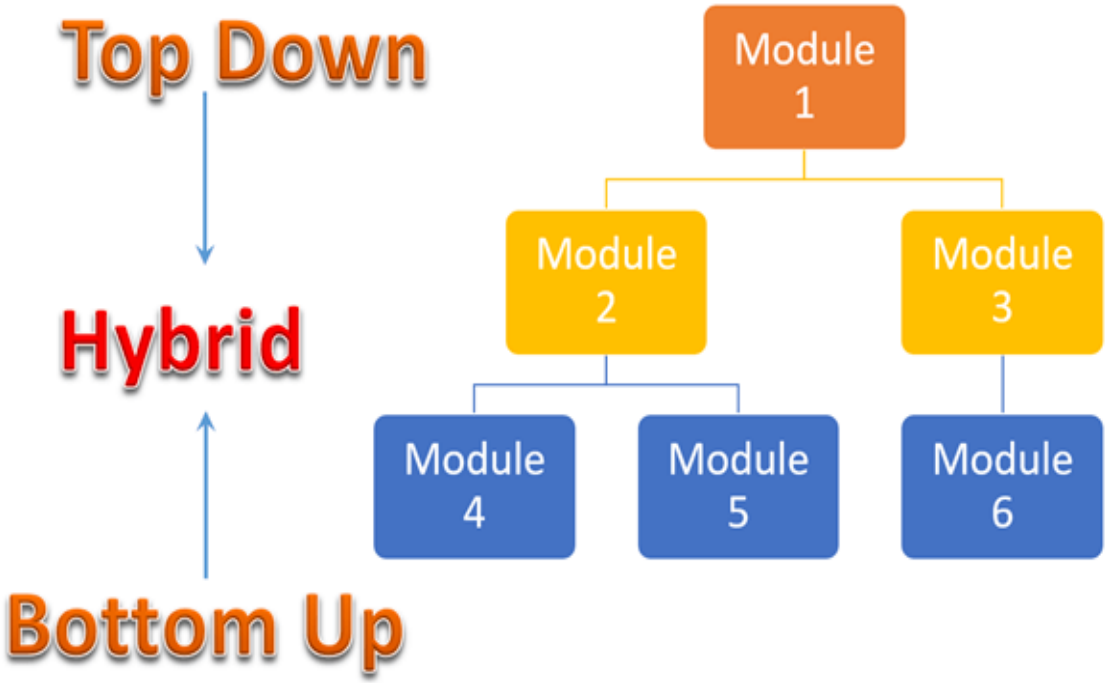




Integration testing

- Integration testing: it is the phase in software testing in which individual software modules are combined and tested as a group.

- Performed by developers or testers
- Top-down approach
- Bottom-up approach



Integration testing – cont.

- Top-down approach
  - Top-down approach is an integration testing technique that is used in order to simulate the behaviour of the lower-level modules that are not yet integrated.
- Fault localization
  - Fault localization is a basic component in fault management systems and it identifies the reason for the fault that can best explain the observed defects.

Top-down Integration testing approach

✧ Advantages
- Fault localization easier
- Few or no drivers needed
- Possibility to obtain an early prototype
- Different order of testing/implementation possible
- Major design flaws found first
  - in logic modules on top of the hierarchy

✧ Disadvantages
- Need lot of stubs / mock objects
- Potentially reusable modules (in bottom of the hierarchy) can be inadequately tested

Integration testing – cont.

- Bottom-up approach
  - Bottom-up approach is a testing strategy in which the modules at the lower level are tested with higher modules until all the modules and aspects of the software are tested properly.

Bottom-up Integration

✧ Advantages
- Fault localization easier
- No need for stubs / fewer mock objects
- Logic modules tested thoroughly
- Testing can be in parallel with implementation

✧ Disadvantages
- Need drivers
- High-level modules (that relate to the solution logic) tested in the last (and least)
- No concept of early skeletal system

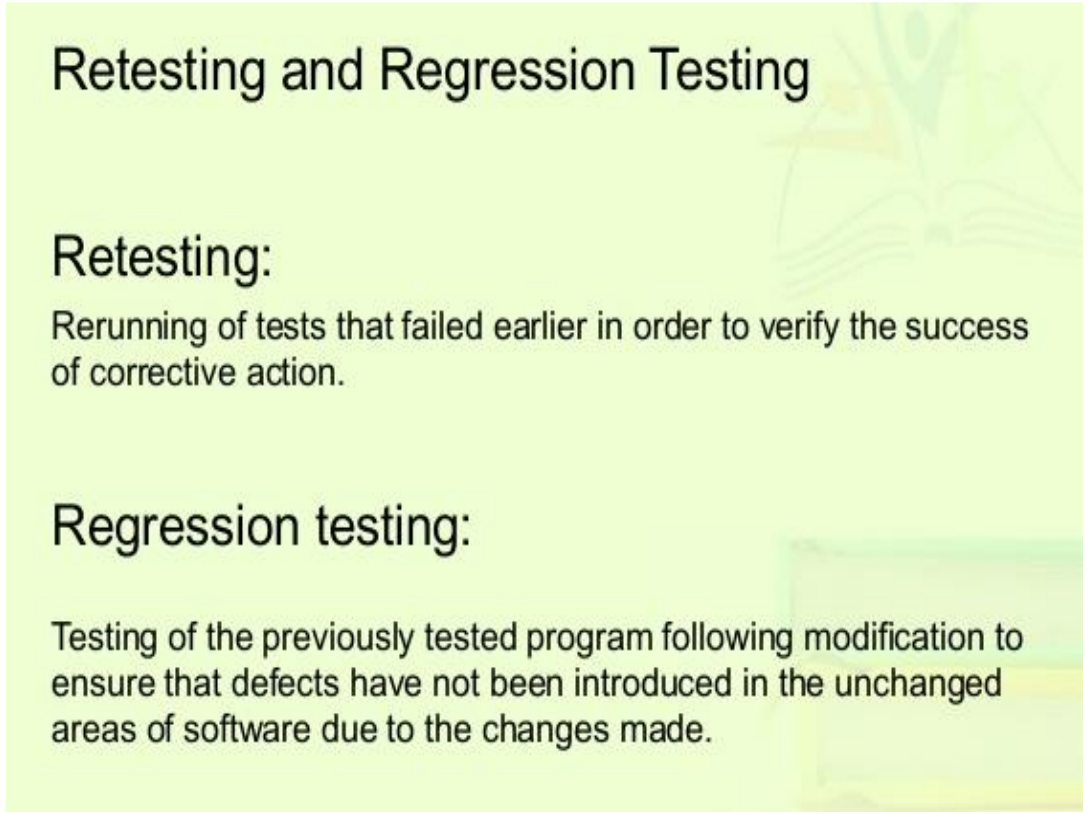
Integration testing – cont.

Top-down vs. Bottom-up

Criteria	Top-Down	Bottom-up
Detection of Major Design Decisions	Detected early	Detected toward the end of the integration process
Observation of System-Level Functions	Observe system-level functions early in the integration process	Only at the end of system integration
Difficulty in Designing Test Cases	Increasingly difficult to design stub behavior and test input	One designs the behavior of a test driver by simplifying the behavior of the actual module
Reusability of Test Cases	Test cases are reused as system-level test cases	All the test cases incorporated into test drivers, except for the top-level test driver, cannot be reused

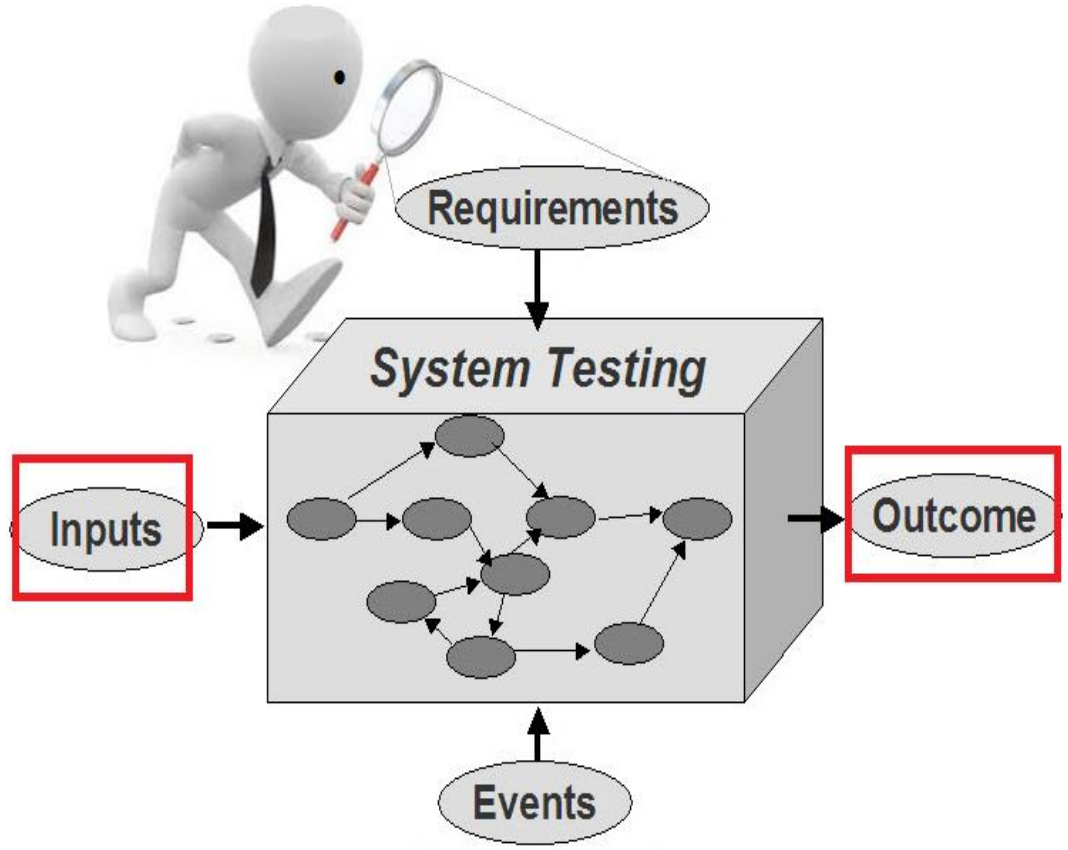
Regression testing

- Regression testing
- Regression testing is a type of testing that is to verify that a code change in the software does not impact the existing functionalities of the product.
- Reduce the side effects
- Retesting vs. regression testing



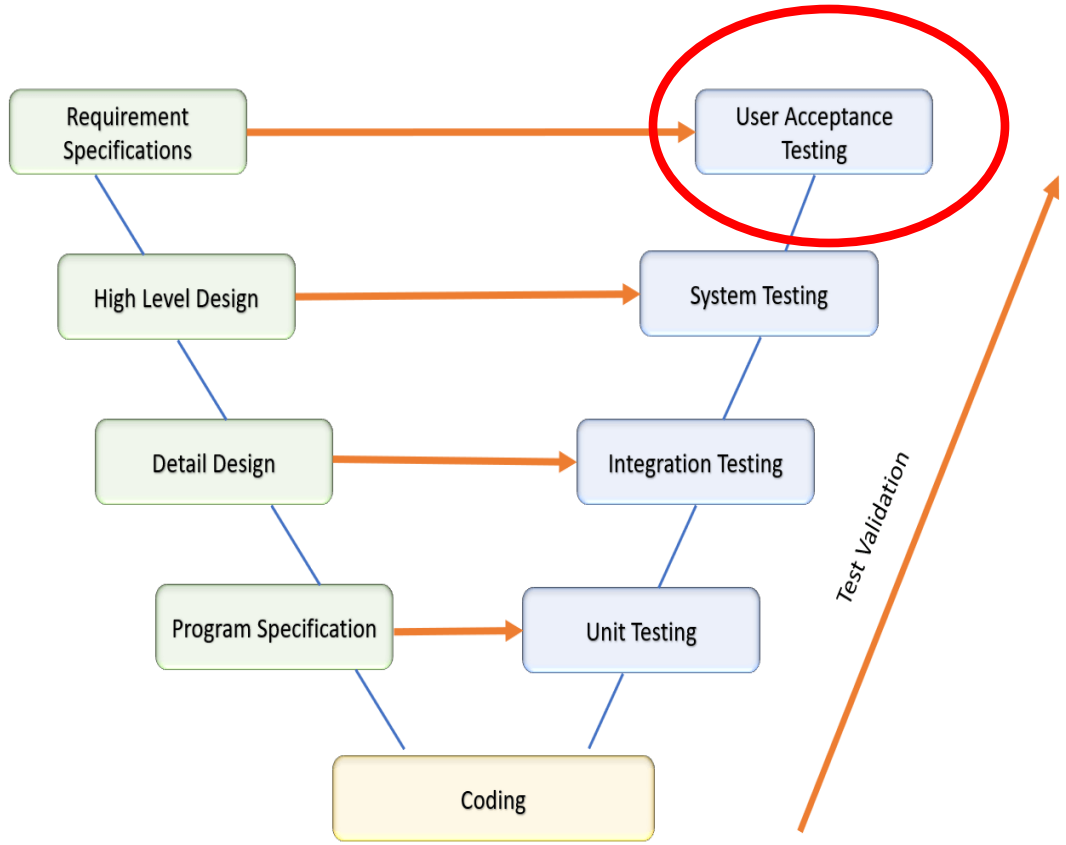
System testing

- System testing
- System testing is testing conducted on a complete integrated system to evaluate the system's compliance with its specified design requirements.
- Sometimes on simulators
- Test procedures
  - Procedures
  - Inputs
  - Expected outputs/results



Acceptance testing

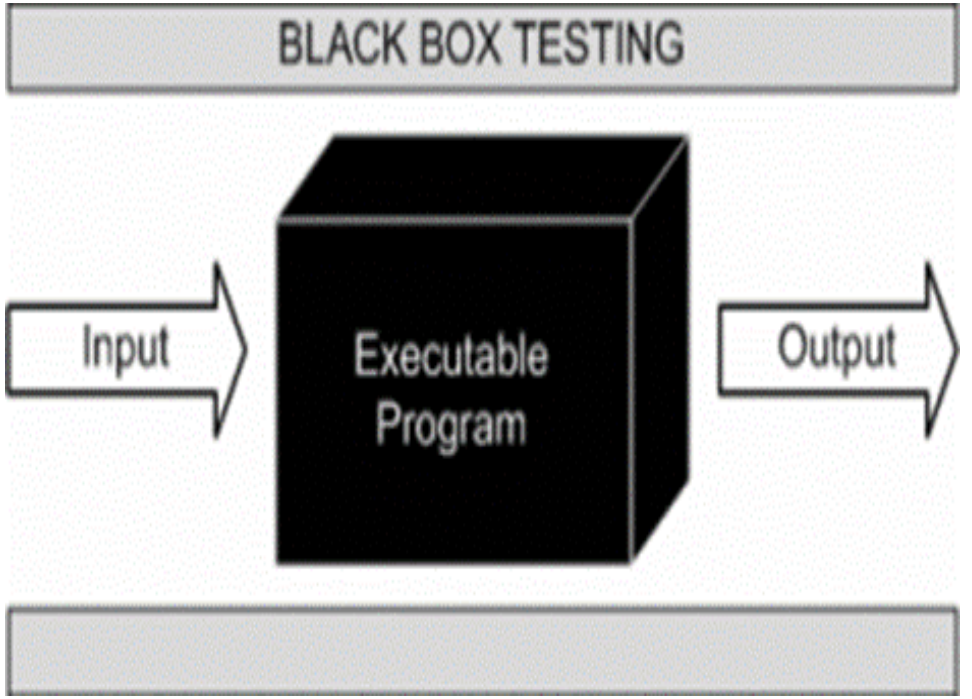
- Acceptance testing
- Acceptance testing is formal testing with respect to user needs, requirements, and business processes conducted to determine whether a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether the system is accepted.
- Customers make decision



Testing techniques

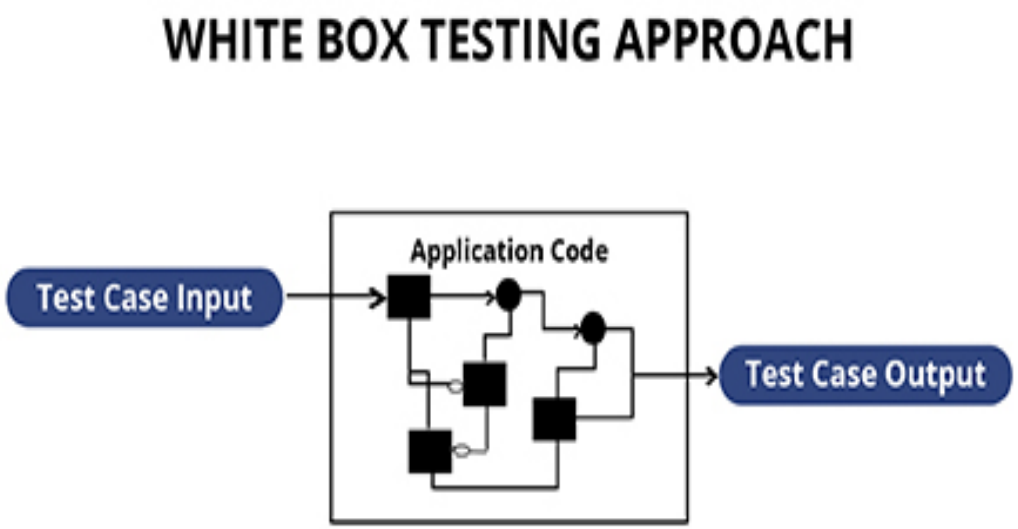
- Black-box testing
- Black-box testing is a method of software testing that examines the functionality of software without peering into its internal structures or workings.

- It can apply to all test levels
  - From unit testing to system testing



Testing techniques – cont.

- White-box testing
- White-box testing is a method of software testing that tests internal structures or workings of software as opposed to its functionality (i.e. black-box testing).



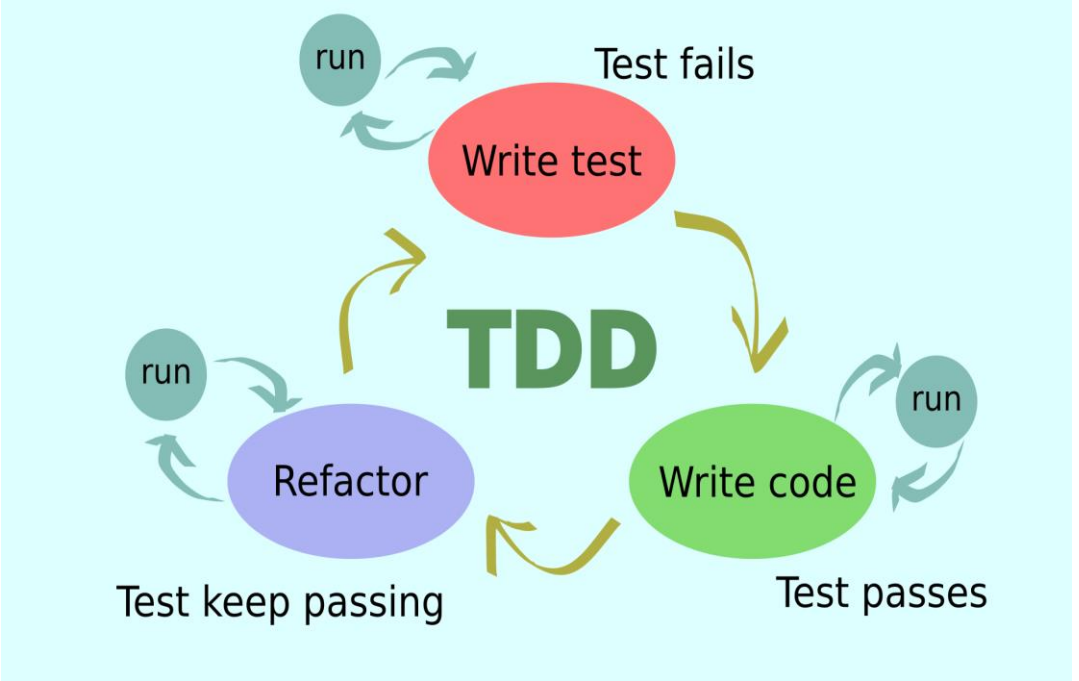
- It can apply to all test levels
  - From unit testing to system testing

Testing techniques – cont.

Comparison Base	Black box testing	White box testing
Meaning	It is a testing approach which is used for testing the software without the knowledge of the internal design or structure of program or application.	It is a testing approach in which internal structure or design is known to the software tester who is going to test the software.
Testing Techniques	Equivalence Partitioning Boundary Value Analysis Decision Table State Transition	Statement Coverage Branch Coverage Path Coverage
Implementation Knowledge	Implementation Knowledge is not vital to perform Black Box Testing.	Implementation Knowledge is vital to perform White Box Testing.
Programming Knowledge	It is not required to carry out Black Box Testing.	It is required to carry out White Box Testing.
Prime Focus	Primarily focus on the functionality of the system under test.	Primarily focus on the testing of the program code of the system under test like branches, code structure, loops, conditions, etc.
Time-Period	It is less exhaustive & time-consuming.	Exhaustive & time-taking method.
Target	The main aim is to check on what functionality is performing by the system under test.	The main aim of is to check on how the system is performing.
Types of Testing	A. Functional Testing B. Non-functional testing C. Regression Testing	A. Path Testing B. Loop Testing C. Condition testing

Test Driven Development

- Test Driven Development (TDD)
  - TDD is a technique for building reliable software that guides software development by writing tests.
  - TDD is not about testing or development (i.e., coding).
  - It is rather about design, where design is evolved through refactoring.
  - Developers write unit tests (Not testers)
- Examples:



Test Driven Development – cont.

- Benefits of TDD (Test - driven development)

- **Early problem discovery**
  - It is better if we can find the error/problem before staging or production stage.
- **Facilitates change**
  - Unit testing allows developers to refactor code anytime. It ensures the code still works correctly.
- **Simplifies integration**
  - Unit testing increases confidence in integration. If everything is tested before integration, integration should be easier.
- **Self documentation**
  - Let the test tell the story.
- **It visualises the design**
  - By doing TDD, unit testing helps developers to design classes, to smell code mess and to refactor the code

Summary

- Software testing
  - Concept
- Software testing plan
- Levels of software testing
  - Unit testing,
  - Integration testing: top-down approach, bottom-up approach
  - System testing
  - Acceptance testing
- Testing techniques
  - Black-box testing
  - White-box testing
- Test driven development (TDD)

Announcement

- Group project: apply what we have learned
  - SDD
  - Implementation

- 9 of you do not have a group. Please let me know if you need help.