For Updated version, please click on this
http://ocw.ump.edu.my

Universiti Malaysia PAHANG
Engineering • Technology • Creativity

BCS3323 – Software Testing and Maintenance

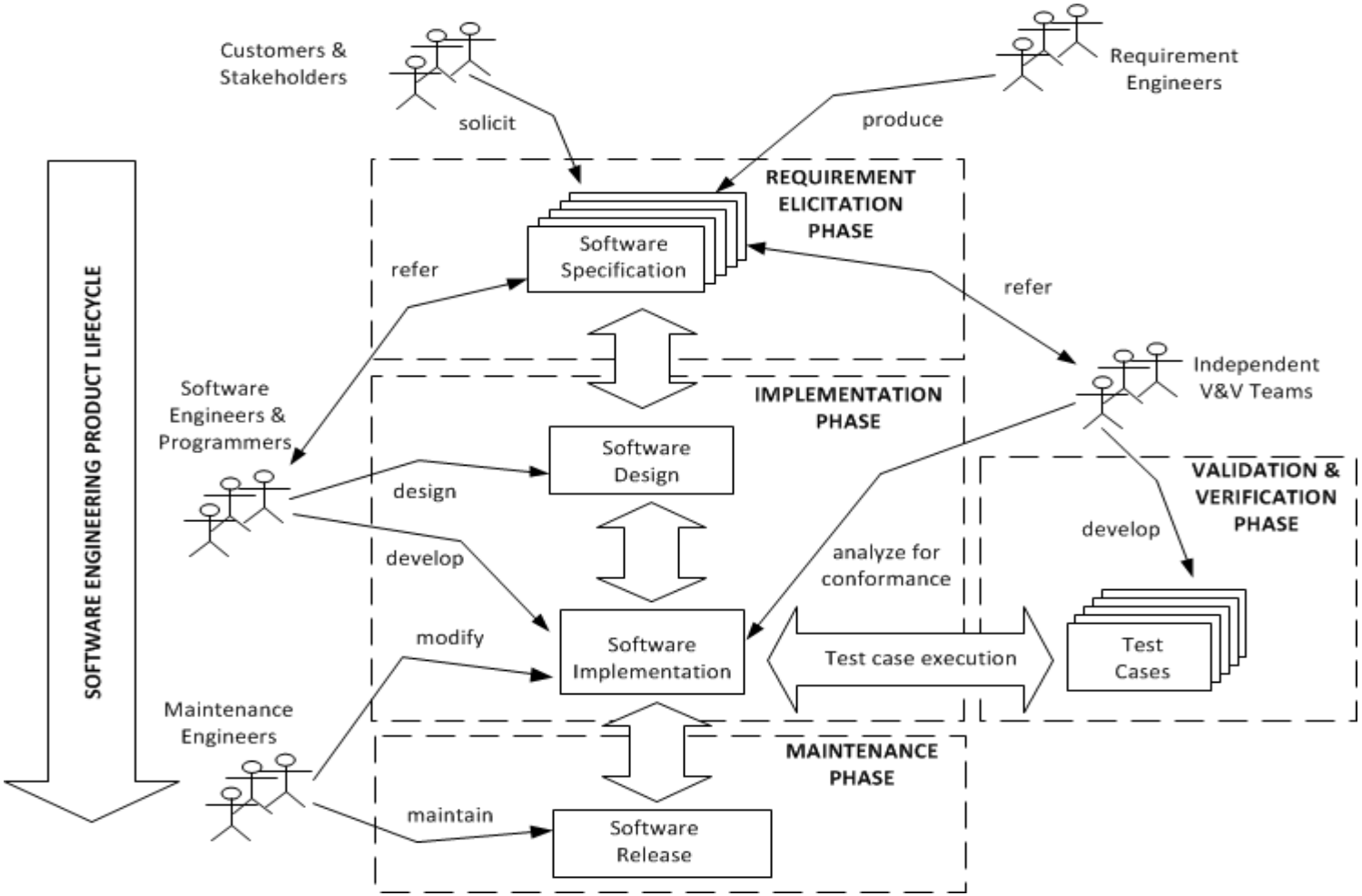
Testing throughout the SDLC

Editors
Prof. Dr. Kamal Z. Zamli
Dr. AbdulRahman A. Alsewari
Faculty of Computer Systems & Software Engineering
alswari@ump.edu.my

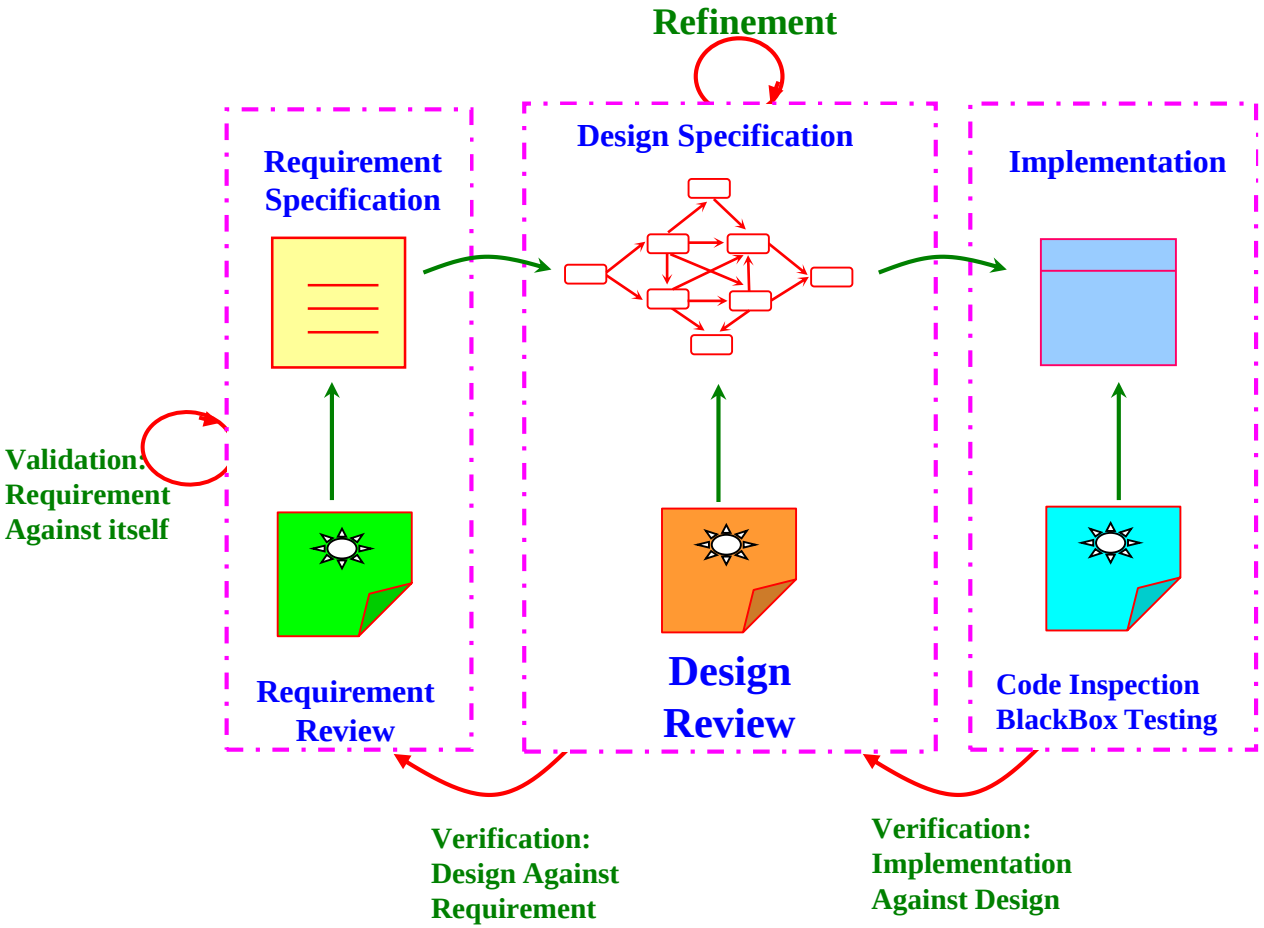
Chapter 2.1

- Aims is students be able to discover
  - The testing phase in each SDLC
  - The difference Testing between SDLCs
  - The testing Levels
- Expected Outcomes
  - Students will propose the proper testing techniques based on SDLC
  - Students will select the testing level for each phase of SDLC.
- References
  - ISTQB
  - MSTB/GTB
  - http://www.softwaretestingclass.com/software-testing-tools-list/
  - http://www.softwaretestinggenius.com/articalDetails.php?qry=572#commentsList

High Level View of Software Engineering



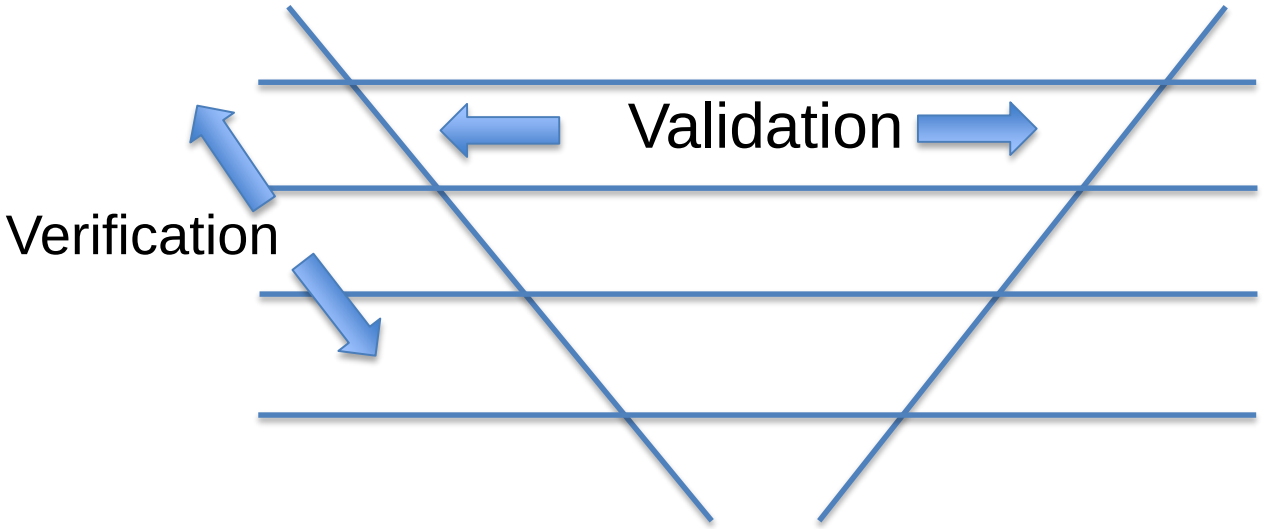
Verification and Validation



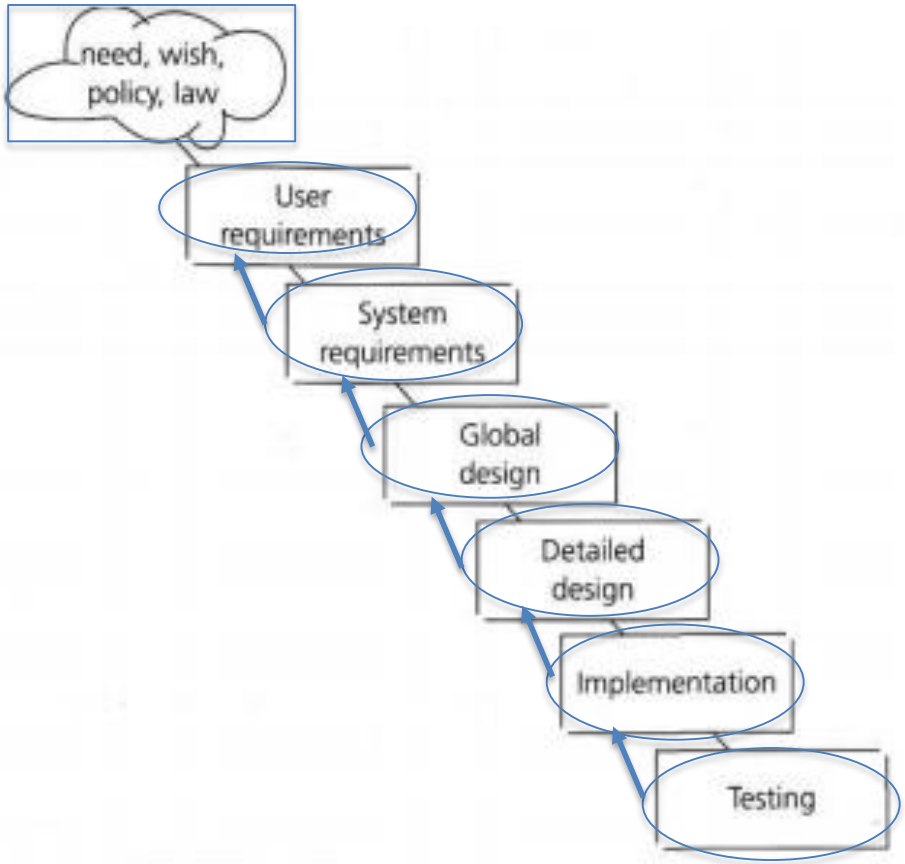
In every software development lifecycle, part of testing is focused on verification and validation

Verification vs Validation

- Verification – Are we building the product right? (i.e. self consistent/traceable).
- Validation – Are we building the right product?



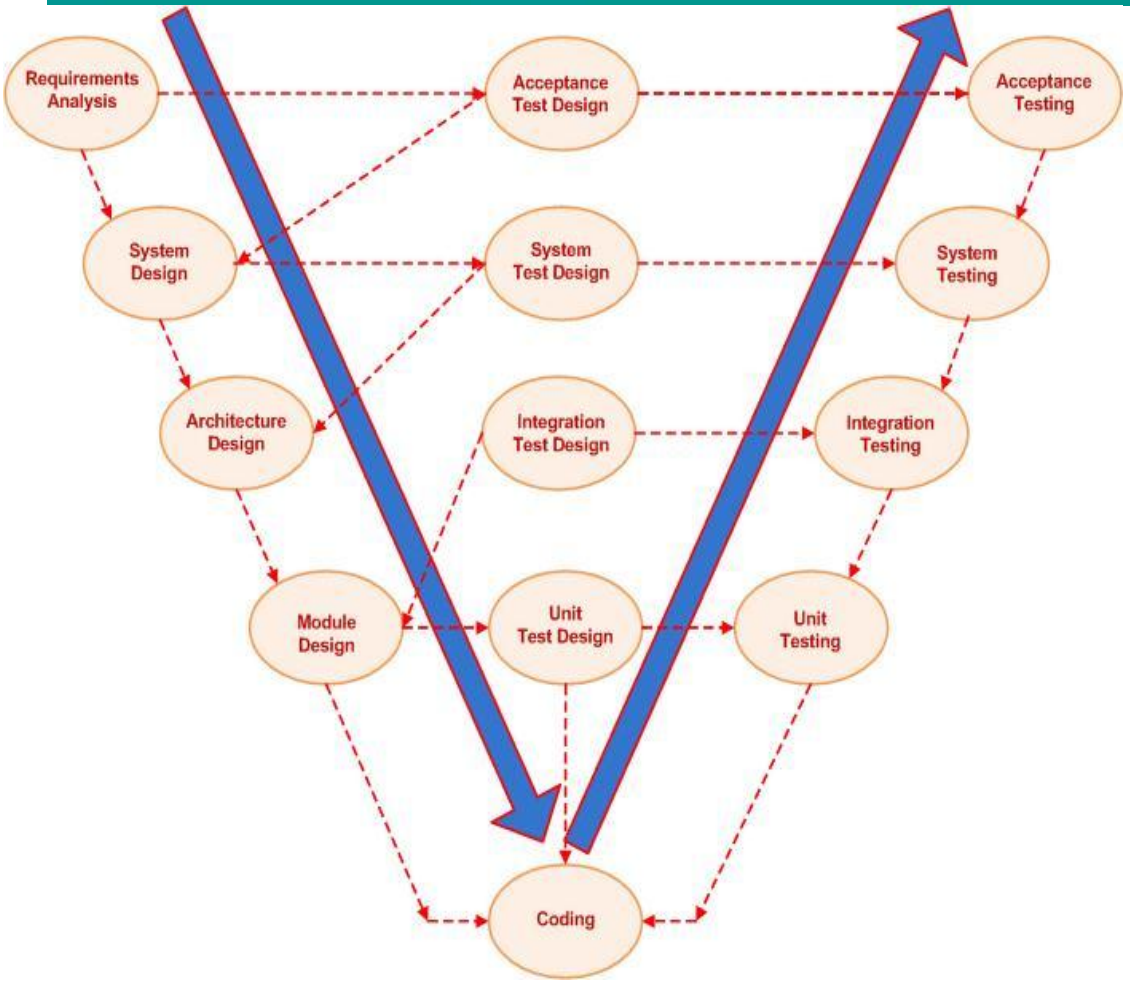
The Waterfall Model



Waterfall model

- Testing tends to happen very late, hence, defects are detected close to live implementation
- Difficult to get feedback backward up.

The V-Model



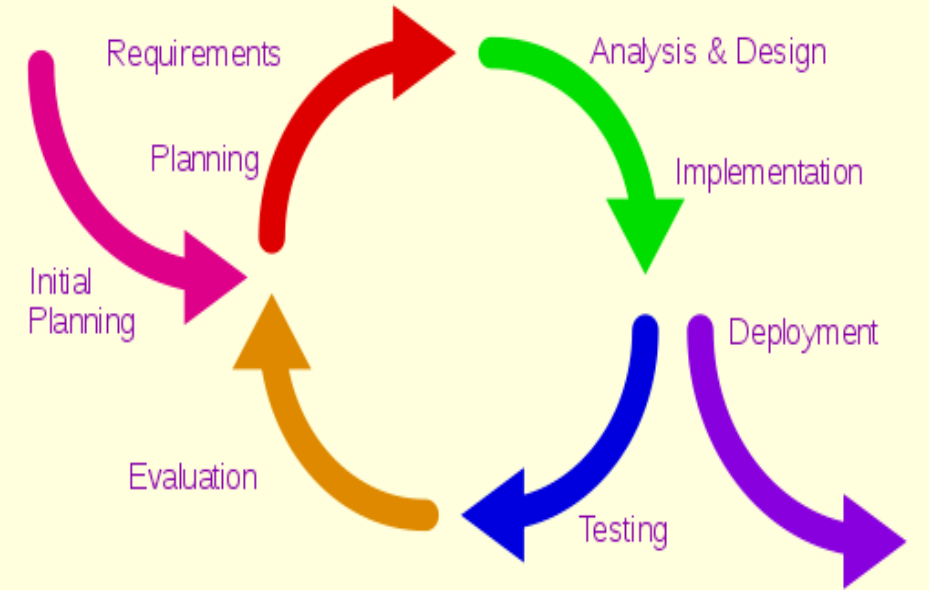
Source: https://en.wikipedia.org: Author m_ajith 2007

Improves the weaknesses of the waterfall model so that testing can start early (even on test basis documentation) and in parallel

Four test levels

- Unit/Component testing - find defects and verifies functionality of software components
- Integration testing – search for defects on interfaces between components, operating system, hardware etc.
- System testing – verification against specified requirements
- Acceptance testing – validation testing against user need.

Iterative Development lifecycle

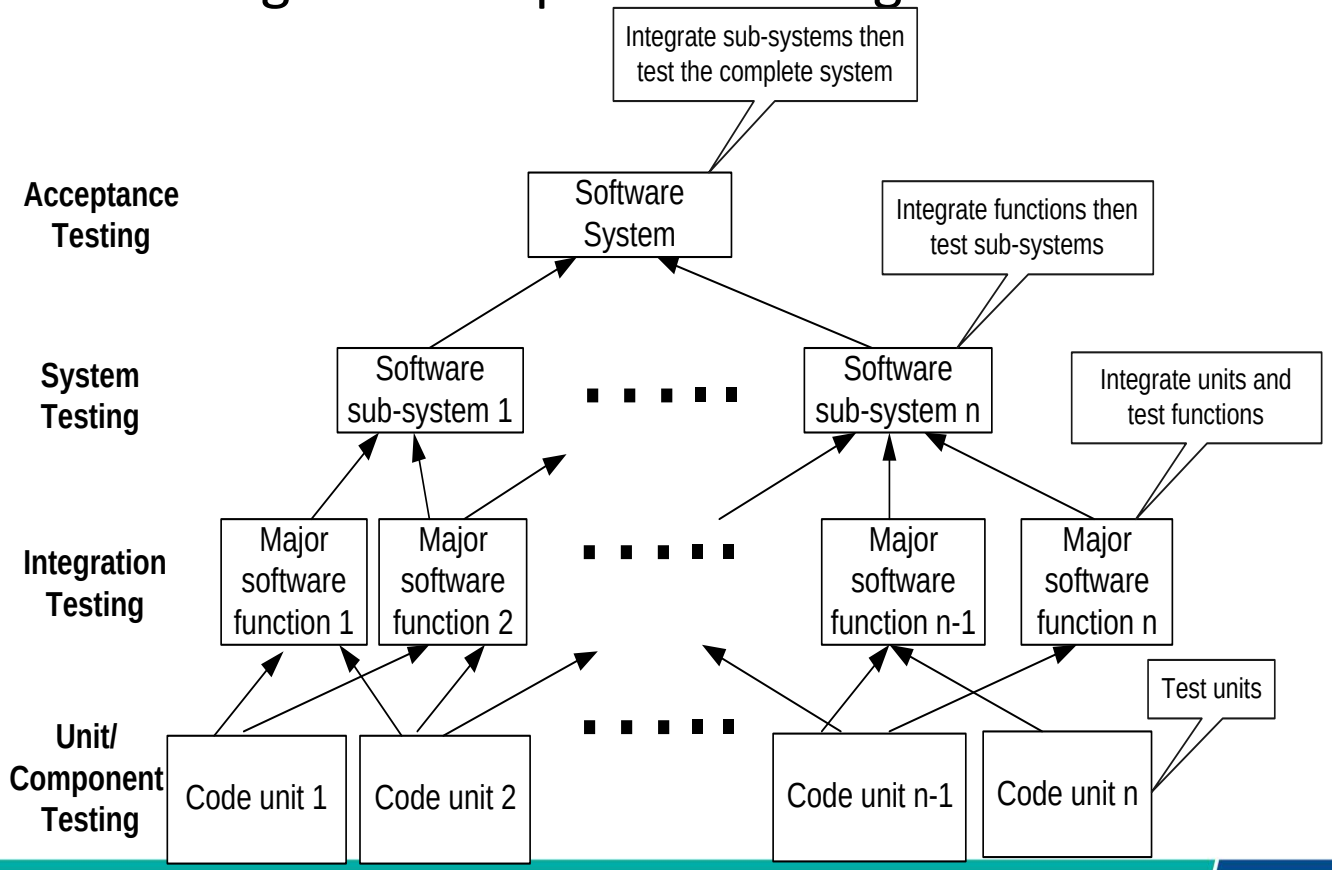


- Work product delivery is divided into increments where each increment adds new functionality.
- Many variants including Rapid Application Development, Agile development, etc.
- Here, testing occurs after each increments.

Levels of Software Testing

Test Levels

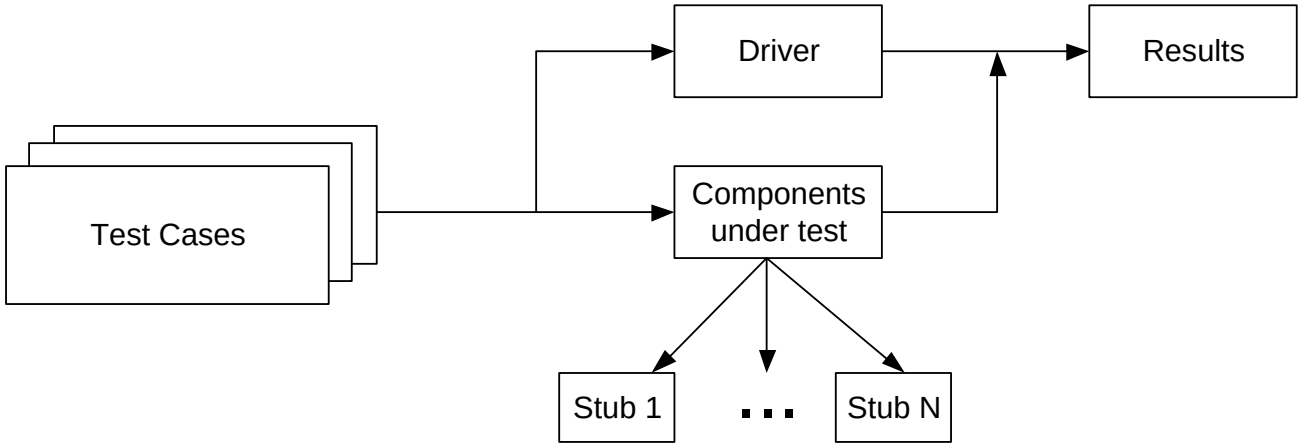
- In developing large software system, testing usually involves several levels consisting of unit testing, integration testing, system testing and acceptance testing



Component Testing

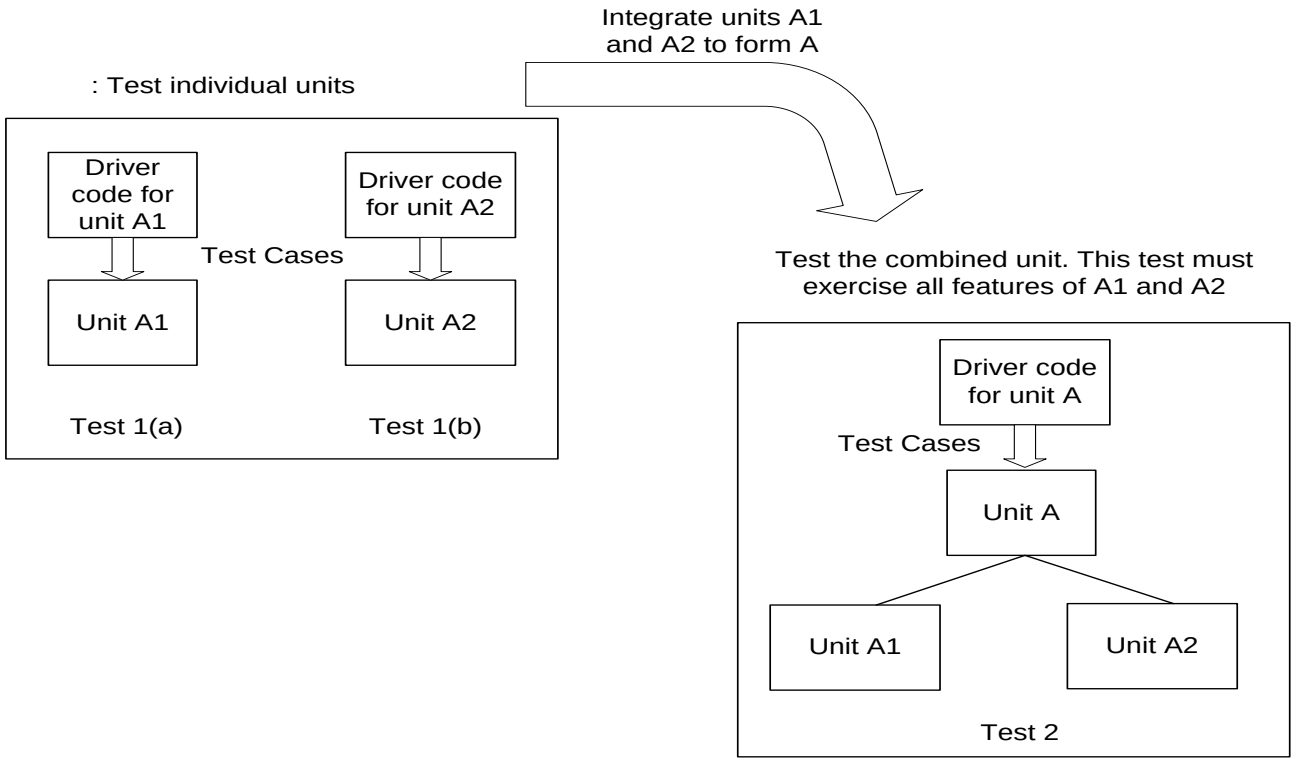
- Component testing is to find for defects in , and verifies the functionalities of the software components (methods, functions, modules, programs, objects, classes, etc.) that are separately testable (e.g. functionally independent from the rest of the code).

Test harness consists **of drivers** that call the target code and **stubs** that represent modules or units it calls.



Integration Testing

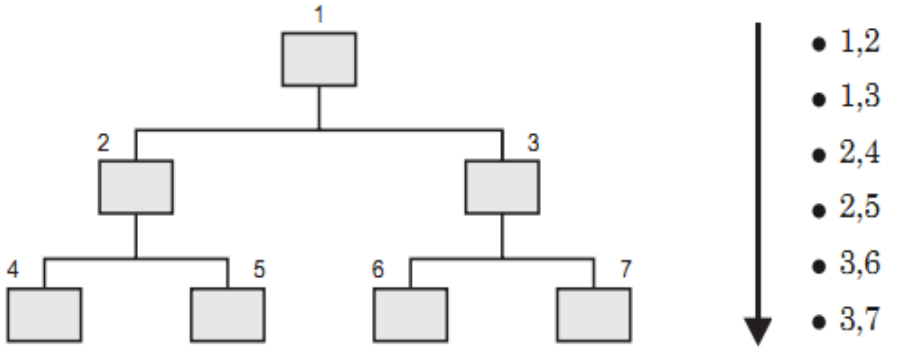
- Integration Testing evaluates interfaces between system components, and interactions with different parts of a system, such as the operating system, file system, hardware, or interfaces between systems.



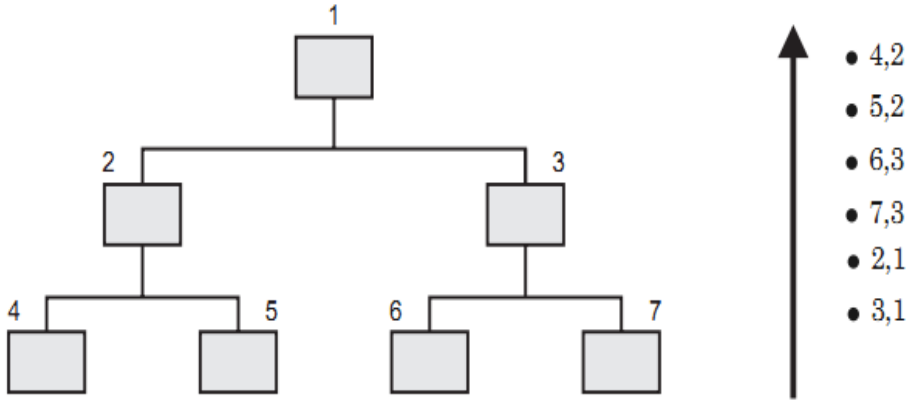
Integration Testing

- Integration testing can be performed using one of the following approaches:
  - Big Bang approach
  - Bottom-up approach
  - Top-down approach

Top-down control structure



Bottom-up integration



System Testing

- System testing is concerned with the behaviour of a whole software/system as defined by the scope of a development project in specification documents.
- System testing include functional requirements and non-functional requirements (e.g. performance and reliability). System testing may address:
  - Stress tests –
  - Volume tests –
  - Configuration tests –
  - Security tests –
  - Compatibility tests
  - Regression tests –
  - Timing tests –
  - Environmental tests –
  - Recovery tests –
  - Maintenance tests –
  - Documentation tests –.
  - Human factor (or Usability) tests

Acceptance Testing 1/2

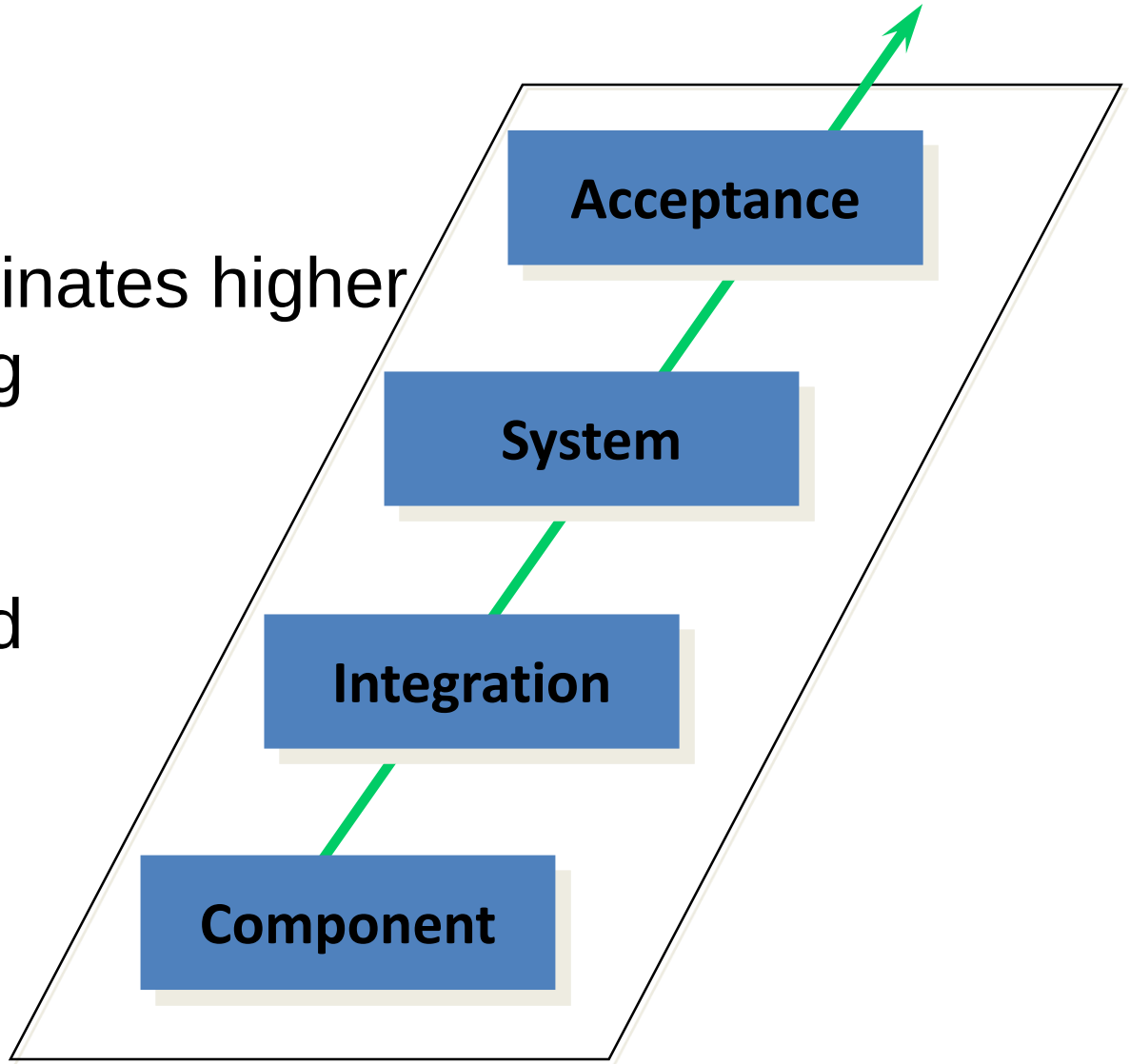
- When the development organization has performed its system testing and has corrected all or most defects, the system will be delivered for acceptance testing. . As such, acceptance testing assesses the system's readiness for deployment and use.

- Acceptance testing may address:
  - Benchmark test
  - Pilot test
  - Alpha test –development site (i.e. pilot test at the development side).
  - Beta test –client site (i.e. pilot test at the client side).
  - Smoke testing

Black box vs White box?

Black box dominates higher levels of testing

White box used predominately at lower levels



On the target of testing

- Functional testing – black box testing – is testing the function of a system or part of the system (modules) in term of what it does (i.e. specification based).
- Non-functional testing – is testing of how well the system works (e.g. performance, load, stress, usability, maintainability, reliability, and portability)
- Structural testing – white box – is testing of the structure of the systemor systems modules

Confirmation vs Regression testing

- Confirmation testing and Regression testing

Testing of changes or the Modifications

  - Confirmation testing - (failed) test is executed the same way as it was first time in order to confirm detected bugs are fixed.
  - Regression testing – verify modification in a software has not cause adverse unintended behavior. Often, associated with impact analysis. Regression testing, done by different testing techniques/test cases/etc.