



Software Quality Assurance & Testing

LECTURE 3: TESTING THROUGHOUT THE SOFTWARE LIFE CYCLE

Prepared By:
Md. Minhaj Hosen, Lecturer, DIU

Topics:

- ▶ **1) Software Development Models**
- ▶ **2) Testing within a life cycle model**
- ▶ **3) Test Levels**
- ▶ **4) Test Types**
- ▶ **5) Maintenance testing**

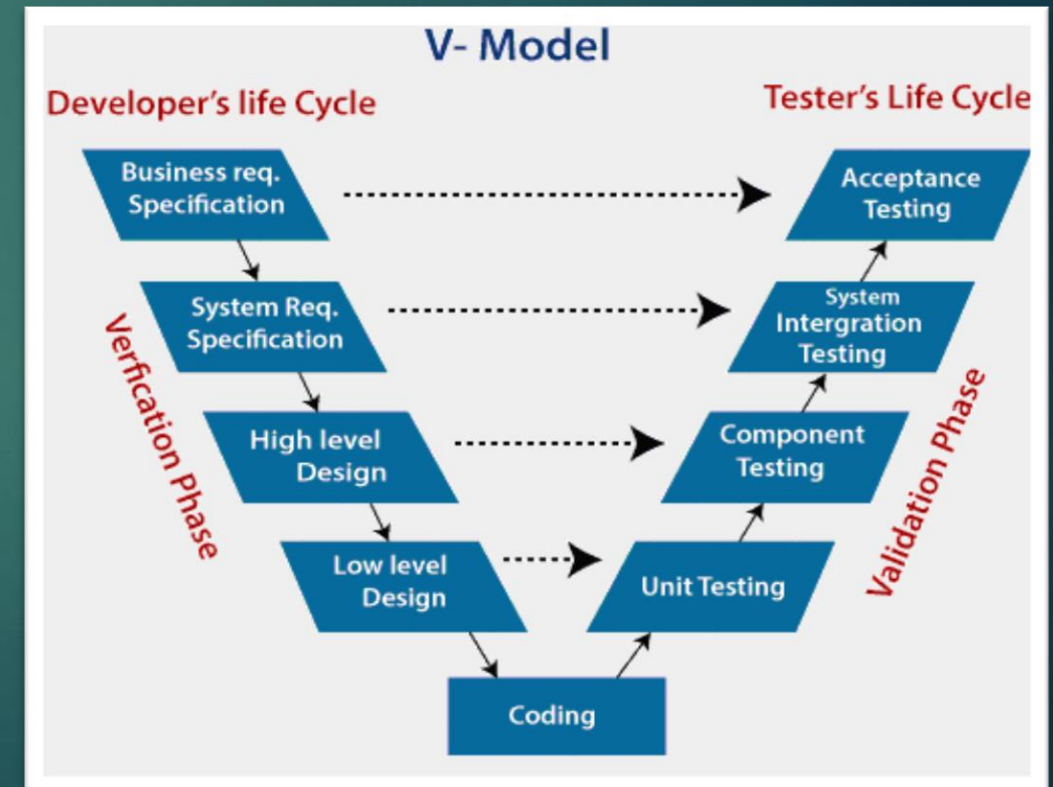
1) Software Development Models

- ▶ a) **Waterfall Model** – It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed fully before the next phase can begin.
- ▶ b) **V-Model** – It is also called as verification and validation model. In V-model for every development activity there is a corresponding testing activity. V Model supports different Test Levels.

Test Level - Test Level is a group of test activities that are organized and managed together.

Different Test Levels -

Component Testing, Integration Testing, System Testing, Acceptance Testing



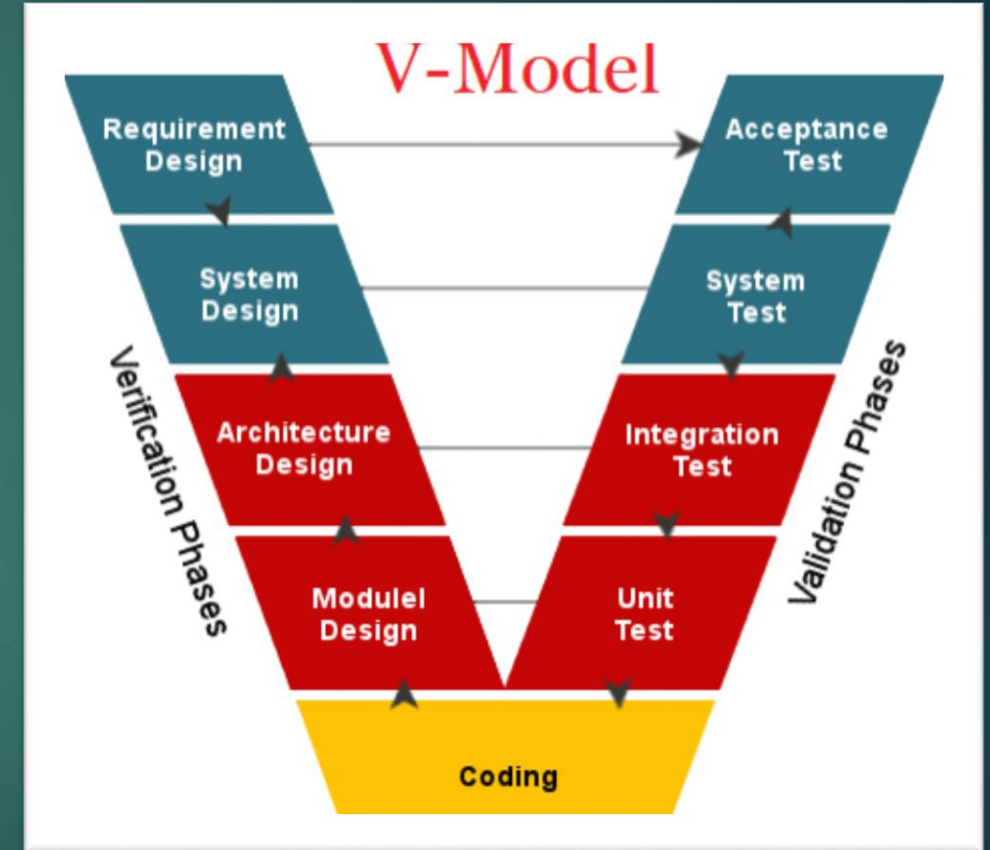
Verification Vs Validation

Verification -

- ▶ Verification checks are **we building the system right?**
- ▶ Checks whether the software conforms to **specifications.**
- ▶ Verification is a **static practice.**
- ▶ Verification uses methods like **inspections, reviews, walkthroughs,** etc.

Validation -

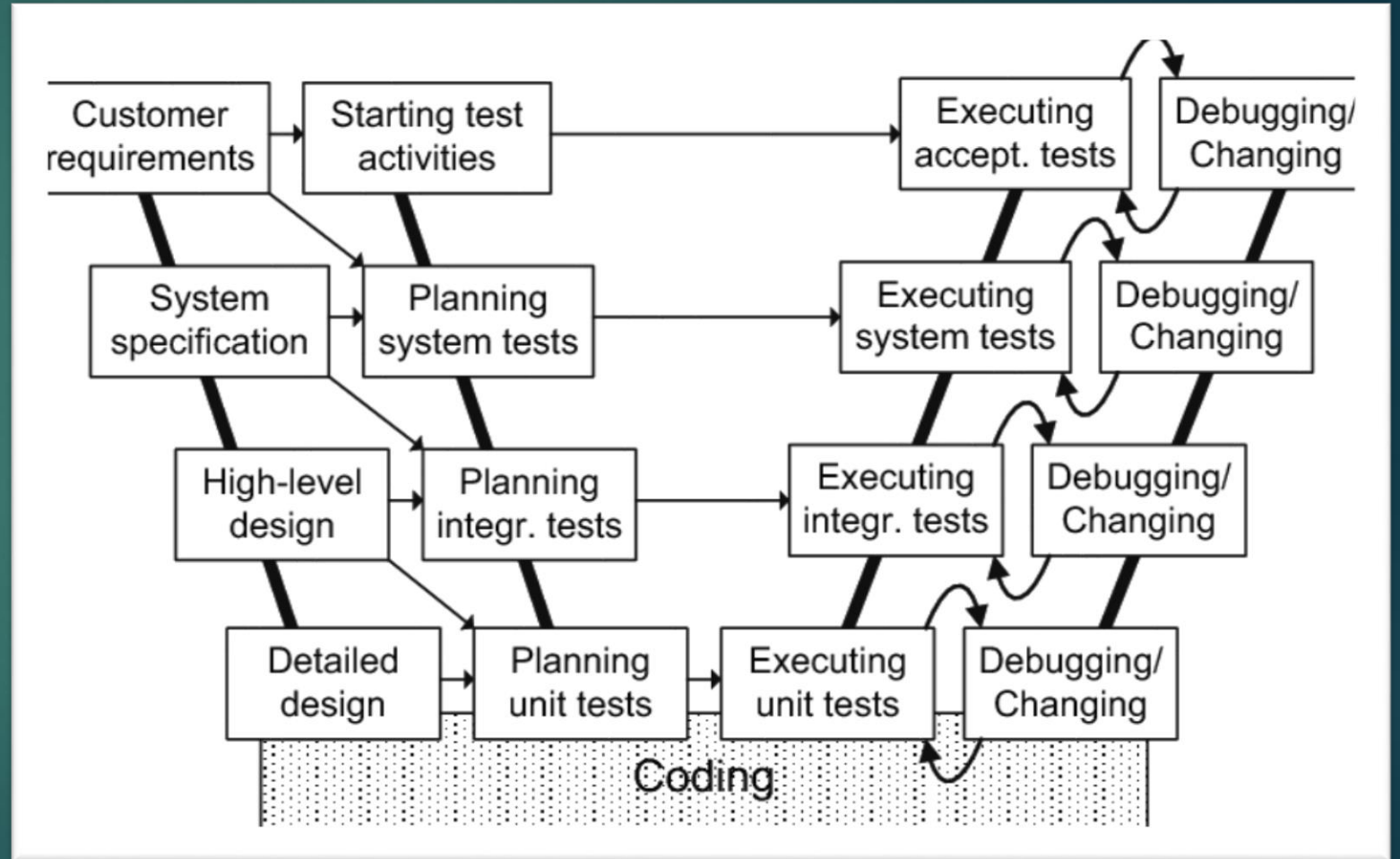
- ▶ Validation checks are **we building the right system?**
- ▶ Check whether software meets the customer expectations and requirements.
- ▶ Validation is a **dynamic mechanism.**
- ▶ Validation uses methods like **black box (functional) testing, gray box testing, and white box (structural) testing** etc.



W-Model

The W-model can be seen as an extension to the general V model

- The W-model states, that certain **quality assurance activities** shall be performed in parallel with the development process



1) Software Development Models

c) Iterative-incremental development models

- ▶ Iterative-incremental development is the process of establishing requirements, designing, building and testing a system, done as a series of shorter development cycles. Examples are: prototyping, rapid application development (RAD), Rational Unified Process (RUP) and agile development models.
- ▶ Agile Development promotes adaptive planning, evolutionary development, early delivery, continuous improvement, and encourages rapid and flexible response to change.

2) Testing within a life cycle model

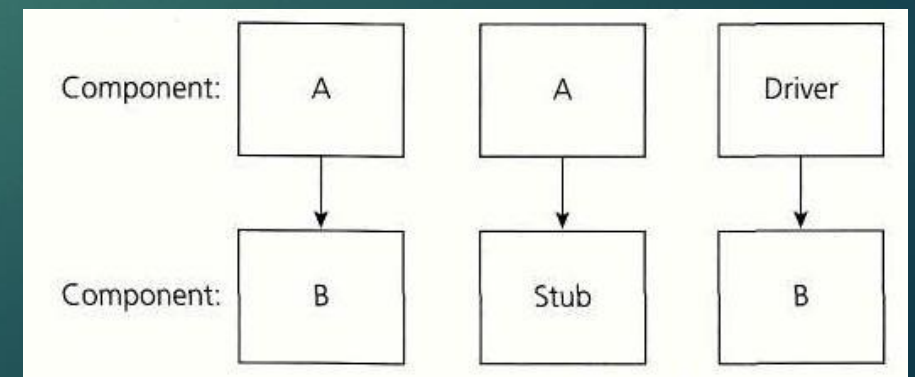
In any life cycle model, there are several characteristics of good testing:

- ▶ For every development activity there is a corresponding testing activity.
- ▶ Each test level has **test objectives** specific to that level.
- ▶ The analysis and design of tests for a given test level should begin during the corresponding development activity.
- ▶ Testers should **be involved in reviewing documents** as soon as drafts are available in the development life cycle.

3) Test levels

a) Component testing

- ▶ Component testing searches for defects in, and verifies the functioning of, software (e.g. modules, programs, objects, classes, etc.) that are separately testable. **Component testing also called as unit and module testing.**
- ▶ Component testing may include testing of functionality and specific non-functional characteristics, such as resource-behaviour (e.g. memory leaks) or robustness testing, as well as structural testing (e.g. branch coverage).
- ▶ The only disadvantage of component testing is that the test cases are derived **from work product such as software design or data model.**
- ▶ For Component testing Stubs and Drivers are used to replace missing components of the software.
- ▶ **Stubs** - A stub is called from a software component which is under test
- ▶ **Driver** - A driver calls a component to be tested



3) Test levels

b) Integration testing

- ▶ Integration testing **tests interfaces between components**, interactions with different parts of a system, such as the operating system, file system, hardware, or interfaces between systems.
- ▶ Component integration testing tests the interactions between software components and is done after component testing;
- ▶ System integration testing tests the **interactions between different systems and may be done after system testing.**
- ▶ Testing of specific non-functional characteristics (e.g. performance) may be included in integration testing.



3) Test levels

c) System testing

- ▶ System testing is concerned with the **behavior of a whole system/product** as defined by the scope of a development project or programme.
- ▶ In system testing, the test environment should correspond to the final target or production environment as much as possible **in order to minimize the risk of environment-specific failures not being found in testing.**
- ▶ System testing may include tests based on risks and/or on requirements specifications, business processes, use cases, or other high level descriptions of system behavior, interactions with the operating system, and system resources.
- ▶ System testing **should investigate both** functional and non-functional requirements of the system.

3) Test levels

d) Acceptance testing

- ▶ Acceptance testing is often the responsibility of the customers or users of a system; other stakeholders may be involved as well.
- ▶ The **goal in acceptance testing is to establish confidence in the system**, parts of the system or specific non-functional characteristics of the system
- ▶ Contract and regulation acceptance testing
- ▶ Contract acceptance testing is performed against a contract's acceptance criteria for producing custom-developed software. Acceptance criteria should be defined when the contract is agreed. Regulation acceptance testing is performed against any regulations that must be adhered to, such as governmental, legal or safety regulations.
- ▶ **Alpha and beta (or field) testing** - Alpha testing is performed at the developing organization's site. Beta testing, or field testing, is performed by people at their own locations. Both are performed by potential customers, not the developers of the product.

4] Test types

a) Testing of function (functional testing)

- ▶ The functions that a system, subsystem or component are to perform may be described in work products such as a requirements specification, use cases, or a functional specification, or they may be undocumented. The functions are “what” the system does.
- ▶ A type of functional testing, **security testing, investigates the functions** (e.g. a firewall) relating to detection of threats, such as viruses, from malicious outsiders. Another type of functional testing, interoperability testing, evaluates the capability of the software product to interact with one or more specified components or systems.

b) Testing of non-functional software characteristics (non-functional testing)

- ▶ Non-functional testing includes, but is not limited to, performance testing, load testing, stress testing, usability testing, maintainability testing, reliability testing and portability testing. It is the **testing of “how” the system works.**
- ▶ Non-functional testing may be performed at all test levels.

4] Test types

c) Testing of software structure/architecture (structural testing)

- ▶ Structural (**white-box**) testing may be performed at all test levels. Structural techniques are best used after specification-based techniques, in order to help measure the thoroughness of testing **through assessment of coverage of a type of structure**.
- ▶ Structural testing approaches can also be applied at system, system integration or acceptance testing levels (e.g. to business models or menu structures).

d) Testing related to changes (confirmation testing (retesting) and regression testing)

- ▶ After a defect is detected and fixed, the software should be retested to confirm that the original defect has been successfully removed. **This is called confirmation. Debugging (defect fixing) is a development activity, not a testing activity.**
- ▶ Regression testing is the repeated testing of an already tested program, after modification, to discover any **defects introduced or uncovered as a result of the change(s)**. It is performed when the software, or its environment, is changed.
- ▶ Regression testing may be performed at all test levels, and applies to functional, non-functional and structural testing.

5) Maintenance testing

- ▶ Once deployed, a software system is often in service for years or decades. During this time the system and its environment are often **corrected, changed or extended**.
- ▶ Modifications include planned enhancement changes (e.g. release-based), corrective and emergency changes, and changes of environment,
- ▶ Maintenance testing for migration (e.g. from one platform to another) should include operational tests of the new environment, as well as of the changed software.
- ▶ Maintenance testing for the retirement of a system may include the testing of data migration or archiving if **long data-retention periods are required**.
- ▶ Maintenance testing may be done at any or all test levels and for any or all test types.