Software Testing in V Model

**Drawback of waterfall STLC:**

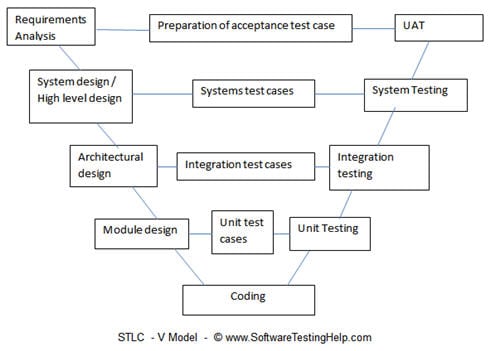
One of the major handicaps of waterfall STLC model was that defects were found at a very later stage of the development process since testing was done at the end of the development cycle. It became very challenging and costly to fix the defects since it was found at a very later stage.

**V Model:**

To overcome this problem of waterfall model, a new development model was introduced called the “V Model”.

V model is now one of the most widely used software development processes. Introduction of the V model has actually proved the implementation of testing right from the requirement phase. V model is also called a verification and validation model.

In the V model, the development and QA activities are done simultaneously. There is no discrete phase called Testing, rather testing starts right from the requirement phase. The verification and validation activities go hand in hand.



V Model STLC Diagram

In a typical development process, the left-hand side shows the development activities and the right hand side shows the testing activities. I should not be wrong if I say that in the development phase both verification and validation are performed along with the actual development activities.

Now let’s understand the figure:

1. **Left-Hand Side**

As said earlier, left-hand side activities are development activities. Normally we feel, what testing can we do in the development phase, but this is the beauty of this model which demonstrates that testing can be done in all phase of development activities as well.

**Requirement analysis:** In this phase, the requirements are collected, analyzed and studied. Here how the system is implemented, is not important but, what the system is supposed to do, is important. Brain storming sessions/walkthrough, interviews are done to have the objectives clear.

* + Verification activities: Requirements reviews.
  + Validation activities: Creation of UAT (User acceptance test) test cases
  + Artifacts produced: Requirements understanding document, UAT test cases.

**System requirements /High-level design:** In this phase, the high-level design of the software is built. The team studies and investigates on how the requirements could be implemented. The technical feasibility of the requirements is also studied. The team also comes up with the modules that would be created/ dependencies, Hardware/software needs

* Verification activities: Design reviews
* Validation activities: Creation of System test plan and cases, Creation of traceability metrics
* Artifacts produced: System test cases, Feasibility reports, System test plan, Hardware-software requirements, and modules to be created, etc.

**Architectural design:** In this phase, based on the high-level design, software architecture is created. The modules, their relationships, and dependencies, architectural diagrams, database tables, technology details are all finalized in this phase.

* Verification activities: Design reviews
* Validation activities: Integration test plan and test cases.
* Artifacts produced: Design documents, Integration test plan and test cases, Database table designs etc.

**Module design/Low-level Design:** In this phase, each and every module of the software components are designed individually. Methods, classes, interfaces, data types etc are all finalized in this phase.

* Verification activities: Design reviews
* Validation activities: Creation and review of unit test cases.
* Artifacts produced: Unit test cases

**Implementation / Code:** In this phase, the actual coding is done.

* Verification activities: Code review, test cases review
* Validation activities: Creation of functional test cases.
* Artifacts produced: test cases, review checklist.

1. **Right Hand Side**

Right-hand side demonstrates the testing activities or the Validation Phase. We will start from the bottom.

**Unit Testing:** In this phase, all the unit test case, created in the Low-level design phase are executed.

Unit testing is a white box testing technique, where a piece of code is written which invokes a method (or any other piece of code) to test whether the code snippet is giving the expected output or not. This testing is basically performed by the development team. In case of any anomaly, defects are logged and tracked.

* Artifacts produced: Unit test execution results

**Integration Testing:**  In this phase, the integration test cases are executed which were created in the Architectural design phase. In case of any anomalies, defects are logged and tracked.

Integration testing is a technique where the unit tested modules are integrated and tested whether the integrated modules are rendering the expected results. In simpler words, It validates whether the components of the application work together as expected.

* Artifacts produced: Integration test results.

**Systems testing:** In this phase all the system test cases, functional test cases and non-functional test cases are executed. In other words, the actual and full fledge testing of the application takes place here. Defects are logged and tracked for its closure. Progress reporting is also a major part of this phase. The traceability metrics are updated to check the coverage and risk mitigated.

* Artifacts produced: Test results, Test logs, defect report, test summary report, and updated traceability matrices.

**User Acceptance Testing:** Acceptance testing is basically related to business requirements testing. Here testing is done to validate that the business requirements are met in the user environment. Compatibility testing and sometimes non-functional testing (Load, stress, and volume) testing are also done in this phase.

* Artifacts produced: UAT results, Updated Business coverage matrices.

**When to use the V Model?**

V model is applicable when:

* The requirement is well defined and not ambiguous
* Acceptance criteria are well defined.
* Project is short to medium in size.
* Technology and tools used are not dynamic.