What is Software Test Estimation?

Test Estimation is a management activity which approximates how long a Task would take to complete. Estimating effort for the test is one of the major and important tasks in Test Management.

Ho we do Estimation-

- Resources
- Time
- Human Skills
- Cost

- **Resources:** Resources are required to **carry out** any project tasks. They can be people, equipment, facilities, funding, or anything else capable of definition required for the completion of a project activity.
- **Time :** Time is the most valuable resource in a project. Every project has a deadline to delivery.
- **Human Skills :** Human skills mean the **knowledge** and the **experience** of the Team members. They affect to your estimation. For example, a team, whose members have low testing skills, will take more time to finish the project than the one which has high testing skills.
- **Cost:** Cost is the project **budget**. Generally speaking, it means **how much money** it takes to finish the project.

How to Estimate?

Step 1) Divide the whole project task into subtasks

Task is a piece of work that has been given to someone. To do this, you can use the Work Breakdown Structure technique.

The modules are divided into sub-modules. Each sub-module is further divided into functionality. It means divide the whole project task into the smallest tasks.

Step 2) Allocate each task to team member

In this step, each task is assigned to the appropriate member in the project team.

Task	Members
Analyze software requirement specification	All the members
Create the test specification	Tester/Test Analyst
Build up the test environment	Test Administrator
Execute the test cases	Tester, Test Administrator
Report defects	Tester

Step 3) Effort Estimation For Tasks

Total Effort = Total Function * Estimate defined per Functions Point

- **Total Effort:** The effort to completely test all the functions of the website.
- **Total Function Points:** Total modules of the website.
- **Estimate defined per Function Points:** The average effort to complete one function points. This value depends on the productivity of the member who will take in charge this task.

Suppose your project team has estimated defined per Function Points of **5 hours/points**.

Step 4) Validate the estimation

Once you create an aggregate estimate for all the tasks. You need to forward it to the management board, who will review and approve it.

The management board will review and discuss your estimation plan with you. You may explain them your estimation logically and reasonably so that they can approve your estimation plan.

Test Plan

What is Test Plan?

A test plan is a technical documentation which details a systematic approach to testing a specific system such as a device, machine or software.

It is a complete planning document containing the object, scope, approach, resources, strategy, risks, work schedule, etc. of all test activities.

The test plan contains a detailed understanding of the workflow and functions of the system and documents how each of those will be tested in order to find out if the system works according to its design, to find bugs, and to determine its actual limitations.

This document is important for determining whether a system or product will be able to meet quality standards before being produced or deployed.

Types of Test Plan-

Manufacturing or production test plan - for preparing a product for assembly or manufacturing, determining its fitness and for verification and quality control.

Regression test plan - usually made for an ongoing development or already-released product to determine if no functionality has been broken or bugs introduced after further development or product software update or upgrade.

Compliance test plan - for verification of a conceptual product or prototype to determine if it can comply with standards before further development

Acceptance test plan - for the test performed at delivery or deployment of the product, especially complex systems, to find out if everything works as it should after installation.

Types of Test Documentation

Test Policy

It is a high-level document describing the principles, methods and important testing goals of the organization.

Test Strategy

It is a high-level document that spots the test levels that need to be executed for the project.

Test Plan

It is a complete planning document containing the object, scope, approach, resources, strategy, risks, work schedule, etc. of all test activities. Everything that a tester needs to do in the project is made available here. Typically, a test plan is drawn up by experienced testing professional.

Requirements Matrix

It is a document that connects the requirements to the test cases for configuration testing as well as other types of testing.

Test Scenario

It is an item or event of a software system that is verified by one or more test cases.

Use Case

It is a less official document which is based on assumptions about what a user will do and which button they will click, thereby allowing testers to check the user paths. Use cases are created in order to address business requirements and objectives.

Test case

It is a document containing a highly detailed and specific description of the steps that a QA engineer needs to perform to test one portion of functionality, comprising a group of input values, execution preconditions, expected execution postconditions and results. It is developed for a test scenario.

Test Data

It exists before a test is executed and is used to execute the test case.

Test data is a commonly used term in a tester's day to day life.

Test Summary Report

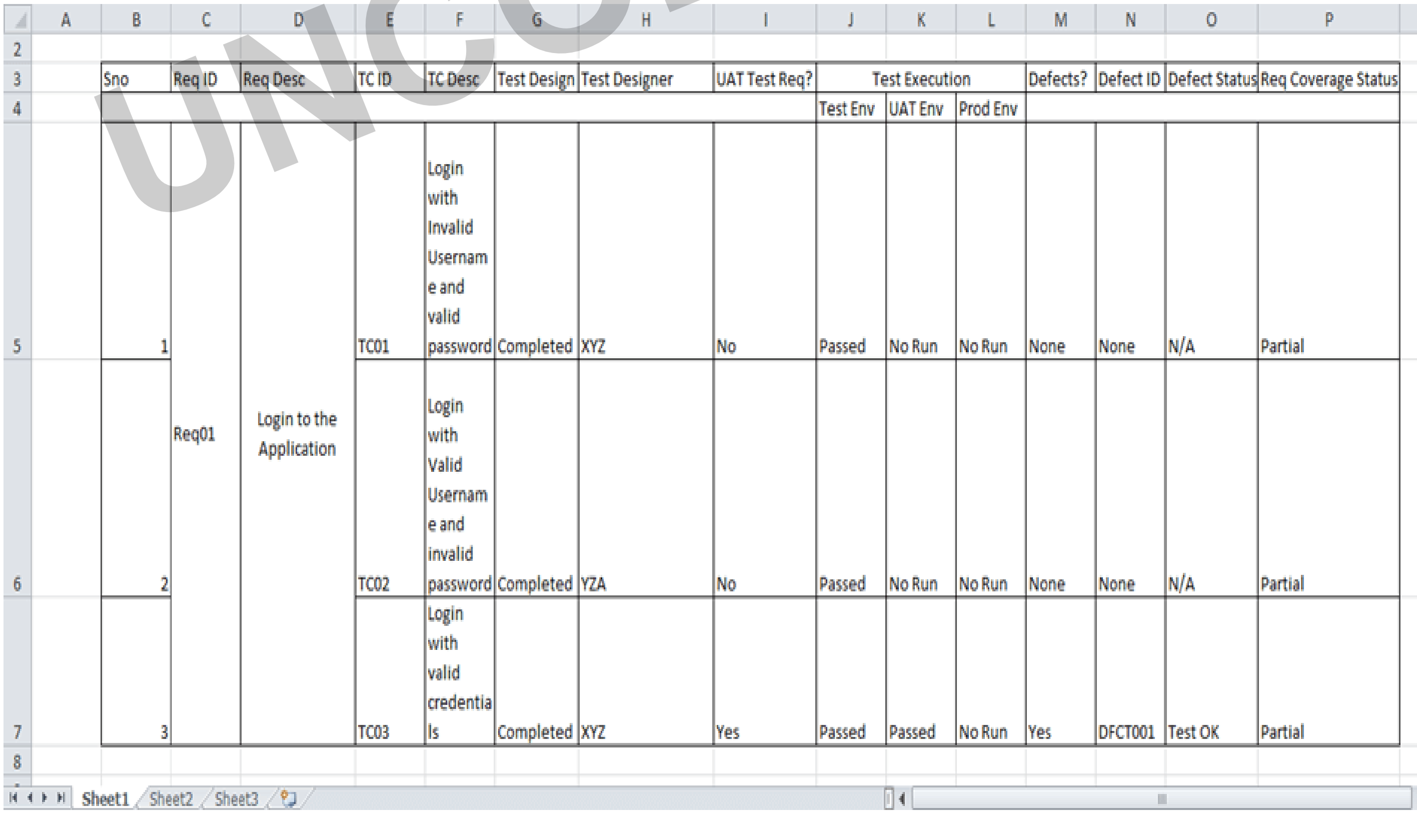
It is a high-level document that summarizes test activities and the test results.

RTM

What is RTM?

Requirement Traceability Matrix (RTM) is used to trace the requirements to the tests that are needed to verify whether the requirements are fulfilled**.**

It's used to prove that requirements have been fulfilled.



Sno	Req ID	Req Desc	TC ID	TC Desc	Test Design	Test Designer	UAT Test Req?	Test Execution			Defects?	Defect ID	Defect Status	Req Coverage Status
								Test Env	UAT Env	Prod Env				
1	Req01	Login to the Application	TC01	Login with Invalid Username and valid password	Completed	XYZ	No	Passed	No Run	No Run	None	None	N/A	Partial
2			TC02	Login with Valid Username and invalid password	Completed	YZA	No	Passed	No Run	No Run	None	None	N/A	Partial
3			TC03	Login with valid credentials	Completed	XYZ	Yes	Passed	Passed	No Run	Yes	DFCT001	Test OK	Partial

Advantage of RTM

- 100% test coverage
- It allows to identify the missing functionality easily
- It allows identifying the test cases which needs to be updated in case of a change in requirement
- It is easy to track the overall test execution status

Types of Software Testing Traceability Matrix (RTM)

- Forward Traceability
- Backward Traceability
- Bi-Directional Traceability(Forward + Backward)

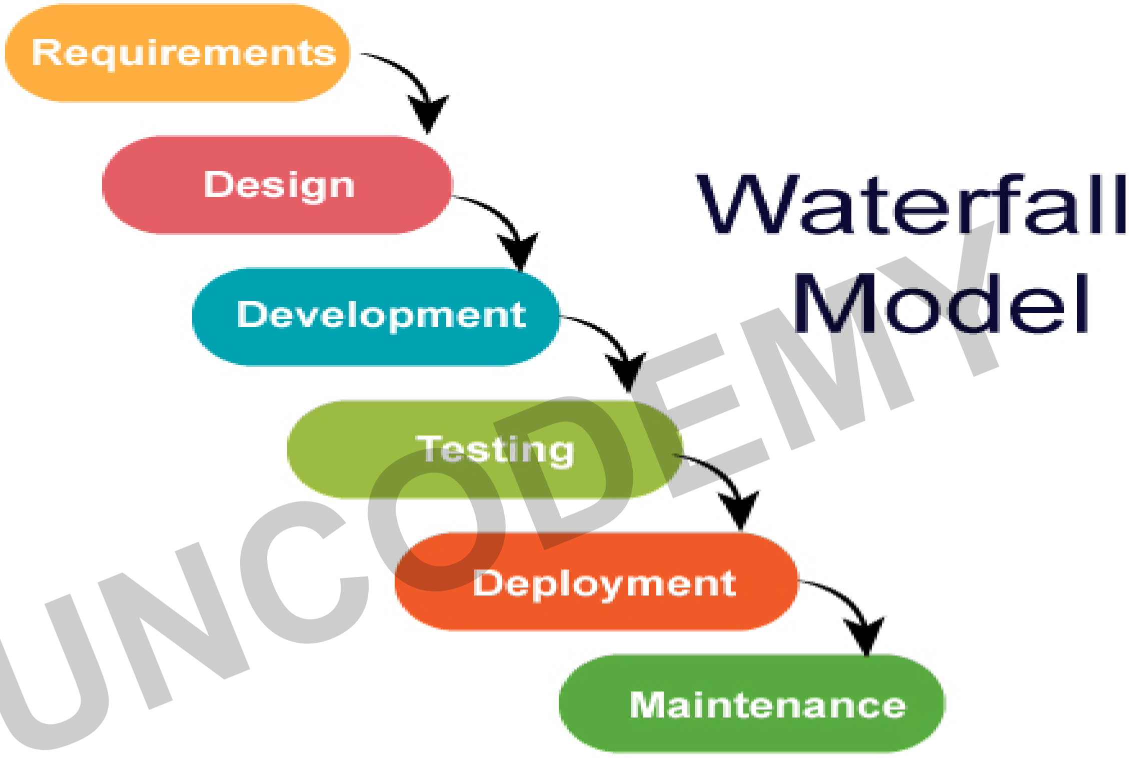
SDLC Models

- Waterfall Model
- V Model
- Agile Model
- Prototype Model
- Spiral Model
- Hybrid Model

Waterfall Model

1. The Waterfall Model was first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**.
2. It is very simple to understand and use.
3. In a waterfall model, each phase must be completed fully before the next phase can begin.

4. This type of software development model is basically used for the project which is small and there are no uncertain requirements.
5. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project.
6. In this model software testing starts only after the development is complete. In waterfall model phases do not overlap.



In the olden days, applications developed in Waterfall Model like CRM Systems, Supply Chain Management Systems etc. would usually take a year or longer to develop.

Advantages of Waterfall Model-

- This model is simple and easy to understand and use.
- It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
- In this model phases are processed and completed one at a time. Phases do not overlap.
- Waterfall model works well for smaller projects where requirements are clearly defined and very well understood.

Disadvantages of Waterfall model-

- Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing.

When we use waterfall-

- This model is used only when the requirements are very well known, clear and fixed.
- Product definition is stable.
- Technology is understood.
- There are no ambiguous requirements
- Ample resources with required expertise are available freely
- The project is short.

Important point

- In Waterfall model, very less customer interaction is involved during the development of the product. Once the product is ready then only it can be demonstrated to the end users.

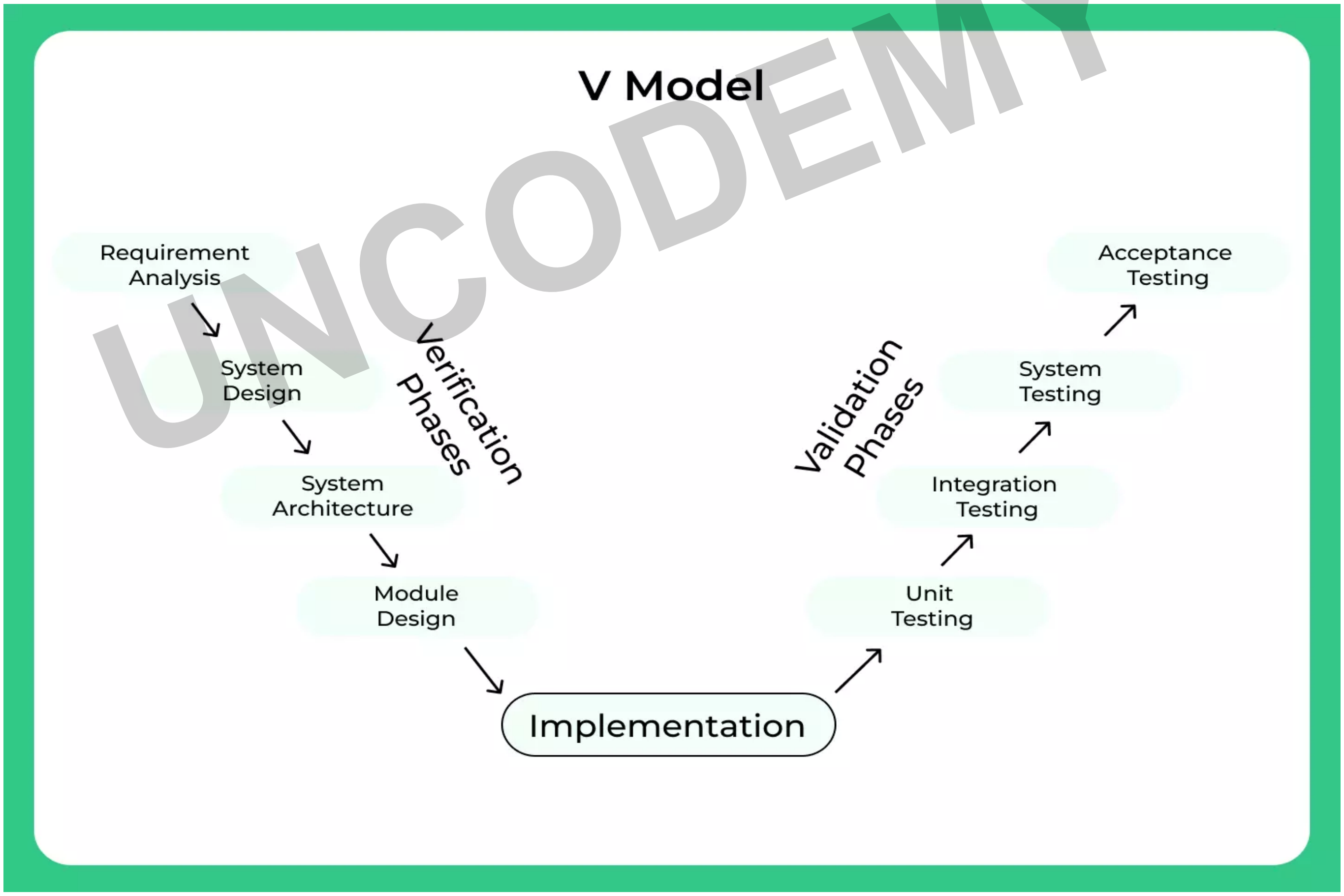
- Once the product is developed and if any failure occurs then the cost of fixing such issues is very high, because we need to update everything from document till the logic.

V model

There are two types of phases in V model:

- Verification Phase.
- Validation Phase.

We called it the V model because the diagram looks like a V, with the two legs of the V representing the verification and validation phases of the project, and the point at the bottom representing the start of the project.



Processes in Verification Phases

- **Requirement Analysis:**
  In this phase, Team collects information from customers about the software.

- **System Design:**
  When the requirements are defined clearly then implement and design the complete hardware and communication setup for developing product and choose the programming language and databases.

- **System Architecture:**
  Architectural specifications are designed in this phase. This is also called **High Level Design (HLD)**. In this stage, communication and transformation of data between the internal modules and the outer world is clearly specified.

- **Module Design:**
  In this phase the system breaks down into small modules. The detailed design of modules is specified, it is also called the **Low-Level Design (LLD)**.

- **Implementation/ Coding Phase:**
  This is the last phase of Verification phase of the V-Shape model. Module design is transformed into the code. The coding is done based on the coding principles and standards in a particular selected programming language.

Processes in Validation Phases

- **Unit Testing:**
  Unit testing is a type of white box testing. These Unit Test Plans are executed to remove bugs at code level. Unit Test Plans are created during the module design phase.

- **Integration Phase:**
  In the integration testing, the integration test cases are executed which were developed in the High-level design phase. Integration testing is a testing process in which unit tested modules are integrated and evaluated. It verifies that the modules work together as expected or not.

- **System Testing:**
  System testing is done corresponds with the system design phase. It tests the functional and non-functional requirements and evaluate the whole system functionality and the communication of the system with external systems.

- **Acceptance Testing:**
  This testing is done to check that the delivered system meets user's requirement or not? Nonfunctional testing such as Load, Stress etc. are also done in this phase.

Advantages of V model-

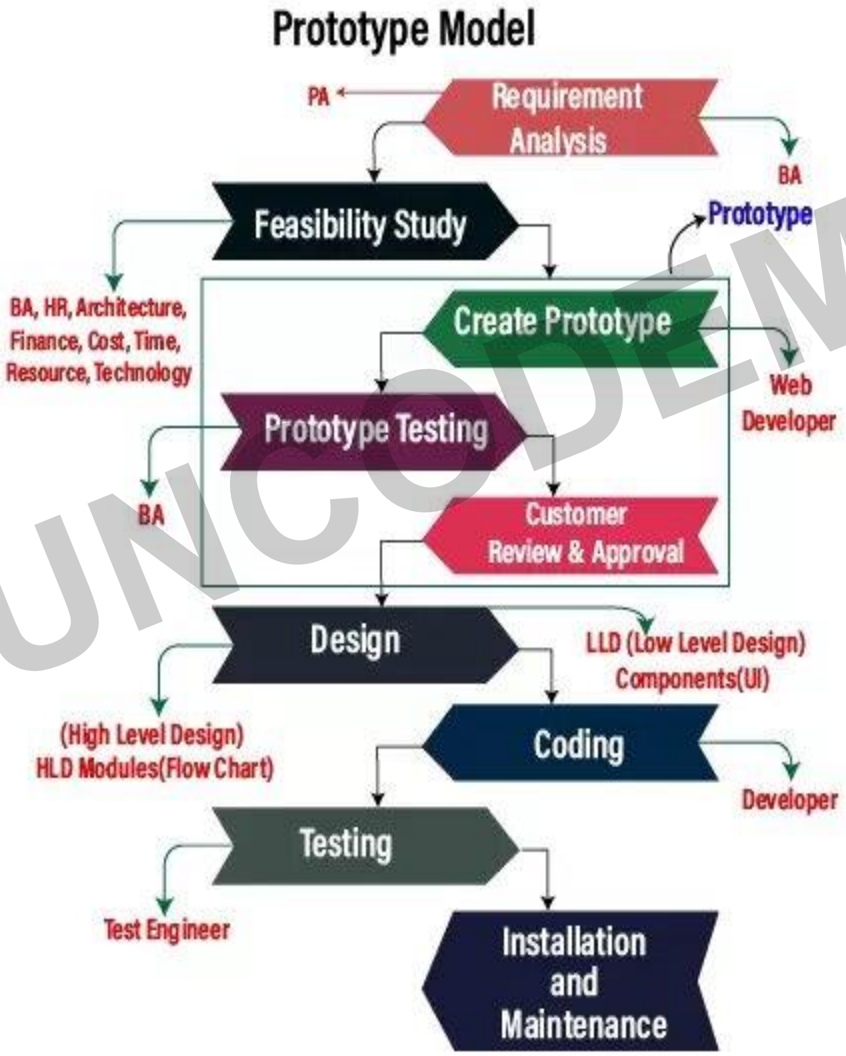
- It works very well for small project according to their requirement.
- This model is very simple, easy and useful.
- This is a high-quality model and all the phases are completed at once.
- This model is use to track the process of project management.
- This model saves a lot of time and efforts.
- Testing is starting at the initial phase so there is no issue of bugs.

Disadvantages of V model-

- This model can not be use for large project.
- This model is not good if customer's requirements are not clear.
- There are lots of risk.
- This model is not easy for complex projects.
- This model contains less flexibility.

Prototype Model

- This methodology creates a working prototype of the program that is tested and improved until it is ready for usage.
- Customers engage with these prototypes and provide input, which is used to guide testing and refinement.
- The customer’s engagement with the prototype helps them better understand the requirements of the needed system and gives them a sense of the quality of the design.
- A prototype model lacks many necessary details and is not a fully functioning system.


Prototype Model
PA
Requirement Analysis
BA
Feasibility Study
Prototype
BA, HR, Architecture, Finance, Cost, Time, Resource, Technology
Create Prototype
Web Developer
Prototype Testing
BA
Customer Review & Approval
Design
LLD (Low Level Design) Components(UI)
(High Level Design) HLD Modules(Flow Chart)
Coding
Developer
Testing
Test Engineer
Installation and Maintenance

Advantages of Prototype model-

1. **Active involvement**-
   This is one of the most prominent advantages of prototype model. With this approach, consumers are actively participating in the development process, making it simpler to tailor the model to their preferences. Due to the users' active participation, problems are found early on, simplifying the procedure.

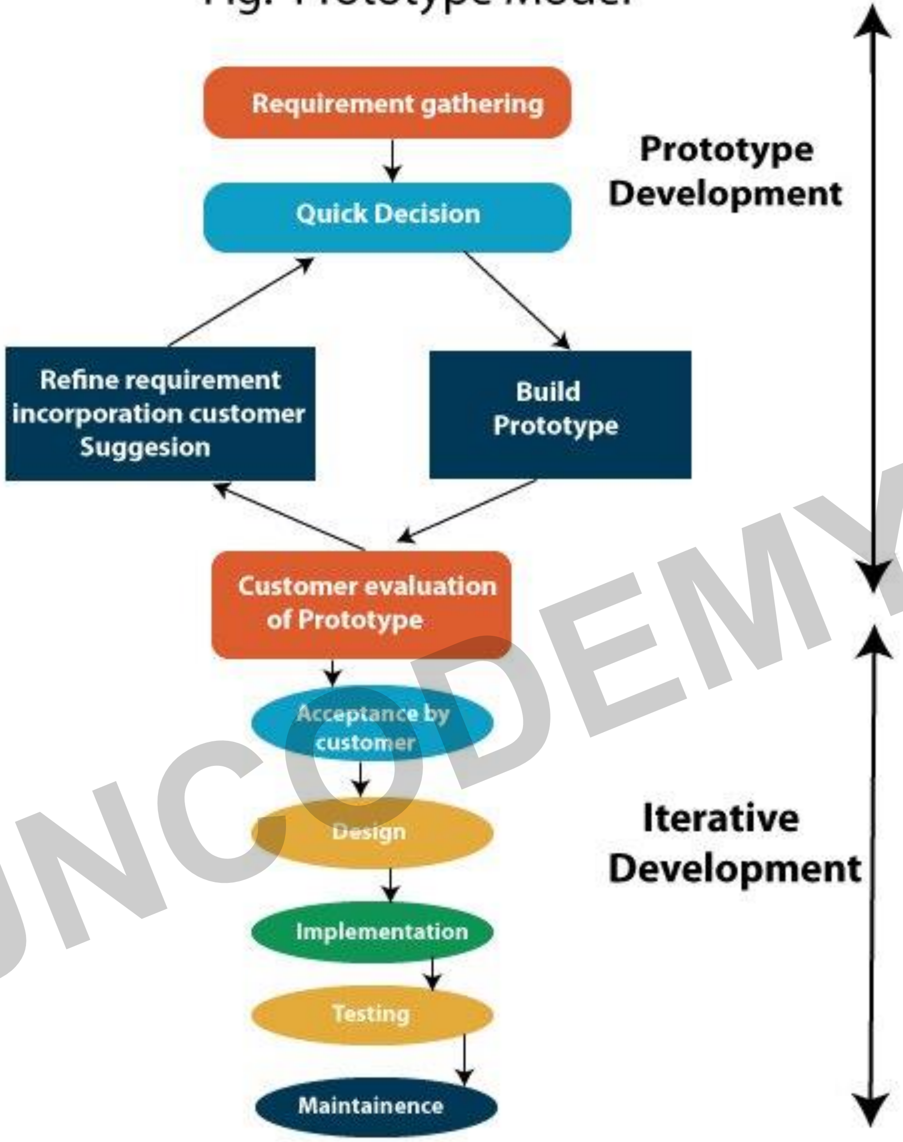
2. **Easy detection of missing functionality-**
   The prototype model's lacking functionality is clearly discernible. The chances of failure are decreased as a result. Additionally, confusing or challenging functions might be found.

3. **Quick Feedback-**
   Feedback from customers is provided much more quickly since they may engage directly with the prototype model. Since they are taken into account while developing the final system, these comments are crucial. Customers may rapidly offer their opinions and indicate adjustments that are necessary for the project, and the developers can subsequently adjust the project as required.

4. **Customer Satisfaction-**
   The prototype model offers much higher levels of client satisfaction. Early on, the consumer has the opportunity to touch and feel the product, which helps them better comprehend its requirements.

Disadvantages of Prototype-

Fig: Prototype Model



- **Time Consuming-**
  Timing is one of the most noticeable disadvantages of prototype model. The creation of the prototype model takes a lot of time. Multiple prototypes are tested before the final product is developed, which takes a lot of time.

- **High Upfront cost-**
  Using a prototype model throughout the last phases of development can help you save money. However, there are up-front expenditures associated with creating a prototype

model. Additionally, since there's a potential that the entire prototype would be discarded, the money spent on producing it can be utterly wasted.

- **Insufficient analysis-**
  There is a potential that the developer may focus on a particular prototype and neglect to do a thorough evaluation of the entire project. As a consequence, the developer may miss better options, forget about important details, and the project as a whole could be poorly designed, necessitating difficult maintenance.

InShort Summary of Advantages:

- Good when requirements/problem not understood.
- Problems detected/corrected early.
- Requirements refined and validated.
- Good for interface design.

InShort Summary of Dis-Advantages:

- Hard to know when to stop.
- Costly.

Hybrid Model

Hybrid Model is a model which is developed by combining two traditional models of SDLC.

The base models can be anyone like a spiral model, V&V model, prototype model, etc.

V & Prototype model-

V Model: In this model, V&V stands for Verification and validation, and in this model, both validation and verification of the system go side by side.

Prototype Model: A beautiful model where a prototype is built before developing the actual design of the system. The designed prototype is sent for testing, design, and customer review.

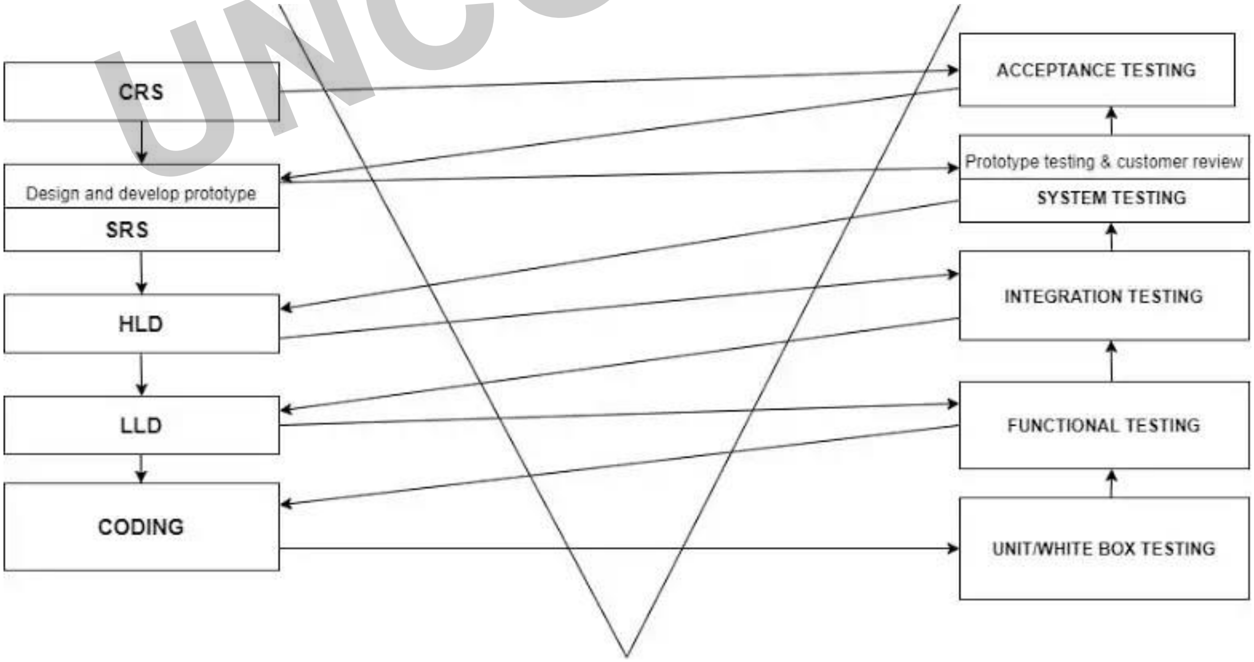
Hybrid model is used when:

The user wants parallel working of the verification and validation process.

Good documentation is needed.

The customer expects high-quality products with lots of testing.

There is no issue of finance and the only requirement is a successful product.



Customer Requirement Specifications (CRS):

Requirements are the basic needs for the development of software and by only requirement, we can plan its working, can decide the processing needs, coding language, team members, number of days to deliver, and much more. The requirements of the system are collected from customers which go for documentation and testing.

CRS document that contains simple, non-technical requirements and is sent to the tester where:

- Tester reviewed the CRS document.
- Write Acceptance Testing by preparing test cases and test plans.

SRS(Software Requirement Specifications) & Prototype Design:

SRS document developed in the previous stage is used to build a SRS document by Business Analyst which contains technical information for its understanding at the business level. SRS contains bar graphs, data flow diagrams, ER diagrams, etc. From the SRS web developers design and develop a prototype which is reviewed and send for testing. in testing testers:

- Review the document.
- Carry out System testing.

HLD:

After getting customer approved prototype it is sent to HLD, where its architecture is designed and understood by developers. Different technical approaches are involved to design the best feasible architecture which includes workflow, dependencies, functionalities of modules, database design, tables, and interface relationships. The designed HLD is passed to the next stage of testing, where testers:

- Review the HLD.
- Write Integration Testing documents and write test cases and test plans to execute the testing process.

LLD:

This phase is for designing the LLD document of the prototype. In this phase, the LLD document is prepared which contains detailed information about the working prototype like the type of programming language, compatibility of all modules, and functioning of modules. After designing the LLD it is sent for testing, where the tester:

- Review the LLD.
- Write functional Testing, test cases, and test plans to carry out the testing process.

Coding:

This phase includes coding of the particular prototypes by the developers using their coding language as directed by their team leaders, the coding must follow all coding standards, and guidelines.

The coded prototype goes for testing, where testers perform Unit Box testing, Integration Testing, System testing and Acceptance testing to find bugs and check extreme conditions of the module.

Advantages of Hybrid:

- The benefit of two models.
- High Output
- Satisfaction

Disadvantages:

- High Cost

White Box Testing:

It is a testing technique that mainly examines program structure and derives test data based on program logic or code. It also referred to names like precise box testing, open box, logic-driven, the path is driven, or structural Testing.

How does it work?

- The steps to perform this Testing is mentioned as follows in a specific order -
- Firstly, all features, components, and programs are to be tested and identified first.
- Create a flow graph and identify /plot all possible paths in the flow graph.
- Identification of all possible paths from the flow graph.
- Write test cases for every single path of the flow path.
- Execute, rinse and repeat test cases.

Three types of white box testing are:

- Unit Testing
- Integration Testing
- System Testing

The benefits of this are explained in the following manner -

- Required knowledge of the internals of the software under test to be tested.
- It allows finding hidden errors to find internal errors because it checks and works by internal functionality.
- It helps to find issues and optimize code to adopt different techniques to test a developed application or website.
- It requires internal knowledge to do Testing, which helps in maximum code coverage.

Why is it Important?

- It is executed at different levels such as system, integration, and unit level of software development.
- One primary goal of it is to verify the working of an application.
- It involves the identification of the operational flow of an application.

- It involves testing a series of predefined inputs against expected or desired outputs so that specific input does not result in the expected output. The bug was obtained, and work on those bugs.
- The functionality of conditional loops.
- Test each statement.
- Testing of each object.
- Test of function individually.
- Detection of Internal Security holes.
- Broken or poorly structured paths in a coding process.
- The flow of specific inputs through a code.
- It is executed in the early stages and does not wait for GUI(Graphical User Interface) availability.

Technique Coverage in White box Testing:

- Statement Coverage - It is a type of testing technique that tests all the defined programming statements.
- Branch Coverage - This type of testing technique ensures that all branches are tested at least once.
- Path Coverage - This testing technique tests all possible paths, covering each statement and branch.
- Statement Testing = (Number of Statements Exercised / Total Number of Statements) x 100 %
- Branch Testing = (Number of decisions outcomes tested / Total Number of decision Outcomes) x 100 %
- Path Coverage = (Number paths exercised / Total Number of paths in the program) x 100 %.

The formula above determines the number of test plans and cases executed for testing components or applications.