Manual Testing:

**SDLC: Software Development Life Cycle**

It’s a step-by-step procedure used to develop a software.

SDLC starts with a new concept and ends when the concept is over.

**Different types of MODELS/PROCEDURES**

1.WaterFall Model

2.Spriral Model

3.V-Model

4.Prototype Model

5.Derived Model

6.Hybrid Model

**1.Water Fall Model**

It is a sequential design process, often used in SDLC, in which the progress is seen as flowing steadily downwards ( like a waterfall ), through the different phases as shown in the figure,

REQUIREMENTS COLLECTION

FEASIBILITY STUDY / ANALYSIS

DESIGN

CODING / PROGRAMMING

TESTING

INSTALLATION

MAINTAINENCE

**Requirements Collection :-**

- done by Business Analysts and Product Analysts

- gathering requirements

- translates business language into software language

**For ex,** let us consider the example of a banking software.

**Feasibility Study :-**

- done by software team consisting of project managers, business analysts, architects, finance, HR, developers but not testers

- architect – is the person who tells whether the product can be developed and if yes, then which technology is best suited to develop it.

- here we check for,

- technical feasibility

- financial feasibility

- resource feasibility

**Design :-**

There are 2 stages in design,

HLD – High Level Design

LLD – Low Level Design

HLD – gives the architecture of the software product to be developed and is done by architects and senior developers

LLD – done by senior developers. It describes how each and every feature in the product should work and how every component should work. Here, only the design will be there and not the code.

**For ex,** let us consider the example of building a house.

**Coding / Programming :-**

- done by all developers – seniors, juniors, freshers

- this is the process where we start building the software and start writing the code for the product.

**Testing :-**

- done by test engineers

- it is the process of checking for all defects and rectifying it.

**Installation :-**

Insatllation:

Type1: centralized installation:- sr.dev installs in centralized server so that everyone can access it

Type2:Distributed installation:-done by Installation engineers: e.g ATM software

- done by installation engineers

- to install the product at a client’s place for using after the software has been developed and tested.

**For ex,** consider the example of a software to be developed and installed at Reliance petrol bunk.

**Maintenance :-**

- here as the customer uses the product, he finds certain bugs and defects and sends the product back for error correction and bug fixing.

- bug fixing takes place

- minor changes like adding, deleting or modifying any small feature in the software product

100 % testing is not possible – because, the way testers test the product is different from the way customers use the product.

**Advantages:**

1.Implementation of this process is simple

**Drawbacks:**

1. Testing is not done at each and every stage bcoz of which there is easy downward flow of the defect

2.Req.changes can’t be handled easily bcoz of which it is less customer friendly model

Note: In WF model, after freezing a particular stage then only we can move to the next stage.

\*It can handle those projects where there is no changes are to be made.

\*It’s a mother of all the models

**2.Sprial Model:** Here the development is carried out in an iterative manner. We use spiral model whenever there is dependency in the development of a s/w.

**DESIGN REQUIREMENTS COLLECTION**

Ra Rb Rc

Dc Db Da Rd

Cc Cb Ca Ta Tb Tc

**CODING TESTING**

For spiral model, the best example that we can consider is the MS-Excel application.

The MS-Excel sheet consists of a number of cells that are the components of Excel sheet.

Here we have to create the cells first (module A). Then we can do operations on the cells like merge cells into two , split cell into half (module B ). Then we can draw graphs on the excel sheet (module C).

**There are 2 types of req. changes**

**1.Major Change :** We carry separate iteration if there are any major changes.

**2.Minor Change:** We don’t follow separate iteration if any minor changes are there, we implement the minor changes in the current iteration only.

**Advantages**

1. Req. changes can be handled easily
2. Testing is carried out in each and every stage
3. More client interaction

**Dis advantages**:

1.Time consuming

**3.V-Model/velocity model/victricity model:** Here development and Testing is carried out parallely

This model came up in order to overcome the drawback of waterfall model – here testing starts from the requirement stage itself.

**1) In the first stage**, the client send the CRS both to developers and testers. The developers translate the CRS to the SRS.

The testers do the following tests on CRS,

1. Review CRS

a. conflicts in the requirements

b. missing requirements

c. wrong requirements

2. Write Acceptance Test plan

3. Write Acceptance Test cases

The testing team reviews the CRS and identifies mistakes and defects and send it to the development team for correcting the bugs. The development updates the CRS and continues developing SRS simultaneously.

**2 ) In the next stage,** the SRS is sent to the testing team for review and the developers start building the HLD of the product. The testers do the following tests on SRS,

1. Review SRS against CRS

a. every CRS is converted to SRS

b. CRS not converted properly to SRS

2. Write System Test plan

3. Write System Test case

The testing team reviews every detail of the SRS if the CRS has been converted properly to SRS.

**3 ) In the next stage,** the developers start building the LLD of the product. The testers do the following tests on HLD,

1. Review HLD

2. Write Integration test plan

3. Write Integration test case

**4 ) In the next stage,** the developers start with the coding of the product. The testing team carries out the following tasks,

1. Review LLD

2. Write Functional test plan

3. Write Functional Test case

After coding, the developers themselves carry out **unit testing** or **also known as white box testing.** Here the developers check each and every line of code and if the code is correct. After white-box testing, the s/w product is sent to the testing team which tests the s/w product and carries out functional testing, integration testing, system testing and acceptance testing and finally deliver the product to the client.

**ACCEPTANCE TESTING**

**CRS**

**SYSTEM TESTING**

**SRS**

**HLD**

**INTEGRATION TESTING**

**LLD**

**FUNCTIONAL TESTING**

**Testing team tests the finished product**

**UNIT TESTING or WHITE-BOX TESTING**

**CODING**

**DEVELOPERS TESTERS**

**CRS: Customer requirement specifications**

**SRS: Software requirement specifications**

**How to handle requirement changes in V&V :-**

Whenever there is change in requirement, the same procedure continues and the documents will be updated.

**Advantages**

1.Testing is done at each and every stage bcoz of which there is less chances of downward flow of the defect

2.Req. changes can be handled easily

3.At the end we get a quality s/w

**Disadvantages**

1.Implementation of this model is complex

2.Experience is required to use/work with this model

3.its difficult for freshers to work with this model

Note: It handles long term project

**How the req. changes can be handled in v-model?**

If a dev team is of 60 people, 40 people work where there are no requirement changes, and 20 people work on the req. changes

**4.Prototype Model:** It’s a dummy application which is functionally unstable. It’s mainly used to improve communication b/w customer and developer

**DEFECTS AND CHANGES**

**MAINTAINENCE**

**INSTALLATION**

**TESTING**

**CODING**

**DESIGN**

**PROTOYPE TESTING**

**REQUIREMENTS COLLECTION**

**DESIGN AND DEVELOPMENT OF PROTOTYPE(Done by Content developer and Web Designer**

**CUSTOMER REVIEW**

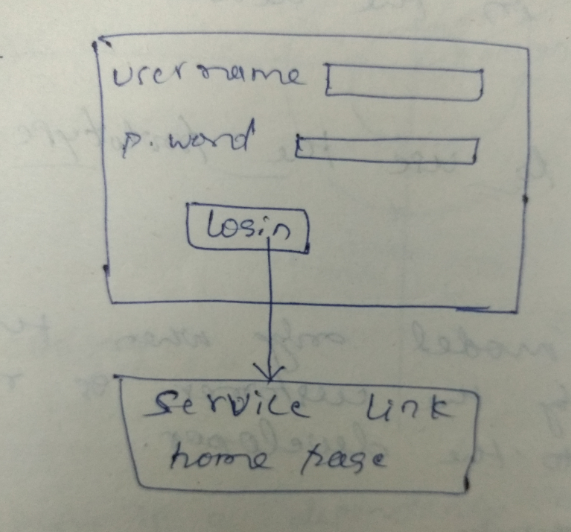
**APPROVAL**

**2 types of prototypes**

**1.Dynamic Prototype: We make use of the tools like dream viewer, visual studio, eclipse to develop dynamic prototype**

**Dynamic Prototype –** similar to a browser, but we can’t enter any information. Only the features are available without entering data. It’s like a dummy page, made out of HTML with tags and links to different pages representing features of the project

**E.g**



**2. Static prototype**

Here we put the images or snapshots in the word document and explain to the customer

**Static Prototype –** entire prototype of the requirement is stored in a word document with explanation and snapshots and instructions on how to go about building the s/w, how the finished product will look like and its working etc.

**Advantages:**

1.The cost of re-work is less

2.It improves the communication b/w the customer and developer

3.customer will get the feel how the application would look like before it is developed

Disadvantages:

1. Initial investment is required
2. There is a delay in the actual application being developed

Note: Its not mandatory to use prototype model. We use this model when the req. given by the customer is not understandable to the developer

**5) Derived model or Customized model** – we can take any of the above 4 models and change it as per business needs and requirements

**6.Hybrid Model:**

When there is a dependency and when the BA does not understand the req. properly from the client then we go with this model i.e when there is a communication problem b/w customer and BA

It combines 2 or more models and modify them as per business requirements.

**A) Hybrid model of Spiral and Prototype development models**

**We go for this model when,**

**1)** Whenever there is dependency, we go for this hybrid model

**2)** When the customer gives requirement in stages, we develop the product in stages using this hybrid model.

**3)** When the customer is new to the s/w domain

**4)** When developers are new to the domain

**5)** When customer is not clear about his own requirements

**Hybrid model of V&V and Prototype model**

**We go for this model when,**

**1)** Testing starts from early stages of product development which avoids downward flow of defects, thus reducing re-work.

**2)** When customer is expecting a very high quality product within stipulated time frame because every stage is tested and developers and testing team work in parallel.

**3)** When client and developers are both new to the domain

**4)** When customer is not clear about his own requirements

In this hybrid model, the testing team is involved in testing the prototype.

**ACCEPTANCE TESTING**

**WHITE-BOX TESTING OR UNIT TESTING**

**FUNCTIONAL TESTING**

**INTEGRATION TESTING**

**CODING**

**LLD**

**HLD**

**CLIENT REVIEWS IT AND APPROVES IT**

**CRS**

**SYSTEM TESTING, TEST PROTOTYPE, FIND DEFECTS AND SENDS IT BACK**

**SRS, DESIGN AND DEVELOPMENT OF PROTOTYPE**

**Changes**

**And**

**defects**

**DEVELOPERS TESTERS**

**Advantages and disadvantages:**

1.Same as spiral and prototype model