**What is Software?**

Consisting of Programs enable a computer to perform specific tasks

Automation of Manual Work

Make life easier / Reliable

**Why we need Software?**

Fast

Reliable

Perform Complex Calculation

**SDLC Concept**  
  
**What is SDLC?**

n A software development life cycle model represents all the activities and work products necessary to develop a software system. Life cycles models make the software development and dependencies visible and manageable.

Major advantages

n Control

n Accountability

n Error detection

Major drawbacks

n Relatively inflexible

n Time-consuming and expensive

n Discourages changes once user requirements are done

SDLC Phases  
  
**SDLC stages are:**

· **Identification:** decision making stage. You have to identify what you need to do. You have to have some kind of idea. Basically idea is generated here. Software development companies use this stage to investigate the market. They sit down and determine what is the present market needs. If they want to develop this software, how this will profit them?

· **Planning & Analysis**: Who is responsible for what part of the project. How you reach to your goal. Basically you need some planning here. Software develop companies use this stage to determine what the developer have to do to develop this software. They come up with some sort of plan and determine the budget to develop this software. Once budget is fixed, they move to the next phase.

· **Requirements:** What is your need? What the customer wants. Software development company will sit down in a meeting, with the client, and the BA (Business Analysis). Client will explain one by one his needs, and BA will write it down each requirement, After getting all requirement from client, BA will analyze the requirement. He will come up with the real requirement, which are valid to implement in software. Once requirements are done, they move to the next phase.

· **Design :** Based on the requirement, they make some sort of interface or design. System architect will design it. After design is done, now move to the next phase.

· **Development:** means coding. Developer will put code under the design. (also you can call construction phase or coding phase), Once the coding is done, now it is comes to the testing phase.

· **Testing:** In this stage developer starts the unit testing module by modules (component) in the application. Unit test is to test the single unit of the code. Then CM (configuration management) will combine all the modules together (which we call build) and deployed them in the test environment for tester to verified those module are integrated and functioning correctly by doing integration testing. Integration testing is whether all the module are combined together and working correctly as per the requirement document this testing is by writing SQL quire. Once tester done with integration test then tester will do smoke testing, to making sure the system stability and all the major functionality is not broken in the application, and determine that application is ready for further testing. Then tester will do the Functional testing, where they will check each and every function of the application is working as per the requirement. It’s a major testing, where 80% of the tests are done. After the functional testing, tester we will do all sort of testing, the most important are regression testing, system testing, Performance testing, Alpha testing, Beta testing. After all the testing is done and the application is proven to be good, now move to the implementation phase.

· **Implementation:** Here in this phase application is handed over to the Client, in order to find out, if the software meets the client expectations. Client will do the UAT (user acceptance testing) testing. Where client will have their own user to test it. If the client satisfies it, they approve it. And job is done. Now the software is moved to the next phase, which is maintenance.

· **Maintenance:** In this phase, as a client he will maintain it. He would hire his own developer, tester to maintain it.

Note: Every textbook has different names for the stages of the SDLC

**SDLC Methodologies**

**What is Methodologies:** Methodologies are like rules or process that you have to follow. And procedure to follow that rules.  
  
**Waterfall:** Water fall methodologies are the sequential execution of the software development and management process. In this stage the Requirement activity and analysis are all completed before the system design activity starts. The goal of this method is to **never turn back** once the Requirement activity and analysis is completed.  
  
**The Stage of the Waterfall Methodologies:**

Concept: Document system concept.

Requirement: Identify system requirements and analyze them before the design phase start.

Design: Break the system into pieces and design overall architecture.

Detailed Design: Design Each piece of code as per the requirement in Detail.

Code: Code the systems components debug and test them individually.

Test: Integrate the all the code or modules and test them.

Deploy: deploy the system.

**Advantage:**

q project object are clear

q project requirement are stable

q Progress of the systems is measurable

q Requirement are strictly signed off

**Disadvantage:**

q Time consuming

q Never backward

q Little room for iteration

q Difficulty response to changes

**Iterative:** **Iterative and Spiral:** is a cycle software development process. It’s developed in response to the weaknesses of the waterfall methodology. It starts with an initial planning and ends with deployment with the cycle interaction in between the phases. The basic idea behind iterative methodology is to develop a software system incrementally.  
  
**Advantage:**

q **The design phase goes much faster**

q **Coding and testing goes much faster,**

q The production in short period of time.

q Avoidness of risk enhancement

q Strong approval and documentation

q Implementation have priority over functionality

q Additional functionality can be edit at later date.

**Disadvantages:**

q **Highly customize limiting re-usability**

q Applied differently for each application

q **Risk of not meeting budget or schedule**

q Possibility to end up implemented at waterfall framework

**Agile**: Small group of people working together, they basically working parallel. Receiving client needs which are requirements. Than developer start doing the coding based on the requirement, on the other hand unit test begins after the coding is done. After the unit test comes to QA start doing the systems test (black box ) to find the bug if he or she find the bug send back to the developer to fix it , it goes on until the product is ready to release.  
  
Advantages  
  
Principles behind agile methodology:

· Customer satisfaction by rapid, and continuous delivery of useful software

· **Working software is delivered frequently**

· **Even late change requirement are welcome**

· Close, daily cooperation between business people and developers

· **Face to face conversation is the best for the communication**

n Disadvantages

**The Teams**

q Users

q Business analysts

q Programmers

q Quality Assurance/ Tester

q Configuration Management

q System stakeholders

**Software Testing Life Cycle**

**(STLC)**

q Requirement Analysis

q Develop Test Plan

q Develop Test Cases

q Execute Test Cases

q Analyze Test Results

q Submit Defect

**How to Analyze Requirement**

q Is this requirement testable?

q Is this requirement follow standard Construct?

q Is this requirement Un-Ambiguous?

q Is this requirement concise?

q Is this requirement Duplicate?

q Is this requirement Traceable

**How to Develop Test Plan?**

Test plan is the guide line of the test process.Test plan is a high level document describing

the activities that will be performed at each specific stage of testing. It basically

summarizes the software items and software features to be tested.

**Overview**

q Introduction

q Project Overview

q **Purpose:** I need to describe the main objectives and purpose of this test. Specifying the major functionality will be impacted in the new release.

**Scope :** I will be describing what particular part I need be testing. (Scope: the totality of work needed to complete a project.)

q **Functions/Features to be tested:** Defining the feature to be tested as part of this release and Identifying each requirement specification associated with each feature.

q **Functions/Features not to be tested:** Defining the feature not to be tested as part of this release and identifying any requirement specification associated with each feature and the reason for not validating.

q **Configuration Management (CM) :** Duty of the CM team is follows:

q They Perform release software builds and installation.

q They Perform monthly review and they maintained of company’s library Repositories

q They Conduct configuration status reporting and configuration audits after every release.

q They Attend and provide input for weekly internal and external change control board meeting (CCB).

q They Perform weekly review

q They Perform change management activity for release.

q **Training :** : if the tester required for trained for new stuff. Training for existing network which will integrate with my application. Also if I need to tested, what kind of tools I need to trained, like QTP, QC, I need to mention here in training section.

q **Status Report :** : Here I will describe what types of status will be reported, to whom, and at what intervals. For example: I will report to my test manage, weekly, the number of test cases completed.

**Approach :** I will be describing what kind of testing will be performed. For example,

q Unite testing:

q Integration testing:

q System testing: is the overall testing.

q UAT testing:

q Performance testing:

**Test Plan Cont’d**  
  
**Test Schedule and Deliverables**: I have to write the task that I’m responsible for it, and then plan it when I’m going to start, and then when I’m going to finish it. What is the task, I need to put it in here, and the rules and responsibility of doing the task.  
  
**Test Schedule:** Start date and End date of your testing, I will mention it here.  
  
**Entry Criteria:** Perquisite of testing.

q testing environment must be ready,

q Tools are setup and ready.

q test plan must be ready,

q Code is reviewed.

q Previous cycle must be completed.

**Exit Criteria:** all the checklists have done. Exit the testing phase of the program.

q All recreational level has been done.

q All test cases developed and have been successfully tested.

q All defect cycle has been reviewed and corrected.

q No severe defect, maybe some cosmetic defect is there.

**Test Deliverable:** You explain or brief summary of which drive or folder you will use to store your test plan, test cases, test data, test result. Mention the specific link of the folder/folders.  
  
Test Plan Cont’d  
  
**Test Infrastructure:** The total resource to build the application.  
  
**Test Resources Requirements**

q **Data:** Test data: The data you will be working to do the test, like: user: jojo, pass: bean. Some people like system administrator or somebody else will provide you test data.

q **Hardware:** Hardware you need.

q **Software:** Software you need.

q **Personnel:** Human resource you need.

q **Test tools:** what kind of testing tools you needed; QTP, QC, LR.

q **Third party vendor:** Supporting you needed for your application. Vendor usually works behind the scene. Example: for ebay, paypal is their vendor.

q **Test Environment:** The environment you need to set for your testing.

q **Interfaces:** How it will be look like

**Test Plan Cont’d**  
  
**Roles and Responsibilities :**  
  
we have to follow some kind of rules, based on the rules

q **Testing Roles:** you have put everybody’s roles and responsibilities. As a tester, what is your roles & responsibility?

q **Project team members:** Put the project team member name.

**Risk Assumptions, Dependencies**

q **Risks:** What will be your risk issue to do this, you have to mention it here. (example: if the tester is not performing enough, project will be delayed)

q **Assumptions:** I’m assuming that I will need training& they will provide me some training. I think I won’t be able to finish it on time. What ever I assume that will happen or need to do.

q **Dependencies:** If this application is depended on any other application. Any Dependency.

**Test Cases**

q What's a 'test case'?

q A test case is a document that describes an input, action, or event and an expected response, to determine if a feature of an application is working correctly.

q A test case should contain particulars such as test case identifier, test case name, objective, test conditions/setup, input data requirements, steps, and expected results.

q Why Writing Test cases?

q Accountability

q Manageability

q Regression

**How to Develop and Execute Test Cases?**

q Test Case ID

q Test Case Description

q Prerequisite

q Input/Action

q Expected Results

q Actual Results

q Pass/Fail

q Date

q Author

Defect Life Cycle

q Tester Find the defect and Submit to the Team Lead (Status = New)

q Team Lead Verified the Defect and Submit to the Developer (Status = Open)

q Developer Fixed the Defect (Status=Fixed)

q Tester Validate the defect and Assigned to Team lead (Status=Verified)

q Team Lead will closed the Defect based on the Tester comments

Documents are helping you to write the test case.

q Design Document

q Use Case: A use case is a document that describes the user action and system response for a particular functionality.

Prototype : it’s the sample document for the company.

q Functional Requirement Specification (FRD) :

7. What is SDLC (Software Development Life Cycle)?  
  
Answer: SDLC (Software Development Life Cycle) is the process of developing software through business needs, analysis, design, implementation and maintenance. Software has to go through various phases before it is born which are as follows:  
  
(i)Generating a Concept – A concept comes from the users of the software. For example, a Pizza Hut may need software to sell pizza. An Indian store may need software to sell its newly arrived movies or grocery. The owner of the company feels that he needs software that would help him in tracking his expenses and income as well as enhance the selling process. This is how the concept is generated. The owner will specifically tell the software company what kind of software he would need. In other words, he will specify his requirements.  
  
(ii) Requirements analysis – After the owner (user) knows his requirements, then it is given to a software team (company) who will analyze the requirement and prepare requirement document that will explain every functionality that are needed by the owner. The requirement document will be the main document for developers, testers and database administrators. In other words, this is the main document that will be referred by everyone. After the requirement documents, other detailed documents many be needed. For example, the architectural design which is a blueprint for the design with the necessary specifications for the hardware, software, people and data resources.  
  
(iii) Development: After the detailed requirement documents (some companies have design documents instead of requirement documents), the developers start writing their code (program) for their modules. On the other hand, the testers in the QA (Quality Assurance) Department start writing Test Plans (one module=1 test plan), test cases and get ready for testing.  
  
(iv) Testing: Once the code (programs) are ready, they are compiled together and to make a build. This build is now tested by the software testers (QA Testers)  
  
(v) Production: After testing, the application (software) goes into production (meaning, it will be handed over to the owner).  
  
(vi) End: And one day, the owner will have say bye to the software either because the business grows and this software does not meet the demand or for some reason, the he does not need the software. That’s the end of it.