**SDLC**

SDLC- Software Development Life Cycle

How to develop software based on client requirement within time and cost.

Process to develop software step by step.

Software—program or collection of program set of instructions that we give to computer. Allows user to interact with computer, computer hardware or programs some tasks.

eg. Calculator

Building

Analysis - product owner u can decide the plan

Design – blueprint of building

Development – the building/product

Testing – quality

Maintenance -

We cannot directly go to build software--- eg. buildling..

SDLC is a framework (skeleton/protocol/rules). It defines what all tasks you should perform to develop and maintain the software.

* **Planning and Requirement Analysis:-**

Here BA(Business Analysist) contact with client.

Discuss about:- budget /cost, time, requirement

If this thing is fine. BA give RFP(Request for proposal) to client.

1. It is like Preliminary analysis
2. Project manager-customer/Product owner/Business Analyst)/end client, sales department and other experts.--> SRS(Softeware Requirements Specification),BRD (Businness Requirement Document)
3. prepare rough document

* **Defining Requirements:-**

the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed during the project life cycle.

### **SRS (** Software Requirement Specification)

### Functional requirements

### Describes what the s/w will do and it will be expected to perform.

### Establishes the basis for an agreement between customers and company on what the s/w product is to do as the requirement is defined clearly so that you can redue later redesign.

### It should also provide estimating on product costs, risks.

### Help prevent s/w failure.

**BRD (**(Businness Requirement Document)

### what is required, rather than on how to achieve it. (Non functional requirements)

such as

**Productivity**( how much a system aids users?

Eg. Walmart---100 users/min-1 sec

**Capacity**(How much traffic must the system be able to handle? – more 100 users

Walmart,amazon(during sales—1000 users)

**Avaibility:-** your system is available in 24/7 or only weekdays..etc..

### Designing

### Bule print for the s/w.(all the dummy thing.How will application looks like..)

### all dia grams.( Use Case Diagram., Activity Diagram, Sequence Diagram.)

usecase dia- actors and their responsibilities

sequence dia- step by step

Class dia-defining variable methods or properties/actions for given behavior.

Activity dia-flow chart

### **Building or Developing**

Here developer will do coding.. using any language---php,java,c .net,.

### 

### **Testing** :- find the defect in product.

**Testing according to requirement specification document.**

* **Deployment:-**

Time to make software live..

### Git repository or local repository build that application and deploy it in a machine and host it from there to central repository.--code review by lead

Integration (Building) using Jenkins

Code review( -2(very bad),-1(bad),0(no change),+1(I have reaview..now I am not enough to review this code),+2(ready to build))

The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).

* Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

### **Maintenance:-**

### resolve issues/errors, enhance the product for better version.

REAL LIFE:-

In the initial phase project manager can prepare a document for the requirements, team leader will prepare a team which is having  test engineers, developer will provided by the project manager, tester will prepare test cases for that particular project

   Analysis phase all the members have a meeting to finalize the technology to develop that project, the employee, time ,...

   Designing phase the project manager like senior level management will give the directions and source code to the team members to develop the actual code,that is guidelines will be given in this phase

    Coding phase developer will develop  the actual code using the source code and they release the application to the tester

    Testing phase they deploy their test cases to that application and prepare a bug profile document if there is any defect/bug in that application and send it back to developer, developer may rectify and releases the application as next build and if the bug not understand it will send to the project lead   in the delivery phase the sr test eng can deploy the application in the client environment

     Maintenance phase if the client get any problem with the application it may solved by the project lead with help of testers and developers

most important and popular SDLC models

* Waterfall Model
* Iterative Model
* Spiral Model
* V-Model

**Waterfall Model**

First Process Model

Linear- sequential life cycle model.

* Requirement Gathering and analysis − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
* System Design − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
* Implementation − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
* Integration and Testing − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
* Deployment of system − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
* Maintenance − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

**When we can use Waterfall Model?**

* Requirements are very well documented, clear and fixed.
* Product definition is stable.
* Technology is understood and is not dynamic.
* The project is short.

**Advantages:-**

* Simple and easy to understand and use
* Phases are processed and completed one at a time.
* Works well for smaller projects where requirements are very well understood.
* Clearly defined stages.
* Well understood milestones.
* Easy to arrange tasks.
* Process and results are well documented.

**Disadvantages of the Waterfall Model:-**

* 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. So, risk and uncertainty is high with this process model.
* It is difficult to measure progress within stages.

**Incremental Model**

In this incremental model, the whole requirement is divided into various builds. During each iteration, the development module goes through the requirements, design, implementation and testing phases. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is ready as per the requirement.

**When to use this model:-**

* Requirements of the complete system are clearly defined and understood.
* Major requirements must be defined; however, some functionalities or requested enhancements may evolve with time

**Advantages:-**

* Results are obtained early and periodically.
* Parallel development can be planned.
* Progress can be measured.
* It supports changing requirements.
* Initial Operating time is less.
* Better suited for large and mission-critical projects.

**Disadvantages :-**

More resources may be required. ( Consider the use the following resources in software development:

1. Time
2. Human Resources – the most crucial of all the resources
3. Computer Resources
4. Money

* )
* Although cost of change is lesser, but it is not very suitable for changing requirements.
* More management attention is required.
* System architecture or design issues may arise because not all requirements are gathered in the beginning of the entire life cycle.

**Spiral Model**

The spiral model has four phases. A software project repeatedly passes through these phases in iterations called Spirals.

### **Design**

The Design phase starts with the conceptual design in the baseline spiral and involves architectural design, logical design of modules, physical product design and the final design in the subsequent spirals.

**V- Model**

The V-model is an SDLC model where execution of processes happens in a sequential manner in a V-shape. It is also known as **Verification and Validation model**.

The V-Model is an extension of the waterfall model and is based on the association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle, there is a directly associated testing phase. This is a highly-disciplined model and the next phase starts only after completion of the previous phase.

**Agile**

It is iterative approach to s/w delivery that builds s/w incrementally from the start of the project instead of trying deliver it all at once near the end.

It works by breaking projects down into little bits of user functionality called user stories. User Stories defines the actual business requirement. Generally created by Business owner.

It works by breaking projects down into little bits of user functionality called user stories. Prioritizing them and ten continuously delivering them in short two week cycles called iterations.

1. Planning meeting:-

Product owner selects a user story based on the priority from the product backlog. Based on the discussion, the scrum team decides the complexity of the story and sizes it as per the Fibonacci series.

2. Daily scrum meeting:- 15 min(beginning of the day)

1. What did the team member do yesterday?
2. What we are going to plan to do today?

3. Review meeting:-

* At the end of the every spring cycle.
* Scrum team meets again and demonstrates implemented user stories to the product owner may cross verify the stories as per its acceptance criteria.

4. Retrospective meeting:-

* Happens after the review meeting.
* Scrum team meets and discusses

1. What went well during the sprint?
2. What did not went well in the sprint?
3. Lessons learnt.

Project Owner – who has the responsibility of managing product backlog. Works with end users and customers and provide proper requirement to the team to build the proper product.

Scrum Master is the facilitator of the scrum team. He / she make sure that the scrum team is productive and progressive.  who works with scrum team to make sure each sprint gets complete on time. Scrum master ensure proper work flow to the team.

Scrum Master key responsibilities involves

* Understand the requirements and turn them into working software
* Monitoring and Tracking
* Reporting and Communication
* Process Check Master
* Quality Master
* Resolve Impediments
* Resolve Conflicts
* Shield the team and performance feedback
* Lead all the meetings and resolve obstacles

Backlog is a collection of userstories.

* Product backlog: It contains a list of all desired features and is owned by the product owner
* Sprint backlog: It is a subset of the product backlog owned by development team and commits to deliver it in a sprint. It is created in Sprint Planning Meeting

Advantages:-

* Functionality can be developed rapidly and demonstrated.
* Resource requirements are minimum.
* Suitable for fixed or changing requirements
* Little or no planning required.
* Easy to manage.
* Gives flexibility to developers.

Burndown chart**:-** how much work left to do.

BurnUp chart**:-** tracks progress towards a projects completion