**SDLC (Software Development Life Cycle):**

SDLC is a procedure to develop the software application. It is consisting of different stages (or) phases.

There are different types of models in SDLC.

1. Waterfall Model
2. Spiral model (or) iterative model
3. V and V model (Verification & Validation)
4. Prototype model
5. Agile method (it is Currently Trending Model)

**1.Waterfall model:**

It is one of the oldest and traditional models of SDLC.

**We go for Waterfall Model:**

* When customer freezes the requirement
* For any short-term projects
* For developing simple applications

**Advantages:**

* We can expect stable applications (if Requirement doesn’t change)
* There will be no disturbance for the team members if the requirement doesn’t change

**Disadvantages:**

* Testing happens only after coding. So, requirement and design are not tested.
* Developers used to do testing before (currently it is done by test engineers).
* If requirement changes, it leads to lot of Rework.

**2.Spiral model (or) iterative model:**

It is one of the oldest and traditional models of SDLC after Waterfall Model. In this model **customer will check the Application module by module (or) Part by Part**. Once 1st module is ready customer will check that module if he will be happy then only, he will give another module to develop the application. If he is not happy, he will discontinue the project.

**We go for this Model because:**

* Customer can see the Application part by part (or) module by module.
* when there is lot of dependency between the modules

**Advantages:**

* Customer see the application partially & he can get confidence
* Requirement changes can be done
* Compare to waterfall model, **Rework is less.**

**Disadvantages:**

* Testing happens only after coding. So, requirement and design are not tested.
* Developers used to do testing before (currently it is done by test engineers).
* If there is any requirement changes, it may delay the projects.
* Customer can cancel the project if he is unhappy.

**3.V and V model (Verification & Validation):**

It means verification and validation model.

**Verification** is a process of checking "are we building product right".

**Validation** is a process of checking "are we building right product".

**We go for V & V Model:**

* When customer needs High quality product.
* For complex applications.
* For long-term projects (more than1 year).

**Advantages:**

* Testing is started at initial stage (i.e., document check).
* Requirement and design are tested.
* The downward flow of defects is less.
* Requirement changes can be done.
* Quality will be high compared to other models
* Rework will be less

**Disadvantages:**

* Documentation work will be more.
* Too much resources are needed.

**4. Prototype Model:**

Prototype is a Dummy Model that is a non-working application.

**We go for Prototype model:**

* When Customer is Confused about Is own Requirement (or) he is not clear about the requirement.
* When developers are new to the Technology
* If Software company is new to the Business
* If Customer is new to the Business
* For Any kind of Experiments this is the best Model

**Advantages:**

* Initial investment is less.
* Customer can see the application very fast (Dummy application)
* Customer can change the requirement initially itself
* Developers will get a clarity in the initial days only.

**Disadvantages:**

* There will be delay in the actual start of the real project
* Investment is done on dummy product
* Too many changes can disturbs the rhythm of the Project.

**5.Agile Model:**

It is a Software Development Model alternate to other models like Waterfall, Spiral, V and V etc.

**Different Technologies of Agile Model:**

1. Scrum
2. XP (Extreme Programming)
3. Crystal Methodologies
4. DSDM (Dynamic Software Development Method)
5. FDD (Feature Driven Development)
6. RUP (Rational Unified Process)
7. RAD (Rapid Application Development)

**We go for Agile Model:**

When customer needs application very fast within less time with good quality.

**1.Scrum:**

Scrum is one of the most popular frameworks for implementing agile.

Scrum is an Agile Methodology in which the project is broken into small pieces or increments called **sprints** where the expected outcome from each sprint is a working software.

**Scrum Elements:**

1. Roles
2. Artifacts
3. Meetings

**Roles:**

There are 3 main roles in a scrum methodology

1. Product Owner
2. Scrum Development Team
3. Scrum Master

**1.Product Owner:**

The Scrum product owner is typically a project's key **stakeholder**. Part of the product owner responsibilities is to have a vision of what he or she wishes to build, and convey that vision to the scrum team.

**Scrum Master:**

The Scrum Master is responsible for making sure a Scrum team lives by the values and practices of Scrum. The Scrum Master can also be thought of as a **process owner** for the team, creating a balance with the project's **product owner**.

The Scrum Master does anything possible to help the team perform at their highest level. This involves removing any **impediments** to progress, facilitating meetings, and doing things like working with the product owner to make sure the product is in good shape and ready for the next **sprint**.

**Scrum Development team:**

A Scrum team in a Scrum environment does not include any of the traditional software engineering roles such as programmer, designer, tester or architect. Everyone on the project works together to complete the set of work they have collectively committed to complete within a **sprint**.

**1.Product Backlog Meeting:**

In this Meeting We will divide the Project into High Level Priorities and Low-level Priorities to work. Product Backlog items are prioritized before we start the sprint cycles.

**2.Sprint Backlog:**

Everything that we have agreed to complete in the current sprint is called as Sprint Backlog. The sprint backlog is a list of tasks identified by the Scrum team to be completed during the Scrum sprint. During the sprint planning meeting, the team selects some of product backlog items, usually in the form of user stories, and identifies the tasks necessary to complete each user story.

**Meetings:**

1. Sprint Planning Meeting
2. Daily Scrum
3. Sprint Review Meeting
4. Sprint Retrospective Meeting
5. Backlog Refinement Meeting

**1.Sprint Planning Meeting:**

It is conducted at the beginning of each sprint. Its duration is typically 2-4 hours. During the sprint planning meeting, the product owner describes the highest priority features to the team. The team asks enough questions that they can turn a high-level user story of the product backlog into the more detailed tasks of the sprint backlog.

**2.Daily Scrum Meeting:**

It is conducted on each day of Sprint. Ideally, a daily scrum meeting is held in the morning, as it helps set the context for the day's work and discuss what was done on the previous day. These scrum meetings are strictly time-boxed to 15 minutes. This keeps the discussion brisk but relevant.

**3.Sprint Review Meeting:**

In Scrum, each sprint is required to deliver a potentially shippable product increment. This means that at the end of each sprint, the team has produced a coded, tested and usable piece of software. So, at the end of each sprint, a sprint review meeting is held. During this meeting, the Scrum team shows what they accomplished during the sprint.

During the sprint review, the project is assessed against the sprint goal determined during the sprint planning meeting.

**4.Sprint Retrospective Meeting:**

The Scrum team should set aside dedicated period at the end of each sprint to reflect on how they are doing and to find ways to improve. This occurs during the sprint retrospective. This meeting typically takes 1-2 hours. The sprint retrospective is the last thing done in a sprint. It can be done after sprint review meeting. The entire team, including both the Scrum Master and the product owner should be part of this meeting.

**5.Backlog Refinement Meeting:**

The Backlog Refinement Meeting is also called Product Backlog Grooming, Backlog Estimation, and even Story Time. The purpose of the Backlog Refinement Meeting is to help the Product Owner get the top of the Product Backlog ready for the next Sprint Planning Meeting.

Backlog refinement includes estimation of effort, clarification of requirements, and decomposition of large Product Backlog Items (often called “epics”) into smaller ones (such as “user stories”).

---------------------------------------------- \*\*\*\*\* -------------------------------------------------------------**STLC (Software Testing Life Cycle):**

It is a procedure to test the application. It is consisting of different stages (or) phases.

**1.System Study:**

Going through the requirement & understanding how the software system works is called as “System Study”

**2.Write/Prepare Test Plan:** it is a document which is prepared by manager for future test activities. Since Manager is having experience, he is the right person to prepare test plan.

This plan has to be followed by Test Engineers.

**3.Write Test Cases:**

Write the Scenarios and Converted that to Test Cases. After writing that the test cases we do review & that will be stored in the **REPOSITORY (JIRA, QC/ALM etc….)**

**4.PrepareRTM (Requirement Traceability Matrix):**

It is a document which is prepared for to check whether every requirement has at least one test case or not. It helps us to cover all scenarios for each & every requirement. If anything is missing, we will come to know about it & we can write Test Case for that.

**5.Execute Test Cases:**

Once Application is given by Developers, we worked on Test Cases. When it is given to test that application, we will compare that application with our expected results and actual results & write that if it is matching the status is “pass”, if it is not matching the status is “failed”. If it is failed, we will raise that defect to the concerned developer. This whole process is called as “Execute test Case” **Here Actual Testing is taking place.**

**6.Defect Tracking:**

Once the Execution of test case is over, some defects will find It has to be reported to developer. Then we should check what is happening to that defect. This is called Defect Tracking.

**7.Test Execution Report:**

Once all the test cases are executed how many are pass & how many are failed will be measured. For this one report will be created. This is called as Test Executed Report.

Till here, customer can involve & he can ask every data & document. According to customer this is the last Stage.

According to company there is one more Stage that is Retrospective Meeting.

**8. Retrospective Meeting / Post-mortem Report:**

All the team members will gather together like Testing team, development team, B.A, Managers and they will discuss about good process which happened & also improvements which is needed. This will be helpful for the future projects

---------------------------------------------- \*\*\*\*\* -------------------------------------------------------------

**Program For Highest (or) Max. Number in an Array:**

**public** **class** ArrayHighestNumber

{

//creating User define method with array as args

**public** **static** **void** max(**int** arr[])

{

**int** max = arr[0];

**for**(**int** i=1; i<arr.length; i++)

{

**if**(arr[i]>max)

{

max = arr [i];

}

}

System.***out***.println("Maximum Number is: " +max);

}

**public** **static** **void** main (String args[])

{

**int** a[] = {116,250,700,1000,550};

*max*(a);

}

}

**Output:** Maximum Number is: 1000

---------------------------------------------- \*\*\*\*\* -------------------------------------------------------------

**Program For Lowest (or) Min. Number in an Array:**

**public** **class** ArrayLowestNumber

{

//creating User define method with array as args

**public** **static** **void** min(**int** arr[])

{

**int** min = arr[0];

**for**(**int** i=1; i<arr.length; i++)

{

**if**(arr[i]<min)

{

min = arr [i];

}

}

System.***out***.println("Minimum Number is: " +min);

}

**public** **static** **void** main (String args[])

{

**int** a[] = {116,250,700,1000,550};

*min*(a);

}

}

**Output:** Minimum Number is: 116

---------------------------------------------- \*\*\*\*\* -------------------------------------------------------------

**Program For 2nd Highest (or) 2nd Max. Number in an Array:**

**public** **class** ArraySecondHighest

{

//Creating User Define Method with array as args

**public** **static** **void** Sec\_Max (**int** arr [])

{

**int** max = arr[0],secmax=arr[0];

**for**(**int** i=1; i<arr.length; i++)

{

**if**(arr[i]>max)

{

secmax = max;

max = arr[i];

}

**else** **if**(arr[i]>secmax)

{

secmax=arr[i];

}

}

System.***out***.println("Maximum Number is: "+max);

System.***out***.println("Second Maximum Number is: "+secmax);

}

**public** **static** **void** main(String[] args)

{

**int** a[]= {116,250,700,1000,550};

*Sec\_Max*(a);

}

}

**Output:** Maximum Number is: 1000

Second Maximum Number is: 700

---------------------------------------------- \*\*\*\*\* -------------------------------------------------------------

**Program For 2nd Lowest (or) 2nd Min. Number in an Array:**

**public** **class** ArraySecondLowest

{

//Creating User Define Method with array as args

**public** **static** **void** Sec\_Min (**int** arr [])

{

**int** min = arr[0],secmin=arr[0];

**for**(**int** i=1; i<arr.length; i++)

{

**if**(arr[i]<min)

{

secmin = min;

min = arr[i];

}

**else** **if**(arr[i]<secmin)

{

secmin=arr[i];

}

}

System.***out***.println("Minimum Number is: "+min);

System.***out***.println("Second Minimum Number is: "+secmin);

}

**public** **static** **void** main(String[] args)

{

**int** a[]= {250,700,1000,550,116};

*Sec\_Min*(a);

}

}

**Output:** Minimum Number is: 116

Second Minimum Number is: 250

---------------------------------------------- \*\*\*\*\* -------------------------------------------------------------