**Subject**

**Software Engineering**

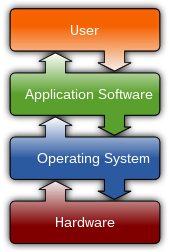
**Ch-1**

**Software Development Process**

***1.1 Software Definition***:

Software is a set of instruction to take input from user and process it and gives output to the user in terms of function and performance as determine by the user to the software.

It is developed to handle as input process output system achieve pre-determined goals.



Software Includes,

1-instructions

2-document

3- Arch. including data structure

***Types of software:***

Types of software

***1] System software:***

Application software

Programming Tools

System software

It directly interact with computer hardware .It is generic software.**System software** is software designed to provide a platform for other software.

System software is classified into three categories;

1. **Operating system:**

O.S is act as interface between user and the hardware and provides different services to user.

1. **System utilities:**

. In this software are included such as booting, formatting, deleting etc.

**c)Device drivers:**

Drivers of the IO devices consist of interfacing program which can interact with IO device.every IO device has its own driver program.

***1.3 Process model:***

A software process model is an abstraction of the software development process. The models specify the stages and order of a process.

So, think of this as a representation of the **order of activities** of the process and the **sequence** in which they are performed.

**A model will define the following:**

* The tasks to be performed
* The input and output of each task
* The pre and post-conditions for each task
* The flow and sequence of each task

The goal of a software process model is to provide guidance for controlling and coordinating the tasks to achieve the end product and objectives as effectively as possible.

There are many kinds of process models for meeting different requirements. We refer to these as **SDLC models** (Software Development Life Cycle models). The most popular and important SDLC models are as follows:

1. Waterfall model
2. V model
3. Incremental model
4. Spiral model
5. RAD model
6. Agile model
7. Iterative model
8. Prototype model

## 1]- Waterfall Model:

In the waterfall method, each step is dependent on the output of the previous step. There's a linear progression to the way these projects unfold.

The waterfall model doesn't include a project's [end user](https://www.techtarget.com/whatis/definition/end-user) or client as much as other development methodologies. Users are consulted during the initial stages of gathering and defining [requirements](https://www.techtarget.com/whatis/definition/technical-requirements), incorporating client [feedback](https://www.techtarget.com/searchitchannel/definition/feedback-loop) after that.

By leaving the client out of the main part of thewaterfall process, the development team moves quickly through the phases of a project.

This methodology is good for teams and projects that want to develop a project according to fixed or unchanging requirements set forth at the beginning of the project. Waterfall projects have a high degree of process definition with little or no output variability. Waterfall is also a good choice if the project is constrained by cost or time

.

### Phases of the waterfall model

When used for a software development process, the waterfall methodology has seven stages:

| **Different phases** | **Activities performed in each stage** |
| --- | --- |
| 1. **Requirement Gathering stage** | * During this phase, detailed requirements of the software system to be developed are gathered from client |
| 1. **Design Stage** | * Plan the programming language, for Example [Java](https://www.guru99.com/java-tutorial.html), [PHP](https://www.guru99.com/php-tutorials.html), .net * or database like Oracle, MySQL, etc. * Or other high-level technical details of the project |
| 1. **Built Stage** | * After design stage, it is built stage, that is nothing but coding the software |
| 1. **Test Stage** | * In this phase, you test the software to verify that it is built as per the specifications given by the client. |
| 1. **Deployment stage** | * Deploy the application in the respective environment |
| 1. **Maintenance stage** | * Once your system is ready to use, you may later require change the code as per customer request |

**Advantages and Disadvantages of Waterfall Model**

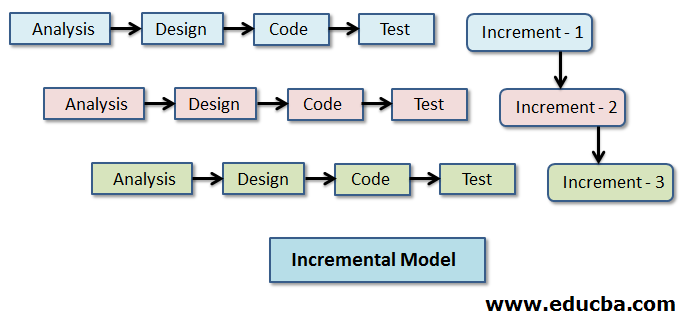
| **Advantages** | **Dis-Advantages** |
| --- | --- |
| * Before the next phase of development, each phase must be completed | * Error can be fixed only during the phase |
| * Suited for smaller projects where requirements are well defined | * It is not desirable for complex project where requirement changes frequently |
| * They should perform quality assurance test (Verification and Validation) before completing each stage | * Testing period comes quite late in the developmental process |
| * Elaborate documentation is done at every phase of the software’s development cycle | * Documentation occupies a lot of time of developers and testers |
| * Project is completely dependent on project team with minimum client intervention | * Clients valuable feedback cannot be included with ongoing development phase |
| * Any changes in software is made during the process of the development | * Small changes or errors that arise in the completed software may cause a lot of problems |

***2] Incremental Model***:

Incremental Model, also known as the successive version model, is a widely adopted model of software development process where the software requirements are divided or broken down into multiple stand-alone modules/increments in the SDLC (Software Development Life Cycle).

Each increment is treated as a sub-project and goes through all phases of the SDLC incremental model. This sounds similar to an iterative model.

However, this model is an enhancement to the iterative model and due to this, the incremental model is also called the Iterative Enhancement Model.  In the incremental model, instead of making one huge leap, we achieve our goals in small steps.



The system is put into production when the first increment is delivered. The first increment is often a core product where the basic requirements are addressed, and supplementary features are added in the next increments. Once the core product is analyzed by the client, there is plan development for the next increment.

**Characteristics of an Incremental module includes**

* System development is broken down into many mini development projects
* Partial systems are successively built to produce a final total system
* Highest priority requirement is tackled first
* Once the requirement is developed, requirement for that increment are fro

**Phases of Incremental Model:**

| **Incremental Phases** | **Activities performed in incremental phases** |
| --- | --- |
| * **Requirement Analysis** | * Requirement and specification of the software are collected |
| * **Design** | * Some high-end function are designed during this stage |
| * **Code** | * Coding of software is done during this stage |
| * **Test** | * Once the system is deployed, it goes through the testing phase |

**Advantages and Disadvantages of Incremental Model**

| **Advantages** | **Disadvantages** |
| --- | --- |
| * The software will be generated quickly during the software life cycle | * It requires a good planning designing |
| * It is flexible and less expensive to change requirements and scope | * Problems might cause due to system architecture as such not all requirements collected up front for the entire software lifecycle |
| * Throughout the development stages changes can be done | * Each iteration phase is rigid and does not overlap each other |
| * This model is less costly compared to others | * Rectifying a problem in one unit requires correction in all the units and consumes a lot of time |
| * A customer can respond to each building |  |
| * Errors are easy to be identified |  |

***3]V-Model:***

The V-model is a type of SDLC model where process executes in a sequential manner in V-shape.

It is also known as Verification and Validation model. It is based on the association of a testing phase for each corresponding development stage.

Development of each step directly associated with the testing phase. The next phase starts only after completion of the previous phase i.e. for each development activity, there is a testing activity corresponding to it.



**Verification:**

 It involves static analysis technique (review) done without executing code. It is the process of evaluation of the product development phase to find whether specified requirements meet.

**Validation:**

It involves dynamic analysis technique (functional, non-functional), testing done by executing code.Validation is the process to evaluate the software after the completion of the development phase to determine whether software meets the customer expectations and requirements.

So V-Model contains Verification phases on one side of the Validation phases on the other side. Verification and Validation phases are joined by coding phase in V-shape. Thus it is called V-Model.

**Design Phase:** 

* **Requirement Analysis:** This phase contains detailed communication with the customer to understand their requirements and expectations. This stage is known as Requirement Gathering.
* **System Design:** This phase contains the system design and the complete hardware and communication setup for developing product.
* **Architectural Design:** System design is broken down further into modules taking up different functionalities. The data transfer and communication between the internal modules and with the outside world (other systems) is clearly understood.
* **Module Design:** In this phase the system breaks down into small modules. The detailed design of modules is specified, also known as Low-Level Design (LLD).

**Testing Phases:**

* **Unit Testing:**

In Unit testing we can divide the project into small parts and test this small parts.

* **Integration testing**:

In integration testing we can integrate this small parts and test this parts.

* **System Testing:**

In this testing test whole system. Checks the all software and hardware are working properly or not.

* **Acceptance Testing:** In this testing test whether the customer requirements are fulfill or not.

**Advantages:**

1. This is a highly disciplined model and Phases are completed one at a time.
2. V-Model is used for small projects where project requirements are clear.
3. Simple and easy to understand and use.
4. This model focuses on verification and validation activities early in the life cycle thereby enhancing the probability of building an error-free and good quality product.
5. It enables project management to track progress accurately.

**Disadvantages:** 

* High risk and uncertainty.
* It is not a good for complex and object-oriented projects.
* It is not suitable for projects where requirements are not clear and contains high risk of changing.
* This model does not support iteration of phases.
* It does not easily handle concurrent events.

***4] Spiral Model:***

The spiral model is a systems development lifecycle ([SDLC](https://www.techtarget.com/searchsoftwarequality/definition/systems-development-life-cycle)) method used for [risk management](https://searchcompliance.techtarget.com/definition/risk-management) that combines the [iterative development](https://www.techtarget.com/searchsoftwarequality/definition/iterative-development) process model with elements of the [Waterfall model](https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model). The spiral model is used by software engineers and is favored for large, expensive and complicated projects.

When viewed as a diagram, the spiral model looks like a coil with many loops. The number of loops varies based on each project and is often designated by the [project manager](https://www.techtarget.com/searchcio/definition/project-management). Each loop of the spiral is a phase in the software development process.

As mentioned before, the spiral model is best used in large, expensive and complicated projects. Other uses include:

* projects in which frequent releases are necessary;
* projects in which changes may be required at any time;
* long term projects that are not feasible due to altered economic priorities;
* medium to high risk projects;
* projects in which cost and risk analysis is important;
* projects that would benefit from the creation of a prototype; and
* projects with unclear or complex requirements.

## Spiral Model Application

The Spiral Model is widely used in the software industry as it is in sync with the natural development process of any product, i.e. learning with maturity which involves minimum risk for the customer as well as the development firms.

The following pointers explain the typical uses of a Spiral Model −

* When there is a budget constraint and risk evaluation is important.
* For medium to high-risk projects.
* Long-term project commitment because of potential changes to economic priorities as the requirements change with time.
* Customer is not sure of their requirements which is usually the case.
* Requirements are complex and need evaluation to get clarity.
* New product line which should be released in phases to get enough customer feedback.
* Significant changes are expected in the product during the development cycle.

**Spiral Model Phases**

| **Spiral Model Phases** | **Activities performed during phase** |
| --- | --- |
| **Planning** | * It includes estimating the cost, schedule and resources for the iteration. It also involves understanding the system requirements for continuous communication between the system analyst and the customer |
| **Risk Analysis** | * Identification of potential risk is done while risk mitigation strategy is planned and finalized |
| **coding** | * It includes testing, coding and deploying software at the customer site |
| **Evaluation** | * Evaluation of software by the customer. Also, includes identifying and monitoring risks such as schedule slippage and cost overrun |

**Spiral Model Advantages**

1. The spiral model is perfect for projects that are large and complex in nature as continuous prototyping and evaluation help in mitigating any risk.
2. Because of its risk handling ability, the model is best suited for projects which are very critical like software related to the health domain, space exploration, etc.
3. This model supports the client feedback and implementation of change requests (CRs) which is not possible in conventional models like a waterfall.
4. Since customer gets to see a prototype in each phase, so there are higher chances of customer satisfaction.

**Spiral Model Disadvantages**

1. Because of the prototype development and risk analysis in each phase, it is very expensive and time taking.
2. It is not suitable for a simpler and smaller project because of multiple phases.
3. It requires more documentation as compared to other models.
4. Project deadlines can be missed since the number of phases is unknown in the beginning and frequent prototyping and risk analysis can make things worse

*5]Prototype Model*:

**Prototyping Model** is a software development model in which prototype is built, tested, and reworked until an acceptable prototype is achieved.

It also creates base to produce the final system or software. It works best in scenarios where the project’s requirements are not known in detail.

It is an iterative, trial and error method which takes place between developer and client.A prototype is a toy implementation of the system.

A prototype usually turns out to be a very crude version of the actual system, possible exhibiting limited functional capabilities, low reliability, and inefficient performance as compared to actual software.

In many instances, the client only has a general view of what is expected from the software product. In such a scenario where there is an absence of detailed information regarding the input to the system, the processing needs, and the output requirement, the prototyping model may be employed.

Advantage of Prototype Model

1. Reduce the risk of incorrect user requirement
2. Good where requirement are changing/uncommitted
3. Regular visible process aids management
4. Support early product marketing
5. Reduce Maintenance cost.
6. Errors can be detected much earlier as the system is made side by side.

## Disadvantage of Prototype Model

1. Difficult to know how long the project will last.
2. Easy to fall back into the code and fix without proper requirement analysis, design, customer evaluation, and feedback.
3. Prototyping tools are expensive.
4. Special tools & techniques are required to build a prototype.
5. It is a time-consuming process.

***6] RAD Model : [Rapid Application Model]***

**RAD Model** or Rapid Application Development model is a software development process based on prototyping without any specific planning.

In RAD model, there is less attention paid to the planning and more priority is given to the development tasks. It targets at developing software in a short span of time.

**Phases of RAD Model:**

When to use RAD Model?

* When the system should need to create the project that modularizes in a short span time (2-3 months).
* When the requirements are well-known.
* When the technical risk is limited.
* When there's a necessity to make a system, which modularized in 2-3 months of period.
* It should be used only if the budget allows the use of automatic code generating tools.

Advantage of RAD Model

* This model is flexible for change.
* In this model, changes are adoptable.
* Each phase in RAD brings highest priority functionality to the customer.
* It reduced development time.
* It increases the reusability of features.

Disadvantage of RAD Model

* It required highly skilled designers.
* All application is not compatible with RAD.
* For smaller projects, we cannot use the RAD model.
* On the high technical risk, it's not suitable.
* Required user involvement.

***1.4] Agile software Model:***

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.

Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks.

Every iteration involves cross functional teams working simultaneously on various areas like −

* Planning
* Requirements Analysis
* Design
* Coding
* Unit Testing and
* Acceptance Testing.

At the end of the iteration, a working product is displayed to the customer and important stakeholders.

**What is Agile?**

Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements.

In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

Here is a graphical illustration of the Agile Model –



The Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and adaptability.

Following are the Agile Manifesto principles −

* **Individuals and interactions** − In Agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
* **Working software** − Demo working software is considered the best means of communication with the customers to understand their requirements, instead of just depending on documentation.
* **Customer collaboration** − As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.
* **Responding to change** − Agile Development is focused on quick responses to change and continuous development.

### Advantages

* Facilitates effective communication between teams and clients
* Enhances project flexibility, easily handling sudden changes anytime
* Cuts overall development time
* Increases customer satisfaction via rapid, continuous delivery of a workable product
* Delivers functioning software frequently, in weeks instead of months

### Disadvantages

* Confusion between teams may develop because documentation wasn’t emphasized. This uncertainty can lead to difficult transitions between phases.
* It is sometimes difficult to gauge how much effort will be needed to start the development life cycle of larger software deliverables.
* If stakeholders project ladder are not on the same page, the project will derail.
* The modeling isn’t for newbies. The sort of decisions involved in Agile require people with experience and solid developer and programming skills.

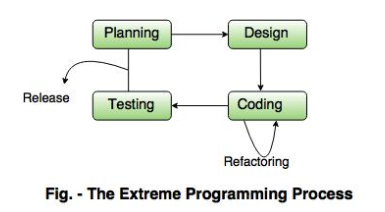
There are many Agile process model there are,

1. XP [Extreme Programming]
2. ASD [Adaptive Software Development]
3. DSDM [Dynamic System Development Method]
4. Scrum
5. Crystal

***1] XP [Extreme Programming]:***

Extreme Programming (XP) is an agile software development framework that aims to produce higher quality software, and higher quality of life for the development team.

XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development.



The XP framework normally involves 5 phases or stages of the development process that iterate continuously:

1. **Planning,**the first stage, is when the customer meets the development team and presents the [requirements](https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/) in the form of user stories to describe the desired result. The team then estimates the stories and creates a release plan broken down into iterations needed to cover the required functionality part after part. If one or more of the stories can’t be estimated, so-called spikes can be introduced which means that further research is needed.
2. **Designing**is actually a part of the planning process, but can be set apart to emphasize its importance. It’s related to one of the main XP values that we’ll discuss below — simplicity. A good design brings logic and structure to the system and allows to avoid unnecessary complexities and redundancies.
3. **Coding**is the phase during which the actual code is created by implementing specific XP practices such as coding standards, pair programming, continuous integration, and collective code ownership (the entire list is described below).
4. **Testing**is the core of extreme programming. It is the regular activity that involves both unit tests ([automated testing](https://www.altexsoft.com/whitepapers/quality-assurance-quality-control-and-testing-the-basics-of-software-quality-management/) to determine if the developed feature works properly) and acceptance tests (customer testing to verify that the overall system is created according to the initial requirements).
5. **Release** is all about constant communication and feedback. The customers and project managers are involved to describe the business logic and value that is expected.

Such a development process entails the cooperation between several participants, each having his or her own tasks and responsibilities. Extreme programming puts people in the center of the system, emphasizing the value and importance of such social skills as communication, cooperation, responsiveness, and feedback. So, these roles are commonly associated with XP:

1. **Customers**are expected to be heavily engaged in the development process by creating user stories, providing continuous feedback, and making all the necessary business decisions related to the project.
2. **Programmers or developers**are the team members that actually create the product. They are responsible for implementing user stories and conducting user tests (sometimes a separate **Tester** role is set apart). Since XP is usually associated with cross-functional teams, the skill set of such members can be different.
3. **Trackers or managers** link customers and developers. It’s not a required role and can be performed by one of the developers. These people organize the meet-ups, regulate discussions, and keep track of important progress KPIs.
4. **Coaches** can be included in the teams as mentors to help with understanding the XP practices. It’s usually an outside assistant or external consultant who is not involved in the development process, but has used XP before and so can help avoid mistakes.

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| No unnecessary programming work | Customer must participate in the process |
| Stable software through continuous testing | Relatively large time investment |
| Error avoidance through pair programming | Relatively high costs |

# *2] Adaptive Software Development (ASD):*

**Adaptive Software Development** is a method to build complex software and system.

Adaptive Software Development (ASD) is a direct outgrowth of an earlier [agile framework](https://www.productplan.com/glossary/agile-framework/), Rapid Application Development (RAD).

Moreover, it aims to enable teams to quickly and effectively adapt to changing requirements or market needs by evolving their products with lightweight planning and continuous learning.

ASD focuses on human collaboration and self-organization. ASD **“life cycle”** incorporates three phases namely:

**1.** Speculation

**2.** Collaboration

**3.** Learning



**1. Speculation:**   
During this phase project is initiated and planning is conducted. The project plan uses project initiation information like project requirements, user needs, customer mission statement, etc, to define set of release cycles that the project wants

**2. Collaboration:**   
It is the difficult part of ASD as it needs the workers to be motivated. It collaborates communication and teamwork but emphasizes individualism as individual creativity plays a major role in creative thinking. People working together must trust each others to

* Criticize without animosity,
* Assist without resentment,
* Work as hard as possible,
* Possession of skill set,
* Communicate problems to find effective solution.

**3. Learning:**   
The workers may have a overestimate of their own understanding of the technology which may not lead to the desired result. Learning helps the workers to increase their level of understanding over the project.   
Learning process is of 3 ways:

* Focus groups
* Technical reviews
* Project postmortem

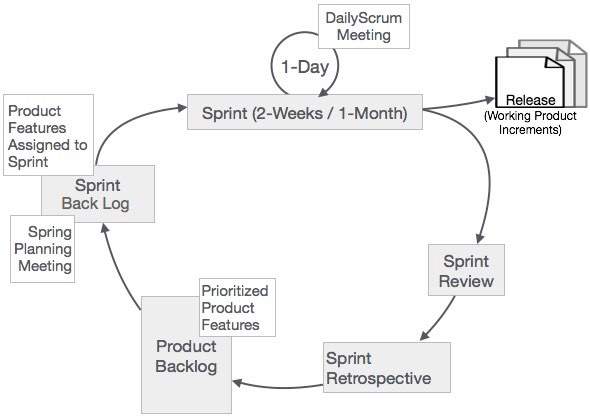
ASD’s overall emphasis on the dynamics of self-organizing teams, interpersonal collaboration, and individual and team learning yield software project teams that have a much higher likelihood of success.

# *3] Scrum :*

**Scrum** is the type of **Agile framework**. It is a framework within which people can address complex adaptive problem while productivity and creativity of delivering product is at highest possible values. Scrum uses **Iterative process**.

**Silent features of Scrum are:**

* Scrum is light-weighted framework
* Scrum emphasizes self-organization
* Scrum is simple to understand
* Scrum framework help the team to work together



**Sprint:**  
A Sprint is a time-box of one month or less. A new Sprint starts immediately after the completion of the previous Sprint.

**Sprint Review:**  
If the product still have some non-achievable features then it will be checked in this stage and then the product is passed to the Sprint Retrospective stage.

**Product Backlog:**  
According to the prioritize features the product is organized.

**Sprint Backlog:**  
Sprint Backlog is divided into two parts Product assigned features to sprint and Sprint planning meeting.

**Scrum is a framework** that helps agile teams to work together. Using it, the team members can deliver and sustain the complex product. It encourages the team to learn through practice, self-organize while working on the problem. Scum is a work done through the framework and continuously shipping values to customers.

It is the most frequent software that is used by the development team. Its principle and lessons can be applied to all kinds of teamwork. Its policy and experiences is a reason of popularity of Scrum framework. The Scrum describes a set of tools, meetings, and roles that help the teams structure. It also manages the work done by the team

**Advantage of using Scrum framework:**

* Scrum framework is fast moving and money efficient.
* Scrum framework works by dividing the large product into small sub-products. It’s like a divide and conquer strategy
* In Scrum customer satisfaction is very important.
* Scrum is adaptive in nature because it have short sprint.
* As Scrum framework rely on constant feedback therefore the quality of product increases in less amount of time

**Disadvantage of using Scrum framework:**

* Scrum framework do not allow changes into their sprint.
* Scrum framework is not fully described model. If you wanna adopt it you need to fill in the framework with your own details like Extreme Programming(XP), DSDM.
* It can be difficult for the Scrum to plan, structure and organize a project that lacks a clear definition.
* The daily Scrum meetings and frequent reviews require substantial resources.

***Difference between Waterfall Model n Spiral Model:***

| **S. No.** | **Waterfall Model** | **Spiral Model** |
| --- | --- | --- |
| **1.** | The[Waterfall model](https://www.geeksforgeeks.org/software-engineering-classical-waterfall-model/) is simple and easy. | The [spiral model](https://www.geeksforgeeks.org/software-engineering-spiral-model/)is a lot more complex. |
| **2.** | The waterfall model works in a sequential method. | While the spiral model works in the evolutionary method. |
| **3.** | In the waterfall model errors or risks are identified and rectified after the completion of stages. | In the spiral model errors or risks are identified and rectified earlier. |
| **4.** | The waterfall model is adopted by customers. | While the spiral model is adopted by developers. |
| **5.** | The waterfall model is applicable for small projects. | While the Spiral model is used for large projects. |
| **6.** | In waterfall model requirements and early stage planning is necessary. | While in spiral model requirements and early stage planning is necessary if required. |
| **7.** | Flexibility to change in waterfall model is Difficult. | Flexibility to change in spiral model is not Difficult. |
| **8.** | There is high amount risk in waterfall model. | There is low amount risk in spiral model. |
| **9.** | Waterfall model is comparatively inexpensive. | While cost of spiral model is very expensive. |
| **10.** | Customer involvement is minimum in Waterfall Model | In the Spiral Model Customer involvement is high. |
| **11.** | It requires least maintenance. | It requires typical maintenance. |
| **12.** | It is based on linear framework type. | It is based on linear and iterative framework type. |
| **13.** | Testing is done after the coding phase in the development life cycle. | Testing is done after the engineering phase in the development cycle. |
| **14.** | Reusability is extremely unlikely. | To a certain extent, reusability is possible. |
| **15.** | Customer control over the administrator is very limited. | Customers have control over the administrator as compared to waterfall model. |

***Difference between Waterfall Model n Agile Model:***

|  |  |
| --- | --- |
| **Agile Model** | **Waterfall Model** |
| It separates the project development lifecycle into multiple sprints. | The process of software development is divided into distinct phases. |
| It uses an incremental approach | It is a sequential design process. |
| It is flexible. | It is a structured process, and may be quite rigid at times. |
| It can be understood as a collection of multiple different projects. | The software can be developed as one single project. |
| Changes can be made in project development requirements after initial planning has been completed. | The requirements can't be changed once the project development starts. |
| It follows an iterative development approach. The process of planning, development, prototyping and software development phases can appear multiple times. | The project development phases such as designing, development, testing, are done only once in the waterfall model. |
| The test plan is reviewed after every sprint. | The test plan is not usually discussed during the test phase. |
| The requirements may change and evolve. | This method can be used for projects that have a definite set of requirements and doesn't changes. |
| The process of testing is performed concurrently with software development. | The 'testing' phase comes after the 'build' phase. |
| It uses a product mindset where software product would satisfy the requirements of end customers. | It shows a project mindset, and focuses on completing the project. |
| It would increase the amount of stress in fixed-price scenarios. | The risk is reduced since the price of contracts are fixed by getting the risk agreement in the beginning. |
| It is preferred to work with small and dedicated teams with high amounts of coordination and synchronization. | There is limited team coordination and synchronization. |
| It works well with time and materials, i.e. non-fixed funding. | Business analysis is done to prepare the requirement before beginning it. |
| Products owner with team prepares requirements just about every day during a project. | It is difficult to initiate any changes in the requirements. |
| Project managers are not needed since project can be managed by the entire team. | The process is generally straightforward hence project manager is required. |