

## Software Engineering : Module 1 Notes

### Software Evolution

Software Evolution is a term that refers to the process of developing software initially, and then timely updating it for various reasons, i.e., to add new features or to remove obsolete functionalities, etc. This article focuses on discussing Software Evolution in detail.

### What is Software Evolution?

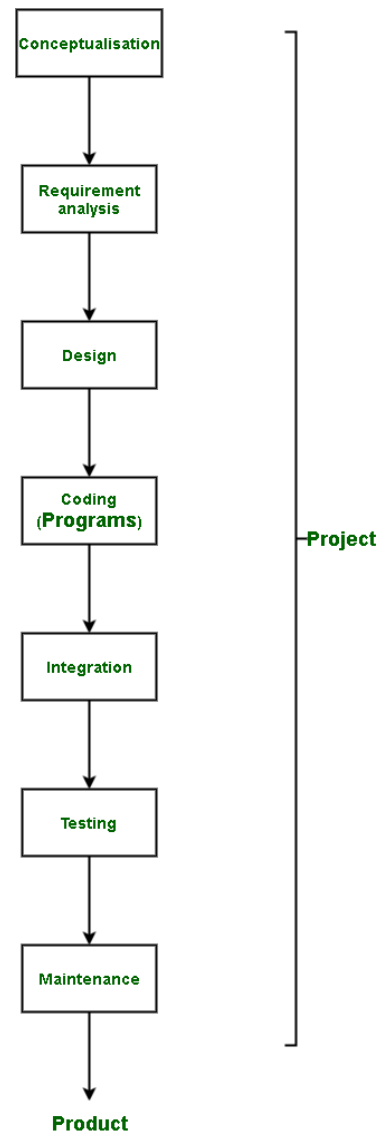
The software evolution process includes fundamental activities of change analysis, release planning, system implementation, and releasing a system to customers.

1. The cost and impact of these changes are accessed to see how much the system is affected by the change and how much it might cost to implement the change.
2. If the proposed changes are accepted, a new release of the software system is planned.
3. During release planning, all the proposed changes (fault repair, adaptation, and new functionality) are considered.
4. A design is then made on which changes to implement in the next version of the system.
5. The process of change implementation is an iteration of the development process where the revisions to the system are designed, implemented, and tested.

### Difference between Program and Product

**1. Program :** The use and need of computers is widespread in today's world. The computer system is however incapable of working on its own and needs certain instructions to do anything. These instructions are provided to the computer system by a computer programmer. A collection of instructions meant to perform a specific task, written by a computer programmer is called a program.

**2. Product :** It is made only on the request of an enterprise. The team of developers undertakes the making of the product (called project). A software product is the finished project which was made after performing the various software development life cycle stages. After the completion of these stages the product comes into existence. It is then made in more numbers, for distribution. The software product is made to solve a specific problem.



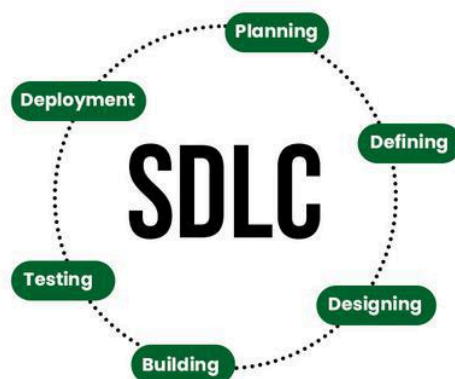
**Difference between Program and Product :**

<b>S.No.</b>	<b>Program</b>	<b>Product</b>
1.	Program contains a set of instructions designed to complete a specific task.	Product is the final manufacture and production of the project.
2.	It is a passive entity as it resides in the secondary memory.	It is an active entity as it is made again and again for the purpose of distribution to users.
3.	It exists at a single place and continues to exist until it is deleted.	A product can exist for a long period of time as long as it is tested for better performance and durability. A specific software product would stop existing if it is destroyed or uninstalled.
4.	It can be made in relatively shorter period of time.	It needs a lot of time to be made. There are several stages (design, coding, testing, maintenance etc.) to form a product.
5.	It is handled by the program managers.	It is handled by product managers.

6.	It does not have any resource requirement, it only requires memory space for storing the instructions.	It needs human, technology and time resources to be functional.
7.	The main focus of the program is whether it is giving a specific desired output or not.	The main focus of a product is its capability to solve the problem that it was made for.
8.	There are no concrete guidelines in writing a program.	In case of product, the firm guidelines are followed.
9.	It may or may not focus on efficiency and cost.	It needs to worry about the cost and efficiency as it would affect the profitability of the enterprise launching the product.

### Software Development Life Cycle (SDLC)

Software development life cycle (SDLC) is a structured process that is used to design, develop, and test good-quality software. SDLC, or software development life cycle, is a methodology that defines the entire procedure of software development step-by-step.



## Waterfall Model – Software Engineering

### 1. Waterfall Model

#### FIGURE SHOULD BE DRAWN HERE

Waterfall model is a famous and good version of SDLC(System Development Life Cycle) for software engineering. The waterfall model is a linear and sequential model, which means that a development phase cannot begin until the previous phase is completed. We cannot overlap phases in the waterfall model.

- **Requirement phase:-** Requirement phase is the first phase of the waterfall model. In this phase the requirements of the system are collected and documented. This phase is very crucial because the next phases are based on this phase.
- **Design phase:-** Design phase is based on the fact how the software will be built. The main objective of the design phase is to prepare the blueprint of the software system so that no problems are faced in the coming phases and solutions to all the requirements in the requirement phase are found.
- **Implementation phase:-** In this phase, hardware, software and application programs are installed and the database design is implemented. Before the database design can be implemented, the software has to go through a testing, coding, and debugging process. This is the longest lasting phase in waterfall.
- **Verification phase:-** In this phase the software is verified and it is evaluated that we have created the right product. In this phase, various types of testing are done and every area of the software is checked. It is believed that if we do not verify the software properly and there is any defect in it then no one will use it, hence verification is very important. One advantage of verification is that it reduces the risk of software failure.
- **Maintenance phase:-** This is the last phase of the waterfall. When the system is ready and users start using it, then the problems that arise have to be solved time-to-time. Taking care of the finished software and maintaining it as per time is called maintenance.

#### Advantages of Waterfall Model

- This model is simple and easy to understand.
- This is very useful for small projects.
- This model is easy to manage.
- The end goal is determined early.
- Each phase of this model is well explained.
- It provides a structured way to do things.
- This is a base model, all the SDLC models that came after this were created keeping this in mind, although they worked to remove its shortcomings.
- In this model, we can move to the next phase only after the first phase is successfully completed so that there is no overlapping between the phases.

#### Disadvantages of Waterfall Model

- In this model, complete and accurate requirements are expected at the beginning of the development process.
- Working software is not available for very long during the development life cycle.
- We cannot go back to the previous phase due to which it is very difficult to change the requirements.
- Risk is not assessed in this, hence there is high risk and uncertainty in this model.
- In this the testing period comes very late.

- Due to its sequential nature this model is not realistic in today's world.
- This is not a good model for large and complex projects.

## 2.Iterative Model

### FIGURE SHOULD BE DRAWN HERE

In the Iterative model we start developing the software with some requirements and when it is developed, it is reviewed. If there are requirements for changes in it, then we develop a new version of the software based on those requirements. This process repeats itself many times until we get our final product.

The different phases of iterative model are as follows:-

1. **Requirement gathering & analysis:** In this phase, all the software requirements of the customer are collected and it is analysed whether those requirements can be meet or not. Besides, it is also checked whether this project will not go beyond our budget.
2. **Design:** In this phase the design of software is prepared. For this, various diagrams like Data Flow diagram, class diagram, activity diagram, state transition diagram, etc. are used.
3. **Implementation:** Now the design of software is implemented in coding through various programming languages. We also call this coding phase.
4. **Testing:** After the coding of the software is done, it is now tested so that the bugs and errors present in it can be identified. To do this, various testing techniques like performance testing, security testing, requirement testing, stress testing, etc. are done.
5. **Deployment:** Finally the software is given to the customer. After this the customer starts using that software in his work environment.
6. **Review:** After the software is deployed in its work environment, it is reviewed. If any error/bug is found or any new requirements come in front of the developer, then again these phases are repeated with new iteration and a new version is developed.
7. **Maintenance:** In this phase we look at customer feedback, solve problems, fix errors, update software, etc.

Advantage of Iterative model:-

- In iterative models, bugs and errors can be identified quickly.
- Under this model, software is prepared quickly with some specifications.
- Testing and debugging the software becomes easier during each iteration.
- We get reliable feedback from users along with blueprints.
- This model is easily adaptable to constantly changing needs.
- During the software development process, additional time is devoted to development and limited time to documentation.
- Risks are identified and resolved during iteration.

Disadvantage of Iterative model:-

- Iterative model is not suitable for small projects.
- Since we have to repeat iterations many times in the software development process due to which we require more resources.
- Since the requirements are constantly changing, we have to make frequent changes in the software.

- Due to constantly changing requirements, the budget of the project also increases and it takes more time to complete it.
- In this model, it is complicated to control the entire process of software development.
- It is very difficult to tell by what date the complete software will be ready.

### 3. V-Model

#### FIGURE SHOULD BE DRAWN HERE

V-Model is an SDLC model, it is also called Verification and Validation Model. V-Model is widely used in the software development process, and it is considered a disciplined model. In V-Model, the execution of each process is sequential, that is, the new phase starts only after the previous phase ends.

- **Requirements analysis:-** This is the first phase of the development cycle, in which the requirements of the product are analysed according to the customer's needs. In this phase, product related requirements are thoroughly collected from the customer. This is a very important phase because this phase determines the coming phases. In this phase, acceptance tests are designed for later use.
- **System design:-** When we have the requirements of the product, after that we prepare a complete design of the system. In this phase, a complete description of the hardware and all the technical components required to develop the product .
- **Architectural design:-** In this phase architectural specifications are designed. It contains the specification of how the software will link internally and externally with all the components. Therefore this phase is also called high level design (HLD).
- **Module design:-** In this phase the internal design of all the modules of the system is specified. Therefore it is called low level design (LLD). It is very important that the design of all modules should be according to the system architecture. Unit tests are also designed in the module design phase.
- **Coding phase:-** In the coding phase, coding of the design and specification done in the previous phases is done. This phase takes the most time.

#### Validation Phases of V-Model

- **Unit testing:-** In the unit testing phase, the unit tests created during the module design phase are executed. Unit testing is code level testing, it only verifies the technical design. Therefore it is not able to test all the defects.
- **Integration testing:-** In integration testing, the integration tests created in the architectural design phase are executed. Integration testing ensures that all modules are working well together.
- **System testing:-** In system testing, the system tests created in the system design phase are executed. System tests check the complete functionality of the system. In this, more attention is given to performance testing and regression testing.
- **Acceptance testing:-** In acceptance testing, the acceptance tests created in the requirement analysis phase are executed. This testing ensures that the system is compatible with other systems. And in this, non-functional issues like:- load time, performance etc. are tested in the user environment.

#### Advantages of V-Model

- This is a simple and easy to use model.
- Planning, testing and designing tests can be done even before coding.

- This is a very disciplined model, in which phase by phase development and testing goes on.
- Defects are detected in the initial stage itself.
- Small and medium scale developments can be easily completed using it.

#### Disadvantages of V-Model

- This model is not suitable for any complex projects.
- There remains both high risk and uncertainty.
- This is not a suitable model for an ongoing project.
- This model is not at all suitable for a project which is unclear and in which there are changes in the requirement.

#### 4. Prototype model

##### FIGURE SHOULD BE DRAWN HERE

Prototype model is an activity in which prototypes of software applications are created. First a prototype is created and then the final product is manufactured based on that prototype.

- The prototype model was developed to overcome the shortcomings of the waterfall model.
- This model is created when we do not know the requirements well.
- The specialty of this model is that this model can be used with other models as well as alone.
- **Requirement gathering:** The first step of prototype model is to collect the requirements, although the customer does not know much about the requirements but the major requirements are defined in detail.
- **Build the initial prototype:** In this phase the initial prototype is built. In this some basic requirements are displayed and user interface is made available.
- **Review the prototype:** When the construction of the prototype is completed, it is presented to the end users or customer and feedback is taken from them about this prototype. This feedback is used to further improve the system and possible changes are made to the prototype.
- **Revise and improve the prototype:** When feedback is taken from end users and customers, the prototype is improved on the basis of feedback. If the customer is not satisfied with the prototype, a new prototype is created and this process continues until the customer gets the prototype as per his desire.

#### Advantages of Prototype model :-

- Prototype Model is suggested to create applications whose prototype is very easy and which always includes human machine interaction within it.
- When we know only the general objective of creating software, but we do not know anything in detail about input, processing and output. Then in such a situation we make it a Prototype Model.
- When a software developer is not very sure about the capability of an algorithm or its adaptability to an operating system, then in this situation, using a prototype model can be a better option.

#### Disadvantages of Prototype model :-

- When the first version of the prototype model is ready, the customer himself often wants small fixes and changes in it rather than rebuilding the system. Whereas if the system is redesigned then more quality will be maintained in it.
- Many compromises can be seen in the first version of the Prototype Model.
- Sometimes a software developer may make compromises in his implementation, just to get the prototype model up and running quickly, and after some time he may become comfortable with making such compromises and may forget that it is completely inappropriate to do so.

## 5. Incremental Model

### FIGURE SHOULD BE DRAWN HERE

In Incremental Model, the software development process is divided into several increments and the same phases are followed in each increment. In simple language, under this model a complex project is developed in many modules or builds.

- **Communication:** In the first phase, we talk face to face with the customer and collect his mandatory requirements. Like what functionalities does the customer want in his software, etc.
- **Planning:** In this phase the requirements are divided into multiple modules and planning is done on their basis.
- **Modelling:** In this phase the design of each module is prepared. After the design is ready, we take a particular module among many modules and save it in DDS (Design Document Specification). Diagrams like ERDs and DFDs are included in this document.
- **Construction:** Here we start construction based on the design of that particular module. That is, the design of the module is implemented in coding. Once the code is written, it is tested.
- **Deployment:** After the testing of the code is completed, if the module is working properly then it is given to the customer for use. After this, the next module is developed through the same phases and is combined with the previous module. This makes new functionality available to the customer. This will continue until complete modules are developed.

### Advantages of Incremental Model

- Important modules/functions are developed first and then the rest are added in chunks.
- Working software is prepared quickly and early during the software development life cycle (SDLC).
- This model is flexible and less expensive to change requirements and scope.
- The customer can respond to each module and provide feedback if any changes are needed.
- Project progress can be measured.
- It is easier to test and debug during a short iteration.
- Errors are easy to identify.

### Disadvantages of Incremental Model

- Management is a continuous activity that must be handled.
- Before the project can be dismantled and built incrementally,
- The complete requirements of the software should be clear.
- This requires good planning and designing.
- The total cost of this model is higher.



## 6.Evolutionary Model

### FIGURE SHOULD BE DRAWN HERE

The Evolutionary development model divides the development cycle into smaller, incremental waterfall models in which users can get access to the product at the end of each cycle.

1. Feedback is provided by the users on the product for the planning stage of the next cycle and the development team responds, often by changing the product, plan, or process.
2. Therefore, the software product evolves with time.
3. All the models have the disadvantage that the duration of time from the start of the project to the delivery time of a solution is very high.
4. The evolutionary model solves this problem with a different approach.
5. The evolutionary model suggests breaking down work into smaller chunks, prioritizing them, and then delivering those chunks to the customer one by one.
6. The number of chunks is huge and is the number of deliveries made to the customer.
7. The main advantage is that the customer's confidence increases as he constantly gets quantifiable goods or services from the beginning of the project to verify and validate his requirements.
8. The model allows for changing requirements as well as all work is broken down into maintainable work chunks.
- 9.

#### Advantages Evolutionary Model

1. **Adaptability to Changing Requirements:** Evolutionary models work effectively in projects when the requirements are ambiguous or change often. They support adjustments and flexibility along the course of development.
2. **Early and Gradual Distribution:** Functional components or prototypes can be delivered early thanks to incremental development. Faster user satisfaction and feedback may result from this.
3. **User Commentary and Involvement:** Evolutionary models place a strong emphasis on ongoing user input and participation. This guarantees that the software offered closely matches the needs and expectations of the user.
4. **Improved Handling of Difficult Projects:** Big, complex tasks can be effectively managed with the help of evolutionary models. The development process is made simpler by segmenting the project into smaller, easier-to-manage portions.

#### Disadvantages Evolutionary Model

1. **Communication Difficulties:** Evolutionary models require constant cooperation and communication. The strategy may be less effective if there are gaps in communication or if team members are spread out geographically.
2. **Dependence on an Expert Group:** A knowledgeable and experienced group that can quickly adjust to changes is needed for evolutionary models. Teams lacking experience may find it difficult to handle these model's dynamic nature.
3. **Increasing Management Complexity:** Complexity can be introduced by organizing and managing several increments or iterations, particularly in large projects. In order to guarantee integration and synchronization, good project management is needed.
4. **Greater Initial Expenditure:** As evolutionary models necessitate continual testing, user feedback and prototyping, they may come with a greater starting cost. This may be a problem for projects that have limited funding.

#### Rapid application development model (RAD)

## FIGURE SHOULD BE DRAWN HERE

The Rapid Application Development Model was first proposed by IBM in the 1980s. The RAD model is a type of incremental process model in which there is an extremely short development cycle. When the requirements are fully understood and the component-based construction approach is adopted then the RAD model is used. Various phases in RAD are Requirements Gathering, Analysis and Planning, Design, Build or Construction, and finally Deployment.

This model consists of 4 basic phases:

1. **Requirements Planning** – This involves the use of various techniques used in requirements elicitation like brainstorming, task analysis, form analysis, user scenarios, FAST (Facilitated Application Development Technique), etc. It also consists of the entire structured plan describing the critical data, methods to obtain it, and then processing it to form a final refined model.
2. **User Description** – This phase consists of taking user feedback and building the prototype using developer tools. In other words, it includes re-examination and validation of the data collected in the first phase. The dataset attributes are also identified and elucidated in this phase.
3. **Construction** – In this phase, refinement of the prototype and delivery takes place. It includes the actual use of powerful automated tools to transform processes and data models into the final working product. All the required modifications and enhancements are to be done in this phase.
4. **Cutover** – All the interfaces between the independent modules developed by separate teams have to be tested properly. The use of powerfully automated tools and subparts makes testing easier. This is followed by acceptance testing by the user.

### When to use the RAD Model?

1. **Well-understood Requirements:** When project requirements are stable and transparent, RAD is appropriate.
2. **Time-sensitive Projects:** Suitable for projects that need to be developed and delivered quickly due to tight deadlines.
3. **Small to Medium-Sized Projects:** Better suited for smaller initiatives requiring a controllable number of team members.
4. **High User Involvement:** Fits where ongoing input and interaction from users are essential.
5. **Innovation and Creativity:** Helpful for tasks requiring creative inquiry and innovation.
6. **Prototyping:** It is necessary when developing and improving prototypes is a key component of the development process.
7. **Low technological Complexity:** Suitable for tasks using comparatively straightforward technological specifications.

### Advantages of Rapid Application Development Model (RAD)

- The use of reusable components helps to reduce the cycle time of the project.
- Feedback from the customer is available at the initial stages.
- Reduced costs as fewer developers are required.
- The use of powerful development tools results in better quality products in comparatively shorter periods.
- The progress and development of the project can be measured through the various stages.
- It is easier to accommodate changing requirements due to the short iteration time spans.
- Productivity may be quickly boosted with a lower number of employees.

### Disadvantages of Rapid application development model (RAD)

- The use of powerful and efficient tools requires highly skilled professionals.

- The absence of reusable components can lead to the failure of the project.
- The team leader must work closely with the developers and customers to close the project on time.
- The systems which cannot be modularized suitably cannot use this model.
- Customer involvement is required throughout the life cycle.
- It is not meant for small-scale projects as in such cases, the cost of using automated tools and techniques may exceed the entire budget of the project.
- Not every application can be used with RAD.

## Agile development models

### FIGURE SHOULD BE DRAWN HERE

In earlier days, the **Iterative Waterfall Model** was very popular for completing a project. But nowadays, developers face various problems while using it to develop software. The main difficulties included handling customer change requests during project development and the high cost and time required to incorporate these changes. To overcome these drawbacks of the Waterfall Model, in the mid-1990s the **Agile Software Development** model was proposed.

**The Agile Model** was primarily designed to help a project adapt quickly to change requests. So, the main aim of the Agile model is to facilitate quick project completion. To accomplish this task, agility is required. Agility is achieved by fitting the process to the project and removing activities that may not be essential for a specific project. Also, anything that is a waste of time and effort is avoided.

**1. Requirement Gathering:-** In this step, the development team must gather the requirements, by interaction with the customer. development team should plan the time and effort needed to build the project. Based on this information you can evaluate technical and economical feasibility.

**2. Design the Requirements:-** In this step, the development team will use user-flow-diagram or high-level UML diagrams to show the working of the new features and show how they will apply to the existing software. Wireframing and designing user interfaces are done in this phase.

**3. Construction / Iteration:-** In this step, development team members start working on their project, which aims to deploy a working product.

**4. Testing / Quality Assurance:-** Testing involves Unit Testing, Integration Testing, and System Testing. A brief introduction of these three tests is as follows:

**5. Unit Testing:-** Unit testing is the process of checking small pieces of code to ensure that the individual parts of a program work properly on their own. Unit testing is used to test individual blocks (units) of code.

- **Integration Testing:-** Integration testing is used to identify and resolve any issues that may arise when different units of the software are combined.
- **System Testing:-** Goal is to ensure that the software meets the requirements of the users and that it works correctly in all possible scenarios.

**5. Deployment:-** In this step, the development team will deploy the working project to end users.

**6. Feedback:-** This is the last step of the **Agile Model**. In this, the team receives feedback about the product and works on correcting bugs based on feedback provided by the customer.

## Principles of the Agile Model

- To establish close contact with the customer during development and to gain a clear understanding of various requirements, each Agile project usually includes a customer

representative on the team. At the end of each iteration stakeholders and the customer representative review, the progress made and re-evaluate the requirements.

- The agile model relies on working software deployment rather than comprehensive documentation.
- Frequent delivery of incremental versions of the software to the customer representative in intervals of a few weeks.
- Requirement change requests from the customer are encouraged and efficiently incorporated.
- It emphasizes having efficient team members and enhancing communications among them is given more importance. It is realized that improved communication among the development team members can be achieved through face-to-face communication rather than through the exchange of formal documents.
- It is recommended that the development team size should be kept small (5 to 9 people) to help the team members meaningfully engage in face-to-face communication and have a collaborative work environment.
- The agile development process usually deploys [Pair Programming](#). In Pair programming, two programmers work together at one workstation. One does coding while the other reviews the code as it is typed in. The two programmers switch their roles every hour or so.

### **Advantages of the Agile Model**

- Working through Pair programming produces well-written compact programs which have fewer errors as compared to programmers working alone.
- It reduces the total development time of the whole project.
- Agile development emphasizes face-to-face communication among team members, leading to better collaboration and understanding of project goals.
- Customer representatives get the idea of updated software products after each iteration. So, it is easy for him to change any requirement if needed.
- Agile development puts the customer at the center of the development process, ensuring that the end product meets their needs.

### **Disadvantages of the Agile Model**

- The lack of formal documents creates confusion and important decisions taken during different phases can be misinterpreted at any time by different team members.
- It is not suitable for handling complex dependencies.
- The agile model depends highly on customer interactions so if the customer is not clear, then the development team can be driven in the wrong direction.
- Agile development models often involve working in short sprints, which can make it difficult to plan and forecast project timelines and deliverables. This can lead to delays in the project and can make it difficult to accurately estimate the costs and resources needed for the project.
- Agile development models require a high degree of expertise from team members, as they need to be able to adapt to changing requirements and work in an iterative environment. This can be challenging for teams that are not experienced in agile development practices and can lead to delays and difficulties in the project.
- Due to the absence of proper documentation, when the project completes and the developers are assigned to another project, maintenance of the developed project can become a problem.

## Spiral Model

### FIGURE SHOULD BE DRAWN HERE

Spiral model is a software development process model. This model has characteristics of both iterative and waterfall models. This model is used in projects which are large and complex. This model was named spiral because if we look at its figure, it looks like a spiral, in which a long curved line starts from the center point and makes many loops around it. The number of loops in the spiral is not decided in advance but it depends on the size of the project and the changing requirements of the user. We also call each loop of the spiral a phase of the software development process.

- **Determining objectives and alternate solutions:** In the first phase, whatever requirements the customer has related to the software are collected. On the basis of which objectives are identified and analyzed and various alternative solutions are proposed.
- **Identifying and resolving risks:** In this phase, all the proposed solutions are assessed and the best solution is selected. Now that solution is analyzed and the risks related to it are identified. Now the identified risks are resolved through some best strategy.
- **Develop and test:** Now the development of software is started. In this phase various features are implemented, that is, their coding is done. Then those features are verified through testing.
- **Review and plan for the next phase:** In this phase the developed version of the software is given to the customer and he evaluates it. Gives his feedback and tells new requirements. Finally planning for the next phase (next spiral) is started.

### Advantages of Spiral Model:-

- If we have to add additional functionality or make any changes to the software, then through this model we can do so in the later stages also.
- Spiral model is suitable for large and complex projects.
- It is easy to estimate how much the project will cost.
- Risk analysis is done in each phase of this model.
- The customer can see the look of his software only in the early stages of the development process.
- Since continuous feedback is taken from the customer during the development process, the chances of customer satisfaction increases.

### Disadvantage of Spiral Model:-

- This is the most complex model of SDLC, due to which it is quite difficult to manage.
- This model is not suitable for small projects.
- The cost of this model is quite high.
- It requires more documentation than other models.
- Experienced experts are required to evaluate and review the project from time to time.
- Using this model, the success of the project depends greatly on the risk analysis phase.

## Comparison of different life cycle models in Software Engineering

Software development existence cycle models are frameworks that manual the development of software program tasks from start to completion. There are several software development existence cycle fashions, every with its personal set of benefits and drawbacks. In this answer, we will compare some of the maximum popular software development life cycle fashions, inclusive of the Waterfall version, the Agile version, and the Spiral version.

### Waterfall Model

The Waterfall model is a linear and sequential model that follows a strict series of steps inside the software improvement system. It includes five levels: Requirements accumulating and analysis, Design, Implementation, Testing, and Maintenance. Each phase has to be finished earlier than transferring on to the next phase. The Waterfall version is useful while necessities are truly defined, and modifications are not likely to arise in the course of the task. However, this version isn't always desirable for projects that require flexibility and steady changes. It also can be hard to perceive issues early on in the manner.

### **The Classical Waterfall Model**

The classical waterfall model is a linear model of software development that was introduced in the 1970s. It divides the process of making software into separate phases that must be done one after the other before going on to the next phase. The model was based on the way that manufacturing and building work is done in steps. There are several phases of the Classical waterfall model.

- **Gathering Requirements:** Project requirements are acquired from the client or stakeholders at this phase. The requirements are analyzed for further scopes, scalability, and potential risk.
- **Designing System:** This phase includes the high-level and the low-level design specification of the system's architecture.
- **Implementation of Software:** During this phase, the physical coding of the software takes place. Programmers develop code in accordance with the design specifications. This stage leads to the development of software modules and components.
- **Testing of Software:** The software is tested properly the defects, bugs and errors. Several kind of testing is performed like Unit testing, Integration testing and system testing.
- **Deployment of Software:** Once the software testing is properly done, the software is deployed to the production environment for the end user.
- **Maintenance of Software:** This is the last and the final phase of the classic waterfall model. In this phase the ongoing maintenance and the support of the software occurs.

### **Advantages of the Classical Waterfall Model**

- **Clear and Structured process:** The model is very straightforward and it is very easy to implement.
- **Documentation:** Each phase requires documentation, which aims in better understanding, knowledge transfer between the team members.
- **Well suited for small project:** Classic waterfall model works good for the small project

### **Disadvantages of the Classical Waterfall Model**

- **Inflexible:** The sequential nature of the model makes it inflexible to changes. If the requirement changes after the project goes to the next stage then it becomes so time consuming and cost effective to rewrite the new changes.
- **Variable demands are hard to meet:** This technique assumes that all client needs can be precisely specified at the outset of the project, yet customers' needs change with time. After requirements definition, amendment requests are tough.
- **Late detection of Defects:** Defects are not detected until the testing phase comes into picture and after that resolving that particular defect becomes costly.
- **Risk management:** The model's structure can lead to a lack of proper risk management.

### **The Iterative Waterfall Model**



The iterative waterfall model is the modified version of the classical waterfall model. The iterative waterfall model follows the sequential software development process. In the traditional Waterfall Model, each phase is finished before going on to the next one, and there isn't no such scope to go back to phases that have already been done. On the other hand, iterative waterfall model uses "iterations" to let comments, changes, and improvements happen during the development process.

- **Gathering Requirements:** Similar to the classical waterfall model, project requirements are acquired from the client or stakeholders at this phase. The requirements are analyzed for further scopes, scalability and potential risk.
- **Designing System:** This phase includes the high level and the low level design specification of the system's architecture.
- **Implementation of Software:** During this phase, the physical coding of the software takes place. Programmers develop code in accordance with the design specifications. This stage leads to the development of software modules and components.
- **Testing of Software:** The software is tested properly the defects, bugs and errors. Several kinds of testing is performed like Unit testing, Integration testing and system testing.
- **Evaluation Phase:** Iteration comes into play at this point. Instead of putting the software into use right after testing, stakeholders look at it. Feedback is collected, and any changes that need to be made are found.
- **Adjustment Phase:** In the adjustment phase, changes are made to the software, design, or requirements based on the comments and evaluations.
- **Reiteration:** Reiteration happens allowing for incremental improvements based on stakeholder feedback and changing requirements.

#### **Advantages of the Iterative Waterfall Model**

- **Incorporating Feedbacks:** In traditional waterfall there was no option for the feedback but Iterative waterfall model gives the privilege of working the feedback provided from one phase to the previous phase.
- **Continuous Improvement:** As the software is run over and over again, it gets better and better over time.
- **More Flexibility:** Compared to the traditional Waterfall Model, the model can better adapt to changes in needs.

#### **Disadvantages of the Iterative Waterfall Model**

- **Increased Complexity:** Keeping track of iterations and multiple rounds can make the project management process more complicated.
- **Time and Cost:** Iterations can take more time and cost more money if they are not handled well.

#### **Agile Model**

The Agile version is an iterative and incremental method to software improvement. This version is primarily based at the Agile Manifesto, which emphasises flexibility, collaboration, and rapid reaction to trade. Agile improvement involves the continuous delivery of running software program in brief iterations, commonly lasting from one to 4 weeks. The Agile version is well-desirable for initiatives with swiftly converting necessities or for groups that cost collaboration and communication. However, this model calls for an excessive degree of collaboration between group individuals, and it may be tough to control large projects.

#### **Advantages of the Agile Model**

- **Flexibility:** Agile projects has the flexible as they can easily change themselves for meeting new needs, goals and perform as per market condition.

- **Frequent Deliverables:** Agile projects help in producing software in shorter iterations, which overall impacts to see real progress and making changes easily.
- **Customer Satisfaction:** Agile helps in delivering useful features at a time, that gives customer more satisfaction.
- **Continuous Improvement:** Agile teams helps in continuously improving their processes and try to become more effective and efficient.

### Disadvantages of the Agile Model

- **Lack of Predictability:** Because of the flexibility of Agile, it is difficult to get exact estimation of project costs, timelines for some long term project.
- **Complex Project Management:** Because of making of Agile in smaller steps, skilled management of project is required to keep project goals in mind.

### Spiral Model

The Spiral version is a chance-driven version that mixes elements of each the Waterfall and Agile fashions. This model involves non-stop chance evaluation and mitigation for the duration of the software development manner. The Spiral version consists of 4 levels: Planning, Risk Analysis, Engineering, and Evaluation. Each section includes a combination of making plans, design, implementation, and trying out. This version is useful whilst managing big or complicated tasks where necessities aren't nicely understood. However, the Spiral model can be time-eating, and it may be hard to decide when to move from one segment to another.

### Selection of Appropriate Existence Cycle Model for a Venture

Selection of right lifecycle model to finish a project is the most vital assignment. It may be decided on via maintaining the benefits and drawbacks of diverse models in mind. The unique troubles that are analysed earlier than choosing an appropriate lifestyles cycle model are given beneath :

- **Characteristics of the software to be developed:** The choice of the life cycle model largely depends on the type of the software that is being developed. For small services projects, the agile model is favoured. On the other hand, for product and embedded development, the Iterative Waterfall model can be preferred. The evolutionary model is suitable to develop an object-oriented project. User interface part of the project is mainly developed through prototyping model.
- **Characteristics of the development team:** Team member's skill level is an important factor to deciding the life cycle model to use. If the development team is experienced in developing similar software, then even an embedded software can be developed using the Iterative Waterfall model. If the development team is entirely novice, then even a simple data processing application may require a prototyping model.
- **Risk associated with the project:** If the risks are few and can be anticipated at the start of the project, then prototyping model is useful. If the risks are difficult to determine at the beginning of the project but are likely to increase as the development proceeds, then the spiral model is the best model to use.
- **Characteristics of the customer:** If the customer is not quite familiar with computers, then the requirements are likely to change frequently as it would be difficult to form complete, consistent and unambiguous requirements. Thus, a prototyping model may be necessary to reduce later change requests from the customers. Initially, the customer's confidence is high on the development team. During the lengthy development process, customer confidence normally drops off as no working software is yet visible. So, the evolutionary model is useful as the customer can experience a partially working software much earlier than whole complete software. Another advantage of the evolutionary model is that it reduces the customer's trauma of getting used to an entirely new system.