**ASSIGNMENT-2**

(Important things to learn)

**TASK-2: Software Development Models**

Discuss the different software development models commonly used for creating software or websites, such as Agile, Waterfall, Scrum, etc., and their characteristics

**What is Software Modeling?**

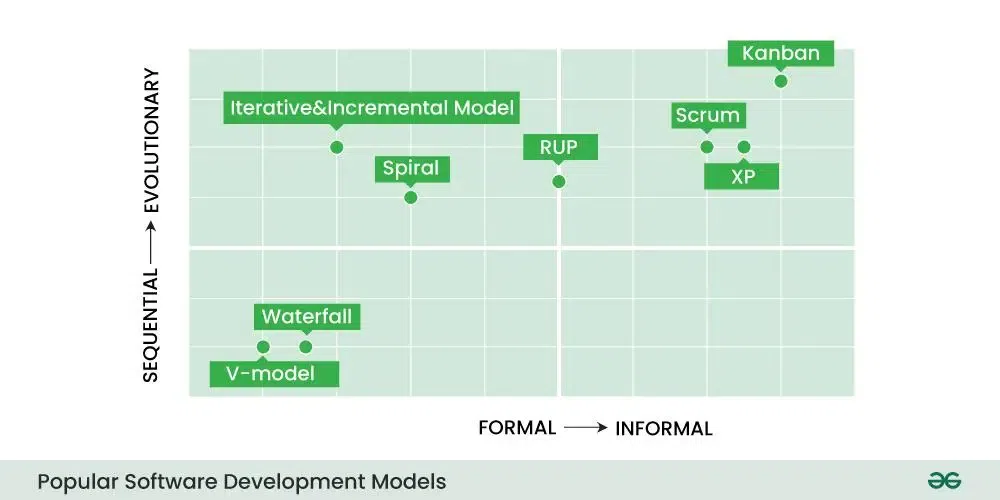
* **Software modeling** is the process of creating abstract representations of a software system. These models serve as blueprints that guide developers, designers, and stakeholders through the system’s structure, behavior, and functionality.

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

* **Software development models** are various processes or methods that are chosen for project development depending on the objectives and goals of the project. Many development life cycle models have been developed to achieve various essential objectives. Models specify the various steps of the process and the order in which they are executed.

## ****Software Development Models****

* Choosing the right model is very important for the development of a software product or application. Development and testing processes are carried out based on the model.



* Different companies, depending on the software application or product, choose the type of development model whichever is appropriate for their application. But these days [‘Agile Methodology](https://www.geeksforgeeks.org/what-is-agile-framework-and-methodology-in-software-development/)‘ is the most popular in the market.

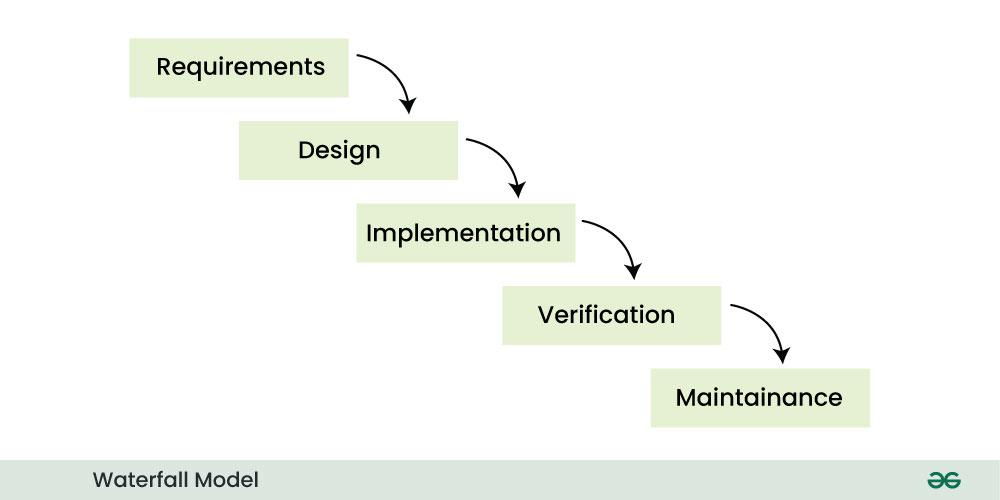
### 1. Waterfall Model

* [Waterfall model](https://www.geeksforgeeks.org/waterfall-model/) is a famous and good version of[SDLC(System Development Life Cycle)](https://www.geeksforgeeks.org/software-development/)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 waterfall model.
* We can imagine waterfall in the following way

*“Once the water starts flowing over the edge of the cliff, it starts falling down the mountain and the water cannot go back up.”*

* Similarly waterfall model also works, once one phase of development is completed then we move to the next phase but cannot go back to the previous phase. In the waterfall model, the output of one phase serves as the input for the other phase.

### Phases of Waterfall model



1. **Requirement phase:-** Requirement phase is the first phase of the [waterfall model](https://www.geeksforgeeks.org/waterfall-vs-agile-software-development-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.
2. **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.
3. **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.
4. **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.
5. **Maintenance phase:-** This is the last phase of 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.
* **Characteristics:**
* **Sequential Phases**: The development process is divided into distinct phases: Requirements, Design, Implementation, Testing, Deployment, and Maintenance.
* **Documentation**: Extensive documentation is produced at each phase.
* **No Overlapping**: Each phase must be completed before the next one begins.
* **Easy to Understand**: The straightforward structure makes it easy to manage and understand.
* **Inflexibility**: Changes are difficult to implement once a phase is completed, making it less adaptive to new requirements or unexpected challenges.

### 2. V-Model((Verification and Validation Model)

### V-Model is an SDLC model, it is also called Verification and Validation Model. V-Model is widely used in the [software development process,](https://www.geeksforgeeks.org/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.

### It is based on the association of testing phase with each development phase that is in V-Model with each development phase, its testing phase is also associated in a V-shape in other words both [software development](https://www.geeksforgeeks.org/software-development/) and testing activities take place at the same time.

### So in this model, Verification Phase will be on one side, Validation Phase will be on the other side that is both the activities run simultaneously and both of them are connected to each other in V-Shape through Coding Phase, hence it is called V-Model.

* **V-Design:** In V-Design the left side represents the development activity, the right side represents the testing activity.

**Phases of V-model**



1. **Requirements analysis:-**This is the first phase of the development cycle, in which the requirements of the product are analyzed 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.
2. **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 .
3. **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).
4. **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.
5. **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.

### I****ntegration 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.

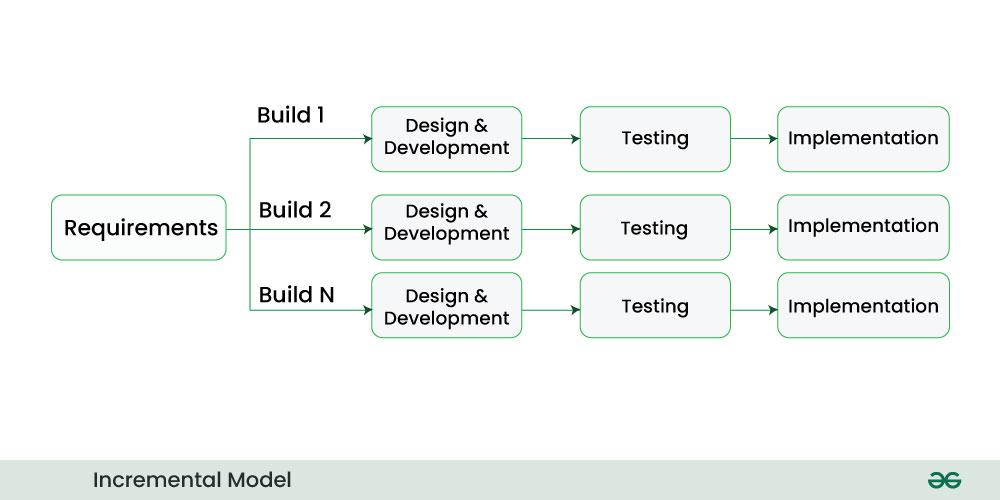
### Advantages of V-Model

* 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.
* **Characteristics:**
* **Parallel Phases**: Corresponding testing phases are associated with each development phase.
* **Strict Documentation**: Extensive documentation and predefined deliverables.
* **Early Testing**: Testing is planned in parallel with development to catch defects early.
* **Rigid Structure**: Like Waterfall, it is less flexible in accommodating changes once a phase is completed.

### 3. Incremental Model

* In Incremental Model, the [software development process](https://www.geeksforgeeks.org/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.
* For example, we collect the customer’s requirements, now instead of making the entire software at once, we first take some requirements and based on them create a module or function of the software and deliver it to the customer. Then we take some more requirements and based on them add another module to that software.
* Similarly, modules are added to the software in each increment until the complete system is created. However, the requirements for making a complex project in multiple iterations/parts should be clear.
* If we understand the entire principle of Incremental methodology, then it starts by developing an initial implementation, then user feedback is taken on it, and it is developed through several versions until an accepted system is developed. Important functionalities of the software are developed in the initial iterations.
* Each subsequent release of a software module adds functions to the previous release. This process continues until the final software is obtained.

### Phases of Incremental Model

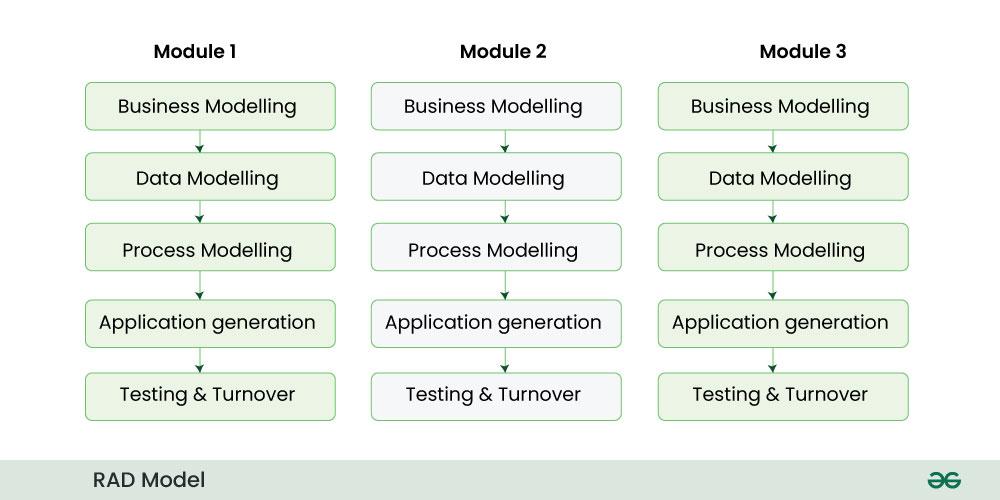


1. **Communication**: In the first phase, we talk face to face with the customer and collect his mandatory requirements. Like what functionalities does the customer want in his software, etc.
2. **Planning**: In this phase the requirements are divided into multiple modules and planning is done on their basis.
3. **Modeling**: 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.
4. **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.
5. **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)](https://www.geeksforgeeks.org/software-development/).
* 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.
* **Characteristics:**
* **Phased Development**: The project is divided into small, manageable segments (increments) that are developed and delivered in phases.
* **Partial Implementations**: Each increment is a part of the entire system, providing partial system functionality and improvements with each phase.
* **Iterative Process**: Each increment builds on the previous, gradually adding more features and functionality.
* **Parallel Development**: Different increments can be developed in parallel, allowing multiple teams to work simultaneously.
* **User Feedback**: Regular user feedback is incorporated into the development process, helping to refine requirements and improve subsequent increments.
* **Flexibility**: More flexible than the Waterfall model as it allows for changes and new requirements to be incorporated in future increments.
* **Risk Management**: Reduces risk by delivering portions of the software early, allowing issues to be identified and addressed sooner.
* **Testing and Integration**: Each increment is thoroughly tested and integrated with the previous increments, ensuring that the complete system functions correctly.

**4. RAD Model**

* RAD model stands for rapid application development model. The methodology of RAD model is similar to that of incremental or waterfall model. It is used for small projects.
* If the project is large then it is divided into many small projects and these small projects are planned one by one and completed. In this way, by completing small projects, the large project gets ready quickly.
* In RAD model, the project is completed within the given time and all the requirements are collected before starting the project. It is very fast and there are very less errors in it.
* The main objective of RAD model is to reuse code, components, tools, processes in project development.

****

**Phases RAD model**

1. **Business modeling:** In this phase, the business model is designed on the basis of whatever functions the business has. If we speak in a little technical language, then we design the business model for the product on the basis of flow of information and distribution of information between different business channels. Here information flow means what type of information drives the business, where the information comes from and where it goes, who generates it, etc. This means that a complete business analysis is done in this phase.
2. **Data modeling:**Using the business model we had prepared, the data objects required for the business are defined.
3. **Process modeling:** The data objects that we defined in the data modeling phase are converted to establish the business information flow. It is necessary to achieve specific business objectives.
4. **Application generation:**In this phase we start building the software based on the output of the above three phases. For this we take the help of automation tools. However, in this phase we do not develop the actual software but make a working prototype.
5. **Testing and turnover:**Whatever prototype we have prepared or whatever components and interfaces we have, they are tested in this phase. Since prototypes are tested separately during each iteration, the overall testing time in rapid application development is reduced.

### Advantage of RAD Model

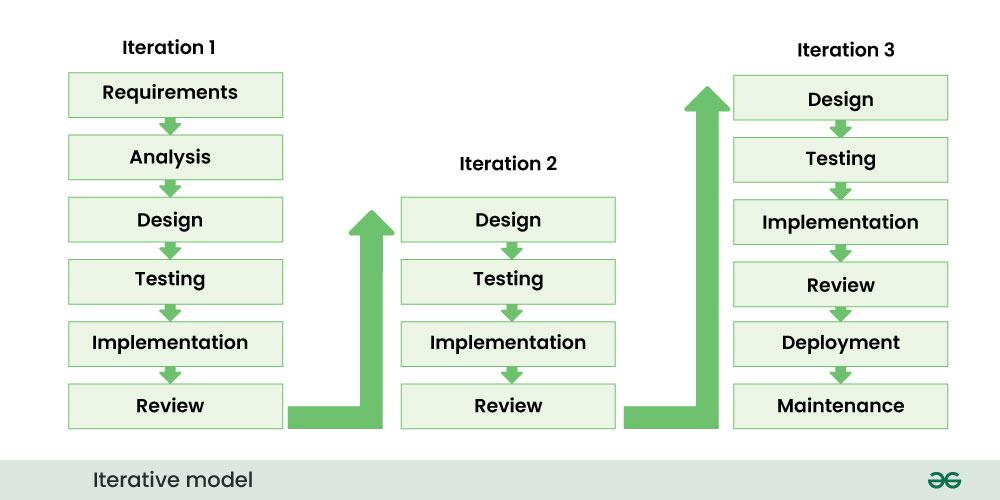
* It reduces the time taken in development.
* In this the components are reused.
* It is flexible and it is easy to make any changes in it.
* It is easy to transfer like scripts because high level abstraction and intermediate codes are used in it.
* There are very few defects in it because it is a prototype by nature.
* In this, productivity can be increased in less time with less people.
* It is cost effective.
* It is suitable for small projects.

### Disadvantages of RAD Model

* In this we need highly skilled developers and designers.
* It is very difficult to manage.
* It is not suitable for project that are complex and takes long time.
* In this, feedback from the client is required for the development of each phase.
* Automated code generation is very expensive.
  + This model is suitable only for component based and scalable systems.
* **Characteristics:**
  + **Prototyping**: Relies heavily on creating prototypes that can be quickly developed, demonstrated, and refined based on user feedback.
  + **User Involvement**: Users are closely involved throughout the development process, providing continuous feedback and validation.
  + **Iterative Development**: The development cycle includes multiple iterations, with each cycle producing a more refined version of the software.
  + **Component-Based Construction**: Uses pre-existing components and modules to speed up development.
  + **Flexibility and Adaptability**: Highly adaptable to changes in requirements and user needs, allowing for modifications at any stage.
  + **Time-Boxed Development**: Each iteration or phase has a fixed timeline, ensuring rapid delivery of functional components.
  + **Focus on Functionality**: Prioritizes the development of functional elements over detailed planning and documentation.
  + **Team Collaboration**: Encourages close collaboration between developers, designers, and users, often using workshops and joint application development sessions.

### 5. Iterative Model

* In 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.
* So, in Iterative model a software is developed by following several iterations. Iteration means that we are repeating the [development process](https://www.geeksforgeeks.org/software-development-process/) again and again. For example, we develop the first version of the software following the SDLC process with some software requirements.
* After the first version is developed, if there is a need to change the software , then a new version is developed with the second iteration. Now again we will see if the new version is enough, if not then we will make changes in it with the third iteration. The iteration will be repeated until the complete software is ready.
* The basic concept of Iterative model is that the software should be developed through repeated cycles or what we also call iteration and only a small part of it should be developed at a time. This model was developed to overcome the drawbacks of the classical waterfall model.



**Phases of iterative model**

1. **Requirement gathering & analysis:**In this phase, all the software requirements of the customer are collected and it is analyzed 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 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.
* **Characteristics :**
* **Repetition of Phases**:Development occurs in iterations, with each iteration going through phases similar to the traditional Waterfall model: planning, design, implementation, and testing.
* **Incremental Development**: The system is developed and delivered in small segments. Each iteration adds incremental improvements and functionality to the system.
* **Refinement through Feedback**: Feedback from the end-users or stakeholders is gathered after each iteration, allowing for modifications and enhancements in the subsequent iterations.
* **Risk Management**: Early iterations tackle high-risk areas, allowing potential issues to be identified and mitigated early in the development process.
* **Flexibility**: The model is adaptable to changing requirements. New features and changes can be incorporated into the system during subsequent iterations.
* **Parallel Development**: Different components or modules can be developed in parallel, increasing efficiency and reducing the overall development time.
* **Frequent Deliveries**: Working versions of the software are delivered regularly, providing tangible progress and allowing for continuous user evaluation and feedback.
* **Focus on Core Architecture**: Initial iterations often focus on building a robust core architecture, which can be expanded and improved in later iterations.
* **Continuous Testing and Integration**: Each iteration involves thorough testing and integration of new components with the existing system, ensuring that the entire system remains functional and coherent.
* **Improvement Over Time**: The system evolves through each iteration, gradually becoming more complete and refined. This allows for continuous improvement and optimization.

**6. Spiral Model**

* Spiral model is a [software development process](https://www.geeksforgeeks.org/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.
* A software project goes through these loops again and again in iterations. After each iteration a more and more complete version of the software is developed. The most special thing about this model is that risks are identified in each phase and they are resolved through prototyping. This feature is also called Risk Handling.
* Since it also includes the approaches of other SDLC models, it is also called Meta Model. It was first developed by Barry Boehm in 1986.



* In Spiral Model the entire process of software development is described in four phases which are repeated until the project is completed.

**Phases of Spiral Model:**

1. **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.
2. **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.
3. **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.
4. **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

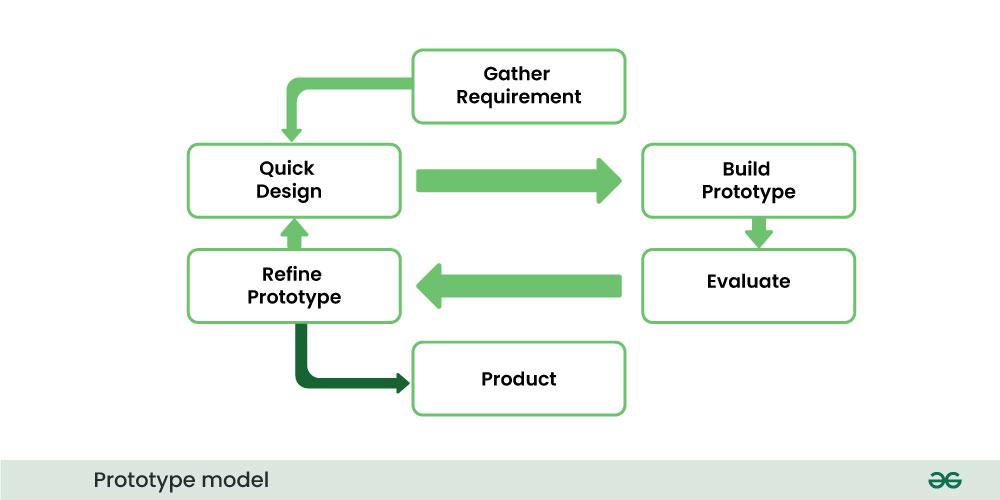
* If we have to add additional functionality or make any changes to the software, then through this model we can do so in the later stages also.
* Spiral model is suitable for large and complex projects.
* It is easy to estimate how much the project will cost.
* Risk analysis is done in each phase of 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.
* **Characteristics:**
* **Risk Management**: Focuses on early identification and mitigation of risks.
* **Iterations**: Development proceeds through iterative cycles (spirals), each addressing a set of objectives and risks.
* **Customer Involvement**: Regular customer feedback after each iteration.
* **Phases**: Each spiral includes planning, risk analysis, engineering, and evaluation.

### 7. Prototype model

* 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.
* One problem in this model is that if the end users are not satisfied with the prototype model, then a new prototype model is created again, due to which this model consumes a lot of money and time.



### Phases of Prototype Model:

1. **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.
2. **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.
3. **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.
4. **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.

### 8. Agile Model

* Agile model is a combination of iterative and incremental models, that is, it is made up of iterative and incremental models.
* In Agile model, focus is given to process adaptability and customer satisfaction.
* In earlier times, iterative waterfall model was used to create software. But in today’s time developers have to face many problems. The biggest problem is that in the middle of software development, the customer asks to make changes in the software. It takes a lot of time and money to make these changes.
* So to overcome all these shortcomings, the agile model was proposed in the 1990s.
* The agile model was created mainly to make changes in the middle of [software development](https://www.geeksforgeeks.org/software-development/) so that the software project can be completed quickly.



* In the agile model, the software product is divided into small incremental parts. In this, the smallest part is developed first and then the larger one.
* And each incremental part is developed over iteration.
* Each iteration is kept small so that it can be easily managed. And it can be completed in two-three weeks. Only one iteration is planned, developed and deployed at a time.

### Agile has the following models

* Scrum
* Crystal methods
* DSDM
* Feature driven development (FDD
* Lean [software development](https://www.geeksforgeeks.org/software-development/)
* Extreme programming (xp)

### Advantages of Agile Model

* In this, two programmers work together due to which the code is error free and there are very few mistakes in it.
* In this the software project is completed in a very short time.
* In this the customer representative has an idea of ​​each iteration so that he can easily change the requirement.
* This is a very realistic approach to software development.
* In this, focus is given on teamwork.
* There are very few rules in this and documentation is also negligible.
* There is no need for planning in this.
* It can be managed easily.
* It provides flexibility to developers.

### Disadvantages of Agile Model

* It cannot handle complex dependencies.
* Due to lack of formal documentation in this, there is confusion in development.
* It mostly depends on the customer representative, if the customer representative gives any wrong information then the software can become wrong.
* Only experienced programmers can take any decision in this. New programmers cannot take any decision.
* In the beginning of software development, it is not known how much effort and time will be required to create the software.
* **Characteristics:**
* **Iterative and Incremental**: Development occurs in small, incremental cycles or iterations.
* **Flexibility**: Highly adaptive to changing requirements even late in the development process.
* **Customer Collaboration**: Continuous customer involvement and feedback throughout the development.
* **Self-Organizing Teams**: Teams are cross-functional and self-organizing, promoting a collaborative work environment.
* **Frequent Deliveries**: Working software is delivered frequently (e.g., every two weeks).