**SDLC Models ( Questions )**

**Software Engineering**

1. **What is the Software Development Life Cycle (SDLC)?\*\*\***

- The Software Development Life Cycle (SDLC) is a step-by-step process used to create software. It includes all the stages needed to develop a software product, from start to finish, and shows the order in which these steps should be done. This helps in organizing and managing the development process effectively.

1. **What are the key phases of the Waterfall Model?**

- The key phases of the Waterfall Model are:

* **Requirement Gathering and Analysis:** Collect and understand what the software needs to do.
* **System Design:** Create a plan or blueprint for how the software will work.
* **Implementation:** Write the actual code to build the software.
* **Integration and Testing:** Combine all parts of the software and test for errors.
* **Deployment of System:** Release the software for users to start using.
* **Maintenance:** Fix issues and update the software after it’s released.

1. **What is the disadvantage of the Waterfall Model?**

-

* No working software is produced until late during the life cycle.
* High amounts of risk and uncertainty.
* Not a good model for complex and object-oriented projects.
* Poor model for long and ongoing projects.
* Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
* It is difficult to measure progress within stages.
* Cannot accommodate changing requirements.

1. **What is the advantage of waterfall model?**

- The Waterfall Model has several advantages, especially for projects with well-defined requirements:

* Simple and easy to understand and use
* Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
* Phases are processed and completed one at a time.
* Works well for smaller projects where requirements are very well understood.
* Clearly defined stages.
* Well understood milestones.
* Easy to arrange tasks.
* Process and results are well documented.

1. **Explain Iterative Model.**

**-** The Iterative model is a way of developing software where you start by creating a basic version of the software with just a few features. After that, you keep improving and adding more features step by step. Each step is called an iteration, and at the end of each iteration, you have a new, better version of the software. This process continues until the software is complete and ready to use.

1. **How does the Iterative Model differ from the Waterfall Model?**

- The Iterative Model is different from the Waterfall Model because it begins with a basic version of the software and then improves it step by step until the whole system is done. In the Waterfall Model, you have to finish one phase completely before moving to the next, with no going back or overlapping. The Iterative Model allows for more flexibility and adjustments along the way.

1. **What is the advantage of the Iterative Model?**

-

* Develop high risk or major function first.
* Each release delivers an operational product.
* Customer can respond to each build.
* Uses “divide and Conquer” break down of task.
* Lower initial delivery cost.
* Initial product delivery is faster.
* Customers get important functionality early.
* Risk of changing requirements is reduced.

1. **What is the disadvantage of Iterative Model?**

**-**

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

1. **Explain Spiral Model.**

**-** The Spiral Model is a way to develop software that combines the step-by-step approach of the waterfall model with the iterative process of making improvements over time. It focuses a lot on identifying and managing risks throughout the development process.

The key feature of the Spiral Model is its emphasis on risk analysis. Before moving forward in each cycle, you assess potential risks and figure out how to handle them. This makes it a good choice for complex projects where risks need to be carefully managed.

1. **When to use Iterative Model?**

**-**

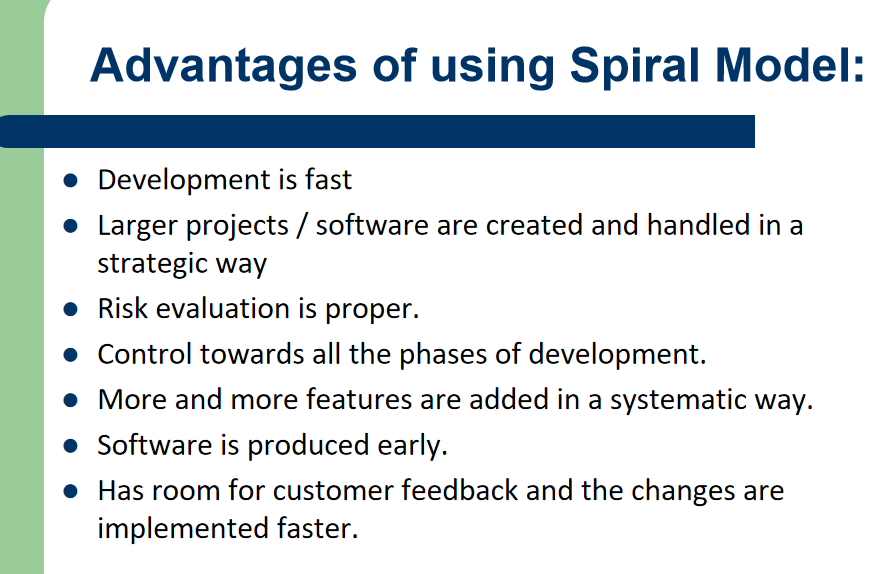
* Most of the requirements are known up-front but are expected to evolve over time.
* A need to get basic functionality to the market early.
* On projects which have lengthy development schedules.
* On a project with new technology.
* Mostly such model is used in web applications.

1. **When to use Spiral?**

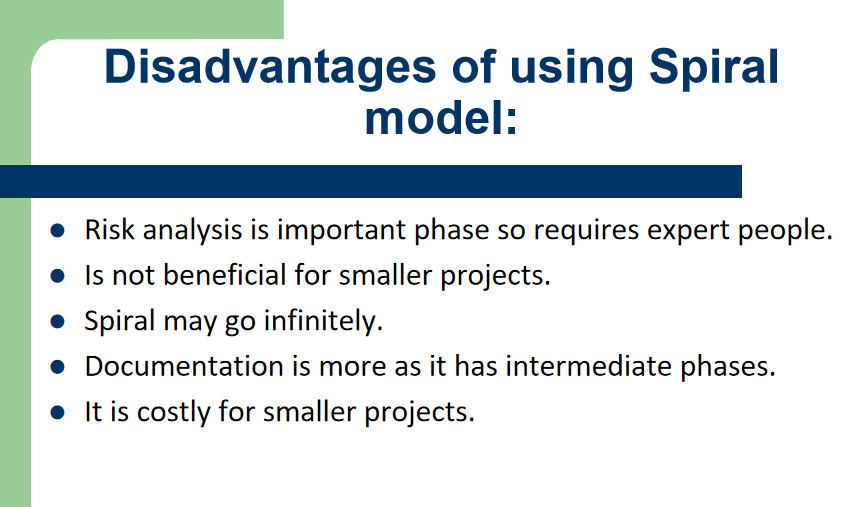
**-**

* When the project is large.
* Where the software needs continuous risk evaluation.
* Requirements are a bit complicated and require continuous clarification.
* Where enough time frame is there to get end user feedback.
* Where releases are required to be frequent.

1. **Advantage of Spiral Model.**

**-**

1. **Disadvantage of Spiral Model.**

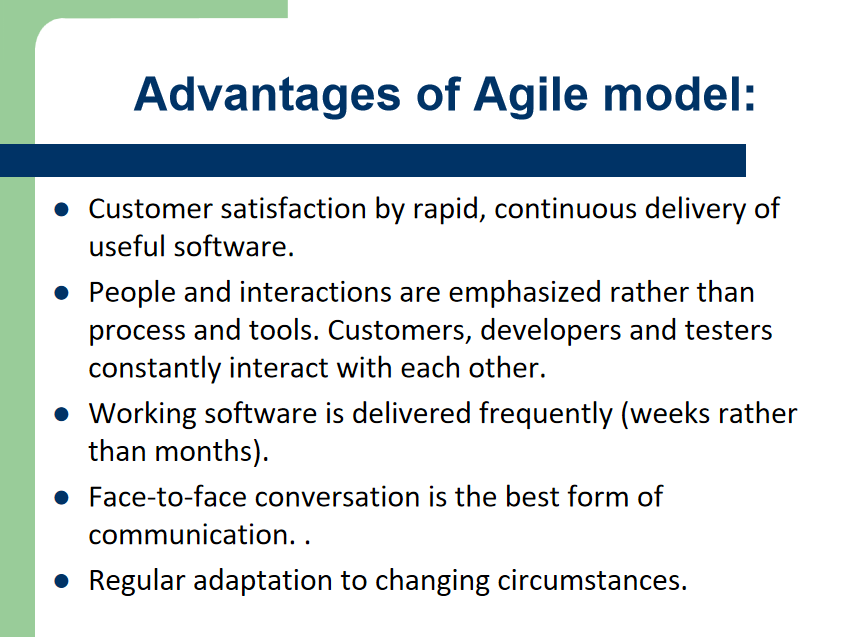
**-** 

1. **Explain Agile Model.**

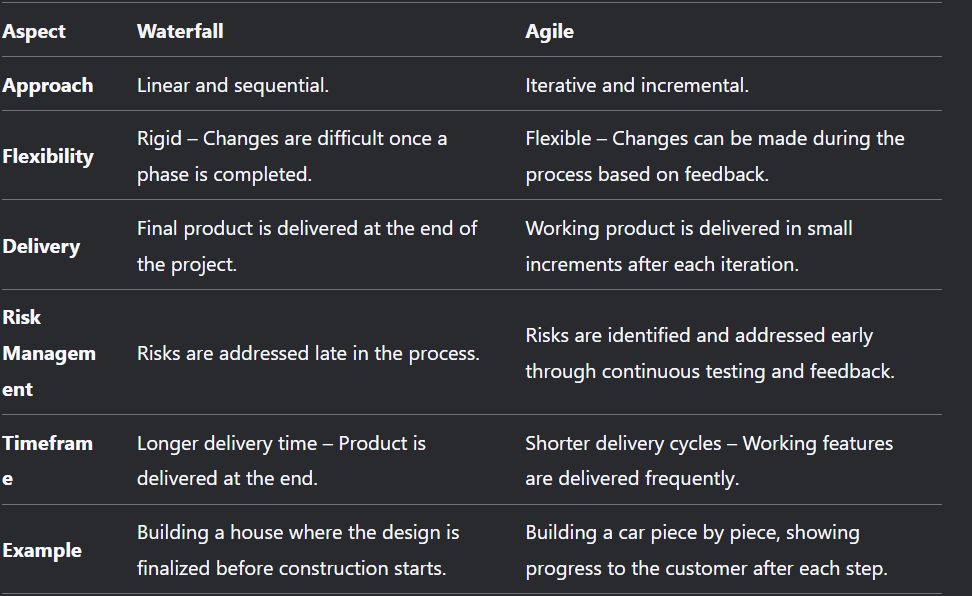
**-** The Agile Model is a way to develop software that focuses on delivering small, working parts of the product quickly and regularly. It combines iterative and incremental approaches, meaning you build the software in small steps and keep improving it over time.

The main goals of the Agile Model are to adapt to changes easily and keep the customer happy by delivering useful software quickly.

1. **Advantage of Agile Model.**

**-** 

1. **Difference between Agile & Waterfall model.**

**-** 

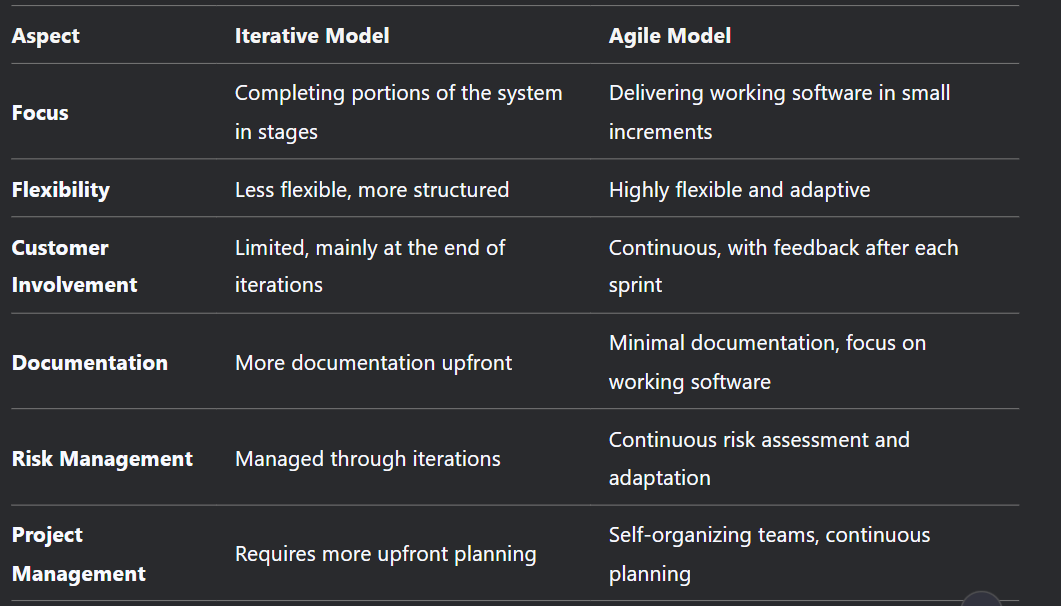
1. **Why is risk analysis important in the Spiral Model?**

**-** Risk analysis is crucial in the Spiral Model because it helps the team spot and handle risks early on. In each cycle of development, the team checks for potential problems and figures out how to deal with them. This ongoing risk assessment allows for adjustments to the project plan, budget, and timeline, helping to keep the project on track and reducing the chances of major issues later. This proactive approach makes the Spiral Model effective for managing complex and risky projects.

1. **What is the key benefit of using the Iterative Model for web applications?**

**-** The main advantage of using the Iterative Model for web applications is that it lets you deliver a basic version of the product early. This initial version can then be improved and expanded over time. This approach is especially helpful for web applications because their requirements often change, and there’s usually a need to launch the product quickly. By starting with a simple version and adding features gradually, you can respond to user feedback and adapt to changing needs more effectively.

1. **What is the difference between the Iterative Model and the Agile Model?**

**-** 

1. **If all the requirements are given, which model is perfect for that? Write the advantages and disadvantages of that model.**

**-** If all the requirements are clearly defined and unlikely to change, the Waterfall Model is the most suitable model for the project.

**Advantages:**

* Simple and easy to follow.
* Well-structured with clear phases.
* Good for small, stable projects.
* Thorough documentation.
* No overlapping phases.

**Disadvantages:**

* Hard to change once a phase is done.
* Working software comes late.
* High risk if requirements change.
* Not good for complex projects.
* Limited customer feedback until the end.
* Integration issues may arise late.

1. **What is SDLC? Explain the framework of SDLC in simple words. ( Combined with slide 2 )**

**-** SDLC (Software Development Life Cycle) is a step-by-step process used to design, develop, and test software. It ensures the software is built correctly and meets user needs.

**Framework of SDLC:**

**Requirement Gathering:** Understand what the customer wants.

Example: For a pizza app, ask for features like ordering and tracking.

**System Design:** Plan how the software will work.

Example: Design the app layout and features.

**Implementation (Coding):** Write the code to build the software.

Example: Code the app to order pizza or track delivery.

**Testing:** Check for bugs and fix them.

Example: Test if the "Order Now" button works.

**Deployment:** Release the software to users.

Example: Launch the app on the App Store.

**Maintenance:** Fix issues and add new features.

Example: Update the app to add "extra cheese" option.

Why is SDLC Important?

* Keeps the team organized.
* Ensures the software meets customer needs.
* Catches problems early.

**Summary:**

SDLC is a roadmap to build software step by step, from understanding needs to releasing and maintaining the product.

1. **When to use agile model.**

The Agile model is like building something step by step, where you can make changes easily along the way. Here’s when it’s useful:

* **When Changes Happen Often**: If you need to keep updating or changing things, Agile is great because it lets you make changes without starting over or spending a lot of extra time or money.
* **Quick Fixes and Updates**: If you want to add something new or fix a mistake, you only need to go back a little bit in your work (like a few hours or days) instead of redoing everything.
* **Less Planning at the Start**: Agile doesn’t need a lot of planning before you begin. It’s good for projects where what people want might change as you go along, especially in fast-moving fields like tech or business.