1.What is Agile model (4m/12m) ?

**Agile model**

1. Agile process model" refers to a software development approach based on iterative development.

2. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning.

3. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

**Iterations in agile model**

Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements.

Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.



Phases of Agile Model:

1. Requirements gathering
2. Design the requirements
3. Construction/ iteration
4. Testing/ Quality assurance
5. Deployment
6. Feedback
   1. **Requirements gathering:** In this phase, you must define the requirements. You should explain business opportunities and plan the time and effort needed tobuild the project. Based on this information, you can evaluate technical and economic feasibility.

**2. Design the requirements:** When you have identified the project, work with stakeholders to define requirements. You can use the user flow diagram or the high-level UML diagram to show the work of new features and show how it will apply to your existing system.

**3. Construction/ iteration:** When the team defines the requirements, the work begins. Designers and developers start working on their project, which aims to deploy a working product. The product will undergo various stages of improvement, so it includes simple, minimal functionality.

**4. Testing:** In this phase, the Quality Assurance team examines the product's performance and looks for the bug.

**5. Deployment:** In this phase, the team issues a product for the user's work environment.

**6. Feedback:** After releasing the product, the last step is feedback. In this, the team receives feedback about the product and works through the feedback.

Principles of Agile model (4 mark)

**Agility Principles:**

The Agile Alliance defines twelve lightness principles for those that need to attain agility:

1. Our highest priority is to satisfy the client through early and continuous delivery of valuable computer software.
2. Welcome dynamical necessities, even late in development. Agile processes harness modification for the customer’s competitive advantage.
3. Deliver operating computer software often, from a pair of weeks to a couple of months, with a preference to the shorter timescale.
4. Business individuals and developers should work along daily throughout the project.
5. The build comes around actuated people. offer them the setting and support they have, and trust them to urge the task done.
6. the foremost economical and effective methodology of conveyancing info to and among a development team is face-to-face speech.
7. working computer software is the primary live of progress.
8. Agile processes promote property development. The sponsors, developers, and users got to be able to maintain a relentless pace indefinitely.
9. Continuous attention to technical excellence and smart style enhances nimbleness.
10. Simplicity—the art of maximizing the number of work not done—is essential.
11. the most effective architectures, necessities, and styles emerge from self– organizing groups.
12. At regular intervals, the team reflects on a way to become simpler, then tunes and adjusts its behavior consequently.
    1. Define Spiral Model ( 12 mark confirm)

**Spiral model** is one of the most important Software Development Life Cycle models, which provides support for **Risk Handling**.

The exact number of loops of the spiral is unknown and can vary from project to project. Each loop of the spiral is called a **Phase of the software development process.**

 The exact number of phases needed to develop the product can be varied by the project manager depending upon the project risks

Each phase of the Spiral Model is divided into four quadrants

* 1. **Objectives determination and identify alternative solutions**
  2. **Identify and resolve Risks**
  3. **Develop next version of the Product**
  4. **Review and plan for the next Phase**

The Spiral model is called a Meta-Model because it subsumes all the other SDLC models.



**Advantages of Spiral Model**:   
Below are some advantages of the Spiral Model.

1. **Risk Handling:** The projects with many unknown risks that occur as the development proceeds, in that case, Spiral Model is the best development model to follow due to the risk analysis and risk handling at every phase.
2. **Good for large projects:** It is recommended to use the Spiral Model in large and complex projects.
3. **Flexibility in Requirements:** Change requests in the Requirements at later phase can be incorporated accurately by using this model.
4. **Customer Satisfaction:** Customer can see the development of the product at the early phase of the software development and thus, they habituated with the system by using it before completion of the total product.

**Disadvantages of Spiral Model**:   
Below are some main disadvantages of the spiral model.

1. **Complex:** The Spiral Model is much more complex than other SDLC models.
2. **Expensive:** Spiral Model is not suitable for small projects as it is expensive.
3. **Too much dependability on Risk Analysis:** The successful completion of the project is very much dependent on Risk Analysis. Without very highly experienced experts, it is going to be a failure to develop a project using this model.

3.Define Waterfall Model

The waterfall model is the simplest model of software development paradigm. It says that all the SDLC phases will linearly function one after another. When the first phase is finished, only the second phase will start and so on. This model assumes that everything is carried out and taken place perfectly as planned in the previous stage and there is no need to think about the past issues that may arise in the next phase. This model does not work smoothly if some issues are left at the previous step. The sequential nature of the model does not allow us to undo or redo our actions. This model is best suited when developers already have designed and developed similar software in the past and are aware of all its domains.

Graphical user interface, text

Description automatically generated

**Iterative Model**

This model leads the software development process in iterations. It projects the process of development in cyclic manner repeating every step after every cycle of SDLC process. The software is first developed on very small scale and all the steps are followed which are taken into consideration. Then, more features and modules are designed, coded, tested, and added to the software on every next iteration. Every cycle produces a software that is complete in itself and has more features and capabilities than the previous one.

Diagram

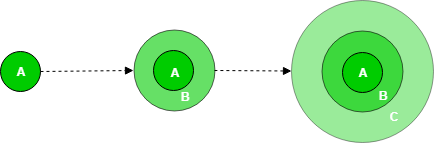
Description automatically generated

After each iteration, the management team can do work on risk management and prepare for the next iteration. Because a cycle includes small portion of whole software process, it is easier to manage the development process but it consumes more resources.

1. Define incremental process model (12 mark)

Incremental process model is also known as the Successive version model.

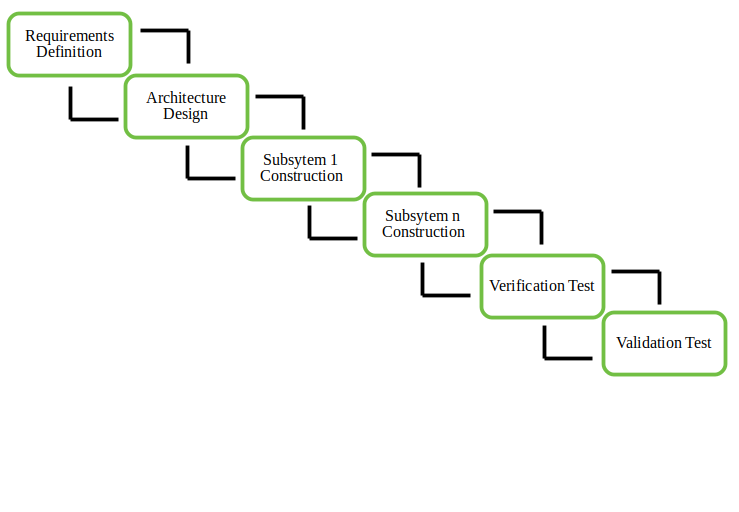
First, a simple working system implementing only a few basic features is built and then that is delivered to the customer. Then thereafter many successive iterations/ versions are implemented and delivered to the customer until the desired system is released.



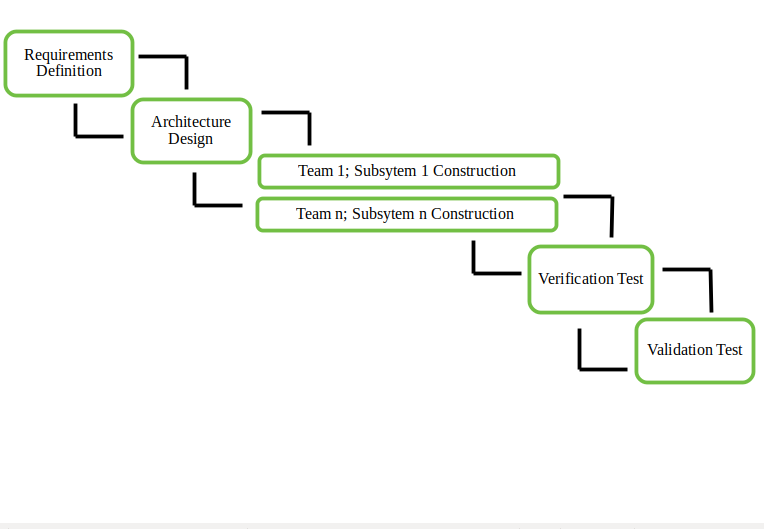
A, B, C are modules of Software Products that are incrementally developed and delivered.

**Types of Incremental model –** 

1. **Staged Delivery Model –**Construction of only one part of the project at a time.



1. **Parallel Development Model –**Different subsystems are developed at the same time. It can decrease the calendar time needed for the development, i.e. TTM (Time to Market) if enough resources are available.



1.Incremental Model is a software development model where the product is, analyzed, designed, implemented and tested incrementally until the product is finished

2.In incremental model requirements and early stage planning is necessary and Incremental model can’t handle large project.

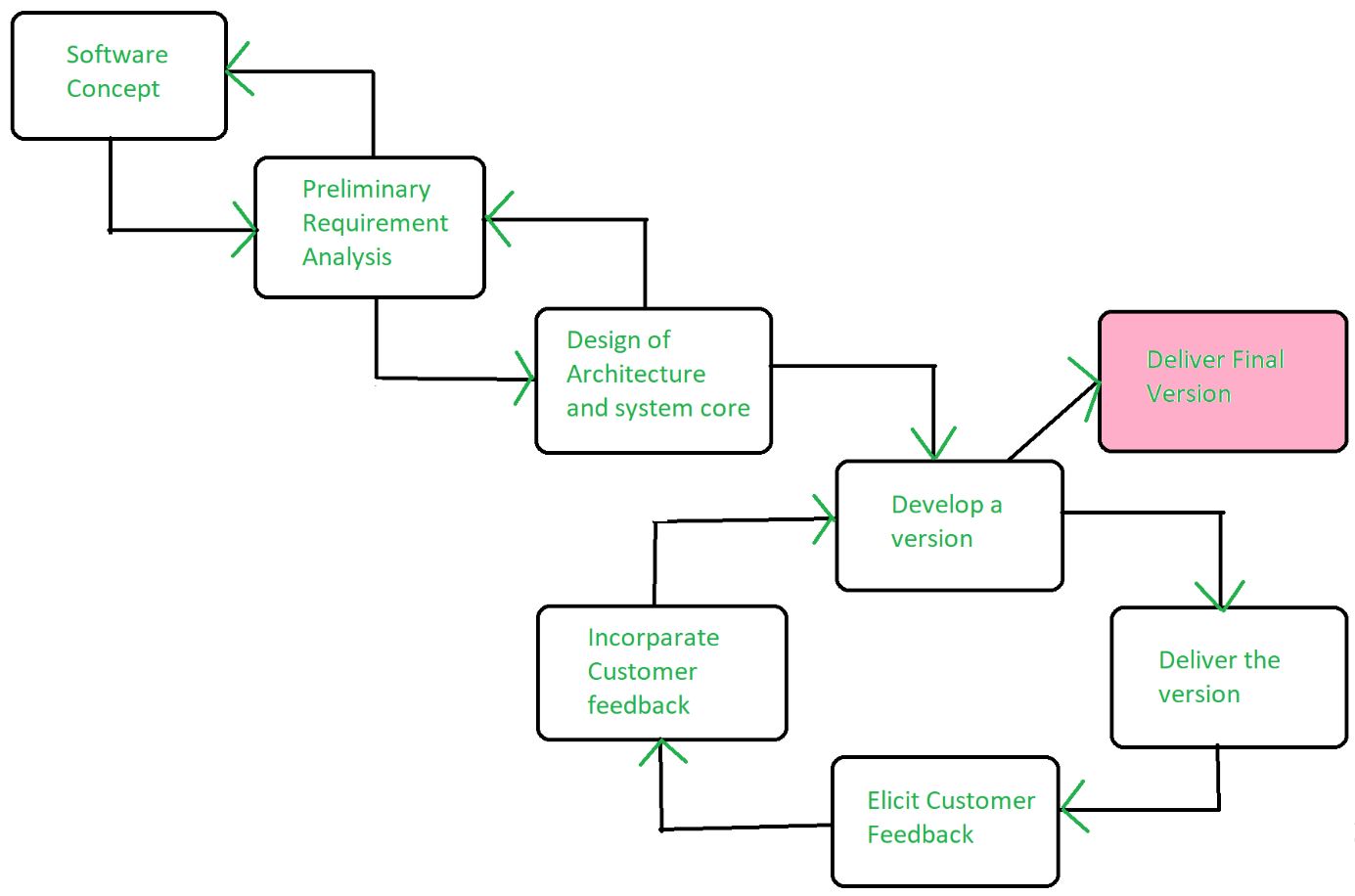
3.In incremental model large team is not required.

4.Flexibility to change in incremental model is Easy.

1. Define Evaluationary model

**Evolutionary model** is a combination of [Iterative](https://www.geeksforgeeks.org/software-engineering-iterative-waterfall-model/)and [Incremental model](https://www.geeksforgeeks.org/software-engineering-incremental-process-model/) of software development life cycle. Delivering your system in a big bang release, delivering it in incremental process over time is the action done in this model. Some initial requirements and architecture envisioning need to be done.

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. Therefore, the software product evolves with time.  
All the models have the disadvantage that the duration of time from start of the project to the delivery time of a solution is very high. Evolutionary model solves this problem in a different approach.



Evolutionary model suggests breaking down of work into smaller chunks, prioritizing them and then delivering those chunks to the customer one by one. The number of chunks is huge and is the number of deliveries made to the customer.

**Application of Evolutionary Model:**

1. It is used in large projects where you can easily find modules for incremental implementation. Evolutionary model is commonly used when the customer wants to start using the core features instead of waiting for the full software.
2. Evolutionary model is also used in object oriented software development because the system can be easily portioned into units in terms of objects.

**Advantages:**

* In evolutionary model, a user gets a chance to experiment partially developed system.
* It reduces the error because the core modules get tested thoroughly.

**Disadvantages:**

* Sometimes it is hard to divide the problem into several versions that would be acceptable to the customer which can be incrementally implemented and delivered.