MODULE 1: Introduction to SE and Process model

Q.1 What is Software Process Framework ?

=>
A Software Process Framework is a structured approach that defines the steps, tasks, and activities involved in software development.

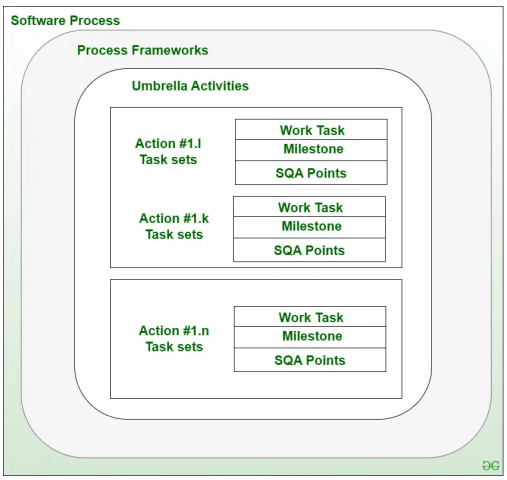
This framework serves as a foundation for software engineering, guiding the development team through various stages to ensure a systematic and efficient process.

A Software Process Framework helps in project planning, risk management, and quality assurance by detailing the chronological order of actions.

It includes task sets, umbrella activities, and process framework activities, all essential for a successful software development lifecycle.

Utilising a well-defined Software Process Framework enhances productivity, consistency, and the overall quality of the software product.

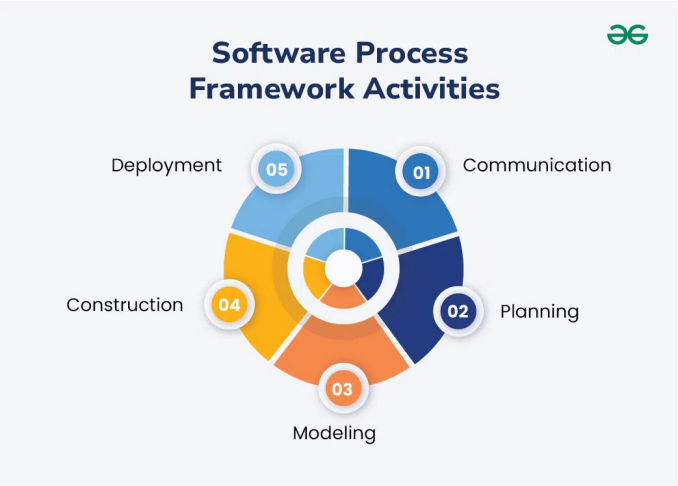
Software Process includes:
Tasks: They focus on a small, specific objective.
Action: It is a set of tasks that produce a major work product.
Activities: Activities are groups of related tasks and actions for a major objective.



The Software process framework is required for representing common process activities. Five framework activities are described in a process framework for software engineering.

1. Communication
2. planning
3. modeling
4. construction
5. and deployment.

These are all examples of framework activities. Each engineering action defined by a framework activity comprises a list of needed work outputs, project milestones, and software quality assurance (SQA) points.



1. Communication:
   Gathering requirements from customers and stakeholders to define the system's objectives.
   - Activities:
     - Requirement Gathering: Meetings, interviews, surveys.
     - Objective Setting: Defining system goals.
   - Explanation: Ensures all stakeholders are aligned on the system's goals.

2. Planning:
   Establishing a detailed work plan, identifying risks, resources, and setting a timeline.
   - Activities:
     - Work Plan, Risk Assessment, Resource Allocation, Schedule Definition.
   - Explanation: Organizes the project and prepares for potential challenges.

3. Modeling:
   Creating designs and models to define the system structure and functions.
   - Activities:
     - Requirement Analysis, System Design.
   - Explanation: Translates requirements into a visual blueprint for development.

4. Construction:
   Building and testing the software to ensure it meets requirements.
   - Activities:
     - Code Generation, Testing, Bug Fixing.
   - Explanation: The actual development and testing of the software.

5. Deployment:
   Delivering the product to users and gathering feedback for improvement.
   - Activities:
     - Product Release, Feedback Collection, Product Improvement.
   - Explanation: Ensures the product meets user expectations.

Q.2 EXPLAIN CMM ?

=>
The SEI capability maturity model is a process meta model developed by software engineering Institute(SEI).

It defines the process characteristic that should exist if an organisation want to establish a software process that is complete the period of the SEI capability.
Maturity model should always be adopted, it argues that software development:

- It must be taken seriously.
- It must be thoroughly.
- It must be controlled uniformly.
- It must be tracked accurately.
- It must be conducted professionally.
- It must focus on the needs of its customer.

The SEI capability maturity module represents two type of meta models:
as a continuous model
as a staged model.

As a Continuous model
It describes the process in two dimensions as shown in the figure. Each process area are assessed against specific goal and practises and is rated according to the following capability levels:

Level 0: Incomplete
the process area is either not performed or does not achieve all goals and objectives defined by SEI capability. Maturity model for level 1 capability.

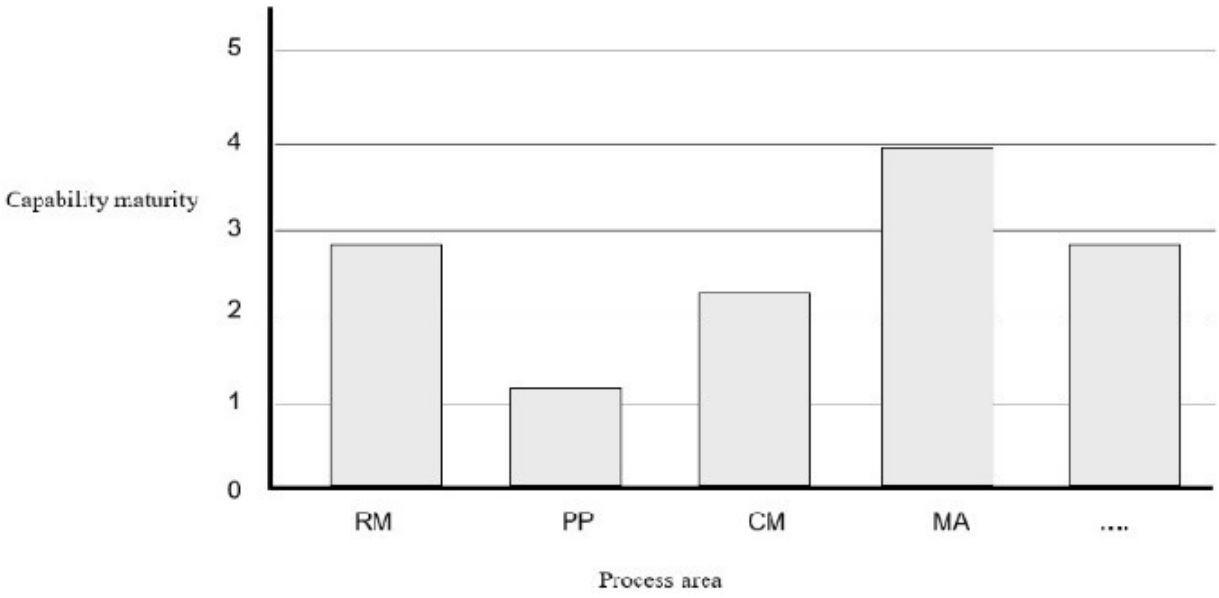
Level 1 :performed
all of the specific goals of the process area have been satisfied. Work task required to produce defined work, product and being conducted.

Level 2: managed
all level one criteria have been satisfied in Edison.
All work associated with the process area confirms to an organisationally defined policy.
All people doing the work have access to adequate resources to get job done.

Level 3: Defined
all level two criteria has been achieved.
All the process is tailored from organisation set of standard process according to organisation guideline.

Level 4 :Quantitatively managed
all level three criteria have been achieved.
Also, the process is control and improve using measurement and quantitative assessment.

Level 5: optimised
all level four criteria has been achieved.
All the process area is adopted all standards to meet the changing customer needs.
The SEI CMM defines each more process area in terms of specific goal and specific practises.



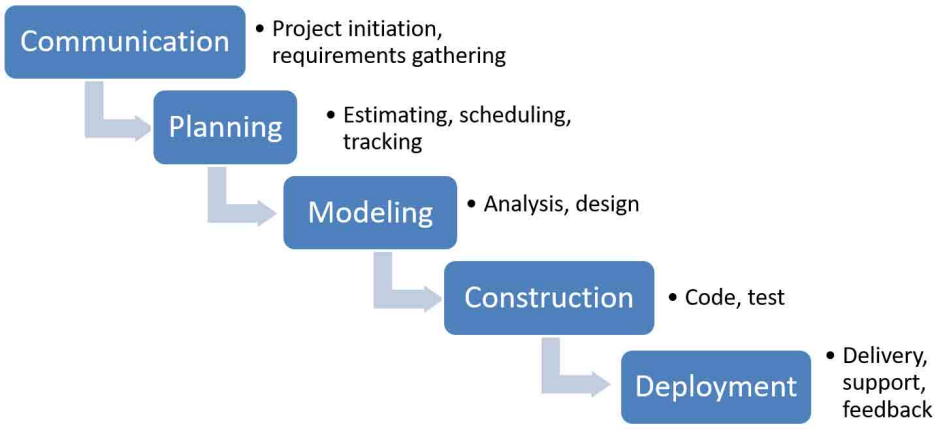
SEI CMM Process area Capability profile

Q.3 EXPLAIN WATERFALL MODEL?

=>
This model is also known as “Linear sequential Model” or “classic life cycle model”.

The **Waterfall Model** is a traditional software development methodology where the process is structured into sequential phases, each of which must be completed before moving on to the next.

It's called "Waterfall" because progress flows in one direction, similar to a waterfall. It’s ideal for projects with clear and fixed requirements.



1. Communication

Definition: This stage focuses on understanding the client’s requirements and defining the project's goals.
Goal: To ensure all stakeholders are aligned on what the software should do.
Activities:

- **Requirement Gathering:** Engaging with stakeholders to gather detailed system requirements.
- **Project Initialization:** Formalizing the project’s objectives and scope, ensuring a clear understanding of the client’s expectations.

2. Planning

Definition: In this stage, the project plan is created, outlining the timeline, resources, and risks involved in the project.
Goal: To ensure the project proceeds in an organized manner with clear expectations.
Activities:

- **Estimation:** Estimating the time, resources, and costs required for the project.
- **Scheduling:** Creating a project timeline with deadlines for each phase.
- **Tracking:** Establishing methods for tracking progress, ensuring that the project stays on course.

3. Modeling

Definition: This phase translates the requirements into a blueprint for the system, ensuring the team understands how to build the software.
Activities:

- **Analysis:** Breaking down the gathered requirements to define how the system should work.
- **Design:** Creating detailed architectural and component designs that serve as a guide for the development phase.

Goal: To create a structured design that guides the development team in coding the system.

4. Construction

Definition: The actual development of the software happens in this phase, followed by rigorous testing to ensure the system works as expected.
Activities:

- **Coding:** Developers write the software based on the design specifications.
- **Testing:** The system is tested for defects and to ensure that it meets the requirements outlined in the initial stages.

Goal: To build a bug-free, functional software product that meets all specified requirements.

5. Deployment

Definition: The software is delivered to the client, and feedback is collected for future improvements.
Activities:

- **Delivery:** The finished product is handed over to the client for use.
- **Support:** Ongoing maintenance and support are provided to address any issues that arise after deployment.
- **Feedback:** Users provide feedback on the software, which is considered for future updates or improvements.

Goal: To ensure the software is successfully delivered and satisfies the client's needs.

Q.4 SHORT NOTE ON AGILE METHODOLOGY.

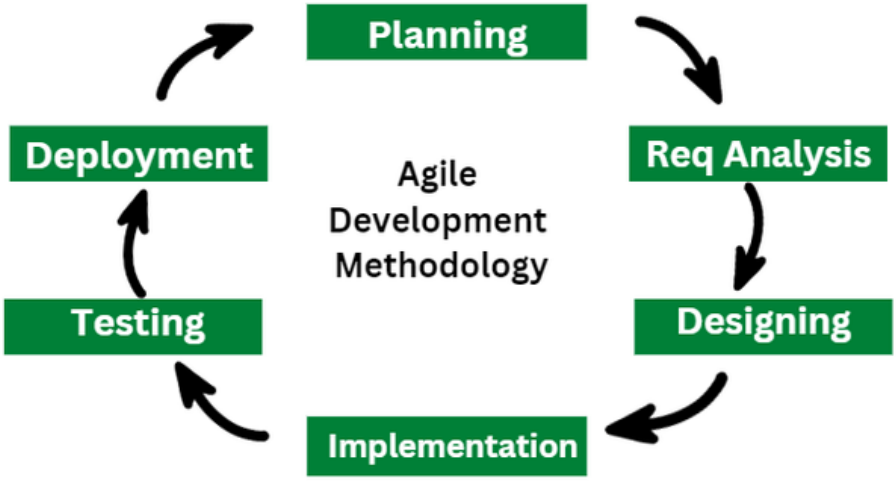
=>

• Agile is a project management and software development approach that aims to be more effective.

1)It focuses on delivering smaller pieces of work regularly instead of one big launch.
2)This allows teams to adapt to changes quickly and provide customer value faster.

• Agile methodologies are iterative and incremental, which means it's known for breaking a project into smaller parts and adjusting to changing requirements.

1)They prioritize flexibility, collaboration, and customer satisfaction.
2)Major companies like Facebook, Google, and Amazon use Agile because of its adaptability and customer-focused approach.



1. Requirement Gathering
In this stage, the project team identifies and documents the needs and expectations of various stakeholders, including clients, users, and subject matter experts.
It involves defining the project's scope, objectives, and requirements.
Establishing a budget and schedule.
Creating a project plan and allocating resources.

2. Design
Developing a high-level system architecture.
Creating detailed specifications, which include data structures, algorithms, and interfaces.
Planning for the software's user interface.

3. Development (Coding)
Writing the actual code for the software. Conducting unit testing to verify the functionality of

individual components.

4. Testing
This phase involves several types of testing:
Integration Testing: Ensuring that different components work together.
System Testing: Testing the entire system as a whole.
User Acceptance Testing: Confirming that the software meets user requirements.
Performance Testing: Assessing the system's speed, scalability, and stability.

5. Deployment
Deploying the software to a production environment.
Put the software into the real world where people can use it.
Make sure it works smoothly in the real world.
Providing training and support for end-users.

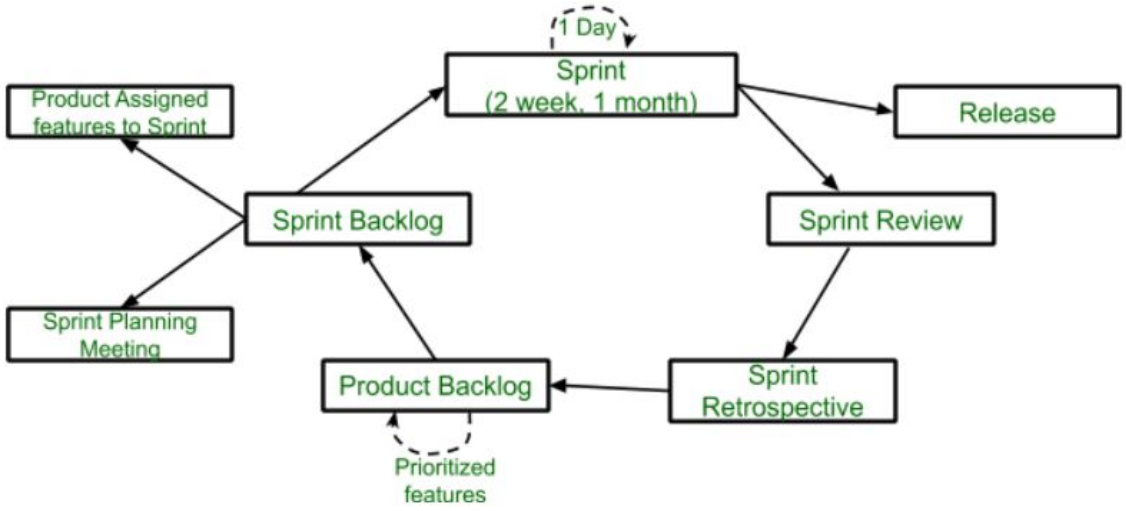
6. Review (Maintenance)
Addressing and resolving any issues that may arise after deployment.
Releasing updates and patches to enhance the software and address problems.

Q.5 SHORT NOTE ON SCRUM METHODOLOGY.

Scrum is a widely used Agile framework for managing complex software development projects.

It focuses on delivering products incrementally through iterative work cycles called sprints, typically lasting 2-4 weeks.

Scrum promotes flexibility, collaboration, and continuous feedback.



Key Elements of Scrum:

- Roles:
  - *Product Owner*: Manages the product backlog and prioritizes tasks based on customer needs.
  - *Scrum Master*: Facilitates the Scrum process, removes impediments, and ensures the team follows Scrum principles.
  - *Development Team*: A self-organizing, cross-functional group responsible for delivering the product increment.

- Artifacts:
  - *Product Backlog*: A prioritized list of features or tasks to be completed.
  - *Sprint Backlog*: A subset of the product backlog to be completed during a sprint.
  - Increment: The working product delivered at the end of each sprint.

- Events:
  - *Sprint Planning*: A meeting to define what will be done in the upcoming sprint.
  - *Daily Scrum*: A brief daily meeting to synchronize the team's progress.
  - *Sprint Review*: A meeting to showcase the work completed during the sprint.
  - *Sprint Retrospective*: A meeting to reflect on the sprint and identify improvements for the next.

Scrum encourages continuous feedback, adaptability to change, and regular delivery of functional software, making it ideal for dynamic and evolving projects.

Q.6 SHORT NOTE ON KANBAN METHODOLOGY.

=>

Kanban is an Agile methodology focused on visualizing and improving workflow efficiency in software development and other processes.

It emphasizes continuous delivery without the need for sprints or time-boxed iterations like Scrum.

Key Principles of Kanban:

1. **Visualizing Workflow:** A Kanban board is used to represent tasks as cards moving through various stages, such as "To Do," "In Progress," and "Done."
   This helps teams track progress and identify bottlenecks.

2. **Limiting Work in Progress (WIP):** Limits are set on the number of tasks in each stage to prevent overloading team members and ensure a steady flow of work.

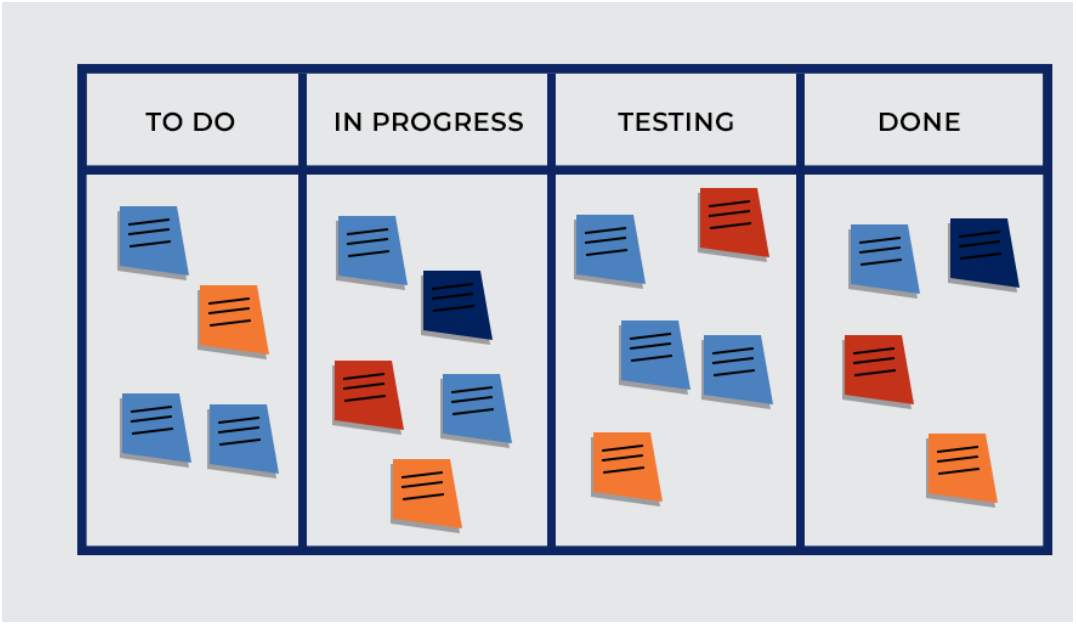
3. **Managing Flow:** Teams focus on optimizing the flow of tasks through the system, ensuring faster and smoother delivery.

4. **Continuous Improvement:** Teams regularly review and refine their processes for more efficiency.

Kanban is flexible, allowing tasks to be completed as they arise, making it ideal for teams that handle varying priorities and need to adapt quickly.

It supports continuous delivery and promotes workflow transparency.

Kanban is an Agile methodology focused on visualizing and improving workflow efficiency in software development and other processes. It emphasizes continuous delivery without the need for sprints or time-boxed iterations like Scrum.



Kanban Boards:

- A **visual tool** used to track work items across different stages, typically organized into columns like "To Do," "In Progress," and "Done."
- Helps teams monitor workflow, identify bottlenecks, and improve task management.

Kanban Cards:

- **Task representations** on the Kanban board, containing details like the task description, assignee, and deadline.
- Cards move through columns as the task progresses from one stage to another.

Benefits of Kanban:

1. **Improved Workflow Visibility:** Teams can easily track progress and identify delays or issues in real-time.
2. **Increased Efficiency:** Limiting Work in Progress (WIP) prevents task overload and ensures steady task completion.
3. **Flexibility:** Kanban allows for changes in priorities without needing to adjust iterations or sprints, making it highly adaptable.

Q.7 SHORT NOTE ON EXTREME PROGRAMMING(XP).

=>

The extreme programming is one of the most commonly used a child process models.

All the process models obey the principle of agility and manifesto of software development.

The XP uses the concept of object oriented programming. This approach is preferred development paradigm.

As in conventional approach, a developer focuses on the framework activities like planning, design, coding, and testing, the XP also has set of roots and practises.

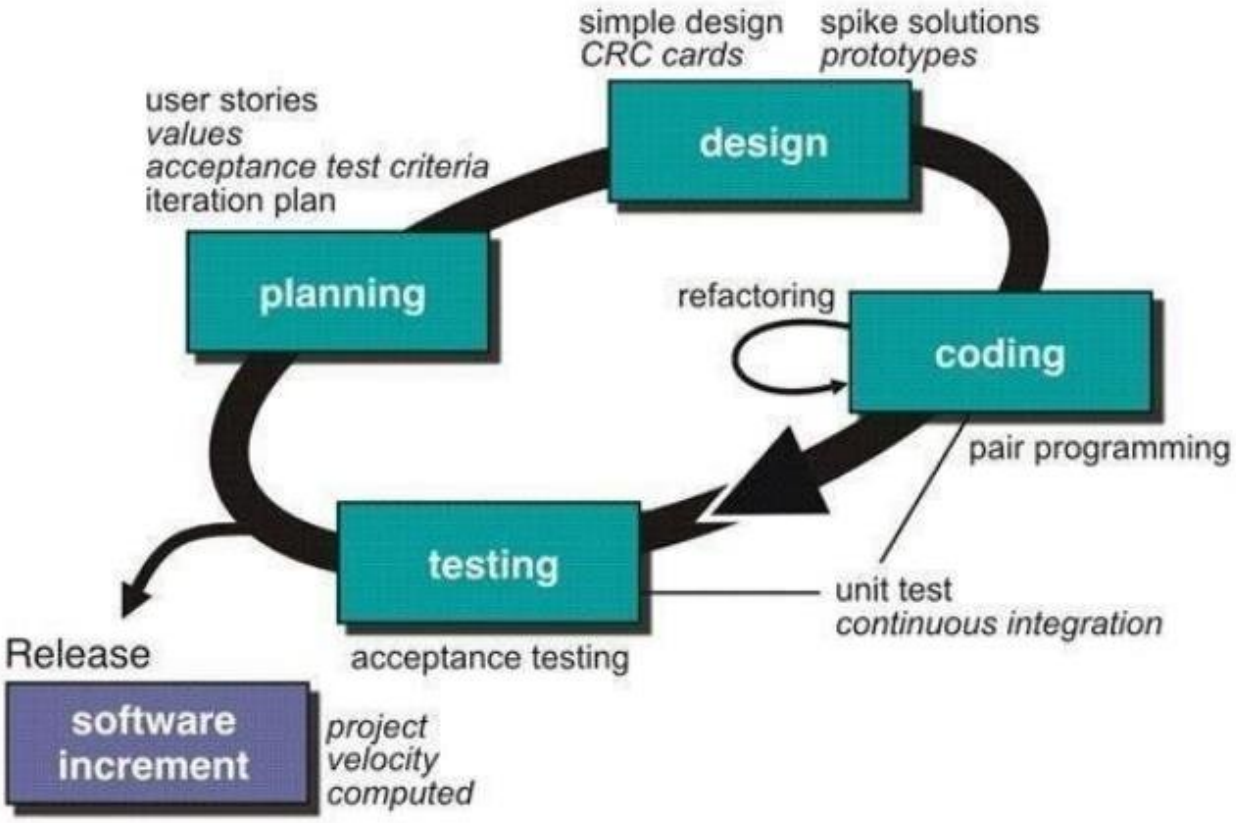


FIG7.1 EXTREME PROGRAMMING (XP) PROCESS

XP VALUES:

Following are set of five values at establish a foundation for all work performed in context with XP

1. Communication
2. Simplicity
3. Feedback
4. Courage
5. Respect

1) For any development process, there must be regular meeting of developer and the customer. There should be a proper communication for requirement, gathering, and discussion of the concepts.

2) The simple design can always be easily implemented in code.
3) The feedback is an important activity which leads to the discipline in the development process.

4) In every development project, there is always a pressure situation. The courage or the discipline will definitely make the task easy.

5) In addition to all the XP values, the agile should calculate respect among all the team members, between other stakeholders and with customer.

Q.8) SHORT NOTE ON SPIRAL MODEL.

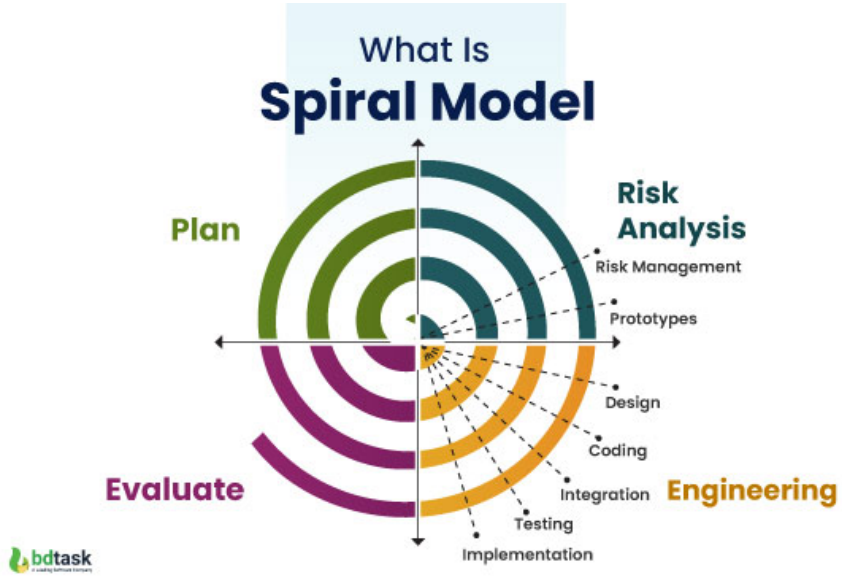
=>

The spiral model is combination of well-known waterfall model and iterative prototyping.
It yields rapid development of more complete version of software.

Using spiral model software is developed by series of evolutionary releases during the initial release.

It may be just paper work for Toto type, but during later release the version got more completed stages.

The spiral model is divided into a set of framework activities defined by software engineering team. Each framework activity represent one segment of spiral as shown in figure



1. COMMUNICATION

A software development process starts with communication between customer and developer

2. PLANNING

It includes complete estimation, example (Cost estimation of project) and scheduling (complete timeline chart for project development) and risk analysis

3. MODELLING

- It includes detail requirement, analysis and project design(algo, flowcharts etc).
- flowchart shows complete flow of program whereas algorithm is step-by-step solution of problem

4. CONSTRUCTION

It includes coding and testing steps:
1)Coding: design details are implemented using appropriate programming language
2)Testing: testing is carried out

5. DEPLOYMENT

It includes software delivery support and feedback from customer. If customer suggest some corrections or demand additional capabilities, then changes are required for such correction or enhancement.

Merits of Spiral Model:

1. **Risk Management:** Allows early identification and mitigation of risks in each iteration.
2. **Flexibility:** Adaptable to changes in requirements, providing room for project refinements.
3. **Customer Feedback:** Ensures continuous customer involvement and feedback, improving product quality.

Demerits of Spiral Model:

1. **Complexity:** Managing multiple iterations and risk assessments can be complex and resource-intensive.
2. **Costly:** The iterative nature and risk analysis make it expensive, especially for small projects.
3. **Requires Expertise:** It requires a highly skilled team to perform risk assessments and manage the process effectively.