

Q1] Requirements engineering is the process of identifying, eliciting, analyzing, specifying, validating and managing the needs and expectations of stakeholders for a software system.

It is an iterative process that involves several steps.

(1) Requirements Elicitation: This step involves interviews, focus groups and other techniques to gather information from stakeholders about the needs and expectations.

(2) Requirement Analysis: This step involves analyzing the gathered information to identify the high-level goals and objectives, any constraints or limitations of the software system.

(3) Requirement Specification: This step involves documenting the requirements in a clear, consistent and unambiguous manner. It involves prioritizing and grouping the requirements into manageable chunks.

(4) Requirement validation: This step involves checking that the requirements are complete, consistent and accurate. It also involves checking that the requirements are testable and that they meet the needs, and expectations of stakeholders.

(5) Requirement Management: This step involves managing the requirements throughout the software development life cycle, including tracking and controlling changes and ensuring that the requirements are still valid and relevant.

Recognizing software requirements in software engineering provides a solid foundation for the development process, which helps to reduce the risk of failure and ensures that the system is delivered on time, within budget and required quality standards.

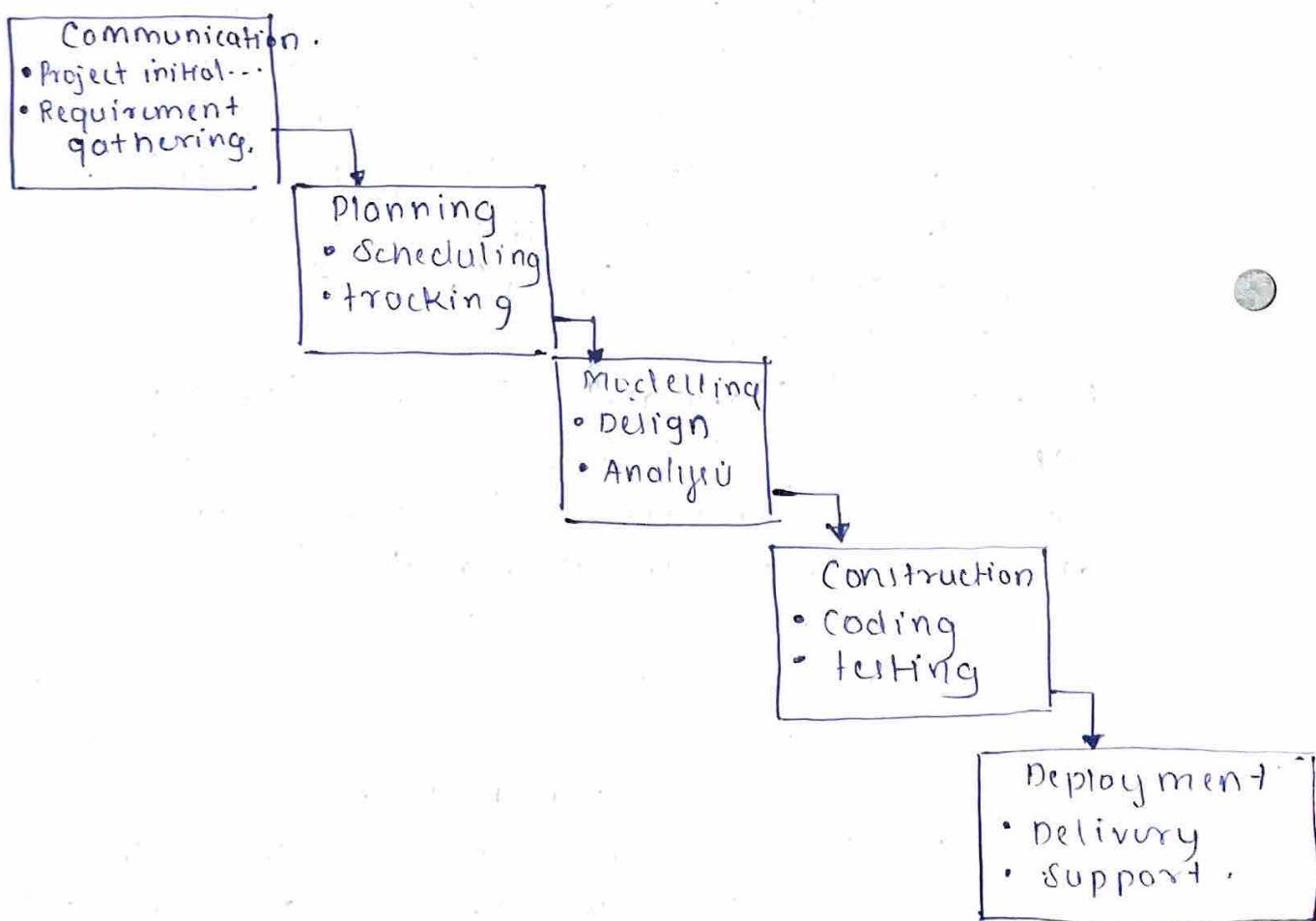
Q2] Software process models is an abstraction of the software development process. It is a representation of the order of activities of the process and the sequence in which they are performed.

The 5 generic process frame activities:-

- 1.) Communication.
- 2.) Planning
- 3.) modelling
- 4.) Construction
- 5.) Deployment.

a) Waterfall model:- It is sequential, plan driven project where you must plan and schedule all your activities before starting the project. It does support iteration so, change can cause confusion.

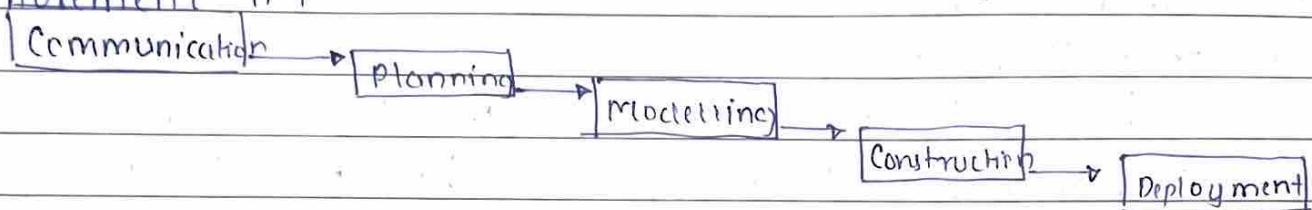
Waterfall model with feedback:



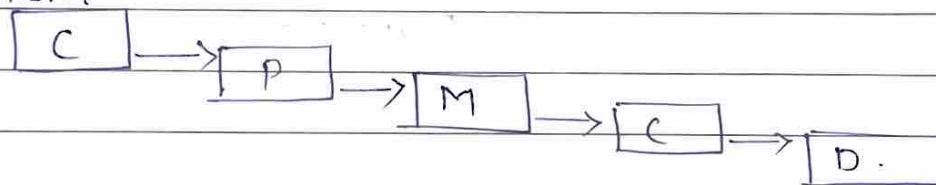
b.

Incremental model: Similar to iterative models, but the software is built in increments, each delivering specific functionality. It is efficient as the developers only focus on what is important and bugs are fixed as they arise, but you need clear and complete definition of the whole sys before you start.

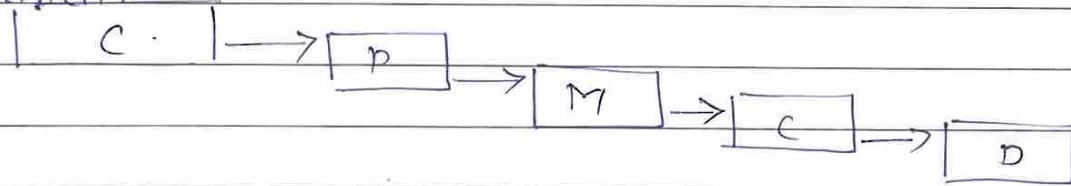
Increment #1



Increment #2.



Increment #3.

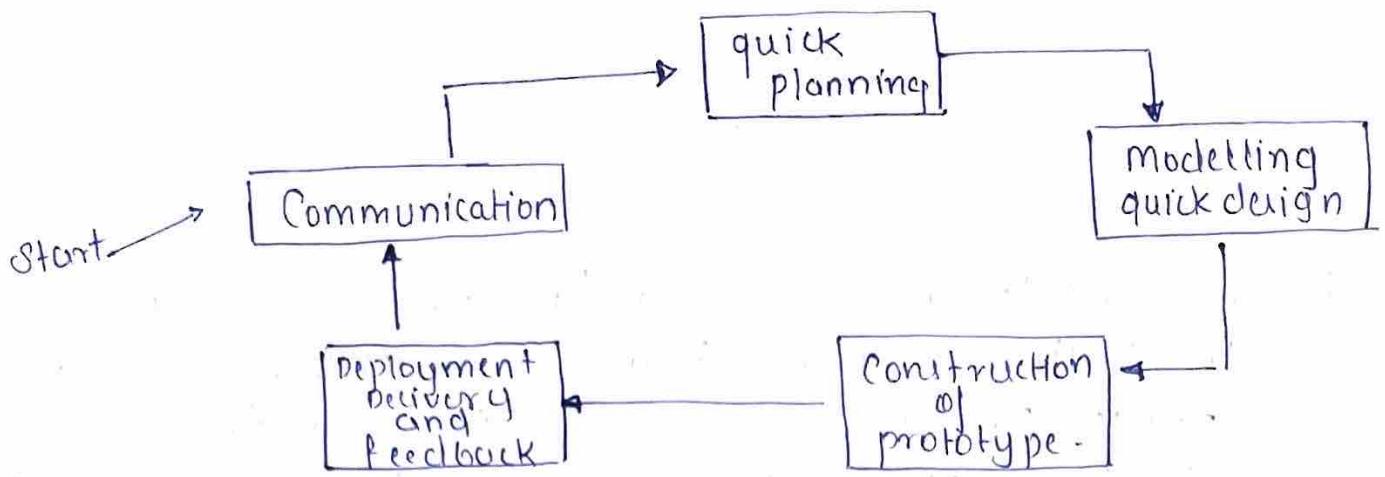


c.

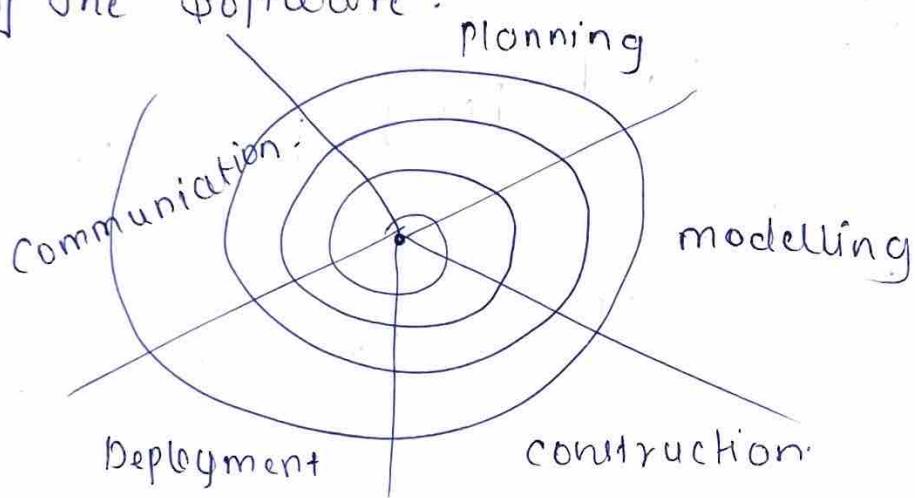
Prototyping model:

follows an evolutionary and iterative approach. Used when requirements are not well understood. focuses on those aspects of software that are visible to the customer / user.

feedback is used to refine the prototype.

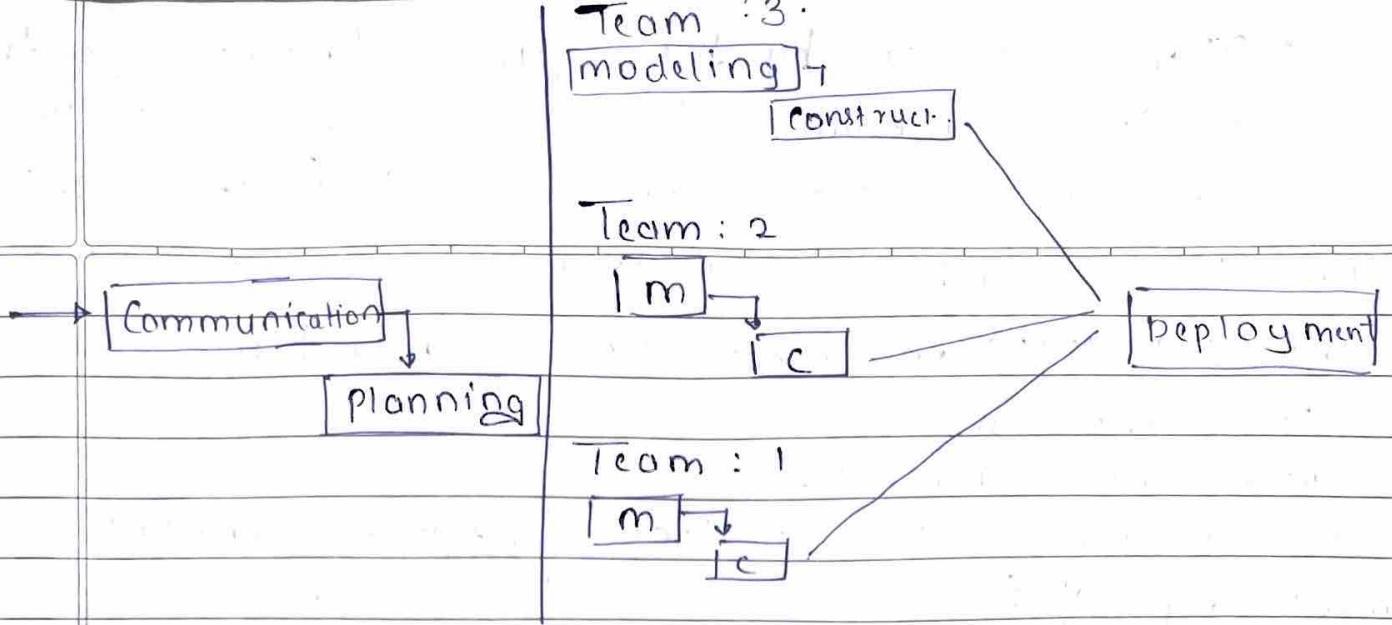


(d) Spiral model :- Combination of waterfall and iterative prototyping follows evolutionary approach.
 Inner spiral focus on identifying software requirements and projects risks.
 Outer spiral take on classical waterfall approach after requirements have been defined, but permit iterative growth of the software.

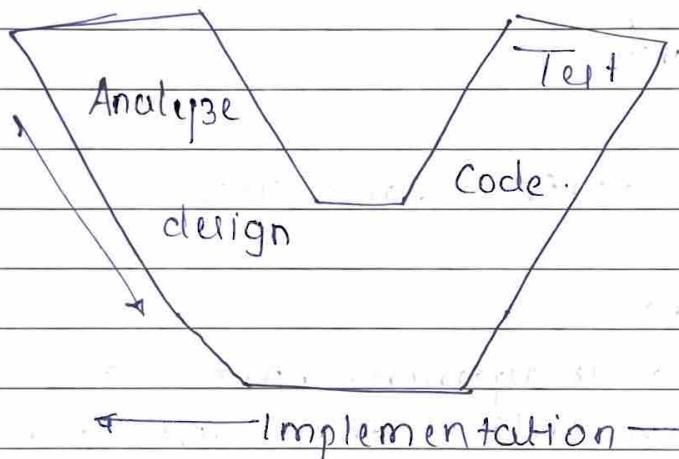


(e) The RAD model (Rapid Action Development)

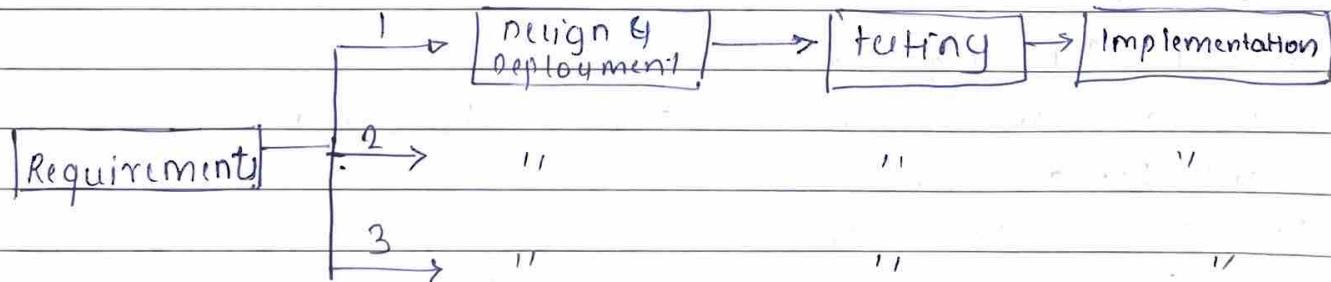
Using RAD model, software product can be developed within a very short period of time. Requirements are divided into diff groups. When all groups are ready with their final product the product of each team is integrated to form a whole product.



⑧ V-modal : It represents a development process that can be considered as an extension of the waterfall mode. Instead of moving down in linear way, the process steps are best upwards after the coding phase.



⑨ Iterative model : Each iteration may include a subset of the software functionality.



(Q3.) CMMI was developed by the Software Engineering Institute (SEI) at Carnegie Mellon University in 1987. It defines the process characteristics that should exist if an organization wants to establish a software process that is complete.

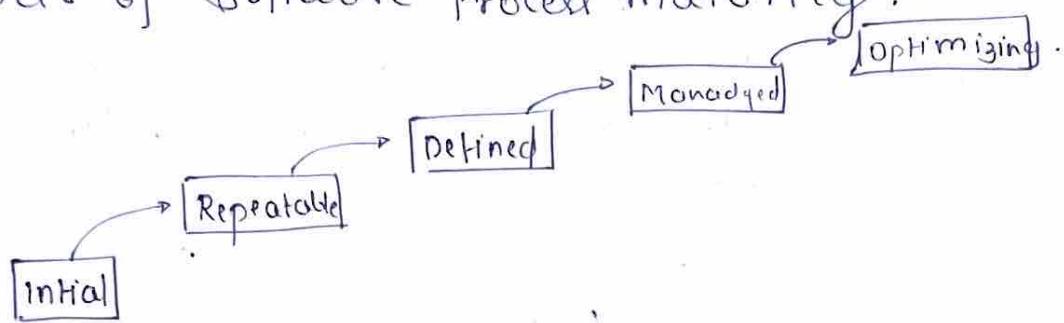
The SEI CMMI presents 2 types of meta models:

As continuous model

As a staged model.

It also provides guidelines to further enhance the maturing of the process used to develop these software products.

Five levels of Software Process maturity:



Level 1: Initial

Few processes are defined, and success depends on individual effort.

Level 2: Repeatable.

Basic project management processes are established to track cost, schedule, and functionality.

Level 3: Defined

All projects use an approved, tailored version of the organization's standard software process for developing and maintaining software.

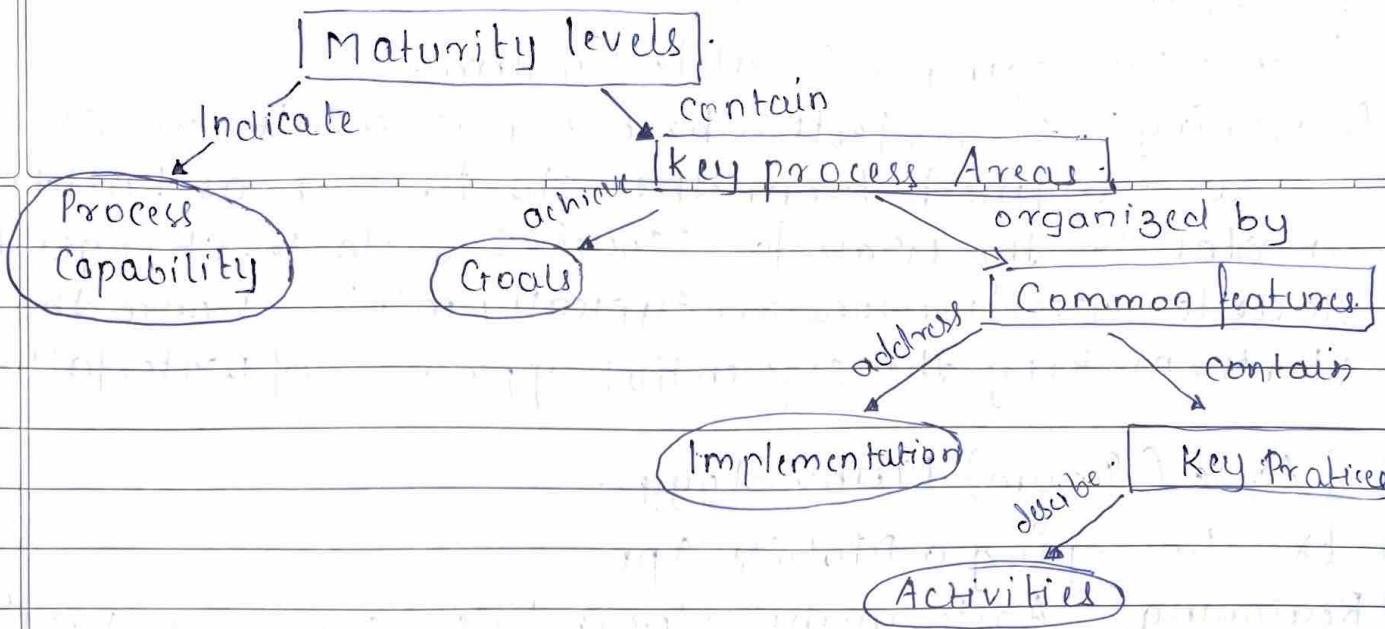
Level 4: Managed.

Detailed measures of the software process and product quality are collected.

Level 5: Optimizing

Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.

Implementation of CMM



q4.7 Prescriptive models

- ① Developed to bring order and structure to the software development process.
- ② Evolutionary software process do not establish the main speed of the evolution Due to this development process becomes slow.
- ③ Defines a distinct set of activities, actions, tasks that lack flexibility, extensibility are required to engineer and high quality high quality software.
- ④ more popular less popular.
 - ⑤ provides complete and full developed systems Time does not full and complete system to be developed.
 - ⑥ Ex: waterfall model, Incremental model Ex: Prototyping, spiral and concurrent models.
 - ⑦ It can accommodate changing requirement. Improvement is required in the product.

Q5] 4] Waterfall Model :-

Ex : Developing a Satellite System.

Reasoning :- for projects where requirements are stable and well defined from the start, such as building a satellite, the Waterfall Model is suitable. Changes to the satellite specifications are typically minimal once the project starts, making the sequential approach of waterfall advantageous.

2] Agile (Scrum) Methodology :-

Ex : Developing a Mobile App.

Reasoning :- Agile methodologies like Scrum are ideal for projects that require frequent updates, iterations and continuous user feedback. This is needed for mobile App. Scrum's iterative approach allows the developers to adapt to user needs and technological shifts during short sprints.

3] Agile (Kanban) Methodology :-

Ex :- Customer Support Team :-

Reasoning :- Kanban, with its focus on continuous flow and visualizing work, is well suited for managing tasks with varying sizes. A customer support team handling a constant stream of incoming issues and requests could benefit from Kanban.

4] Spiral Model :-

Ex :- Medical Device Development.

Reasoning :- Projects involving critical safety concerns, like medical device development benefit from the risk-driven approach of the Spiral model. The iterative and risk focused nature of the spiral model aligns well with the need for careful analysis and testing in such projects.

Q6] The Waterfall and Agile models differ significantly in terms of project planning and process tracking

1) Waterfall model :-

- (a) Project Planning :- It follows a sequential approach. Planning is done extensively at the beginning of the project, where all requirements are gathered and documented before development begins.
- (b) Project tracking :- Progress is tracked through predefined milestones. Each phase must be completed before moving to the next, making it challenging to accommodate changes later in the project.

2) Agile Methodologies :-

- (a) Project Planning : Agile approaches (like Scrum or Kanban) emphasize flexibility and adaptability. Planning is done in iterations, with focus on delivery smaller, functional increments of the project.
 - (b) Progress tracking : Agile relies on frequent iterations and feedback loops. Teams track progress using tools like burndown charts, velocity measurement and daily stand-up meetings. This allows for course correction and adjustment as needed.
- Waterfall model follows a rigid, linear planning process while Agile methodologies promote iterative planning.

Q7.] Project metrics refers to quantitative measures used to assess various aspects of a software development project's progress, quality, efficiency and performance. It provides valuable insights into the project's health.

① Waterfall :

- (a) Development speed : longer development cycles might lead to slower speed compared to Agile methodologies.
- (b) Adaptability to change : It is less adaptable to

- change due to its sequential nature, which can lead to challenges when change is required.
- (c) Customer satisfaction: Since changes can't be easily incorporated mid-process, customer satisfaction might be impacted if initial requirements do not meet their evolving needs.
- (d) Agile (Scrum):
- (a) Development Speed: Short iterations (sprints) can result in faster development speed and more frequent releases.
 - (b) Adaptability to change: Agile methodologies like scrum are designed to embrace change, allowing teams to adjust requirements between sprints.
 - (c) Customer Satisfaction: frequent feedback loops and incremental improvement often lead to higher customer satisfaction.
- (e) Agile (Kanban):
- (a) Development Speed: It is focused on continuous delivery which results in steady development speed as work items are pulled through the process.
 - (b) Adaptability to change: It is flexible and allows for changes to be introduced at any point, making it adaptable to shifting requirements.
 - (c) Customer Satisfaction: Continuous delivery and focus on flow can lead to better customer satisfaction by quickly addressing their needs.

Q8. [] Here factors help in evaluating and selecting a suitable model for a given project, considering its unique characteristics and goals: Each model has its strengths and weaknesses making the comparison relevant for making informed decisions in software development.