FR. CONCEICAO RODRIGUES COLLEGE OF ENGG.

Fr. Agnel Ashram, Bandstand, Bandra (W) Mumbai 400 050.

SEMESTER / BRANCH: V/COMPUTER Engineering

SUBJECT: Software Engineering **(CSC502)/ First Assignment**

Date: 19-08-23 Due Date : 25-08-23

CSC502.1: Recognize software requirements and various process models. (Understanding)
CSC502.2: Develop project Plan, schedule and track the progress of the given project (Applying)

Questions :

1. What is the significance of recognizing software requirements in the software engineering process?
2. Describe the main characteristics of different process models used in software development.
3. How does the Capability Maturity Model (CMM) contribute to improving software development processes?
4. Explain the differences between prescriptive process models and evolutionary process models.
5. Provide examples of situations where using a specific process model would be more suitable.
6. Compare and contrast the Waterfall model and Agile methodologies in terms of project planning and progress tracking.
7. Apply process metrics to evaluate the efficiency and effectiveness of Waterfall , Agile (both Scrum & Kanban) methodologies, considering factors such as development speed, adaptability to change and customer satisfaction.
8. Justify the relevancy of the fallowing comparison for software development models.

Features	Water fall Model	Incremental Model	Prototyping Model	Spiral Model
Requirement Specification	Beginning	Beginning	Frequently Changed	Beginning
Understanding Requirements	Well Understood	Not Well Understood	Not Well Understood	Well Understood
Cost	Low	Low	High	Expensive
Availability of reusable component	No	Yes	Yes	Yes
Complexity of System	Simple	Simple	Complex	Complex
Risk Analysis	Only at beginning	No risk analysis	No risk analysis	Yes
User involvement in all phases of SDLC	Only at beginning	Intermediate	High	High

Guarantee of Success	Less	High	Good	High
Overlapping Phases	Absent	Absent	Present	Present
Implementation Time	Long	Less	Less	Depends on Project
Flexibility	Rigid	Less flexible	Highly flexible	Flexible
Changes Incorporated	Difficult	Easy	Easy	Easy
Expertise Required	High	High	Medium	High
Cost Control	Yes	No	No	Yes
Resource Control	Yes	Yes	No	Yes

Rubrics :

Indicator	**Average**	**Good**	**Excellent**	**Marks**
Organization (2)	Readable with some mistakes and structured (1)	Readable with some mistakes and structured (1)	Very well written and structured (2)	
Level of content(4)	Minimal topics are covered with limited information (2)	Limited major topics with minor details are presented(3)	All major topics with minor details are covered (4)	
Depth and breadth of discussion(4)	Minimal points with missing information (1)	Relatively more points with information (2)	All points with in depth information(4)	
Total Marks(10)				

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

(5) Requirement Management: This step involves managing the requirements throughout the software development life cycle, including tracking and controlying 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.

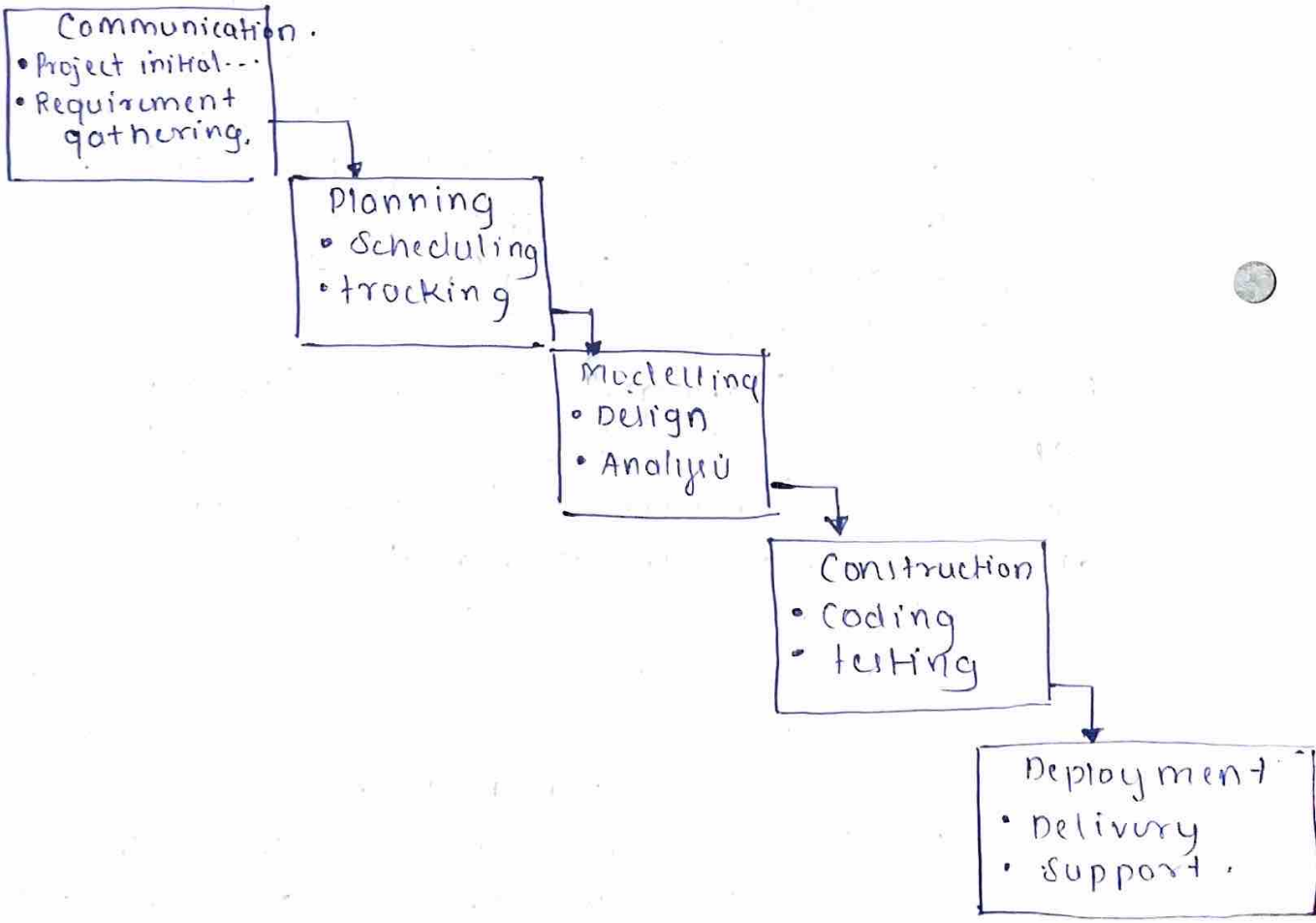
Q2] Software process models is an abstraction of the software development process. It is a representation of the order of activities of the process and the sequence in which they are performed.

The 5 generic process frame activities:-

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

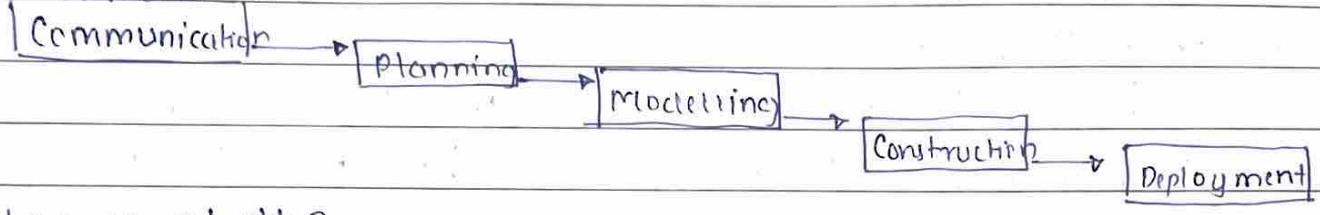
(a) Waterfall model:- It is sequential, plan driven-process where you must plan and schedule all your activities before starting the project. It does support iteration so, changes can cause confusion.

Waterfall model with feedback:

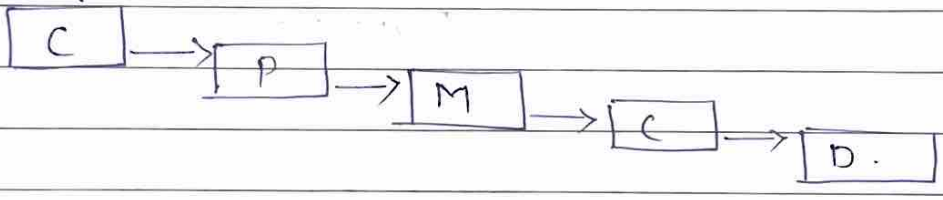


(b) Incremental model :- Similar to iterative models, but the software is built in increments, each delivering specific functionality. It is efficient as the developers only focus on what is imp and bugs are fixed as they arise, but gov need clear and complete definition of the whole sys before you start.

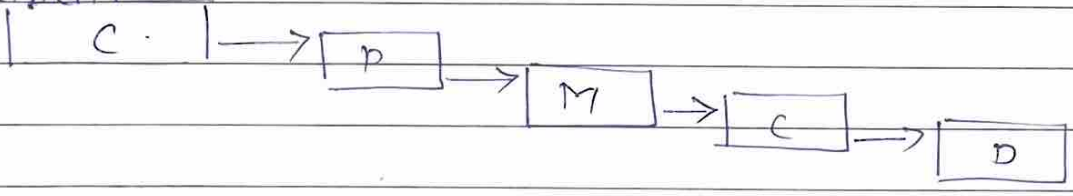
Increment #1



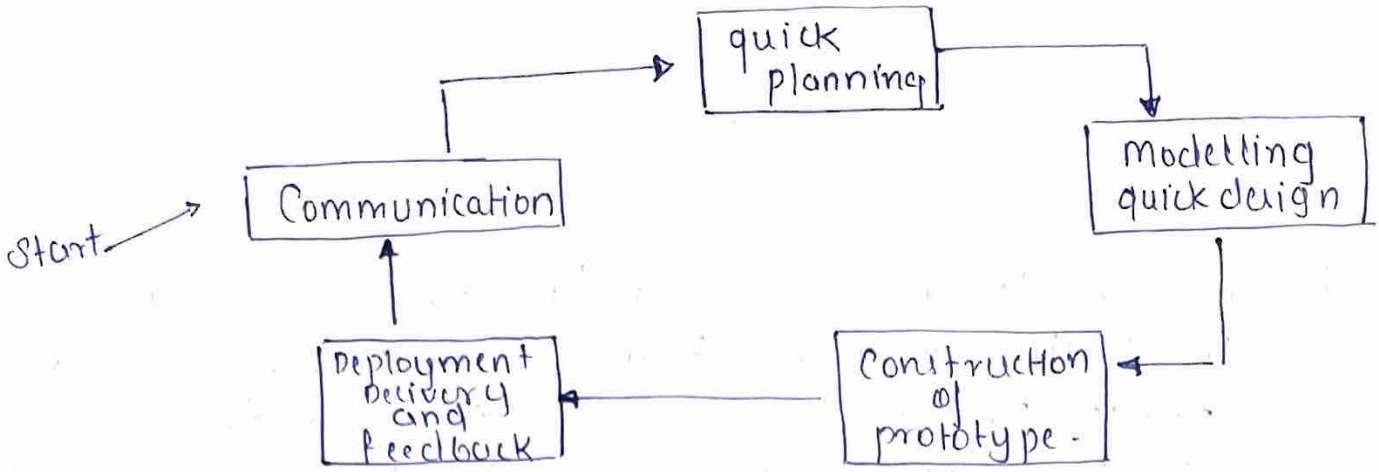
Increment # 2.



Increment # 3.



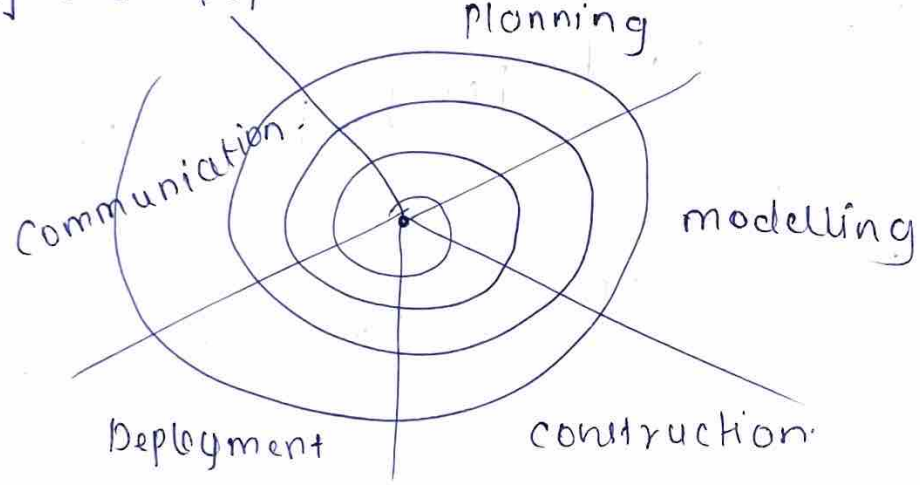
(c) Prototyping model :-
focuses on evolutionary and iterative approach.
Used when requirements are not well understood
focuses on those aspects of software that are visible to the customer / user.
feedback is used to refine the prototype.



(d) Spiral model :- Combination of waterfall and iterative prototypi follows evolunoary approach.

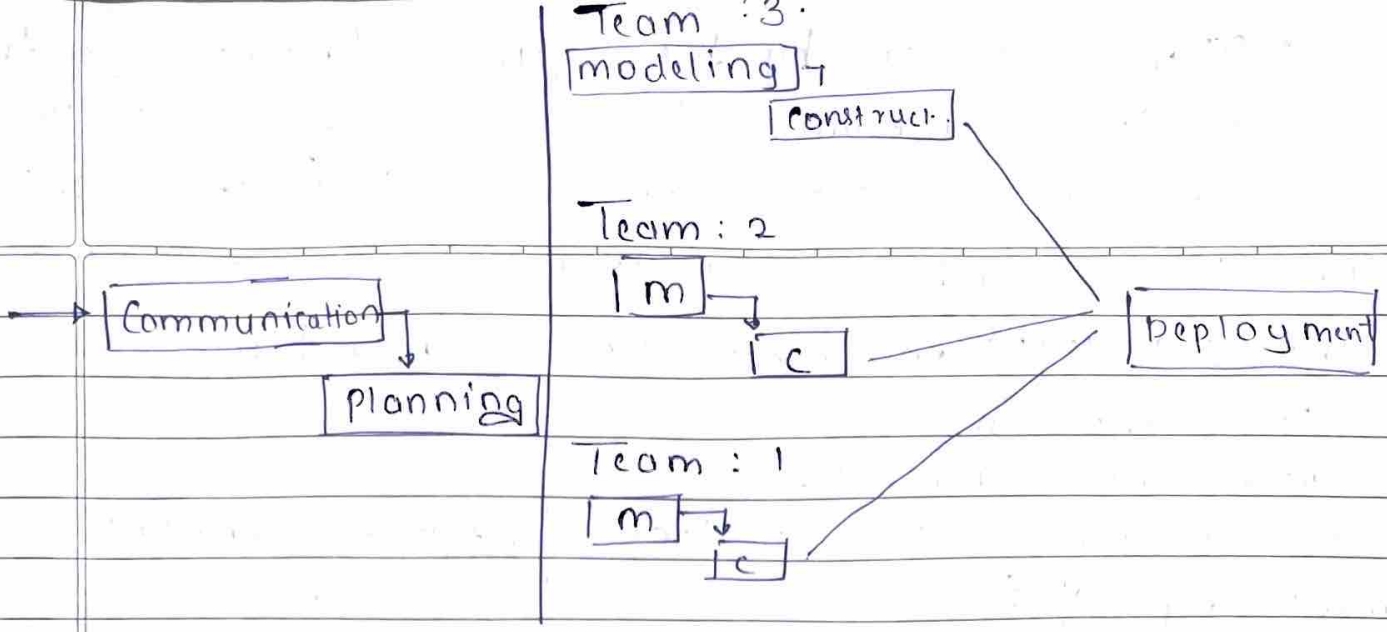
Inner spiral focuses on identifying software requirements and projects risks.

Outer spiral take on classical waterfall approach after requirements have been defined, but permit iterative growth of the software.

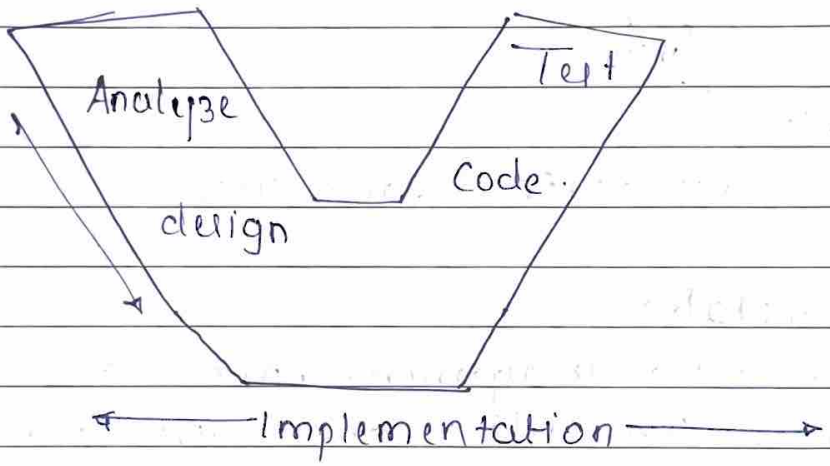


(e) The RAD model (Rapid Action Development)

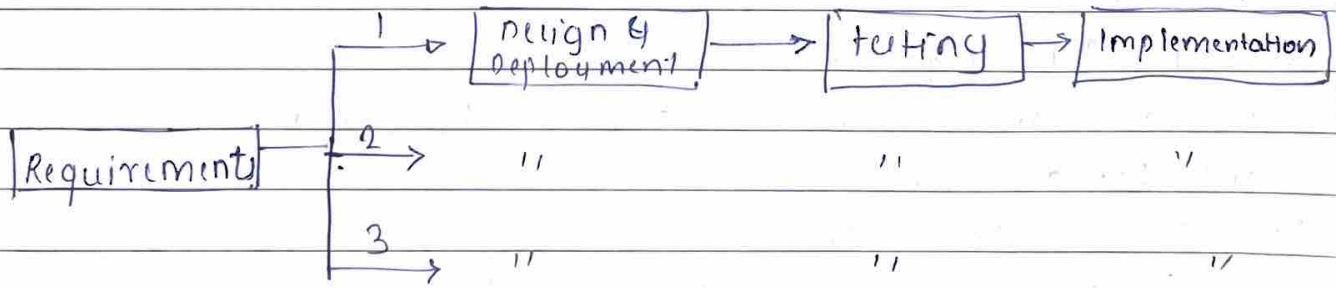
Using RAD model, software product can be developed withing a very short period of time, Requirements are divided into diff groups. When all groups are ready with their final product the product of each team is integrated to form a whole product.



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



(g) Iterative model : Each iteration may include a subset of the software functionality.



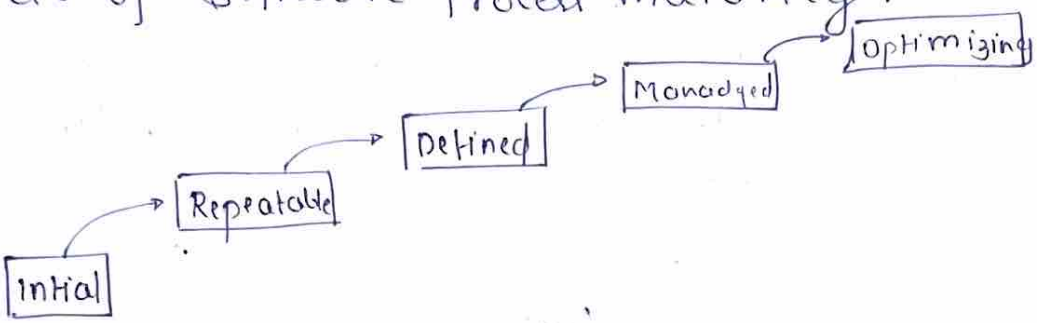
Q3.] CMM was developed by the Software Engineering Institute (SEI) at Carnegie mellon University in 1987. It defines the process characteristics that should exist if an organization wants to establish a software process that is complete.

The SEI CMM presents 2 types of metal models:

As continous 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 process are defined, and sucess depends on individual effect.

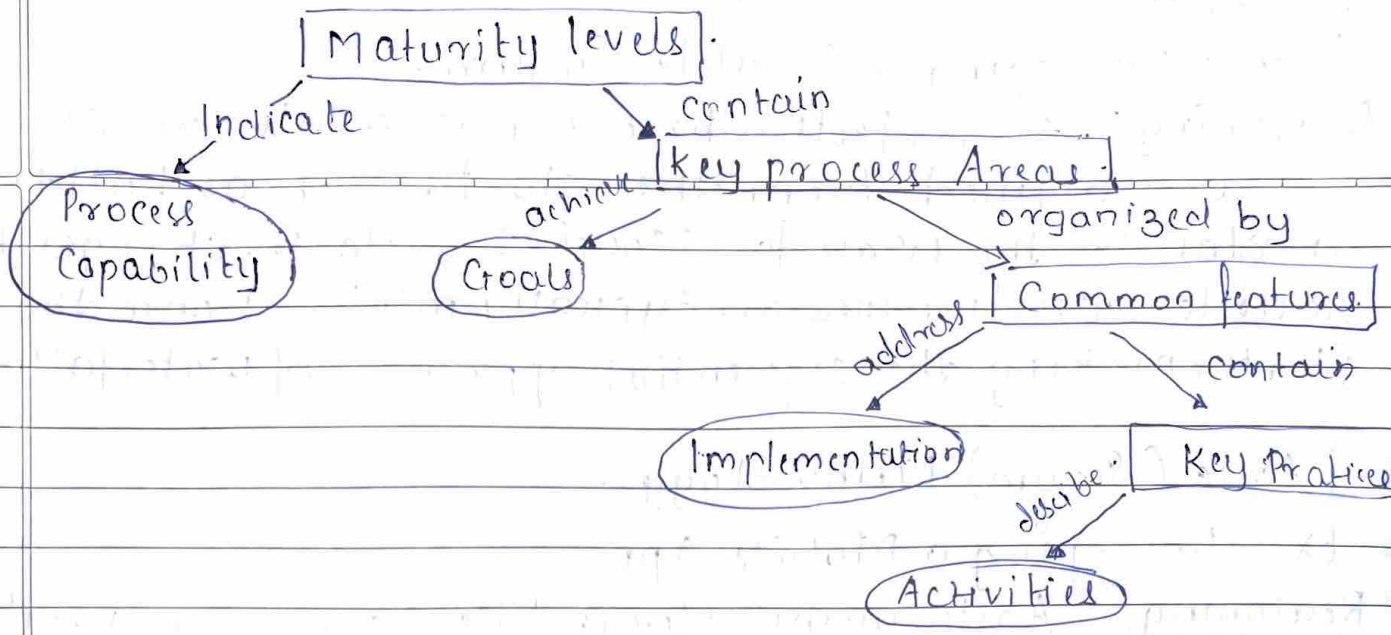
Level 2: Repetable.
Basic project management process are established to track, cost, schedule, and functionality.

Level 3: Defined
All projects used an approved, tailored version of the organizations 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] Prescriptive models	Evolutionary Models
① Developed to bring order and structure to the software development process	Evolutionary software process do not establish the max speed of the evolution Due to this development process becomes slow.
② Defines a distinct set of activities, actions, tasks that are required to engineer high quality software.	Evolutionary process models lack flexibility, extensibility and high quality.
③ more popular	less popular.
④ Provides complete and full developed systems	Time does not allow full and complete system to be developed.
⑤ Ex. Waterfall modal, Incremental model	Ex: Prototyping, spiral and concurrent models.
⑥ It can accomodate changing requirement.	Improvement is required in the product.

Q5] 1] 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 satelleite specifications are typically minimal once the projects 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 continous user feedback. This are need 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 task with varying and sizes. A customer support team handling a constant stream of incoming issues and request could benfit 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 coretul 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) Projects tracking :- Progress is tracked through predefined milestones. Each phase must be completed before moving to the next, making it challenging to accomodate 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 metrices refers to quantitative measures used to assess various aspects of a software development projects 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 meth Agile methodologies

(b) Adaptibility to change :- It is less adaptable to

change due to its sequential nature, which can lead to challenge 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

(2) Agile (scrum):

(a) Development Speed: Short iterations (sprints) can result in faster development speed and more frequent releases.

(b) Adaptibility to change: Agile methodologies like scrum are designed to embrace change, allowing teams to adjust requirements between sprints.

(c) Customer Satisfication: frequent feedback loops and incremental improvement often lead to higher customer satisfaction.

(3) 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: Continous 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 this comparison relevant for making informed decisions in software development.