denny lopes - 9618
FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

1812

SE Assignment 1

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

It is an iterative process that involves several steps:

(1) Requirements Elicitation: This step involves interviews, surveys, 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.

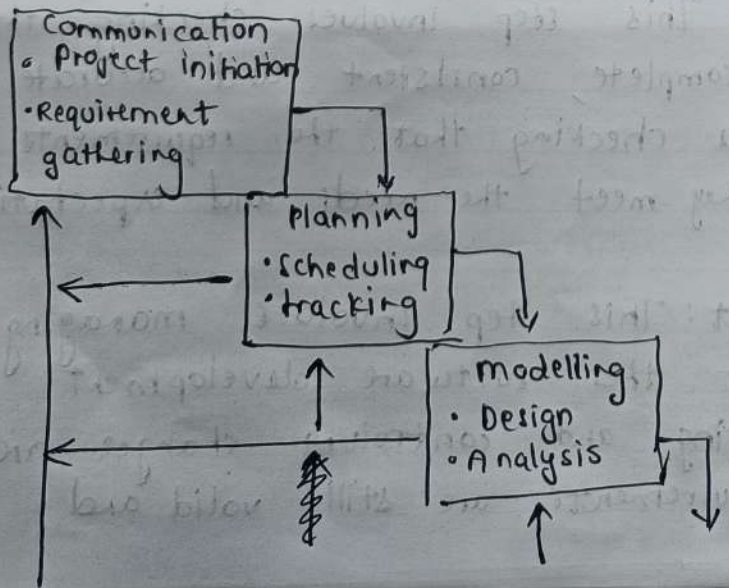
Q2) Software process models is an abstraction of the software development process. It is a representation of the order of activities of the process and the sequence in which they are performed.

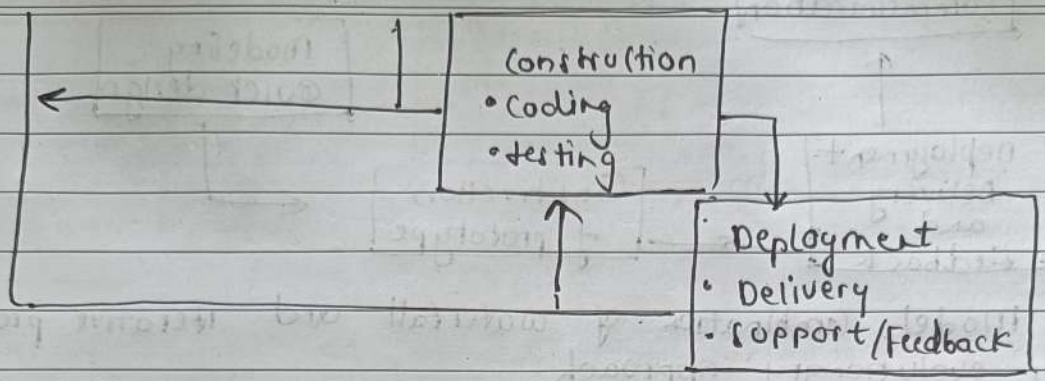
The 5 generic process framework activities:

① Communication
② Planning
③ Modelling
④ Construction
⑤ Deployment

ⓐ Waterfall model: It is sequencial 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.

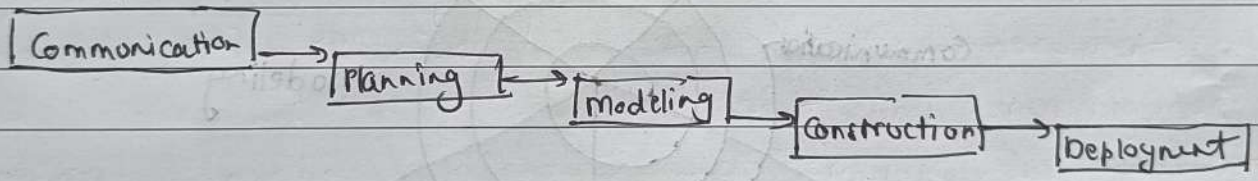
Waterfall model with feedback:



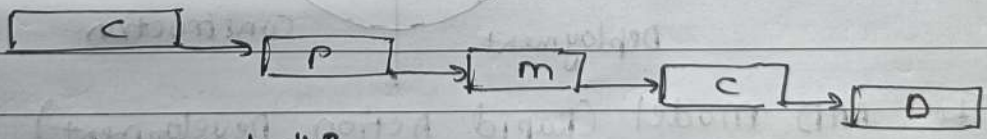


(b) Incremental model : similar to iterative models, but the software is built in increments, each delivering specific functionality. It is efficient as the developers only focus on what is imp and bugs are fixed as they arise, but you need a clear and complete definition of the whole sys before you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

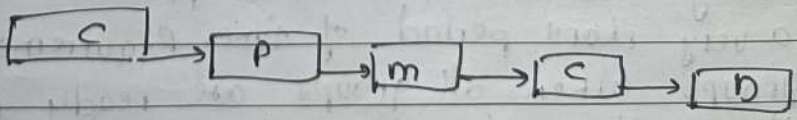
Increment # 1



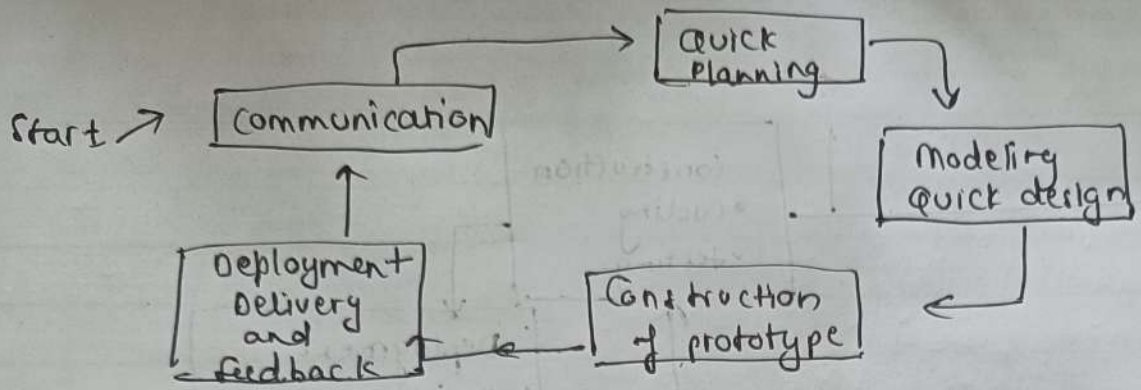
Increment #2



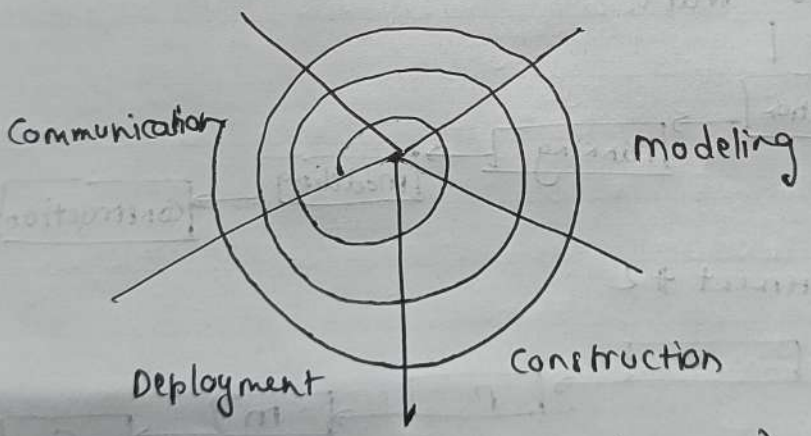
Increment #3



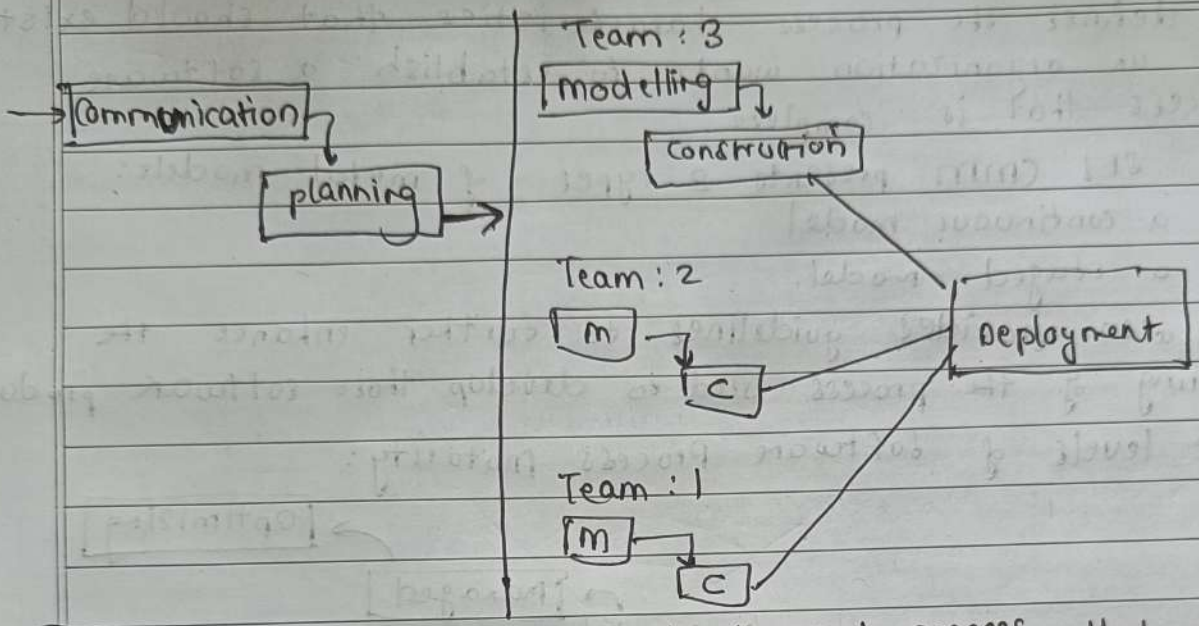
(c) Prototyping model:
follows an evolutionary and iterative approach.
Used when requirements are not well understood.
Focuses on those aspects of software that are visible to the customer/user.
Feedback is used to refine the prototype.



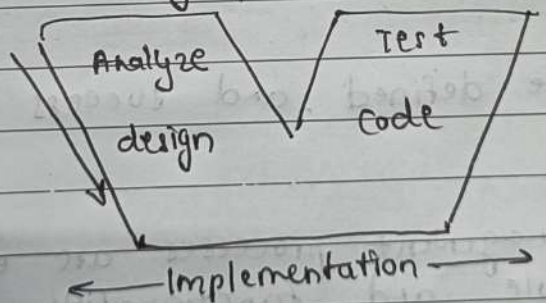
(d) Spiral Model: Combination of waterfall and iterative prototyping follows evolutionary approach.
Inner spiral focuses on identifying software requirements and project risks.
Outer spiral take on a classical waterfall approach after requirements have been defined, but permit iterative growth of the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



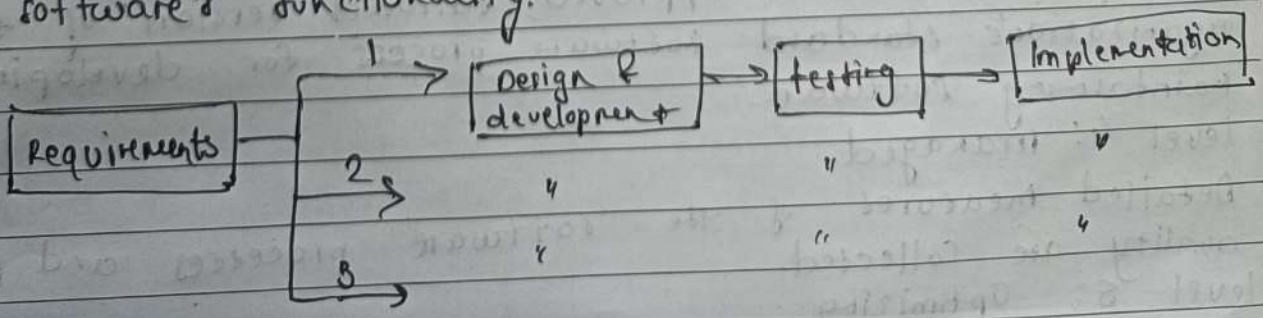
(e) The RAD Model (Rapid Action Development)
Using RAD model, software product can be developed within a very short period of time. Requirements are divided into diff groups. When all groups are ready with their final product the product of each team is integrated to form a whole product.



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



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

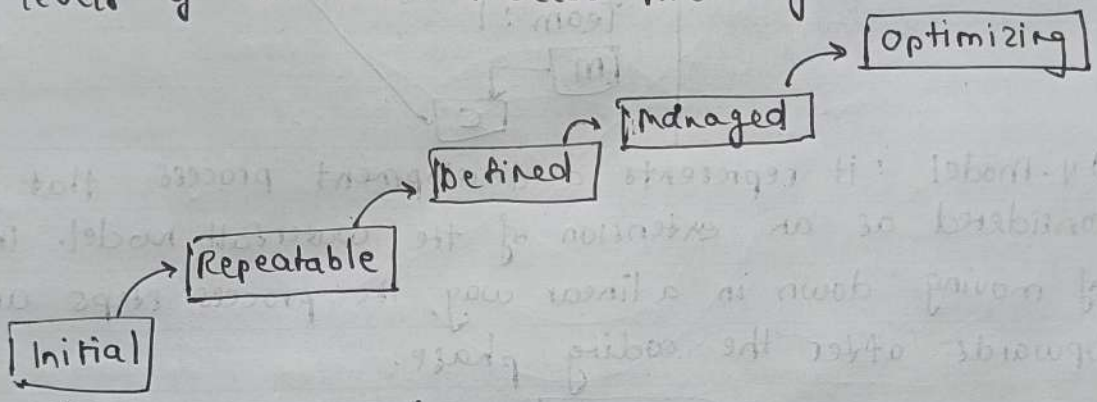


Q3) CMM was developed by the Software Engineering Institute (SEI) at Carnegie Mellon University in 1987. It defines the process characteristics that should exist if an organization wants to establish a software process that is complete.

The SEI CMM presents 2 types of metal models:
As a continuous model
As a staged model.

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

Five levels of software process maturity:



Level I: 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.

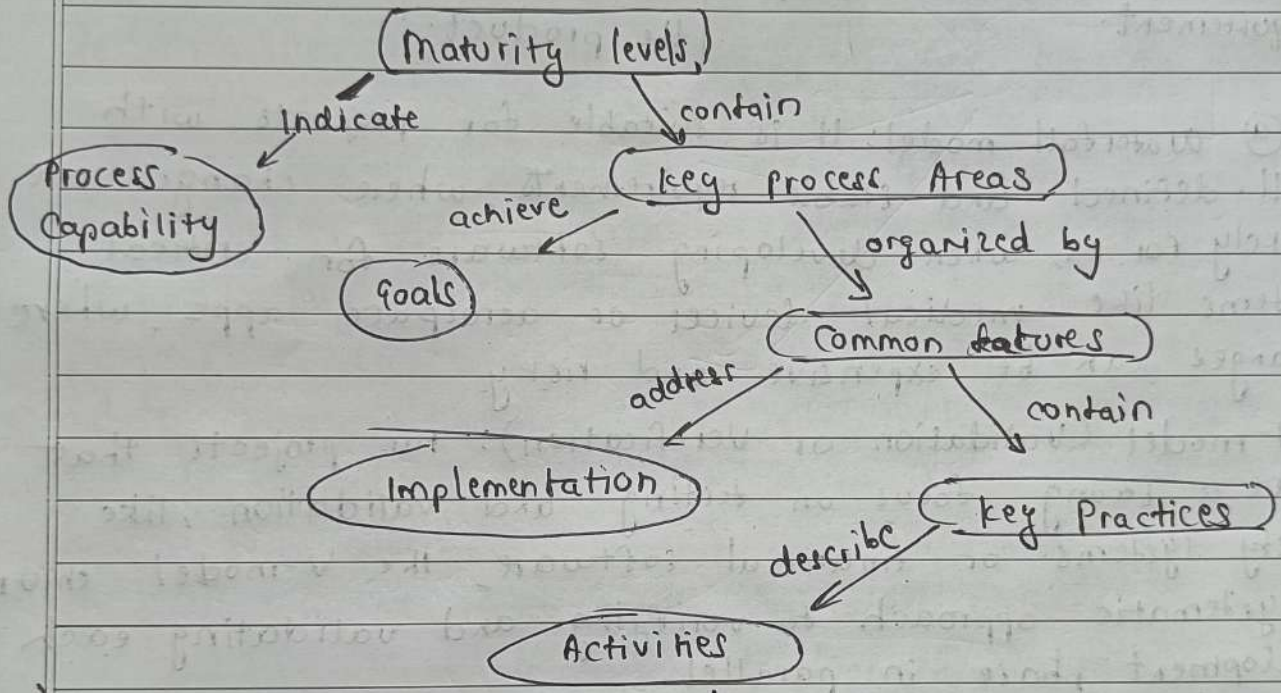
level 3: Defined.
All projects use an approved, tailored version of the organization's standard software process for developing and maintaining software.

level 4: Managed.
Detailed measures of the software processes 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 processes 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.

⑤ Ex: Water fall Model, incremental model.

⑥ It can accomodate charging requirement.

Time does not allow a full and complete system to be developed.

Ex: Prototyping, spiral and Concurrent models.

Improvement is required in the product.

Q5) ⓐ Waterfall model: It is suitable for projects with well-defined and stable requirements, where changes are unlikely. For ex, when developing software for critical systems like medical devices or aerospace apps, where changes can be expensive and risky.

ⓑ V-model (Validation or Verification): For projects that require a strong focus on testing and validation, like safety systems or financial software, the V-model ensures a systematic approach to verifying and validating each development phase in parallel.

ⓒ Incremental model: When a project has a tight schedule but the full set of requirements can't be defined upfront, the incremental model works wells. It's suitable for projects involving e-commerce platforms where features can be developed and delivered in stages.

ⓓ RAD Model: It is useful in scenarios where there's a need for rapid software development and quick delivery, typically in business-focused projects. For ex, Costomer relationship management (CRM) system for a sales team that requires constant updates and improvements to stay competitive.

(e) ~~The~~ Prototype Model: It is beneficial when the client's requirements are not well-defined or are subject to frequent changes. When creating a user interface for a new mobile app, using a prototype allows stakeholders to visualize the design early on and provide feedback for refinements before the final product is developed.

(f) Spiral model: In situations where the project involves a high level of risk assessment and continuous refinement such as large scale government projects, this model provides a structured approach for iterative development while ~~mang~~ managing risks effectively.

(g) Agile Model: Agile methodologies like scrum are beneficial when requirements are constantly evolving and the development team needs to adapt quickly. Its useful for projects involving web development, mobile apps where iterative development and customer feedback are crucial.

Q(d) The waterfall and Agile models differ significantly in terms of project planning and progress tracking.

(1) Waterfall model:

(a) Project Planning: It follows a sequencial approach. Planning is done extensively at the beginning of the project, where all requirements are gathered and documented before development begins.

(b) Progress 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:

ⓐ Project Planning: Agile approaches (like scrum or kanban) emphasize flexibility and adaptability. Planning is done in iterations, with focus on delivering smaller, functional increments of the project.

ⓑ Progress tracking: Agile relies on frequent iterations and feedback loops. Teams track progress using tools like burndown charts, velocity measurements, and daily stand-up meetings. This allows for course correction and adjustments as needed.

Waterfall model follows a rigid, linear planning process with limited room for changes, while Agile methodologies promote iterative planning and ongoing adjustments based on regular feedback, fostering a more adaptable approach to project management.

⑦) Project metrics refer 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:

ⓐ Development speed: longer development cycles might lead to slower speed compared to Agile methodologies.

ⓑ Adaptibility to change: it is less adaptable to change due to its sequencial nature, which can lead to challanges when change is required.

© 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 satisfaction: Frequent feedback loops and incremental improvements 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: Continuous delivery and focus on flow can lead to better customer satisfaction by quickly addressing their needs.

Q8) These 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.