

Class Assignment 2

Question 1: Discuss Elicitation analysis in requirement engineering.

Solution: Elicitation Analysis in Requirement Engineering

Elicitation analysis is a critical step in the requirement engineering process, focusing on the effective gathering, analysis, and prioritization of the requirements from all relevant stakeholders. This phase aims to understand and document the needs, constraints, and goals of stakeholders, ensuring that the system being developed aligns with their expectations. Elicitation techniques commonly used include interviews, surveys, focus groups, brainstorming sessions, and document analysis. These methods help to uncover both explicit and implicit needs, clarify ambiguities, and highlight potential conflicts early in the process.

The role of elicitation analysis extends beyond just collecting requirements; it is also about understanding the context in which the system will operate, identifying potential risks, and resolving conflicting requirements through discussions and negotiations. By properly analyzing the gathered information, the team ensures that the final requirements are clear, complete, and feasible, ultimately leading to more successful and efficient system development.

Question 2: Discuss specification in requirement engineering.

Solution: Specification in Requirement Engineering

Specification in requirement engineering refers to the formal process of documenting the system's requirements in a clear, comprehensive, and detailed manner. This step takes the requirements gathered during the elicitation phase and translates them into a structured format that can be easily interpreted by both technical and non-technical stakeholders. The specification document serves as a formal contract between the stakeholders and the development team, ensuring that everyone is aligned on the project's objectives and scope.

The specification includes two main types of requirements: functional and non-functional. Functional requirements define the specific behaviors, actions, and functions that the system must perform, while non-functional requirements describe the system's quality attributes, such as performance, security, and usability. The specification document serves as a baseline for the design and development of the system and is used throughout the lifecycle of the project, ensuring that the developed system meets the intended goals.

Question 3: Describe Waterfall model.

Solution: Waterfall Model

The Waterfall model is a traditional, linear approach to software development in which the process is divided into distinct, sequential phases. These phases are Requirement Analysis, Design, Implementation, Testing, Deployment, and Maintenance. In this model, each phase must be completed before the next one begins, and there is minimal overlap between phases. The Waterfall model is known for its simplicity and structure, making it ideal for projects with well-defined requirements and minimal changes during the development process.

While the Waterfall model offers a clear, step-by-step approach, it has limitations, particularly in dynamic or uncertain projects. The lack of flexibility to accommodate changes during development can lead to challenges if requirements evolve or unexpected issues arise. Despite these drawbacks, the Waterfall model remains a popular choice for projects where requirements are stable and well-understood, such as in government or highly regulated industries.

Question 4: Explain Scrum framework.

Solution: Scrum Framework

Scrum is an agile framework that emphasizes iterative development, flexibility, and collaboration. It is designed for teams working on complex projects, where requirements and solutions evolve over time. Scrum divides work into small, time-boxed iterations known as Sprints, which typically last between 2 to 4 weeks. At the start of each Sprint, a team selects a set of items from the product backlog, a prioritized list of tasks, to work on during the Sprint. The goal is to deliver small, incremental improvements to the product at the end of each Sprint.

Scrum involves several key roles, including the Product Owner, who is responsible for managing the product backlog and ensuring that the team works on the highest-priority items; the Scrum Master, who helps the team follow Scrum practices and removes any obstacles; and the Development Team, which is responsible for delivering the work. Scrum also uses artifacts such as the Product Backlog, Sprint Backlog, and Increment, which help in tracking progress and managing the project.

Question 5: Explain Kanban framework.

Solution: Kanban Framework

Kanban is a lean methodology that focuses on visualizing and optimizing workflows to improve efficiency and productivity. Unlike Scrum, which organizes work into iterations, Kanban operates on a continuous flow basis, where work is managed as it moves through various stages of the process. The primary tools in Kanban are the Kanban board and work items, which are represented by cards. These cards are placed in columns on the board that represent different stages of the workflow, such as "To Do," "In Progress," and "Completed."

Kanban's key principles include limiting Work in Progress (WIP), focusing on continuous delivery, and optimizing flow. By limiting WIP, teams can avoid overloading resources and can more easily identify

bottlenecks in the process. Kanban helps teams improve productivity and maintain focus on the most critical tasks without being constrained by rigid iterations or schedules. It is an excellent framework for teams working on projects that require flexibility and continuous improvement.

Question 6: Describe 5 different Effort estimation models.

Solution: Effort Estimation Models

Effort estimation is an essential part of project planning and helps in determining the amount of time and resources needed to complete a project. Below are five common models used to estimate effort in software development:

- **Expert Judgment:** This model relies on the expertise and experience of individuals or groups who have a good understanding of similar projects. Experts make estimations based on their knowledge and intuition, making this approach subjective and dependent on the availability of experienced professionals.
- **Function Point Analysis (FPA):** FPA is a method that quantifies the functional components of the software, such as inputs, outputs, inquiries, and files. By counting these elements, an estimate of the effort can be made, which is often used in large-scale software development projects.
- **COCOMO (Constructive Cost Model):** COCOMO is an empirical model that uses historical data and factors such as project complexity, team experience, and product size to estimate the effort required. COCOMO offers three levels of estimation: basic, intermediate, and detailed.
- **Use Case Points (UCP):** UCP estimates effort based on the use cases of a system. Each use case is assigned a weight based on its complexity, and actors (users or systems interacting with the system) are assigned complexity points. The total effort is then estimated based on these weights.
- **Wideband Delphi:** This is a group-based estimation technique where experts provide individual estimates, discuss variations, and eventually converge on a consensus estimate. It allows for diverse perspectives to be considered in the estimation process.

Question 7: Explain various Code Refactoring techniques.

Solution: Code Refactoring Techniques

Code refactoring involves restructuring existing code without altering its external behavior. The goal is to improve code quality, readability, maintainability, and efficiency. Several common code refactoring techniques include:

- **Renaming Variables and Methods:** Giving meaningful names to variables and methods helps improve code readability and makes the code more understandable.
- **Extract Method:** This technique involves taking a section of code from a larger method and turning it into its own method. It enhances modularity and makes the code easier to understand and test.

- **Introduce Parameter Object:** Instead of passing many parameters to a method, they are grouped into a single object. This reduces method complexity and improves clarity.
- **Encapsulate Field:** This involves making fields private and providing getter and setter methods to access them. It improves data encapsulation and ensures better control over data access.
- **Replace Magic Numbers with Constants:** Instead of using unexplained literal values, we replace them with named constants. This improves code clarity and makes future updates easier.