SET: 2

KONERU LAKSHMAIAH EDUCATION FOUNDATION
(Deemed to be University, Estd. u/s. 3 of UGC Act 1956)

Academic year: 2021-22
Sem-In Examinations-I,
B. Tech II nd Year, 1st Semester
Subject Code: 21CS2111RA Subject Name Software Engineering

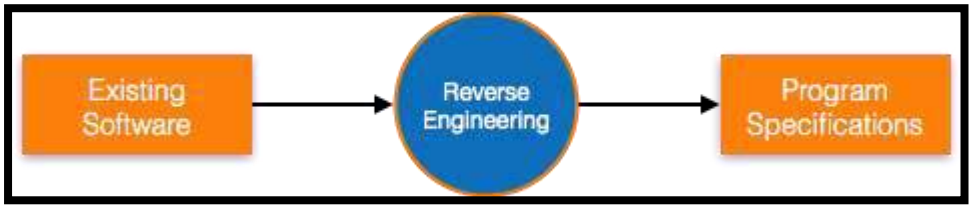
Answer Key

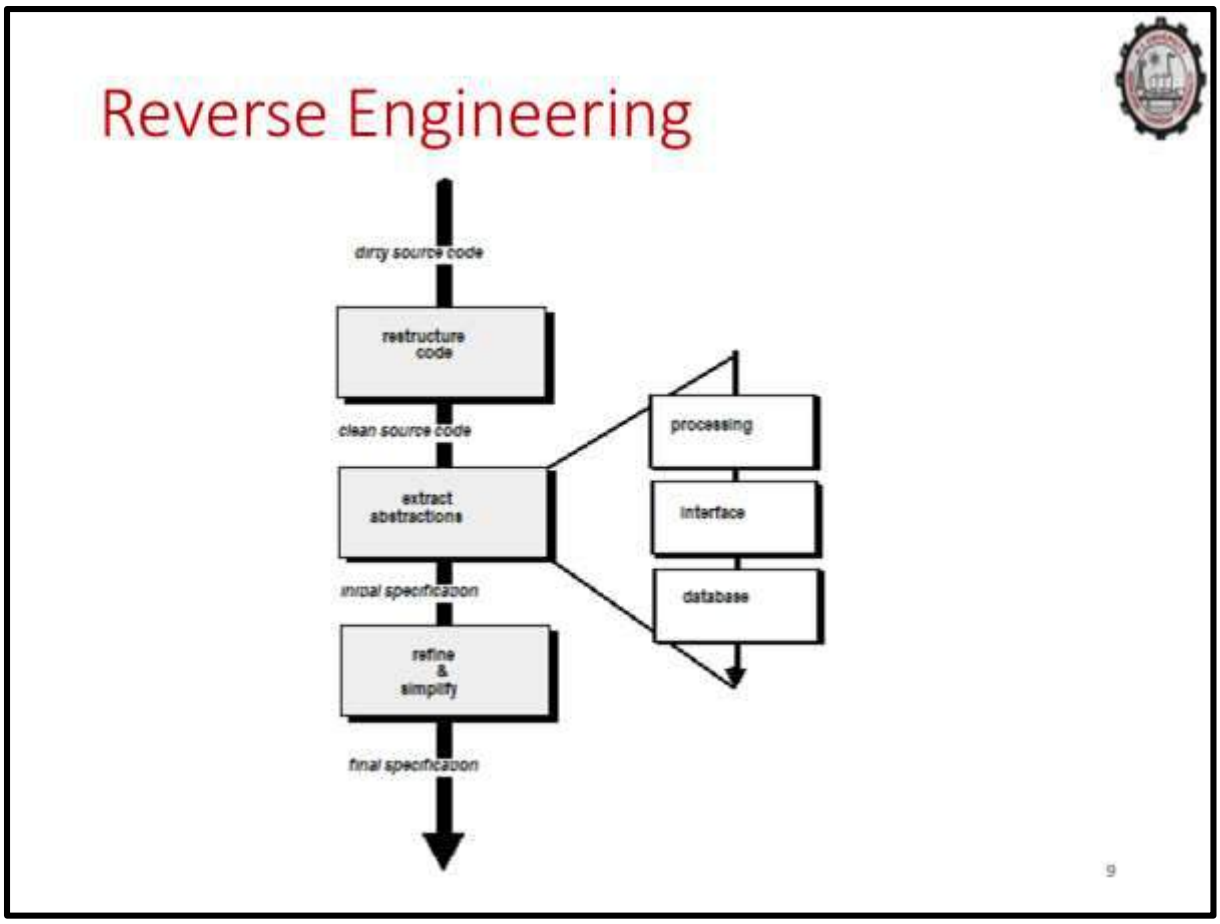
Time: 2 hours **Max. Marks: 50**

CO-1

1. **Define reverse engineering and sketch the process of reverse engineering with a help of neat diagram**

- It is a process to achieve system specification by thoroughly analyzing, understanding the existing system. This process can be seen as reverse SDLC model, i.e. we try to get higher abstraction level by analyzing lower abstraction levels.
- An existing system is previously implemented design, about which we know nothing. Designers then do reverse engineering by looking at the code and try to get the design. With design in hand, they try to conclude the specifications. Thus, going in reverse from code to system specification.





---

2. **Define Software and Software Engineering**

Software is:

(1) instructions (computer programs) that when executed provide desired features, function, and performance;

(2) data structures that enable the programs to adequately manipulate information and

(3) documentation that describes the operation and use of the programs

[Software engineering is] the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.

Software Engineering:

The application of a Systematic, Disciplined, Quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.

3.**Examine the perspective approach for the structure and the order in Software Development**

The name 'prescriptive' is given because the model prescribes a set of activities, actions, tasks, quality assurance and change the mechanism for every project.

What is the prescriptive process model?

A prescriptive process model is a model that describes "how to do" according to a certain software process system. A prescriptive model prescribes how a new software system should be developed

- Waterfall Model.
- Incremental Process Model.
- Evolutionary Process Model.
- Concurrent model

---

4. Outline the essence of software engineering practices

- How does the practice of software engineering fit in the process activities mentioned above? Namely,
  - communication,
  - planning,
  - modeling,
  - construction
  - deployment.

The essence of problem solving is outlined in 4 points:

1. *Understand the problem* (communication and analysis).

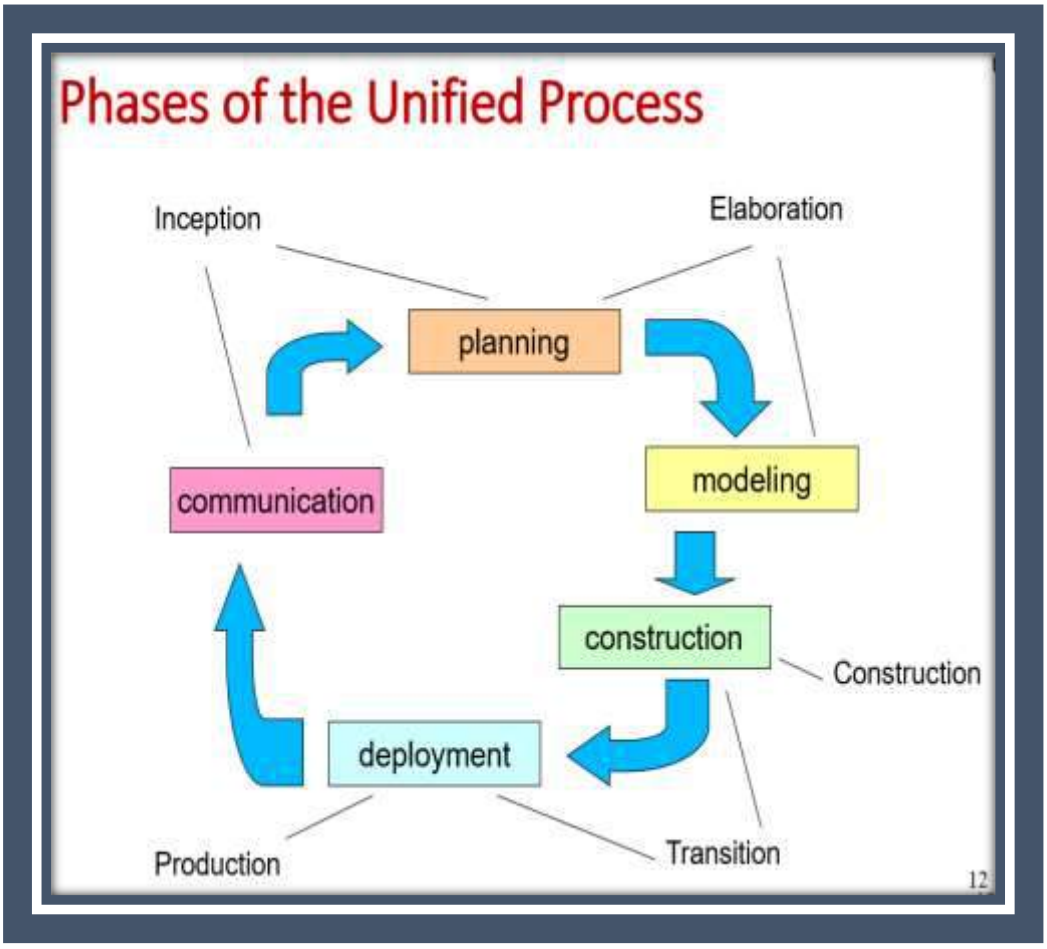
2. *Plan a solution* (modeling and software design).

3. *Carry out the plan* (code generation).

4. *Examine the result for accuracy* (testing and quality assurance).

5. a) Classify the unified process in detail

- Unified process developed which is a framework for object-oriented software engineering using UML
    - Draws on the best features and characteristics of conventional software process models
    - Emphasizes the important role of software architecture
    - Consists of a process flow that is iterative and incremental, thereby providing an evolutionary feel
- Consists of 5 phases:
    - Inception
    - Elaboration
    - Construction
    - Transition
    - Production



- Encompasses both customer communication and planning activities of the generic process
- Business requirements for the software are identified
- A rough architecture for the system is proposed
- A plan is created for an incremental, iterative development
- Fundamental business requirements are described through preliminary use cases
    - A use case describes a sequence of actions that are performed by a user
- Encompasses both the planning and modelling activities of the generic process
- Refines and expands the preliminary use cases
- Expands the architectural representation to include five views
    - Use-case model
    - Analysis model
    - Design model
    - Implementation model
    - Deployment model
- Often results in an executable architectural baseline that represents a first cut executable system
- The baseline demonstrates the viability of the architecture but does not provide all features and functions required to use the system
- Encompasses the construction activity of the generic process
- Uses the architectural model from the elaboration phase as input
- Develops or acquires the software components that make each use-case operational
- Analysis and design models from the previous phase are completed to reflect the final version of the increment
- Use cases are used to derive a set of acceptance tests that are executed prior to the next phase
- Encompasses the last part of the construction activity and the first part of the deployment activity of the generic process
- Software is given to end users for beta testing and user feedback reports on defects and necessary changes
- The software teams create necessary support documentation (user manuals, trouble-shooting guides, installation procedures)

- At the conclusion of this phase, the software increment becomes a usable software release
- Encompasses the last part of the deployment activity of the generic process
- On-going use of the software is monitored
- Support for the operating environment (infrastructure) is provided
- Defect reports and requests for changes are submitted and evaluated
-

---

5 b) Explain how umbrella activities are useful while developing a software

What are umbrella activities in software engineering?

- Software engineering is a collection of interconnected phases. These steps are expressed or available in different ways in different software process models. Umbrella activities are a series of steps or procedures followed by a software development team to maintain the progress, quality, changes, and risks of complete development task

Need for umbrella activities

- In general, umbrella activities are applied throughout a software project and help a software team manage and control progress, quality, change, and risk. Since the software engineering process is not a rigid regimen that must be followed precisely by a software team, the process has a lot of room for adaptation
- Complete the five process framework activities and help team manage and control progress, quality, change, and risk.
- Software project tracking & control: assess progress against the plan and take actions to maintain the schedule.
- Risk management: assesses risks that may affect the outcome and quality.
- Software quality assurance: defines and conduct activities to ensure quality.
- Technical reviews: assesses work products to uncover and remove errors before going to the next activity.
- Measurement: define and collects process, project, and product measures to ensure stakeholder's needs are met.
- Software configuration management: manage the effects of change throughout the software process.
- Reusability management: defines criteria for work product reuse and establishes mechanism to achieve reusable components.
- Work product preparation and production: create work products such as models, documents, logs, forms and lists.

---

6) a) Explain any two specialized process models in detail

Student can write any two specialized process models

Specialized Process Models: Specialized process models use many of the characteristics of one or more of the conventional models presented so far, however they tend to be applied when a narrowly defined software engineering approach is chosen. They include

1. Component Based Development
2. Formal Methods Model
3. Aspect-Oriented Software Development
4. Unified process

CBD

- The component based development model incorporates many of the characteristics of the spiral model.
- The process to apply when reuse is a development objective
- It Consists of the following process steps
    - Available component-based products are researched and evaluated for the application domain in question
    - Component integration issues are considered
    - A software architecture is designed to accommodate the components
    - Components are integrated into the architecture
    - Comprehensive testing is conducted to ensure proper functionality
- Capitalizes on software reuse, which leads to documented savings in project cost and time

FDM

- Encompasses a set of activities that leads to formal mathematical specification of computer software
- Enables a software engineer to specify, develop, and verify a computer-based system by applying a rigorous, mathematical notation
- Ambiguity, incompleteness, and inconsistency can be discovered and corrected more easily through mathematical analysis
- Offers the promise of defect-free software
- Used often when building safety-critical systems

ASOE

It provides a process and methodological approach for defining, specifying, designing, and constructing aspects

As modern computer based systems become more sophisticated and complex there are certain *concerns by the customer*

- required properties or areas of technical interest.
- Span the entire architecture
- High-level properties of a system (e.g; security, fault tolerance)
- Other concern affect functions (e.g; the application of business rules)
- While others are systemic (e,g; task synchronization or memory management)

---

6 b) How incremental model is employed in spiral model

Spiral model is a software development model and is made with features of incremental, waterfall or evolutionary prototyping models. Incremental Model is a software development model where the product is, analyzed, designed, implemented and tested incrementally until the product is finished

The various phases of incremental model are as follows:

1. Requirement analysis: In the first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team. To develop the software under the incremental model, this phase performs a crucial role.

2. Design & Development: In this phase of the Incremental model of SDLC, the design of the system functionality and the development method are finished with success. When software develops new practicality, the incremental model uses style and development phase.

3. Testing: In the incremental model, the testing phase checks the performance of each existing function as well as additional functionality. In the testing phase, the various methods are used to test the behavior of each task.

. Implementation: Implementation phase enables the coding phase of the development system. It involves the final coding that design in the designing and development phase and tests the functionality in the testing phase. After completion of this phase, the number of the product working is enhanced and upgraded up to the final system product

When we use the Incremental Model?

- When the requirements are superior.
- A project has a lengthy development schedule.

- When Software team are not very well skilled or trained.
- When the customer demands a quick release of the product.
- You can develop prioritized requirements first.

Advantage of Incremental Model

- Errors are easy to be recognized.
- Easier to test and debug
- More flexible.

. Implementation: Implementation phase enables the coding phase of the development system. It involves the final coding that design in the designing and development phase and tests the functionality in the testing phase. After completion of this phase, the number of the product working is enhanced and upgraded up to the final system product

When we use the Incremental Model?

- When the requirements are superior.
- A project has a lengthy development schedule.
- When Software team are not very well skilled or trained.
- When the customer demands a quick release of the product.
- You can develop prioritized requirements first.

Advantage of Incremental Model

- Errors are easy to be recognized.
- Easier to test and debug
- More flexible.

Definition: The spiral model is similar to the incremental development for a system, with more emphasis placed on risk analysis. The spiral model has four phases: Planning, Design, Construct and Evaluation. A software project repeatedly passes through these phases in iterations (called Spirals in this model).

Description: These phases are - Planning: This phase starts with the gathering of business requirements. In the subsequent spirals as the product matures, identification of system requirements and unit requirements are done in this phase. This also includes understanding of system requirements by continual communication between the customer and the analyst. At the end of the spiral the product is deployed.

Design: Design phase starts with the design in the baseline spiral and involves architectural, logical design of modules, physical product design and final design in the successive spirals. Construct: Construct phase refers to development of the final software product at every spiral. In the spiral when the product is just thought and the design is being developed, a Proof of Concept (POC) is developed in this phase to get the users' feedback. Then in the successive spirals with higher clarity on requirements and design a working model of the software called build is developed with a version number. These versions are sent to the users for feedback.

CO2

7. Sketch the life cycle activities of Extreme Programming with a neat diagram

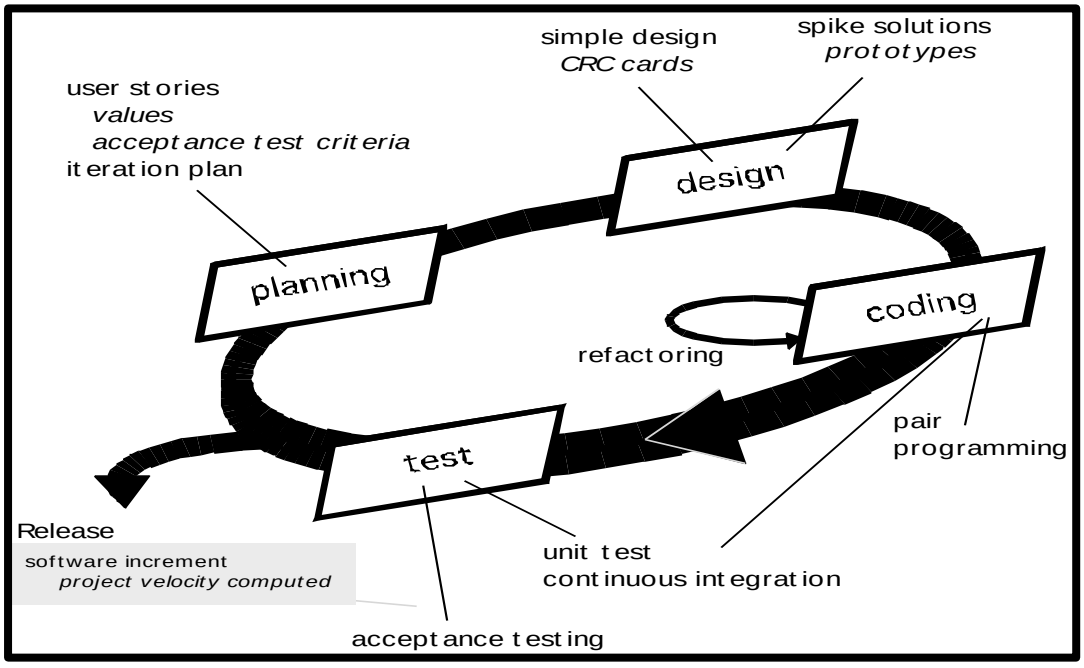
Extreme Programming Process Cycle

Extreme Programming is iterative and incremental and is driven by Time-Boxed Cycles.

Extreme Programming has the following activity levels −

- Product Life Cycles
- Releases
- Iterations
- Tasks
- Development
- Feedback

The Extreme Programming process cycle is illustrated below.



Development

The developers form pairs, which can be a continuous and dynamic activity.

Each Pair −

- Verifies their understanding of the story.
- Determines detailed implementation approach, ensuring simple design.
- Begins test-driven development, i.e., write unit test, implement code to pass the unit test, refactor to make the code simple.
- Integrates their code to the system code base at appropriate intervals.
- Reviews the progress frequently.

Feedback

Pairs constantly communicate within themselves and outward to the team as well. On-line customer is also involved in the communication on a continuous basis. Certain teams resort to daily stand-up meetings to discuss the overall team status quickly and the possible re-synchronization and micro-planning if necessary.

The iteration and release reviews provide overall status and points for process adjustment and improvement.

- Development episodes may cause rethinking of tasks.
- Task development may cause rethinking of stories.
- Story re-estimation may cause iteration changes or recovery.
- Iteration results may cause changes to release plan.

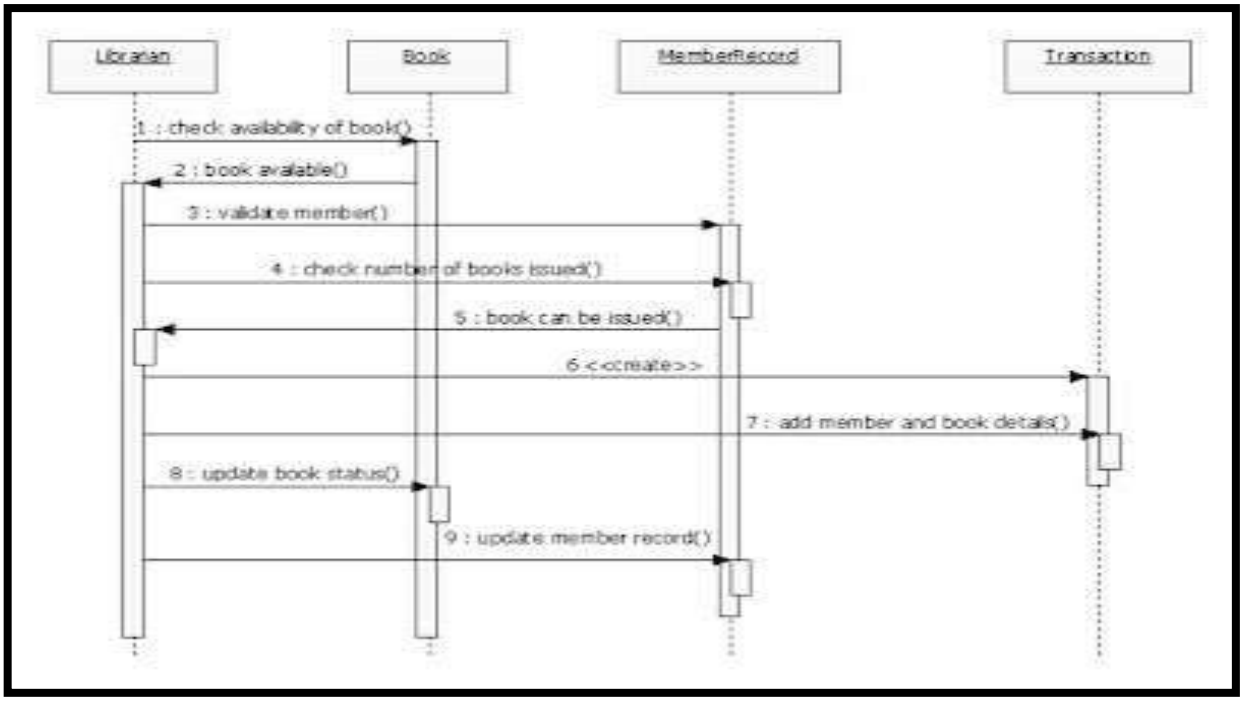
---

8. Illustrate the process of incorporating quality in generating requirements for any social networking website

Quality requirements (QRs) **describe the desired quality of software**, and they play an important role in the success of software projects.

1. Identify the community.
2. Define the features and functions.
3. Choose the right technology.
4. A must have structure. ...
5. Design Activity Stream. ...
6. Create Status Update Features. ...
7. Quality Viewing Data options. ...
8. You need to attract the right users.

9. Generate a sequence diagram for return of a book in Library Management System



10.Validation activities take place at the beginning of every software process iteration, Outline all of them.

The validation process involves activities like unit testing, integration testing, system testing and user acceptance testing.

Unit testing is testing the smallest testable unit of an application. It is done during the coding phase by the developers.

Integration testing is performed using the black box method. This method implies that a testing team interacts with an app and its units via the user interface.

System testing is the process in which a quality assurance (QA) team evaluates how the various components of an application interact together in the full, integrated system or application.

User acceptance testing (UAT), also called application testing or end-user testing, is a phase of software development in which the software is tested in the real world by its target users.

---

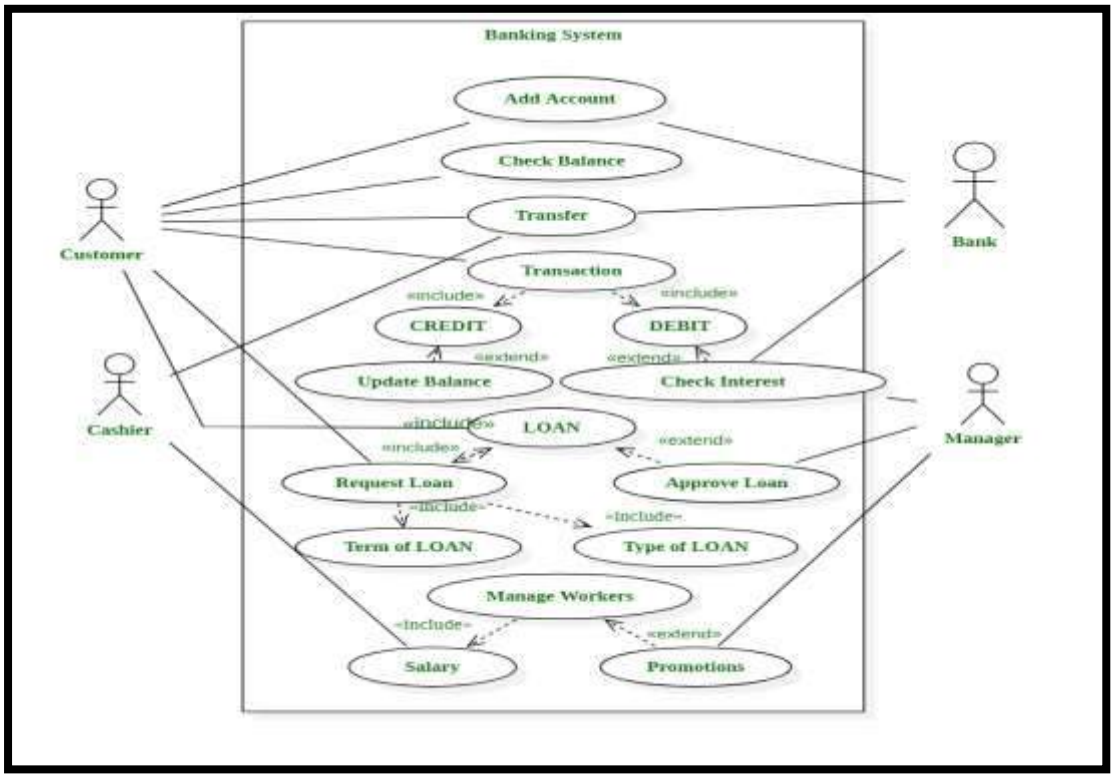
11 a. Mark out the types of requirements identified by Quality Function Deployment

Quality Function Deployment (QFD) is process or set of tools used to define the customer requirements for product and convert those requirements into engineering specifications and plans such that the customer requirements for that product are satisfied.

Key steps in QFD :

1. Product planning :
   - Translating what customer wants or needs into set of prioritized design requirements.
   - Prioritized design requirements describe looks/design of product.
   - Involves benchmarking – comparing product's performance with competitor's products.
   - Setting targets for improvements and for achieving competitive edge.
2. Part Planning :
   - Translating product requirement specifications into part of characteristics.
   - For example, if requirement is that product should be portable, then characteristics could be light-weight, small size, compact, etc.
3. Process Planning :
   - Translating part characteristics into an effective and efficient process.
   - The ability to deliver six sigma quality should be maximized.
4. Production Planning :
   - Translating process into manufacturing or service delivery methods.
   - In this step too, ability to deliver six sigma quality should be improved.

11 b. Identify and draw the role of use case diagram for Online Banking case study while describing the steps involved in it.



Some possible scenarios of the system are explained as follows :

1. A Customer is required to create an account to avail services offered by Bank. Bank verifies detail and creates new account for each new customer. Each customer is an actor for the Use-Case Diagram and the functionality offered by Online Banking System to Add Account is Use-Case.

2. Each customer can check the balance in bank account and initiate request to transfer an account across distinct branches of Bank. Cashier is an employee at bank who supports service to the customer.
3. A customer can execute cash transactions where the customer must either add cash value to bank account or withdraw cash from account. Either of two or both that is credit as well as debit cash, might be executed to successfully execute one or multiple transactions.
4. After each successful transaction customer might or might not want to get details for action. Manager can check interest value for each account corresponding to transaction to ensure and authenticate details.
5. A customer can also request loan from bank where customer must add request for loan with the appropriate details.
6. The type of loan in accordance with purpose or the need for loan and term or duration to pay back the loan must be provided by customer.
7. The manager of each branch of bank has choice to either accept or approve loan to initiate process further or just reject request for loan based on terms and conditions.
8. The record for each employee of bank is maintained by bank and bank manages all employees of each branch of bank. The manager of each branch has choice to offer bonus to employees. Note here that each employee is paid as part of management of staff but promotion or bonus might or might not be offered certainly to each employee.
9.

---

12 a. Narrate how is Refactoring useful in the agile development process.

Refactoring is the practice of continuously improving the design of existing code, without changing the fundamental behavior. In Agile, teams maintain and enhance their code on an incremental basis from Sprint to Sprint. If code is not refactored in an Agile project, it will result in poor code quality, such as unhealthy dependencies between classes or packages, improper class responsibility allocation, too many responsibilities per method or class, duplicate code, and a variety of other types of confusion and clutter. Refactoring helps to remove this chaos and simplifies the unclear and complex code.

Challenges:
Though refactoring brings a lot of benefits to the code quality of the software, there are multiple challenges that dissuade developers in Agile projects from continuously refactoring the code. Following are a few challenges that are mostly seen in Agile projects

- Time Constraint: Time is the biggest challenge for doing refactoring in Agile projects as the Sprints are time-boxed with a defined set of deliverables.
- Reluctance: If the code is working fine without any refactoring done, there will be an inclination towards not revisiting the code. This is primarily because of the mindset that there is no error and hence no need to do additional activities i.e. refactoring.
- Integration with branches: To integrate the code across different branches post refactoring is considered a challenge
- Fear factor: Developer often fears that refactoring will introduce bugs and break the existing functionality which is working fine
- Re-Testing: In case automated test suites are not available, the developer is discouraged to do refactoring with the additional effort of manual testing to check the functionality

- Backward Compatibility: Backward compatibility often prevents developers from starting refactoring efforts.

12 b. **Set out a plan for establishing the ground work in requirement engineering**

Establishing the groundwork in requirement engineering

1. Identifying the stakeholders.
2. Recognizing Multiple Viewpoints.
3. Working toward Collaboration.
4. Asking the first questions.

1. Stakeholder Requirements, often referred to as user needs or user requirements, describe what users do with the system, such as the activities that users must be able to perform.
2. To ensure as far as possible that the system to be implemented meets the needs and expectations of all involved stakeholders, it is necessary to understand their various viewpoints and manage any inconsistencies and conflicts.
3. A collaborative engineering approach allows an organization to design anywhere, build anywhere, and service everywhere. Key stakeholders throughout the supply chain process work collaboratively to create solutions, resolve conflicts, and make decisions on critical actions for any given design and manufacturing project.
4. The first step is to gather, analyze and develop requirements from the Concept of Operations, stakeholder needs, objectives, and other external requirements. Once requirements are documented, they are prioritized, de-conflicted, and validated with the stakeholders.
