

Register No.

2	0	1	7	2	0	5	9
---	---	---	---	---	---	---	---

BTech Degree Examination December 2022

Fifth Semester

Information Technology

20ITT53 – SOFTWARE ENGINEERING

(Regulations 2020)

Time: Three hours

Maximum: 100 marks

Answer all Questions

Part – A ($10 \times 2 = 20$ marks)

1. Write any two characteristics of the software. [CO1,K1]
2. Compare the features of Incremental model over waterfall model. [CO1,K2]
3. Specify the problems that are identified during requirement elicitation process. [CO2,K2]
4. Name some requirements of Quality function deployment (QFD). [CO2,K1]
5. Differentiate between static model and Dynamic model in UML diagrams. [CO3,K2]
6. Provide a complete set of CRC model index cards for the Students Record information system. [CO3,K3]
7. List any four architectural design styles. [CO4,K2]
8. State the golden rules for an good user interface design. [CO4,K2]
9. In what way, does the verification testing differ from validation testing? [CO5,K2]
10. Specify the need of Software Configuration Management System. [CO5,K2]

Part – B ($5 \times 16 = 80$ marks)

11. a. Elaborate the need of evolutionary approach for software development. Explain in brief about the common models used in evolutionary approach. (16) [CO1,K2]
- (OR)
- b. i) Diagrammatically explain about the SCRUM process. (8) [CO1,K2]
- ii) "Refactoring is encouraged in Extreme programming (XP)"-Justify (8) [CO1,K2]
12. a. Summarize the tasks associated with requirement gathering process and elaborate the importance of it. (16) [CO2,K3]
- (OR)
- b. Elaborate the procedures in collecting the requirements. Explain how the gathered requirements are validated and how will you analyze whether the requirements are sufficient. (16) [CO2,K3]

13. a. Develop a complete use-case, activity and class diagram for the following (16) [CO3,K3] activities.

- a) Making a cash deposit in an ATM.
- b) Making a cash withdrawal from an ATM.
- c) Change of PIN number.

(OR)

b. Name the common elements of class-based modelling. Illustrate the need of each (16) [CO3,K3] element with suitable examples.

14. a. Define software architecture. Elaborate the features of different architectural (16) [CO4,K3] styles with neat diagram.

(OR)

b. Provide the various dimensions of Design model. Illustrate with examples on (16) [CO4,K3] different elements of design model.

15. a. i) Discuss about the elements involved in software configuration (8) [CO5,K2] management.

ii) Elaborate the features of SCM process. (8) [CO5,K2]

(OR)

b. Elaborate the various Test strategies handled for conventional and object (16) [CO5,K2] oriented software. Explain with an example.

Bloom's Taxonomy Level	Remembering (K1)	Understanding (K2)	Applying (K3)	Analysing (K4)	Evaluating (K5)	Creating (K6)
Percentage	2	44	54			

B.Tech Degree Examination December 2022

Fifth semester

Information Technology

20ITT53-Software Engineering

(Regulations 2020)

Answer key

PART – A ($10 \times 2 = 20$ marks)

1. Two characteristics of Software: [CO1,K1]

Reliability, Correctness, Integrity, Efficiency, Usability, Security, Safety, Interoperability, Reusability, Portability, Maintainability, Extensibility, Testability, Scalability.

2. Compare the features of Incremental model over waterfall model [CO1,K2]

Sr. No.	Incremental Model	WaterFall Model
1	Incremental Model is the development model in which the entire model is divided into various sub development phase where corresponding testing phase for each development phase is practices. In other words we can say that for every stage in the development cycle, there is an associated testing phase and corresponding testing phase of the development phase is planned in parallel.	On other hand Waterfall model there is first development of the application and after which the different testing of application take place. In other words we can say that in WaterFall the complete process is divided into several phases among which one phase should be completed in order to reach the next phase and testing is almost at end phase of the development.
2	As mentioned above that in Incremental Model the execution of the phases i.e., development and testing happens in a sequential manner so type of Incremental Model is Sequential/Parallel in nature.	On other hand WaterFall Model is a relatively linear sequential design approach as each phase should be completed in order to reach the next phase. So type of this model is Continuous in nature.
3	In Incremental Model each development phase get tested at its own level and hence no pending testing occurs in this model also if any validation requires to be implemented then it could be implemented at that phase.	On other hand in case of WaterFall Model the testing occurs after development is completed and thus if any missing validation is identified to be implemented then first that phase of development needs to be recognized and then that validation get implemented.

4	As sequential phases need to be functional hence the cost is higher as compared to that of WaterFall Model also the complexity is more than WaterFall.	On other hand in WaterFall Model due to linear development only one phase of development is operational and hence cost and complexity is low as compared to that of Incremental Model.
5	In Incremental Model the probability of total number of defects in the development of application is low as testing is done in parallel to the development.	On other hand in WaterFall Model the probability of total number of defects in the development of application is high as testing is done post development.

3. Problems that are identified during requirement elicitation process [CO2,K2]

1. Understanding large and complex system requirements is difficult
2. Undefined system boundaries
3. Customers/Stakeholders are not clear about their needs
4. Conflicting requirements are there
5. Changing requirements is another issue
6. Partitioning the system suitably to reduce complexity
7. Validating and Tracing Requirements.
8. Identifying Critical requirements.

4. Requirements of Quality Function Deployment (QFD): [CO2,K1]

Normal requirements
Expected requirements
Exciting requirements

5. Differentiate between Static Model and Dynamic model: [CO3,K3]

The most notable difference between static and dynamic models of a system is that while a dynamic model refers to runtime model of the system, static model is the model of the system not during runtime. Another difference lies in the use of differential equations in dynamic model which are conspicuous by their absence in static model. Dynamic models keep changing with reference to time whereas static models are at equilibrium of in a steady state

6. CRC model for Student Record Information [CO3,K3]

Class Name : Student Record	
Responsibilities: Student Rollno Name Address Phone Number Course name Department Date of Birth	Student Information

7. **Architectural Design Styles :** [CO4,K2]
 Data-centered architecture
 Data-flow architecture
 Layered architecture
 Call and return architecture

8. **Golden rules for user interface design** [CO4,K2]
 • 1.Place the user in control.
 • 2. Reduce the user's memory load.
 • 3. Make the interface consistent.

9. **Validation and verification difference** [CO5,K2]

Verification	Validation
The verifying process includes checking documents, design, code, and program	It is a dynamic mechanism of testing and validating the actual product
It does not involve executing the code	It always involves executing the code
Verification uses methods like reviews, walkthroughs, inspections, and desk- checking etc.	It uses methods like Black Box Testing, White Box Testing , and non-functional testing
Whether the software conforms to specification is checked	It checks whether the software meets the requirements and expectations of a customer
It finds bugs early in the development cycle	It can find bugs that the verification process can not catch
Target is application and software architecture, specification, complete design, high level, and database design etc.	Target is an actual product
QA team does verification and make sure that the software is as per the requirement in the SRS document.	With the involvement of testing team validation is executed on software code.
It comes before validation	It comes after verification

10. **Needs of Software Configuration Management System** [CO5,K1]

- There are multiple people working on software which is continually updating
- It may be a case where multiple version, branches, authors are involved in a software config project, and the team is geographically distributed and works concurrently

- Changes in user requirement, policy, budget, schedule need to be accommodated.
- Software should be able to run on various machines and Operating Systems
- Helps to develop coordination among stakeholders
- SCM process is also beneficial to control the costs involved in making changes to a system

PART – B (5 × 16 = 80 marks)

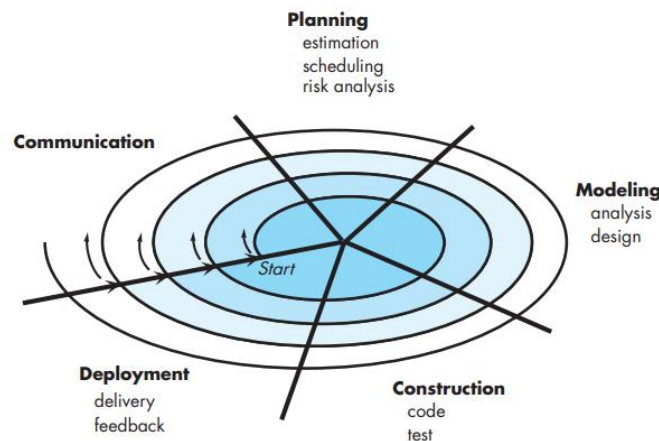
11. Need of Evolutionary approach and its common models.

[CO1,K2]

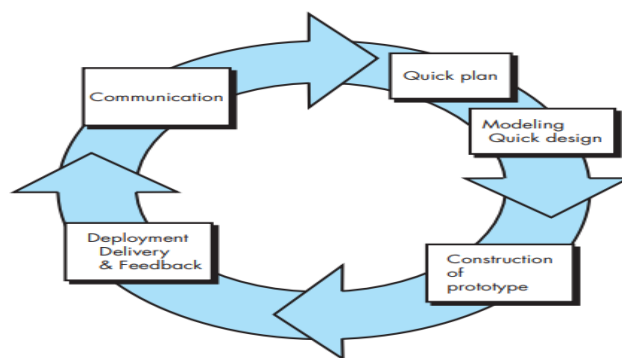
a.

Definition and need of Evolutionary approach -(4)

Explanation and diagram for Spiral model -(6)



Explanation and diagram for Prototype model -(6)

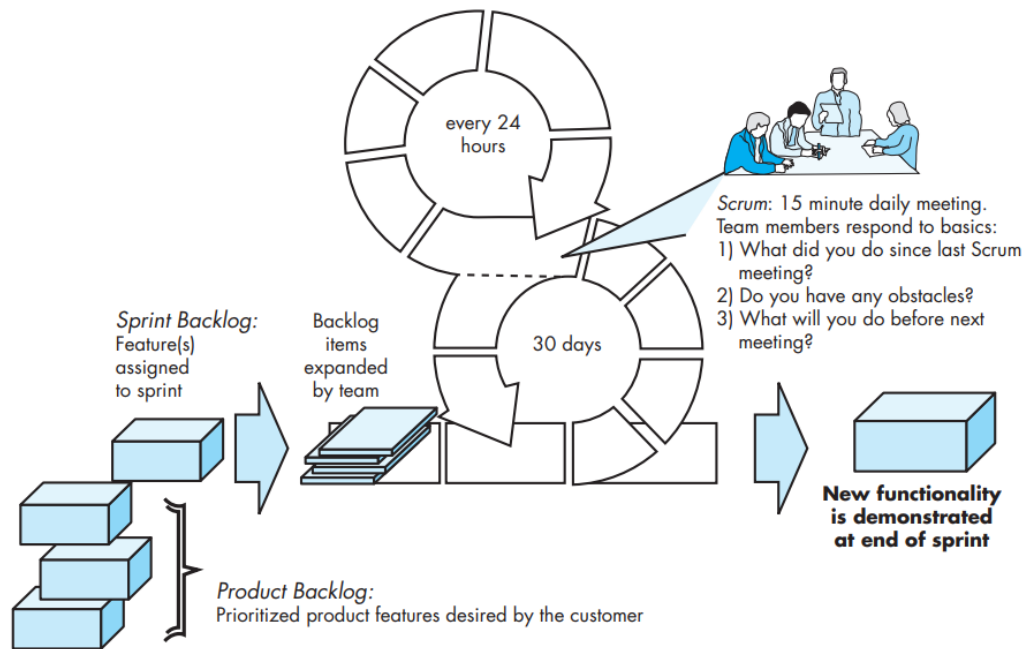


(or)

11. Scrum Process:

[CO1,K2]

b.i)



-(4)

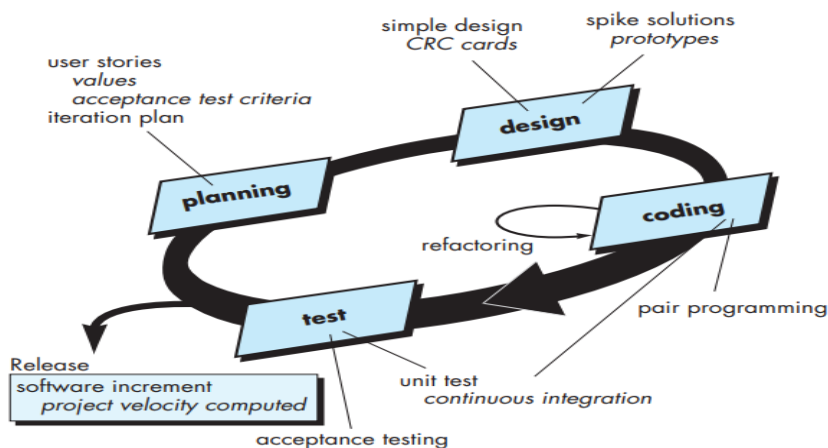
Scrum principles are consistent with the agile manifesto and are used to guide development activities within a process that incorporates the following framework activities: requirements, analysis, design, evolution, and delivery. Within each framework activity, work tasks occur within a process pattern (discussed in the following paragraph) called a sprint. The work conducted within a sprint (the number of sprints required for each framework activity will vary depending on product complexity and size) is adapted to the problem at hand and is defined and often modified in real time by the Scrum team.

Explain about Backlog , Sprints, Scrum meetings, Demos

-(4)

11. Refactoring is encouraged in Extreme Programming (XP) [CO1,K2]

b.
ii)



-(4)

Explaining about Design, Coding, Testing phase with Refactoring principles (4)

12) a) Tasks in Requirements gathering :

[CO2,K3]

- Inception (2)
- Elicitation(3)
- Elaboration(3)

- Negotiation(2)
- Specification(2)
- Validation(2)
- Requirements Management(2)

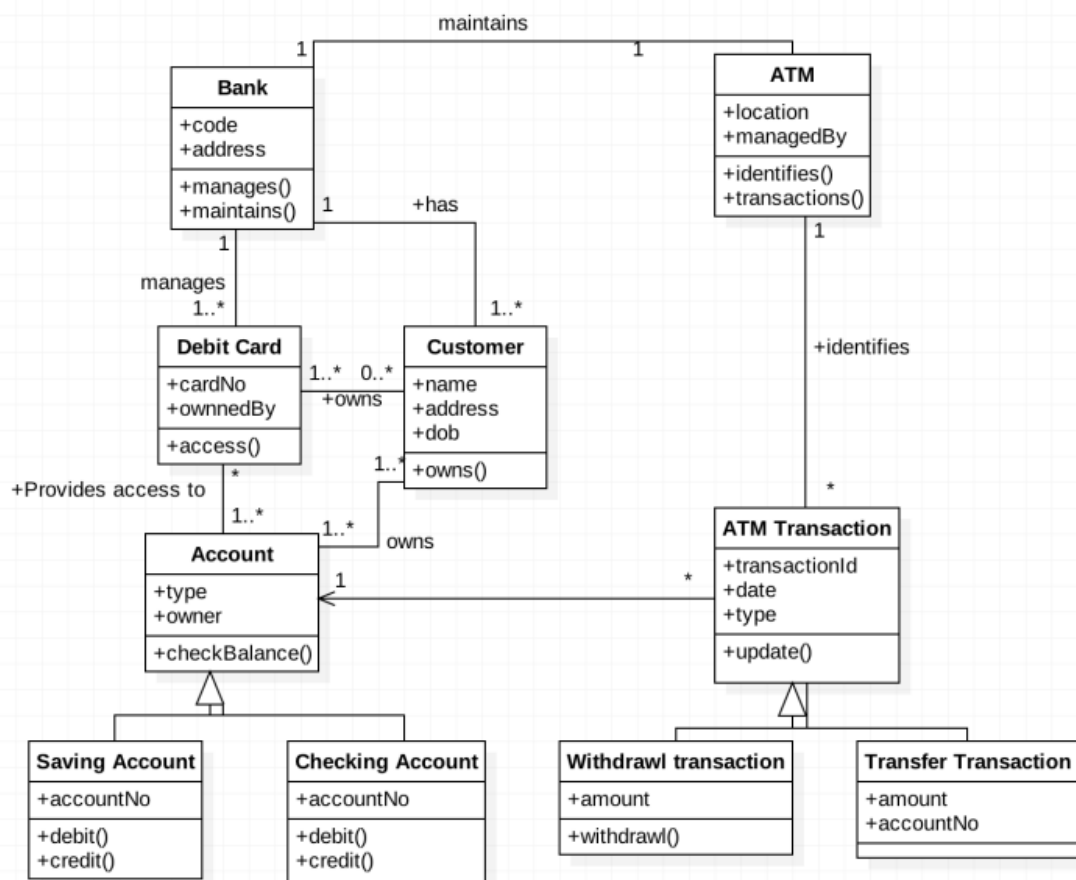
(or)

12 b) Procedures in Collecting the requirements :

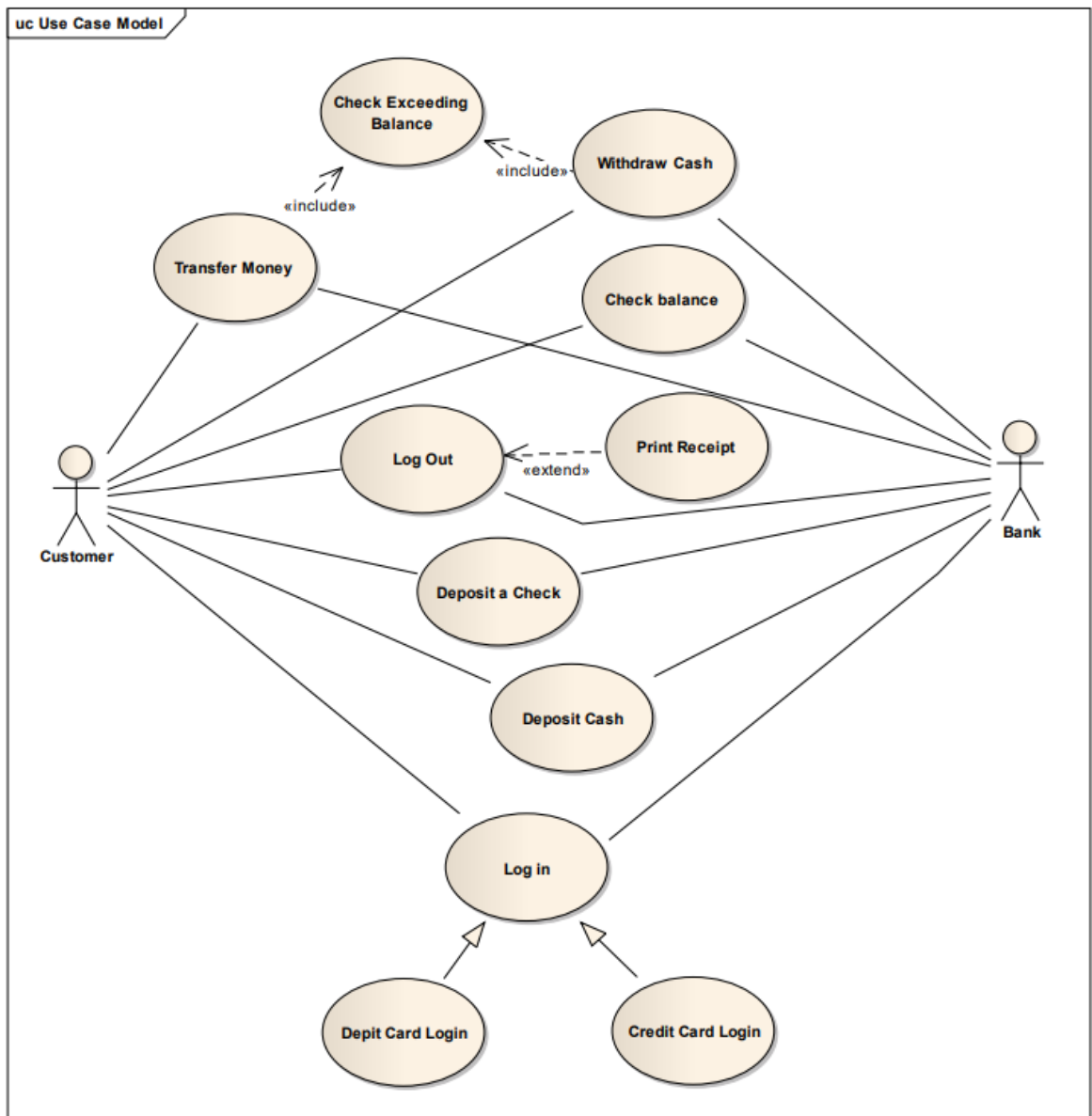
[CO2,K3]

- 1) Initiating the requirement Engineering Process
 - ➔ Identifying the Stakeholders
 - ➔ Recognizing multiple viewpoints
 - ➔ Working toward collaborations
 - ➔ Asking the first questions (4)
- 2) Eliciting requirements
 - ➔ Collaboration requirements gathering
 - ➔ Quality function deployment
 - ➔ User scenarios (4)
 - ➔ Elicitation work products (4)
- 3) Developing Use-Cases
- 4) Building the Analysis model (4)
- 5) Negotiating requirements
- 6) Validating requirements (4)

13 a) Class Diagram (6)

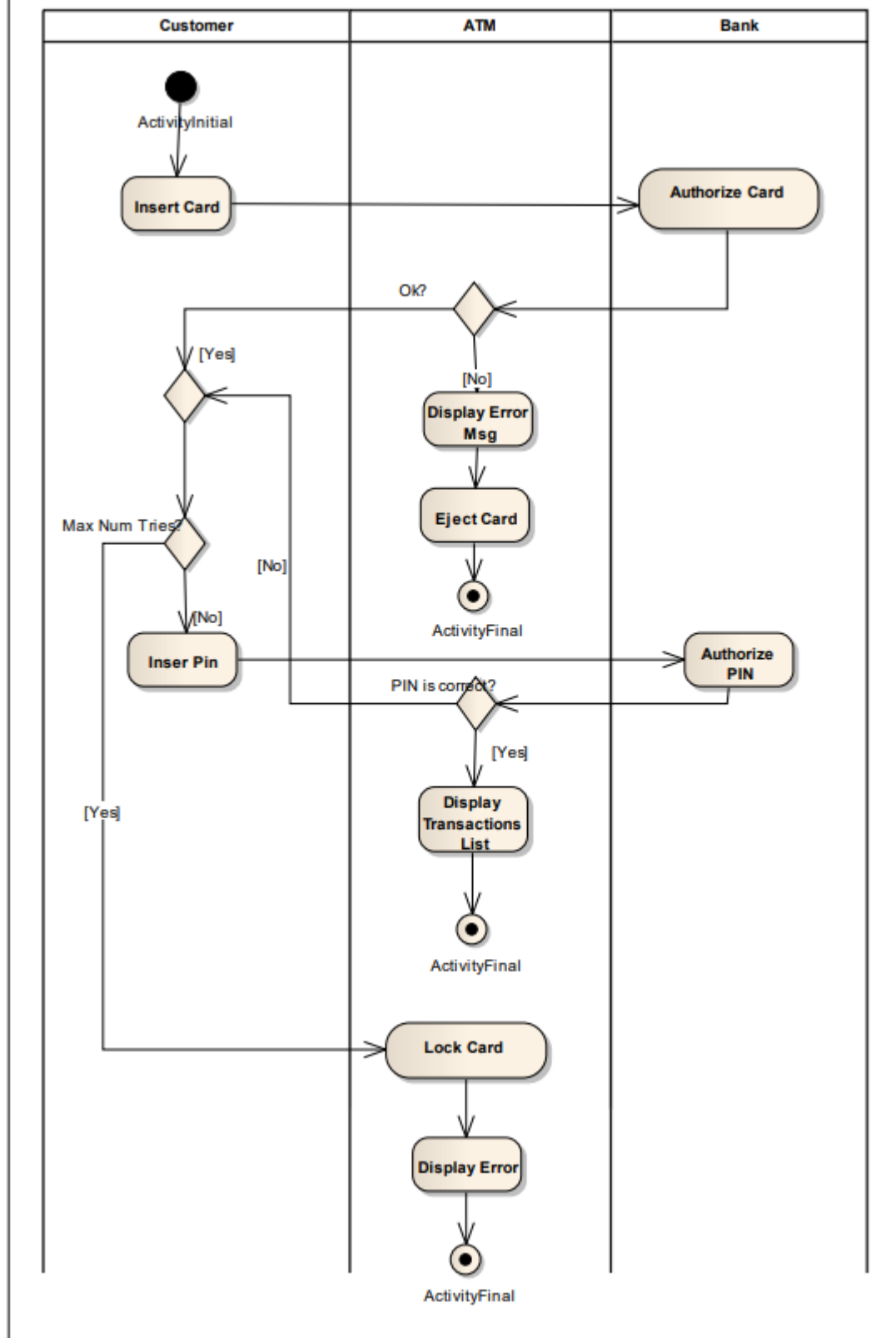


Use Case Diagram (6)



Activity Diagram (4)

act Login Activity Diagram



(or)

13) b) Elements of Class based modelling: [CO3,K3]

Class based modelling represents the object. The system manipulates the operations.

The elements of the class-based model consist of classes and object, attributes, operations, class – responsibility – collaborator (CRS) models.

Classes (4)

Classes are determined using underlining each noun or noun clause and enter it into the simple table.

Classes are found in following forms:

External entities: The system, people or the device generates the information that is used by the computer-based system.

Things: The reports, displays, letter, signal are the part of the information domain or the problem.

Occurrences or events: A property transfer or the completion of a series or robot movements occurs in the context of the system operation.

Roles: The people like manager, engineer, salesperson are interacting with the system.

Organizational units: The division, group, team are suitable for an application.

Structures: The sensors, computers are defined a class of objects or related classes of objects.

Attributes (4)

Attributes are the set of data objects that are defining a complete class within the context of the problem.

For example, 'employee' is a class and it consists of name, Id, department, designation and salary of the employee are the attributes.

Operations (4)

The operations define the behavior of an object.

The operations are characterized into following types:

- The operations manipulate the data like adding, modifying, deleting and displaying etc.
- The operations perform a computation.
- The operation monitors the objects for the occurrence of controlling an event

CRS Modeling (4)

- The CRS stands for Class-Responsibility-Collaborator.
- It provides a simple method for identifying and organizing the classes that are applicable to the system or product requirement.

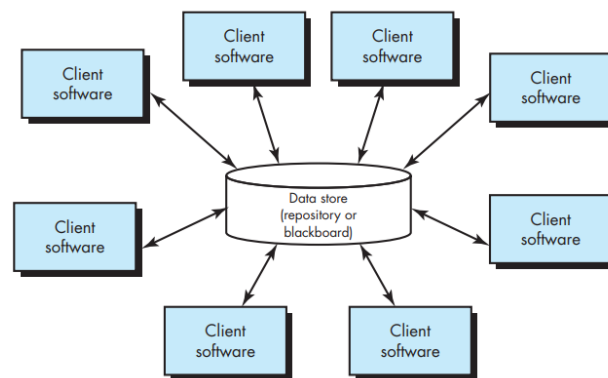
- Class is an object-oriented class name. It consists of information about sub classes and super class
- Responsibilities are the attributes and operations that are related to the class.
- Collaborations are identified and determined when a class can achieve each responsibility of it. If the class cannot identify itself, then it needs to interact with another class.

14 a) Different Architectural Styles with diagram:

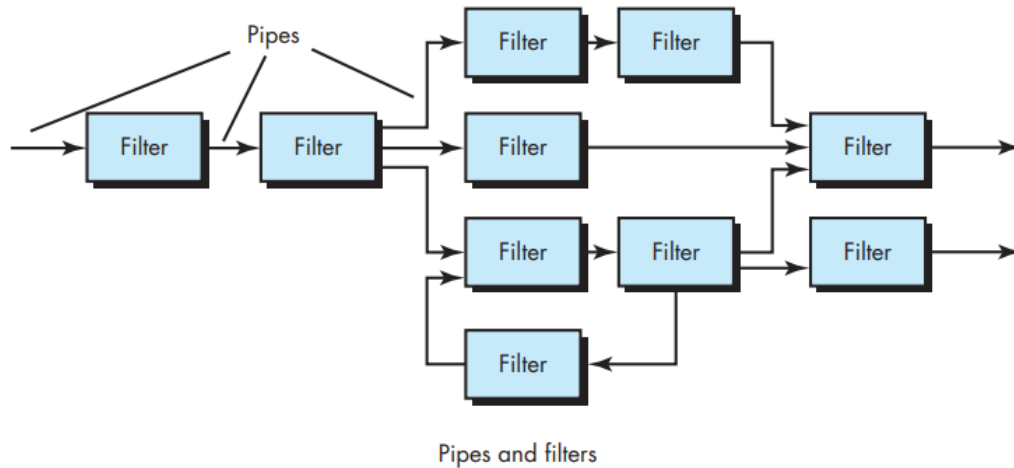
[CO4,K3]

The software architecture of a program or computing system is the structure of the system, which comprise software components, the externally visible properties of those components, and the relationships among them. (2)

1) Explanation for Data – Centered Architecture with diagram (4)



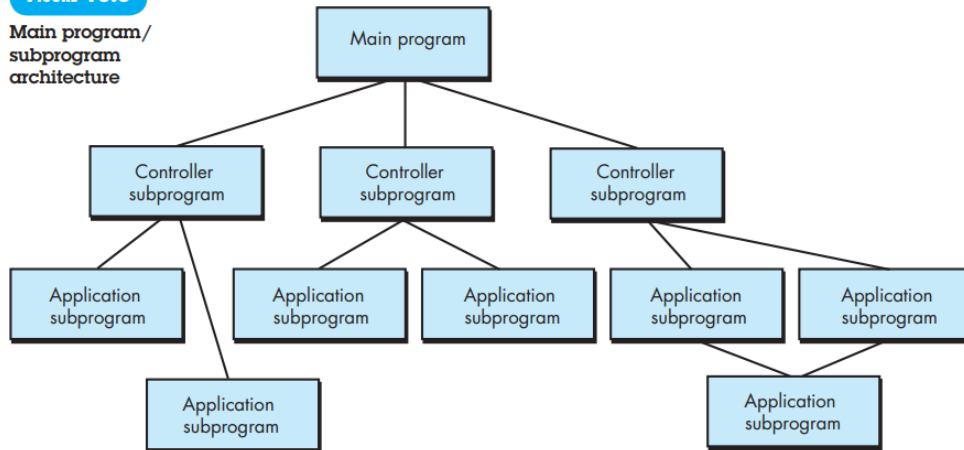
2) Explanation for Data – Flow Architecture with Diagram (4)



3) Explanation for Call and return architecture with Diagram (4)

FIGURE 13.3

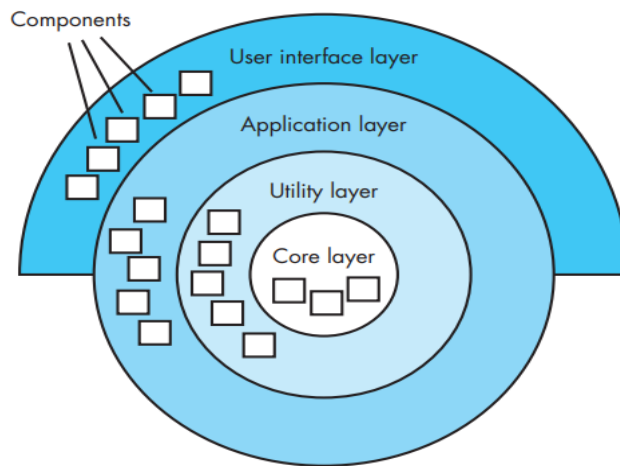
Main program/
subprogram
architecture



4) Explanation for object – oriented architecture and layered architecture with diagram (4)

FIGURE 13.4

Layered
architecture



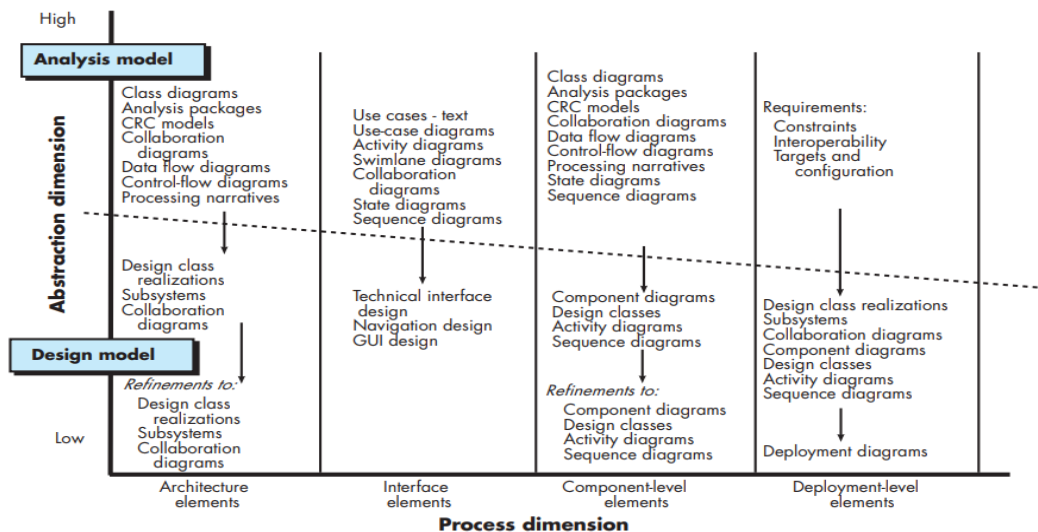
(or)

14) b) Dimension of Design model:

[CO4,K3]

FIGURE 12.4

Dimensions of the design model



(4)

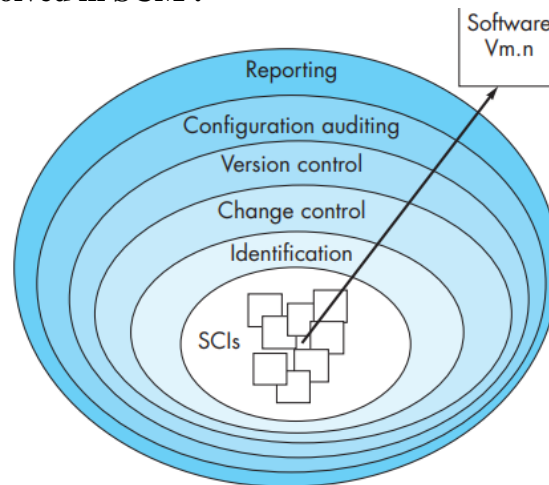
Elements of Design model:

- ➔ Data Design Elements
- ➔ Architectural Design Elements
- ➔ Interface Design Elements
- ➔ Component-level Design elements
- ➔ Deployment-level Design elements

(12)

15 a) i) Elements involved in SCM :

[CO5,K2]



(2)

Explaining the following topics,

- ➔ Version control
- ➔ Change control
- ➔ Impact management
- ➔ Configuration auditing
- ➔ Reporting

(6)

15)a) ii) Features of SCM process:

1. Versioning
2. Dependency tracking and change management (4)
3. Requirements tracing
4. Configuration management
5. Audit trails

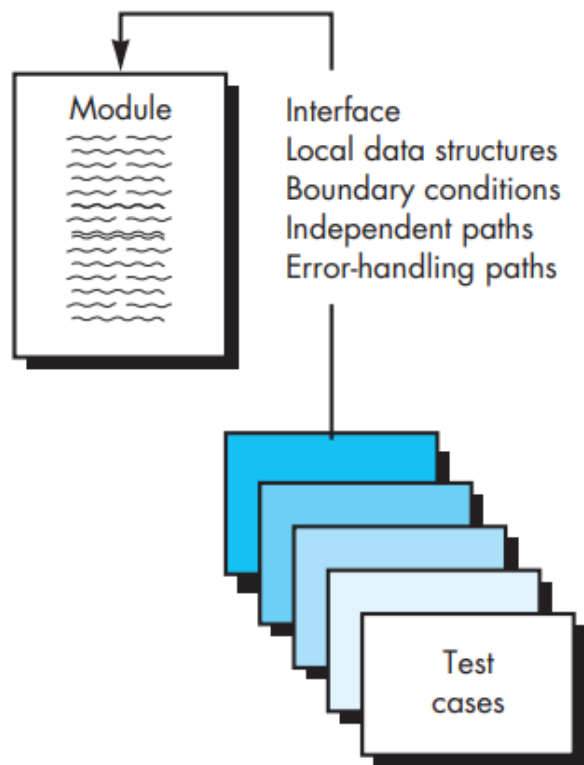
(4)

(or)

15)b) Test strategies handled for Conventional and object-oriented software: [CO5,K2]

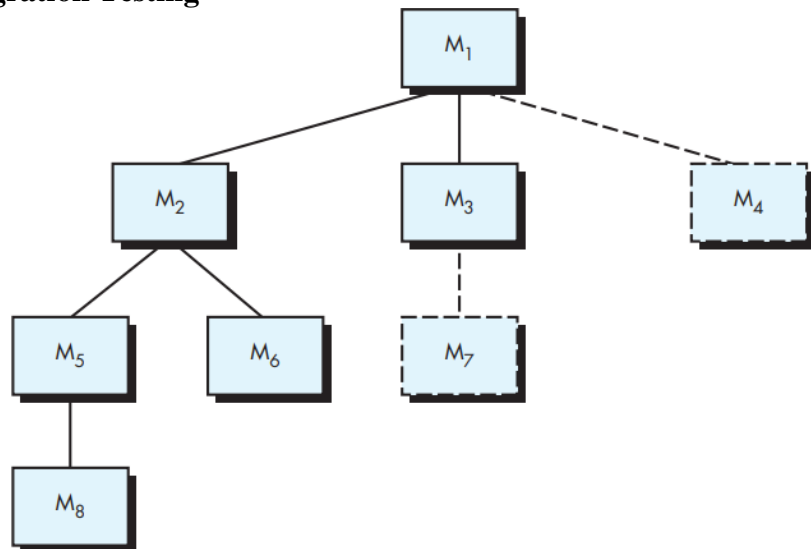
1) Conventional Software: (12)

Unit Testing

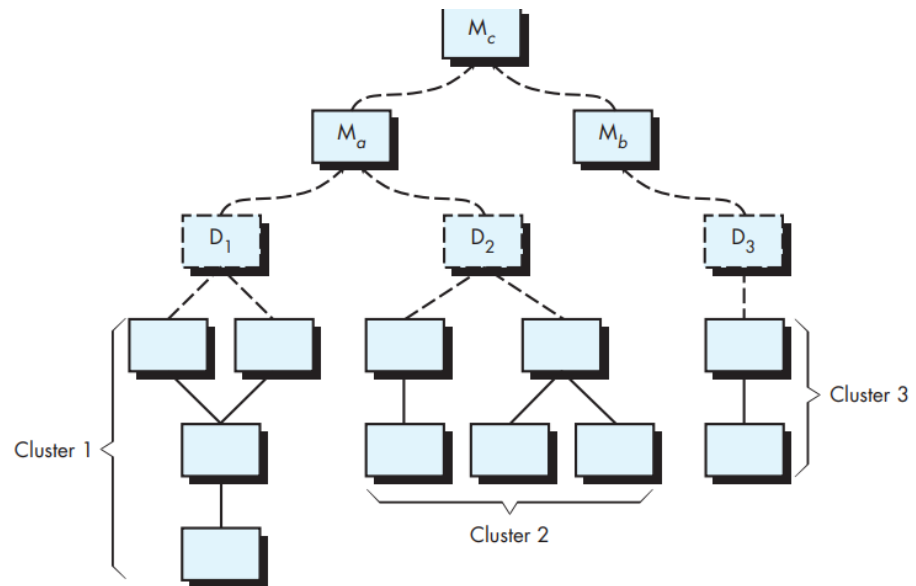


Integration Testing

Top-down
integration



Bottom-up
integration



Regression Testing
Smoke Testing

- 2) Object-oriented software: (4)
Unit Testing in the OO context
Integration Testing in the OO Context

Bloom's Taxonomy Level	Remembering (K1)	Understanding (K2)	Applying (K3)	Analysing (K4)	Evaluating (K5)	Creating (K6)
Percentage	2	44	54	--	--	--

