

Register No.

BE Degree Examination November 2023

Fifth Semester

Computer Science and Engineering

20CSC51 – AGILE METHODOLOGIES
(Regulations 2020)

Time: Three hours

Maximum: 100 marks

Answer all Questions

Part – A ($10 \times 2 = 20$ marks)

1. Write the IEEE definition of software engineering. [CO1.K1]

2. State the objectives of inception and elaboration in requirement engineering process. [CO1.K1]

3. List the four core values of agile software development process. [CO2.K1]

4. Identify the 3C's of user story and write the user story format. [CO2.K1]

5. Indicate the four variables of XP and sketch its relationship. [CO3.K2]

6. Mention the various principles involved in kanban process. [CO3.K1]

7. Interpret the need for system testing. [CO4.K1]

8. Distinguish between Alpha and Beta testing. [CO4.K2]

9. Define indirect measure. Why are such measures common in software metric works? [CO5.K2]

10. Mention the advantages and disadvantages of size measure. [CO5.K1]

Part – B ($5 \times 16 = 80$ marks)

11. a. i) Discuss the distinct tasks involved in requirement elicitation phase with a neat sketch. (8) [CO1,K2]

ii) Predict which process model is suitable for risk management. Discuss in detail with an example with its advantages and disadvantages. (8) [CO1,K2]

(OR)

- b. The case study on library management system gives us the complete (16) [CO1,K2] information and daily transaction done in a library. We need to record and retrieve the details of book in library which mainly focus on operation like adding a newbook, new member, searching books and returning books. For the above real time scenario, draw the following

 - 1) Use case diagram
 - 2) Class diagram
 - 3) Level 0 and Level 1 data flow diagram

12. a. Elaborate on the 12 principles of agile software development process and (16) [CO2,K2] explain each of them.

(OR)

- b. i) Discuss in detail about various scrum roles in developing a project. (6) [CO2,K2]
- ii) Explain about Burndown chart. Discuss with an example, how ideal line, actual line and capacity line insights into different types of risks. (10) [CO2,K2]
13. a. Describe the extreme programming process for the projects where the requirement changes constantly, with a neat process diagram. (16) [CO3,K2]
- (OR)
- b. i) Summarize in detail about Kanban's values and its need. (8) [CO3,K2]
- ii) Examine the seven wastes of software development process and state the ways to eliminate. (8) [CO3,K2]
14. a. i) Demonstrate the purpose of integration testing. Explain various strategies available in integration testing. (10) [CO4,K2]
- ii) Show the various steps involved in change control procedure of software configuration management with a neat sketch. (6) [CO4,K2]
- (OR)
- b. What is black box testing? Summarize on different types of black box testing with suitable examples. (16) [CO4,K2]

15. a. i) Consider 7 functions with their estimated lines of code as given below: (10) [CO5,K3]

Function	LoC
Func 1	2340
Func 2	5380
Func 3	6800
Func 4	3350
Func 5	4950
Func 6	2140
Func 7	8400

Average productivity based on historical data is 620 LoC per month and labour rate is Rs.8,000 per month. Find the following

- 1) Cost per LoC
 - 2) Total estimated effort
 - 3) Total estimated cost
- ii) 1) A web engineering team has built an e-commerce webApp that contains 145 individual pages. Of these pages 65 are dynamic; that is, they are internally generated based on end-user input. What is the customization index for this application? (6) [CO5,K3]
- 2) For a project ABC, defect reported by the customer is given as 12 and defect determined during modeling phase were 30. Find the defect removal efficiency for these two phases.

(OR)

- b. i) Consider the following FP components and their complexity

(6) [CO5,K3]

Function type	Estimated cost	Complexity
EI	50	4
EO	40	5
EQ	35	4
ILF	6	10
EIF	4	7

Assume the value adjustment factor is average and 14 algorithms have been computed. Compute the estimated function point values.

- ii) Assume, you are managing a project which is divided into 4 months of (10) execution. Calculate the following at the end of second month and identify whether the project is on schedule and on budget.
- 1) Schedule Variance (SV)
 - 2) Schedule Performance Index (SPI)
 - 3) Cost Variance (CV)
 - 4) Cost Performance Index (CPI)

Month / Values	1	2	3	4
Planned value	12,10,000	8,00,000	22,00,000	5,00,000
Earned value	10,00,000	9,50,000	-	-
Actual cost	13,50,000	7,00,000	-	-

Bloom's Taxonomy Level	Remembering (K1)	Understanding (K2)	Applying (K3)	Analysing (K4)	Evaluating (K5)	Creating (K6)
Percentage	8	74	18	-	-	-

KONGU ENGINEERING COLLEGE, PERUNDURAI 638 060

BE Degree Examination December 2023

Fifth Semester

Computer Science and Engineering

Course Code : : 20CSC51 Course Name : **Agile Methodologies**

(Regulations 2020)

End Semester Examination – Answer Key

Part A

1. Software Engineering is defined as the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (**Definition: 2 Marks**)
2. The objective of Inception is to establish a basic understanding of the problem and the nature of the solution and objective of Elaboration is to Create an analysis model that represents information, functional, and behavioral aspects of the requirements. (**Objective of Inception: 1 Mark, Objective of Elaboration: 1 Mark**)
3. The four Agile values of Software Development Process
(Each Value 0.5 marks: $4*0.5=2$ Marks)
 - ↳ Individuals and interactions over processes and tools.
 - ↳ Working software over comprehensive documentation.
 - ↳ Customer collaboration over contract negotiation.
 - ↳ Responding to change over following a plan.

4. 3 C's of User Story (**1 Mark**)

- ↳ Card- Annotated with Estimates ,values, notes, etc.,
- ↳ Conversation- Details of the story come out through conversation
- ↳ Confirmation-Acceptance tests are defined to confirm stories

User Story format (1 Mark)

- ↳ As [a user], I want [to perform this action] so that [I can accomplish this goal].

5. XP Variables (List: **1 Mark**, Relation: **1 Mark**)

1. Cost
 2. Scope
 3. Quality
 4. Time
- 111 Mark*

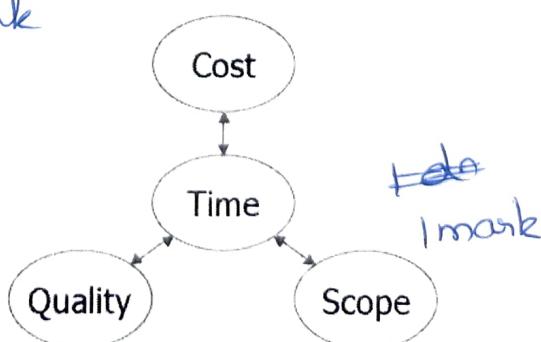


Figure 1 : The relationships between the variables

6. Kanban Principles:

(Each Principle 0.5 marks: $4*0.5=2$ Marks)

- ➔ Start with what you do now
- ➔ Agree to pursue Increment, Evolutionary change
- ➔ Respect the current Process, Roles and Responsibilities
- ➔ Encourage Acts of Leadership

- 7. The primary purpose of System testing is to evaluate the overall functionality and performance of a complete and fully integrated software solution. It tests if the system meets the specified requirements and if it is suitable for delivery to the end-users. (2 Marks)

8. Alpha and Beta Testing (2 Marks)

Alpha testing	Beta testing
The alpha test is conducted at the developer's site.	The beta test is conducted at end-user sites.
Alpha testing is done by testers and quality analysts inside the organization	Beta testing is done by real users who will be actually using the software

- 9. Indirect measures is defined as the measures that examines the quality of the software product itself (e.g. functionality, complexity, efficiency, reliability, maintainability). (1 Mark)

They are common in software metrics works as software metrics are used to measure performance, plan upcoming work tasks, track productivity and aims for better management. (1 Mark)

10. Advantages of Size measures (1 Mark)

- ➔ It is simple to measure
- ➔ It is well-designed.

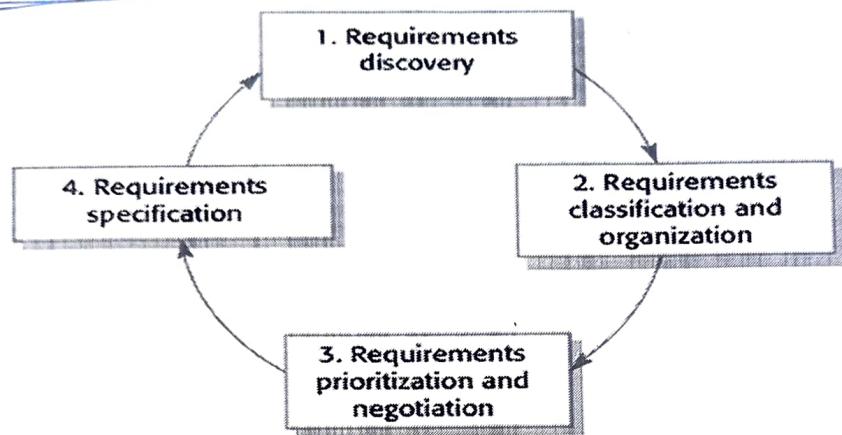
Disadvantages of Size measures (1 Mark)

- ➔ It is very difficult to estimate LOC in early stage of development.
- ➔ This measure is dependent on the programming language

Part B

11. a) i) Requirement Elicitation Phase (Diagram 3 Marks, List: 1 Mark, Explanation: 4 Marks)

Diagram 3 Marks



Requirement Elicitation and Analysis (List: 1 Mark)

- Requirement Elicitation Techniques
- Interviews
- Surveys
- Questionnaires
- Task analysis
- Domain Analysis
- Brainstorming
- Prototyping
- Observation

Explanation: 4 Marks

- ↳ Gather the requirement from the User
- ↳ Classify the requirements
- ↳ Customers, users and other stakeholders are asked to rank/prioritizes the requirement
- ↳ Negotiation can also be done —Agree on a deliverable system that is realistic for developers and customers.
- ↳ Prepare Software requirement specification. It is a kind of document which is created by a software analyst after the requirements collected from the various sources.
- ↳ After requirement specifications developed, the requirements discussed in this document are validated.

ii) Spiral Model is best suitable for Risk Management.

(Model Identification: 1 Mark, Diagram 2 Marks, Explanation: 3 Marks, Advantages -1 mark & Disadvantages: 1 Mark)

Model Identification: 1 Mark

- Best suited for Risk Management: Spiral model

Diagram 2 Marks

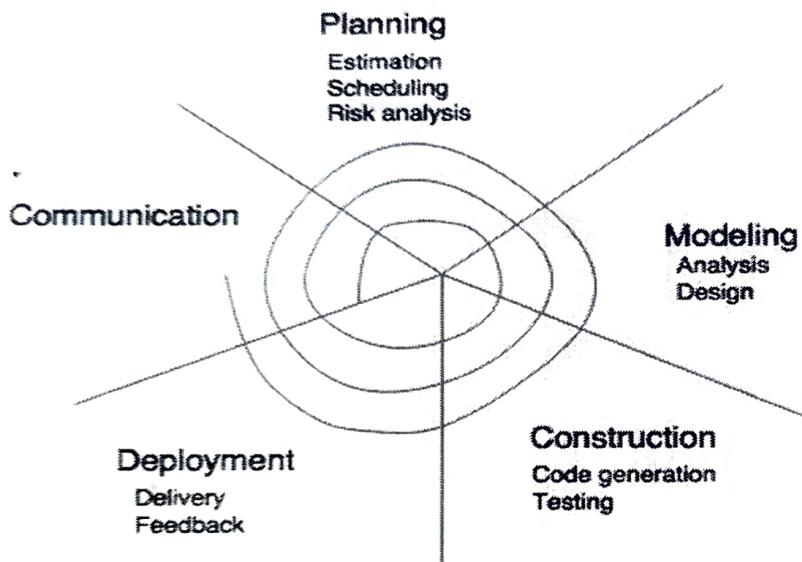


Fig. - The Spiral Model

Explanation: 3 Mark

- **Communication** – Gather Requirements from Customers
- **Planning** – Plan to be followed will be created. It describes the technical task to be conducted, risk ,required resources, work schedule etc.,
- **Modeling** – A model will be created to better understand the requirements
- **Construction** – Code generated and tested
- **Deployment** - Complete or partial complete version of software is given to the customer to evaluate and they give feed back based on the evaluation

Advantages: (1 Mark)

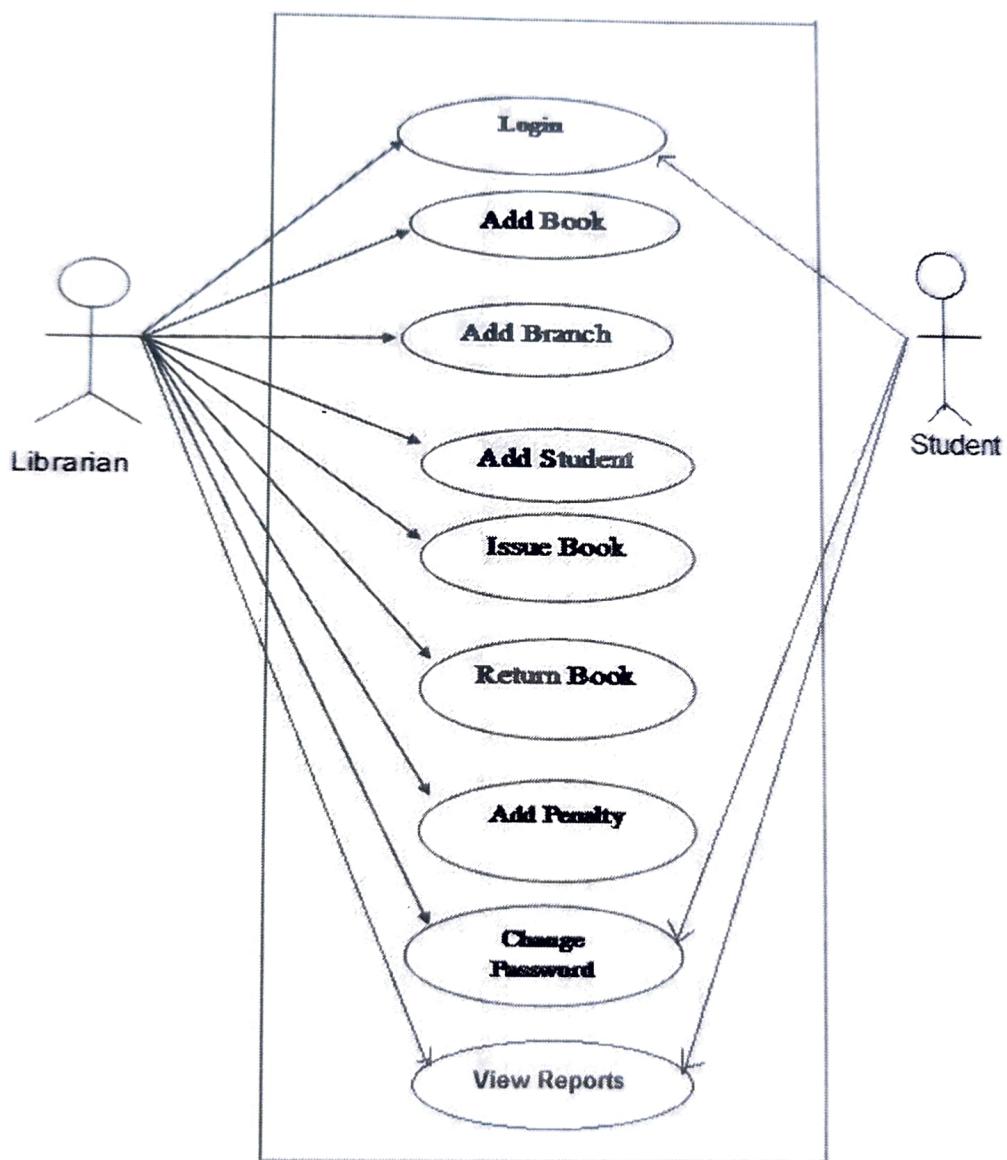
- It reduces high amount of risk.
- It is good for large and critical projects.
- It gives strong approval and documentation control.
- In spiral model, the software is produced early in the life cycle process.

Disadvantages: (1 Mark)

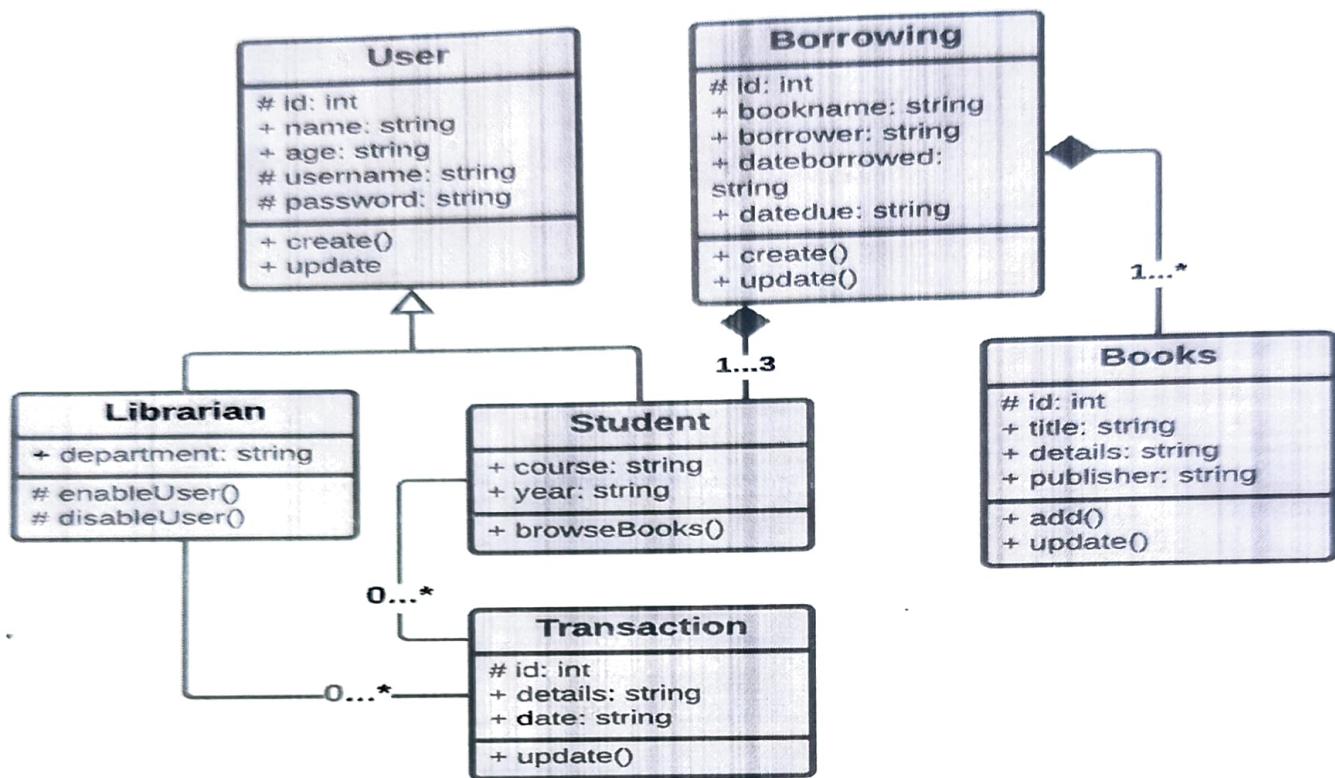
- It can be costly to develop a software model.
- It is not used for small projects.

b) Library Management System

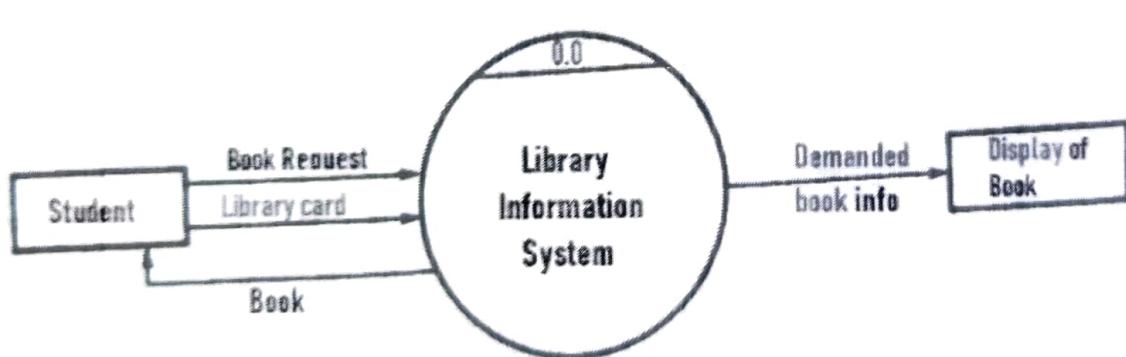
Use Case Diagram (4 Marks)



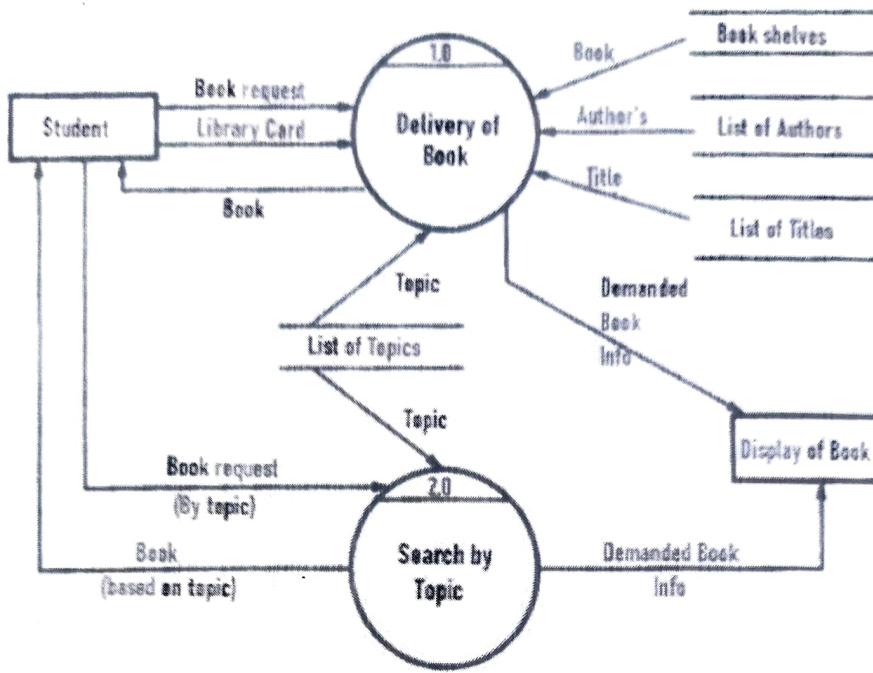
Class Diagram(6 Marks)



Level - 0 DFD (2 Marks)



Level – 1 DFD (4 Marks)



Level 1 DFD

12. a) 12 principles of Agile software development process (4 Marks for Explanation, 12 Marks for 12 Principles)

1. Customer Satisfaction through Continuous Delivery of Value:
2. Welcome Changing Requirements, Even Late in Development:
3. Deliver Working Software Frequently (in Weeks Rather Than Months)
4. The Most Efficient and Effective Method of Conveying Information to and Within a Development Team Is Face-to-Face Conversation:
5. Collaboration between Business Stakeholders and Developers:
6. Build Projects around Motivated Individuals; Give Them the Environment and Support They Need:
7. Working Software Is the Primary Measure of Progress:
8. Maintain a Sustainable Pace of Work for the Team:
9. Continuous Attention to Technical Excellence and Good Design:
10. Simplicity—the Art of Maximizing the Amount of Work Not Done—is Essential:

11. Self-Organizing Teams Encourage Great Architectures, Requirements, and Designs to Emerge:

12. Regular Reflection on Team Effectiveness:

b) i) Various scrum roles in developing a project

(3 Roles: Each role 2 Marks: $3 \times 2 = 6$ Marks)

There are three main roles on a Scrum project:

- ❖ Product Owner
- ❖ Development Team or Team Members
- ❖ Scrum Master

Product Owner: (2 Mark)

- Owns the backlog, strives to prioritize and takes all the key product decisions.
- This person turns all user requirements into actionable work for the development team

Development Team: (2 Mark)

- Cross functional team of developers, testers, designers, and other technical members who are needed for the development of the product.

Scrum Master: (2 Mark)

- A person who ensures the entire team has everything they need to ship a user story on time.
- This person communicates progress and ensures there aren't any roadblocks.
- Every day, the team holds a short face-to-face meeting called the Daily Scrum to update each other on the progress they've made, and to discuss the roadblocks ahead.

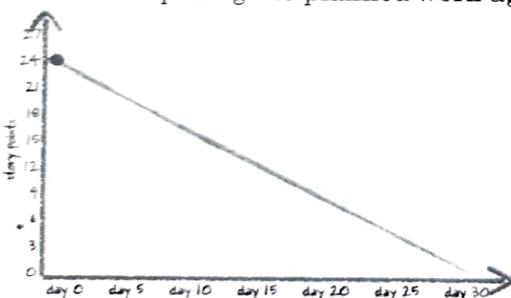
ii) Burn down chart in agile project management

Definition: (2 Marks)

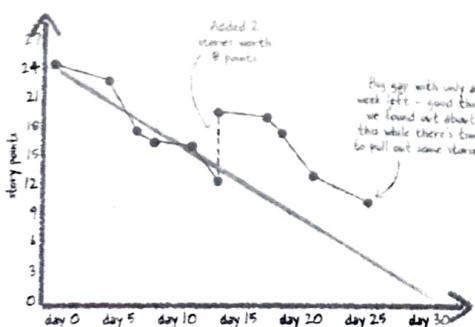
It is a visual representation of remaining amount of work for any specific sprint at Scrum team level

Burn down chart description: (2 Marks)

- Keeping the team running on schedule
- Comparing the planned work against the team progression



Example with Insights: Ideal Line, Actual Line and Capacity Line: (6 Marks)



13. a) Extreme Programming Process (Diagram: 4 Marks, Explanation: 12 Marks)

Diagram: 4 Marks

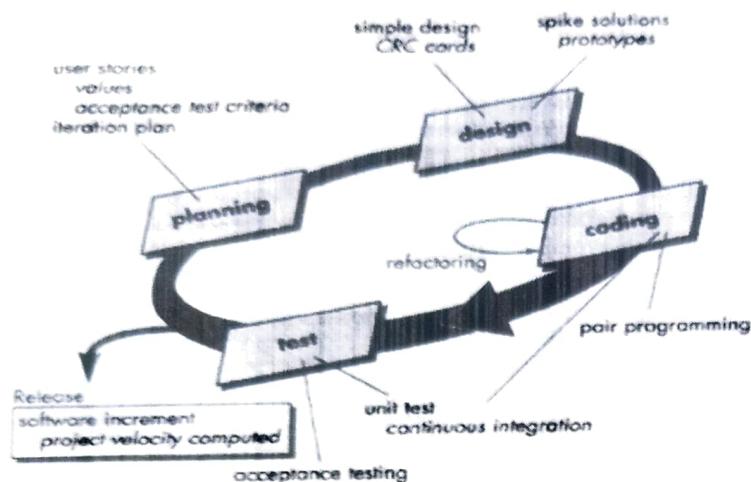


Fig. Extreme Programming Process

Explanation (12 Marks: 4 *3 =12 Marks)

Planning

The first phase of Extreme Programming life cycle is planning, where customers or users meet with the development team to create 'user stories' or requirements.

- The customer defines what he/she wants the system to do and the developers estimate how long it will take to implement the desired behavior.
- In the planning phase both parties negotiate which of the desired features can be done within the given bounds (time, resources).

Designing

- Use of Software Class Responsibilities and Collaboration (CRC) Cards. This cards allow all members of the project team to contribute ideas, and collate the best ideas into the design

Coding

- Coding constitutes the most important phase in the Extreme Programming life cycle.
- Developing the Pair programming
- Strict adherence to 40-hour workweeks with no overtime.

Testing

- Extreme program integrates testing with the development phase rather than at the end of the development phase.

- All codes have unit tests to eliminate bugs, and the code passes all such unit tests before release.
- Another key test is customer acceptance tests, based on the customer specifications.

b) i) Kanban Values (Any 8 Values with explanation: 8 Marks)

- ➔ Transparency
- ➔ Balance
- ➔ Collaboration
- ➔ Customer Focus
- ➔ Work flow
- ➔ Leadership
- ➔ Understanding
- ➔ Agreement
- ➔ Respect

Transparency : By an open exchange of information and a clear and unambiguous vocabulary you create transparency in all areas.

Balance : It is necessary to balance the different requirements, views and abilities of all participants among each other.

Collaboration : The Kanban method improves the way people work together.

Customer Focus: The customers and the value (asset) they receive are the natural centre of interest of all persons involved in the company.

Workflow: Work represents a continuous or occasional flow of values.

Leadership : Leadership is required at all levels to generate value and achieve an improved state.

Understanding: Understanding means first and foremost self-awareness, both from the individual employee as well as from the entire organization to move forward.

Agreement : In an agreement, all parties agree to pursue goals together.

Respect : Respect for people in the form of appreciation, understanding and consideration is the foundation on which the other values are based

Kanban Values are needed for successful completion of software with customer satisfaction.

ii) Seven Wastes of Software Development

(List: 1 Mark, Explanation for each waste 1 Mark: $7 \times 1 = 7$ Marks)

- ❖ Partially done work (List: 1 Mark)
- ❖ Extra processes
- ❖ Extra features
- ❖ Task switching
- ❖ Waiting
- ❖ Motion
- ❖ Defects

7 wastes \rightarrow 7 marks

❖ Partially done work (1 Mark)

- Work is delivered only if it is at “done”. If it’s not 100% completed, then it is not delivered.
- Any activity that doesn’t deliver value is waste.

❖ Extra processes (1 Mark)

- Processing beyond the standard.

❖ Extra features (1 Mark)

- Creating more information or tests than needed.
- When the team builds a feature that user haven’t asked, that is considered as waste.

❖ Task switching (1 Mark)

- Unnecessary movements between processes.

❖ Waiting (1 Mark)

- People or items that wait for a work cycle to be completed

❖ Motion (1 Mark)

- Unnecessary movement with a process
- When the team doesn’t sit together, people literally have to stand up and walk to their team members in order to have a discussion.

❖ Defects (1 Mark)

- Test-driven development prevents a lot of defects.
- Defects can be identified from Unit Test, Integration Test and System test.

14. a i) Integration Testing (10 Marks)

Purpose: 2 Marks,

Each Strategies with diagram (2 Marks) and Explanation (2 Marks): ($2 \times 4 = 8$ Marks)

Purpose: 2 Marks

- Integration testing is a technique to uncover errors associated with interfacing.
- The objective is to take unit-tested components and perform integration testing.

Each Strategies with diagram (2 Marks) and Explanation (2 Marks): ($2 \times 4 = 8$ Marks)

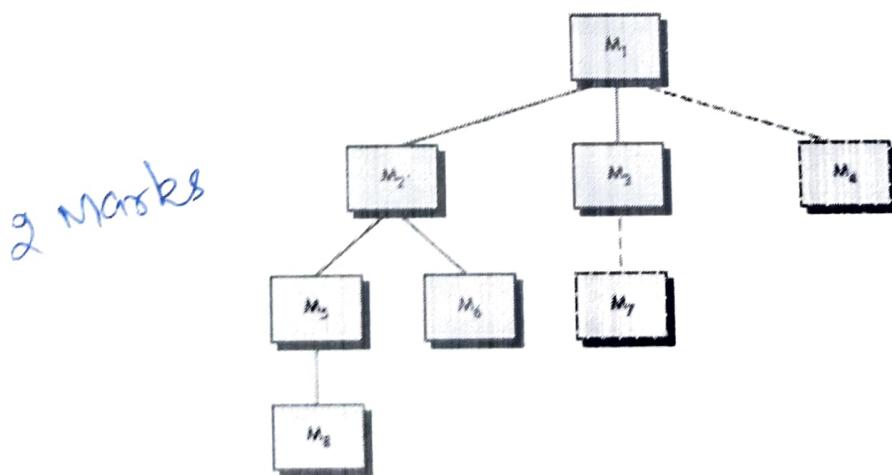
Top-down integration

Bottom-up integration

Top-down integration. (4 Marks)

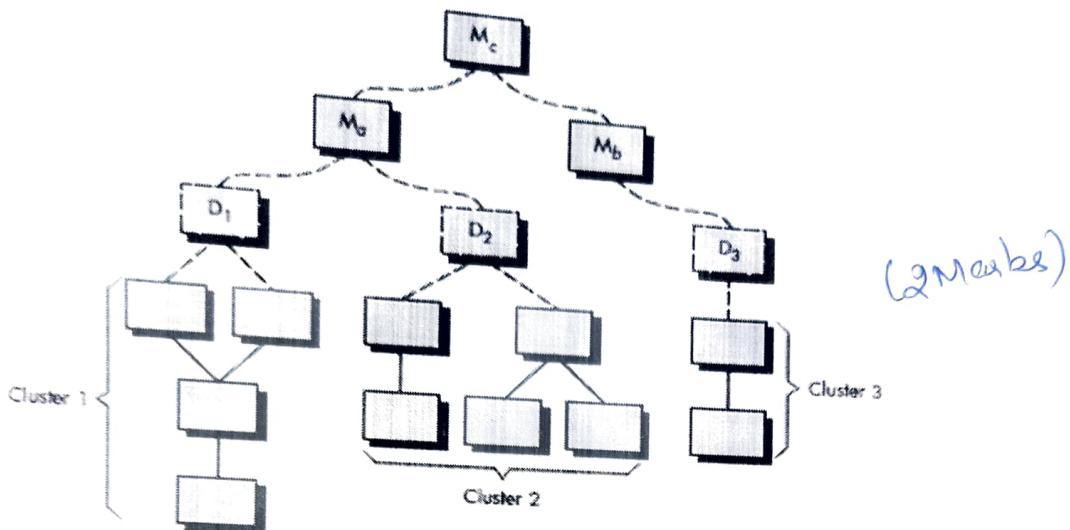
- Modules are integrated by moving downward through the control hierarchy, beginning with the main control module.

2 Marks • Modules subordinate to the main control module are incorporated into the structure in either a depth-first or breadth-first manner.



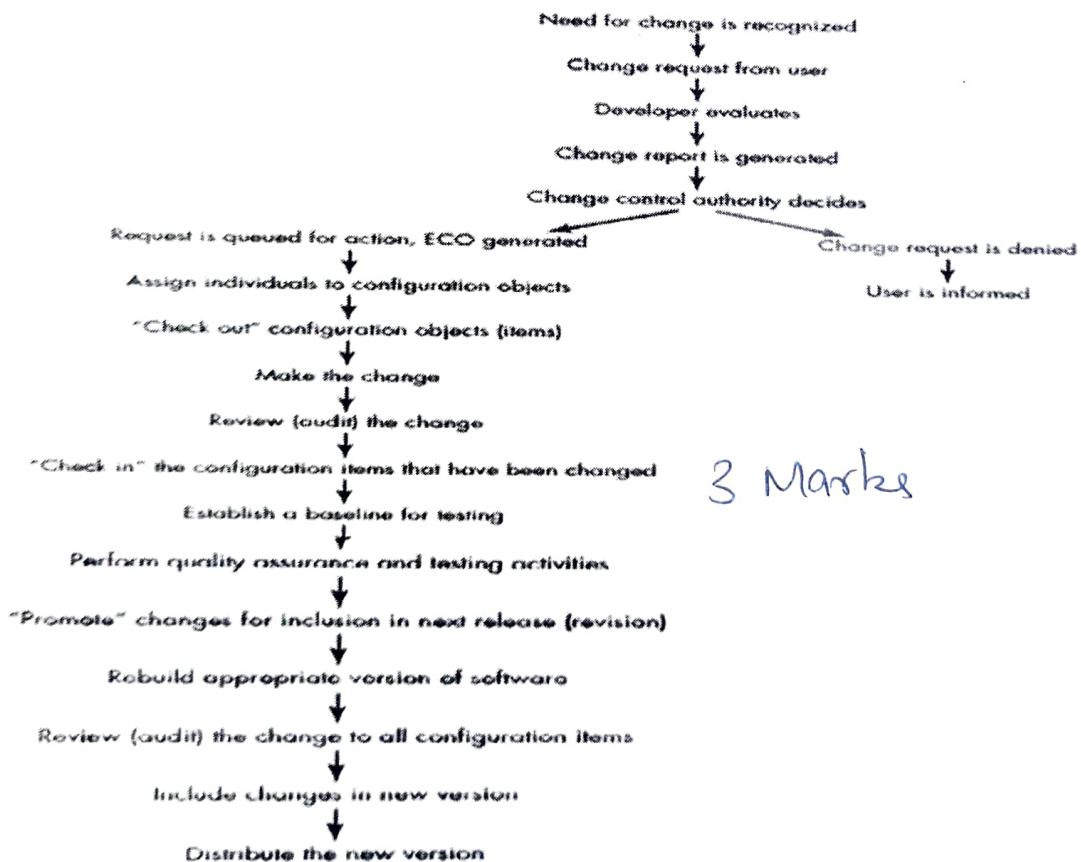
Bottom-up integration (4 Marks)

- Bottom-up integration testing starts with testing the lowest-level or most independent modules of a system and gradually integrates them with higher-level or more dependent modules.



ii) Change Control (Diagram:3 Marks, Explanation: 3 Marks) (6 M)

Diagram:3 Marks



Explanation: 3 Marks

- Change control is the process through which all requests to change the approved baseline of a project, programme or portfolio are captured, evaluated and then approved or rejected.

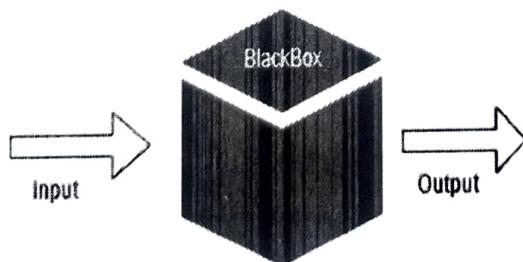
- A change request is submitted and evaluated. The results of the evaluation are presented as a change report, which is used by a change control authority (CCA).
- An Engineering change order (ECO) is generated for each approved change.
- A version control system updates the original file once the change has been made.
- Prior to an SCI becoming a baseline, only informal change control need be applied.
- The developer of the configuration object (SCI) in question may make whatever changes are justified by project and technical requirements.
- Once the object has undergone technical review and has been approved, a baseline can be created.
- Once an SCI becomes a baseline, project level change control is implemented.
- Now, to make a change, the developer must gain approval from the project manager (if the change is “local”) or from the CCA if the change affects other SCIs.
- When the software product is released to customers, formal change control is instituted.

b. Black box testing and its types (16 M)

(Blackbox Testing Explanation with Diagram: 4 Marks, 4 types: $4 \times 3 = 12$ Marks)

Blackbox Testing Explanation with Diagram: 4 Marks

- In Black-box testing, it is not necessary for the tester to have knowledge on internal structure of the code.
- Black-box testing , also called **behavioral testing or functional testing**, focuses on the **functional requirements of the software**.

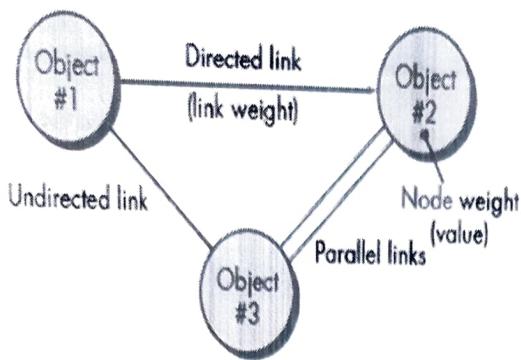


Types: (4 types: $4 \times 3 = 12$ Marks)

Graph-Based Testing Methods

- The symbolic representation of a graph.
- **Nodes** are represented as circles connected by links
- A **directed link** (represented by an arrow) indicates that a relationship moves in only one direction.
- A **bidirectional link** implies that the relationship applies in both directions.
- **Parallel links** are used when a number of different relationships are established between graph nodes.

(3 Marks)



Equivalence Partitioning

- Equivalence partitioning is a black-box testing method that divides the input domain of a program into classes of data from which test cases can be derived.

1. If an input condition specifies **a range, one valid and two invalid equivalence classes** are defined.
2. If an input condition requires **a specific value, one valid and two invalid equivalence classes** are defined.
3. If an input condition specifies **a member of a set, one valid and one invalid equivalence class** are defined.
4. If an input condition is **Boolean, one valid and one invalid class** are defined.

(3 Marks)

Boundary Value Analysis

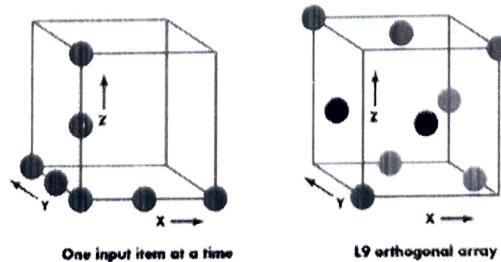
- Rather than selecting any element of an equivalence class, BVA leads to the selection of test cases at the **"edges"** of the class.
- If an input condition specifies a range bounded by values a and b, test cases should be designed with values a and b and just above and just below a and b.
- If an input condition specifies a number of values, test cases should be developed that exercise the **minimum and maximum numbers**. Values just above and below minimum and maximum are also tested.

(3 Marks)

Orthogonal Array Testing

- Orthogonal array testing can be applied to problems in which the input domain is relatively small but too large to accommodate exhaustive testing.
- The orthogonal array testing method is particularly useful in finding region faults—an error category associated with faulty logic within a software component.
- One input item at a time - results in relatively limited coverage of the input domain.
- When orthogonal array testing occurs, an L9 orthogonal array of test cases is created. In this case, test cases (represented by black dots in the figure) are “dispersed uniformly throughout the test domain.”

(3 Marks)



(10 marks)

Solution:

Total Estimation of LOC = 33,360

Average productivity = 620 LOC per month

Labour Rate = Rs. 8000 per month

$$1) \text{ Cost of LOC} = \frac{\text{Average Labour Cost}}{\text{Avg productivity}}$$
$$(3 \text{ Marks}) = \frac{8000}{620} = 12.9 = \text{Rs. } 13$$

$$2) \text{Total Estimated Effort} = \frac{\text{Total Estimation}}{\text{Avg productivity}}$$
$$(3 \text{ Marks}) = \frac{33360}{620}$$
$$= 53.8 \approx 54 \text{ Persons}$$

$$3) \text{Total Estimated cost} = \text{Cost of LOC} * \text{Total estimation}$$
$$(4 \text{ Marks}) = 13 * 33360$$
$$= 4,33,680$$

— X —

ii) Solution:- Formula : 1 Mark (3 mark)

Solution: 2 Marks

Customization index = $\frac{\text{Number of static web pages}}{\text{Number of Dyn. W.P + No. of Static W.P.}}$

Total: 145
 Dynamic : 65
 Hence static is $145 - 65 = 80$

$$C = \frac{80}{65+80} = \frac{80}{145} = 0.55$$

2) Solution:- Formula: 1 Mark (3 mark)

Solution: 2 Marks

Defect Removal Efficiency = $\frac{E}{E+D}$

Where

E - No. of Errors found before delivery

D - No. of defect found after delivery

$$E = 30$$

$$D = 12$$

$$= \frac{30}{30+12} = \frac{30}{42}$$

$$= 0.71$$

b) i) Solution

Formula: 2 marks (6 marks)

Calculation: 4 Marks

Function Point = CAF * VAF + 0.65 + 0.01 * $\sum F_i$

Where, $\frac{V}{\text{Total count}}$

CAF - Complexity Adjustment factor

VAF - Value Adjustment factor

$\sum F_i$ = Sum of all 14 questionnaires

Complexity Adjustment is average

∴ Scale is 3

$$\begin{aligned}\therefore \text{Total Count} &= 50 * 4 + 40 * 5 + 35 * 4 + \\ &\quad 6 * 10 + 4 * 7 \\ &= 628\end{aligned}$$

$$\sum F_i = 14 * 3 = 42$$

$$\begin{aligned}\therefore \text{Function Point } \} &= 628 * (0.65 + 0.01 * 42) \\ &= 628 * 1.07 \\ &= 671.9 \\ &= 672\end{aligned}$$

— X —

ii) Solution

(10 marks)

Task	Planned value	Earned value	Actual cost
1	12,10,000	10,00,000	13,50,000
2	8,00,000	9,50,000	7,00,000
3	22,00,000	20,10,000	-
4	5,00,000	-	-
		BCWP = 19,50,000	ACWP = 2050000

1) Schedule Variance (SV)

$$SV = BCWP - BCWS \quad \begin{matrix} \text{Earned Value} \\ \text{Planned value} \end{matrix} \quad (2.5 \text{ Marks})$$

$$= 19,50,000 - 20,10,000$$

$$= -60,000 \quad (\text{Behind the schedule as -ve})$$

2) SPI - Schedule Performance Index

$$SPI = \frac{BCWP}{BCWS} = \frac{19,50,000}{20,10,000} = 0.97 \quad (2.5 \text{ Marks})$$

3) Cost Variance (CV)

$$CV = BCWP - ACWP \quad (2.5 \text{ Marks})$$

$$= 1950000 - 2050000 = -100000 \quad (\text{Over Budget})$$

4) Cost Performance Index (CPI) = $\frac{BCWP}{ACWP}$ (2.5 Marks)

$$= \frac{19,50,000}{20,50,000} = 0.95$$

Conclusion: Project is Not in On Schedule & Not in On budget.