Register No.	2	0	C	S	R	2	3	6

BE Degree Examination December 2022

Fifth Semester

Computer Science and Engineering

20CSC51 – AGILE METHODOLOGIES

(Regulations 2020)

Time: Three hours Maximum: 100 marks

Answer all Questions.

Part – A (10 × 2 = 20 marks)

1. Identify the activities of process framework for software engineering process. [CO1,K1]

2. Define Requirement engineering and list its phases. [CO1,K1]

3. Compare product backlog and sprint backlog. [CO2,K2]

4. List the manifesto for Agile software development. [CO2,K1]

5. Why are informative workspaces important in XP? [CO3,K2]

6. State Lean Thinking and mention its values. [CO3,K1]

7. What guidelines lead to a successful software testing strategy? [CO4,K1]

8. Develop a checklist for use during configuration audits. [CO4,K2]

9. Outline the characteristics of an effective project manager. [CO5,K1]

10. Mention the different risk components and risk drivers. [CO5,K1]

Part – B (5 × 16 = 80 marks)

11. a. i) Evolutionary process model produce an increasingly more complete version of the software with each iteration. Justify the statement with proper explanation. (10) [CO1,K2]

ii) Discuss any four problems that occur when requirements has to be elicited form different customers. (6) [CO1,K2]

(OR)

b. Illustrate the class based modeling with suitable example. Draw the class diagram and develop the activity and sequence diagram for the same example. (16) [CO1,K2]

12. a. Summarize the principles of agile development process and discuss them. (16) [CO2,K2]

(OR)

b. User stories help to build features the users will-use. Justify the statement with suitable example and explain how it is effective. (16) [CO2,K2]

13. a. i) Why team resist changes and how do the primary practices of XP help teams do that? Discuss the answer. (8) [CO3,K2]

ii) With a simple example explain the use of value stream map for a real feature that goes through the traditional project management cycle. (8) [CO3,K2]

(OR)

b. i) How do the teams visualize the workflow using a Kanban board in software development process? Discuss. (8) [CO3,K2]

ii) Outline and define briefly the principles of XP. (8) [CO3,K2]

14. a. i) Use your own words and describe the difference between verification and validation testing. Do both make use of test-case design methods and testing strategies? Explain. (10) [CO4,K2]

ii) Draw a neat diagram that depicts the change control process and summarize it. (6) [CO4,K2]

(OR)

b. i) Show the procedure to compute the cyclomatic complexity in testing with suitable example. (8) [CO4,K2]

ii) Illustrate multiclass testing with a banking application. (8) [CO4,K2]

15. a. For the following projects calculate the Schedule Variance (SV), Cost Variance(CV), Schedule Performance Index (SPI) and Cost Performance Index (CPI) at the end of second month and identify whether the (16) [CO5,K3]

(i) project is on schedule (ii) project is on budget

Year	Planned value	Earned value	Actual cost
1	11,10,000	10,00,000	12,50,000
2	6,00,000	7,50,000	5,00,000
3	25,00,000	-	-
4	8,00,000	-	-

(OR)

b. (i) Compute the function point value for a project with the following information domain characteristics: (16) [CO5,K3]

(a) No.of user inputs=32

(b) No.of user output=60

(c) No.of user inquires=24

(d) No.of files=8

(e) No.of external interfaces=2

Assume all complexity adjustment values are average (4, 5, 4, 10, 7) and 14 algorithms to be computed with an average complexity – 3.

(ii) Compute cost per function point (FP).

(iii) Total estimated project cost

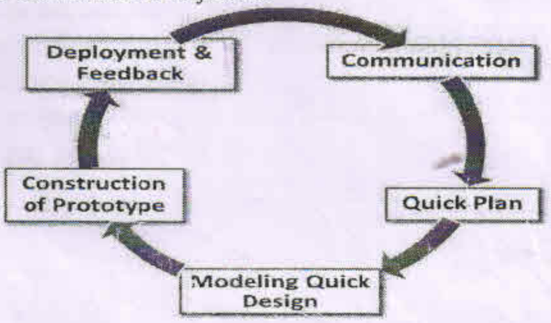
(iv) Total estimated project effort.

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

BE Degree Examination December 2022
Fifth Semester
Computer Science and Engineering
20CSC51 – Agile Methodologies
(Regulations 2020)

Part – A (10*2=20marks)

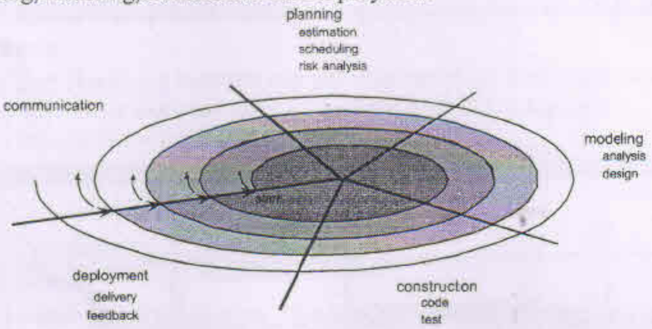
<table>
<tr><td>1</td><td></td><td>Identify the activities of process framework(Any 4)
• Communication
• Planning
• Modelling
• Construction
• Deployment</td><td>2 mark</td></tr>
<tr><td>2</td><td></td><td>Define Requirement Engineering and list its phases.
<b>Requirement Engineering</b> – Understanding of Requirement
<b>Phases/Task:</b> (Any 4)
Inception
Elicitation
Elaboration
Negotiation
Specification
Validation
Requirement Management</td><td>1 mark
1 mark</td></tr>
<tr><td>3</td><td></td><td>Compare Product Backlog and Sprint Backlog.(Any 2)
<table>
<tr><th>S.No</th><th>Product Backlog</th><th>Sprint Backlog</th></tr>
<tr><td>1</td><td>Created by Product Owner</td><td>Created during Sprint Planning</td></tr>
<tr><td>2</td><td>Specific to the product</td><td>Specific to the current sprint</td></tr>
<tr><td>3</td><td>It includes list of all user stories required to complete the product</td><td>It includes only the user stories belong to the current sprint</td></tr>
<tr><td>4.</td><td>long term goal</td><td>short-term goal</td></tr>
</table></td><td>2 mark</td></tr>
<tr><td>4</td><td></td><td>List the manifesto for Agile software development.
• Individuals and Interaction
• Working Product
• Customer Collaboration
• Responding to change</td><td>2 mark</td></tr>
<tr><td>5</td><td></td><td>Why are informative workspaces important in XP?
• Better Decision
• It is just like task board in scrum
• Information shared here for whole team</td><td>2 mark</td></tr>
<tr><td>6</td><td></td><td>State Lean Thinking and mention its values
Lean Thinking : Process of Eliminating waste
Lean Values :(Any 4)
• Eliminate waste
• Amplify Learning
• Decide as late as possible
• Deliver as fast as possible
• Empower the team
• Build integrity in
• See the whole</td><td>1 mark
1 mark</td></tr>
<tr><td>7</td><td></td><td>What guidelines lead to a successful software testing strategy?
• Unit Testing
• Integration Testing</td><td>2 mark</td></tr>
</table>

		• Validation and System testing	
8		Develop a checklist for use during configuration Audit.(Any 4) • Has Change Specified in ECO been made? • Have Technical Review completed? • Has all Software Engineering standards Applied? • Changes been highlighted in SCI? • Has Change Date, Author been specified? • Have SCM Procedures Followed? • All Related SCI's been properly updated?	2 mark
9		Outline the characteristics of an effective project manager.(Any 4) • Effective communication • Negotiation • Time Management • Leadership • Open-Mindness • Technical Expertise • Critical Thinking	2 mark
10		Mention the different Risk Components and Risk drivers. **Risk Components:** • Performance Risk • Cost Risk • Support Risk • Schedule Risk **Risk drivers:** Reason that affect the Performance/Cost/Schedule etc..of a project[ie., Risk Components]	1 mark 1 mark
		PART B (5*16=80 marks)	
11.a)	i)	Evolutionary Process model produce an increasingly more complete version of the software with each iteration. Justify the statement with proper explanation **Evolutionary Process models = iterative +incremental** This model is designed to accommodate a product that evolves over time They allow developing more complete versions of the software with each iteration. **Advantages :** • Customer requirements are clearly specified, • Risk analysis is better • It supports changing environment, • Initial Operating time is less, • Best suited for large project and critical project **Disadvantages:** • Not feasible for smaller projects, • Cost, • Highly skilled persons are required **Common evolutionary process models are:** 1)The prototyping model 2) The spiral model **Prototyping Model:** Mockup or model of a software Product Used when Customer defines a set of objectives  Deployment & Feedback Communication Quick Plan Modeling Quick Design Construction of Prototype	1 mark 1 mark 1 mark 1 mark 3 mark

3 mark

Spiral Model:

- It combines the best feature of classical life cycle and the iterative nature of prototype
- It includes new element : Risk element
- Starts in middle and continually visits the basic tasks of communication, planning,modeling,construction and deployment



11.a) ii) Discuss any four problem that occur when requirements has to be elicited from different customers

Following are the problems that are encountered during elicitation

3 mark

- ❖ Problem of scope
- ❖ Problem of understanding
- ❖ Problems of volatility

To overcome the above problem , we must approach the ***requirement gathering in an organized way***

Requirement Elicitation

Ask the customer, the users and others

3 mark

- ❖ what the objectives for the system or product are,
- ❖ what is to be accomplished,
- ❖ how the system or product fits into the needs of the business.
- ❖ How the system or product to be used on day to day basis

11.b) Illustrate the class based modelling with suitable example. Draw the class Diagram and develop the activity and sequence diagram for the same example

Elements of class based model are

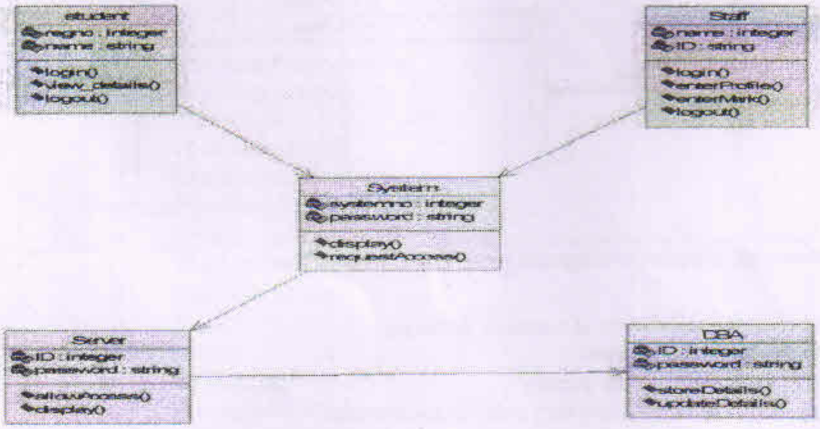
1 mark

- Classes and Objects
- Attributes
- Operations
- Class-Responsibility-Collaborator(CRC) models
- Collaboration diagrams
- Packages

Example: Student Management System(Student can give his/her own example)

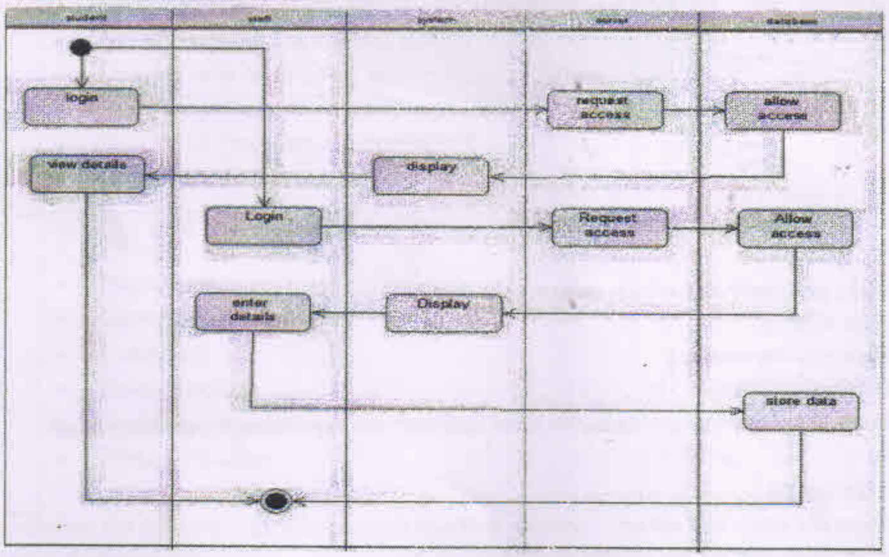
Class Diagram

5 mark



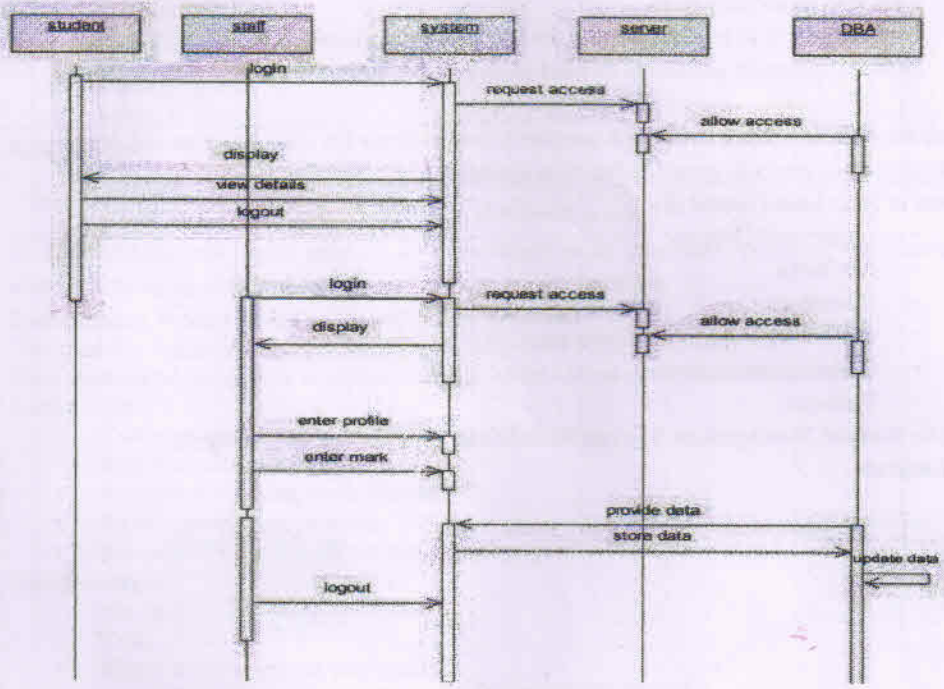
Activity Diagram

5 mark



Sequence Diagram

5 mark



12.a Summarize the principles of agile development process and discuss them.

12*1 = 12 mark

1. Early and continuous delivery of valuable Software
2. Welcome changing requirement
3. Delivery working software frequently
4. Face – to – Face conversation within a development team
5. Business people and developers must work together daily
6. Build projects around motivated individuals
7. Working software is the primary measure of progress
8. Support sustainable development
9. Continuous attention
10. Simplicity
11. Self-Organizing Team
12. At regular intervals, the team reflects on how to become more effective and adjusts its behaviour accordingly

		Explanation for any 4 principles	
b		User Stories help to build features the users will use. Justify the statement with suitable example and explain how it is effective **User Stories:** • User Stories Help Build Features Your Users will use • A User story is a deceptively simple tool. • It's a quick and simple description of a specific way that a user will use the software	2 mark
		User story Format: • Most User Stories are between one and four sentences long- most teams have a rule of thumb that a user story has to fit on the front of a 3X5 index card • Mad Libs- Style for writing user stories • As a < type of user >, I want < some goal > so that < some reason >.	3 mark
		User Story Example and Explanation: (Student's may write his/her own example) Nominate a video for an achievement As a returning user with a large friends list, I want to nominate one friends video For an achievement So that all of our mutual friends can vote To give him a star	5 mark
		User Stories Advantages: (Any 3) • User Stories also give teams an easy way to manage their backlog • It's easy for the Product Owner to review the stories with the users and Stakeholders – to figure out which stories are valuable • Each Story is small, which make it easier to add new ones or change their order at any time • To Start the sprint – Product owner and team can pull of the stories out of the backlog • Most team will then break down the stories into task and start to estimate how long those tasks will take • The task for each story would go into the "ToDo" Column of the taskboard • Once the team member taken the task it moved to "In-Progress" Column of the taskboard	2 mark
		Condition of Satisfaction : (Any 2) • Refer to as "acceptance criteria" • It is defined for each user story • It can fit on the back side of the same 3X5 index card • It is valuable to developer • User story is "Done" only when all the conditions are satisfied	2 mark
		Condition of satisfaction Example: (Own Example) • A user can nominate a video for achievement • A user's friend is notified when his video gets notified • A user can see all of the videos his friend have nominated	2 mark
13.a	(i)	Why team resist changes and how do the primary practices of XP help teams do that? Discuss the answer XP Primary Practices Categories: • Programming Practices • Test First Programming • Pair Programming	2 mark
		• Integration Practices • 10 minutes build • Continuous Integration	2 mark
		• Planning Practices(Any 2) • Iterative • Weekly Cycle • Quarterly Cycle • Slack	2 mark
		• Team Practices • Sit Together • Informative Workspace	2 mark

13.a (ii) With a simple example explain the use of value stream Map for a real feature that goes through the traditional project management cycle

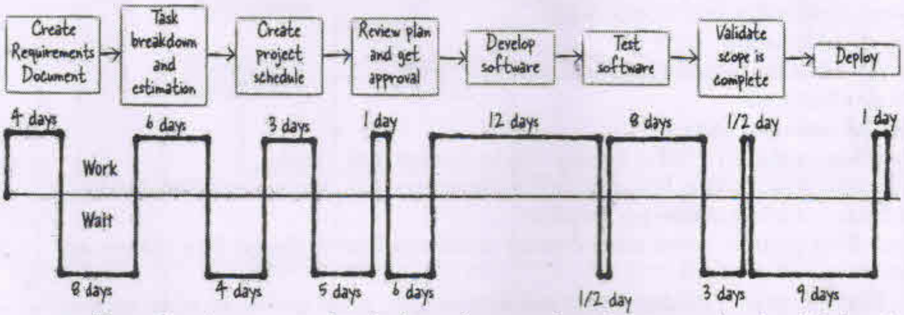
Value Stream Map:

- In Lean Software Development, Mary and Tom Poppendieck recommend a simple **pencil-and-paper exercise to help you find waste.**
- It's called a **value stream map**, and you can build one for any process.
- Like many techniques used in conjunction with Lean, it originated in manufacturing, but it makes **sense for software teams as well**

2 mark

Value Stream Map Example and Explanation:(Own Example)

6 mark



- The value stream map clearly shows how much wait time was involved during the process.
- It took a total of 71 days from the time the team started work on it to the time that it was deployed.
- Of those 71 days, 35.5 were spent waiting rather than working.

13.b (i) How do the teams visualize the workflow using a Kanban board in software development process?Discuss

Kanban

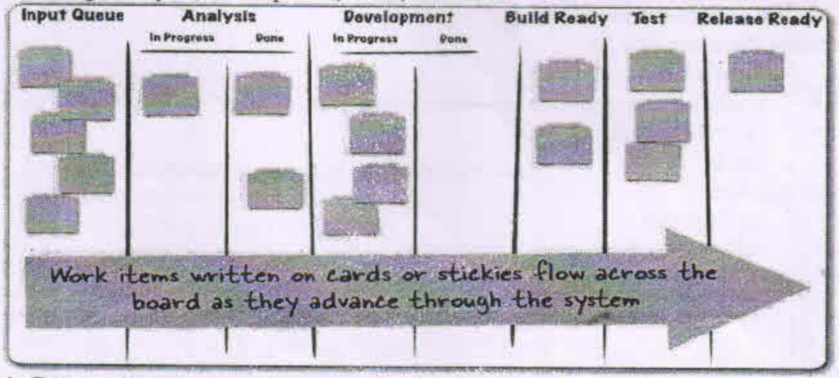
2 mark

- It is a **method for process improvement** used by agile teams.
- It is designed to help you to visualize work Flow, work in progress and work efficiently
- It is a workflow management method

Visualize the work Flow

4 mark

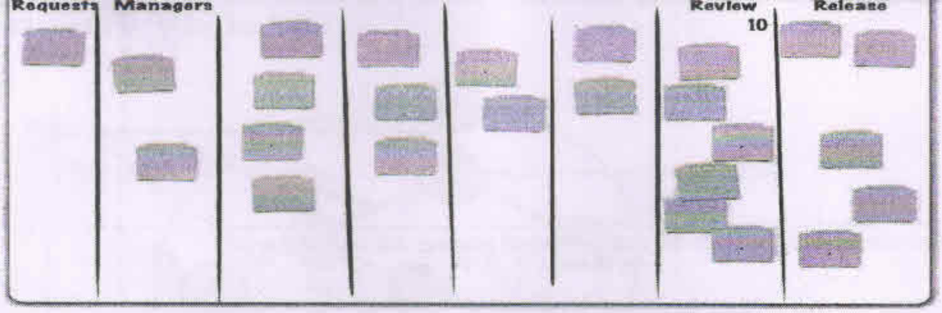
- When a team wants to adopt Kanban, the first thing that they do is visualize the workflow by creating a kanban board.
- For example, one of the first kanban boards in David Anderson's book, Kanban, has these columns: Input Queue, Analysis (In Prog),Analysis (Done), Dev Ready, Development (In Prog), Development (Done), Build Ready, Test, and Release Ready.
- This board would be used by a team that follows a process where each feature goes through analysis, development, build, and test.


Input Queue
Analysis
In Progress
Done
Development
In Progress
Done
Build Ready
Test
Release Ready
Work items written on cards or stickies flow across the board as they advance through the system

Limit in Progress:

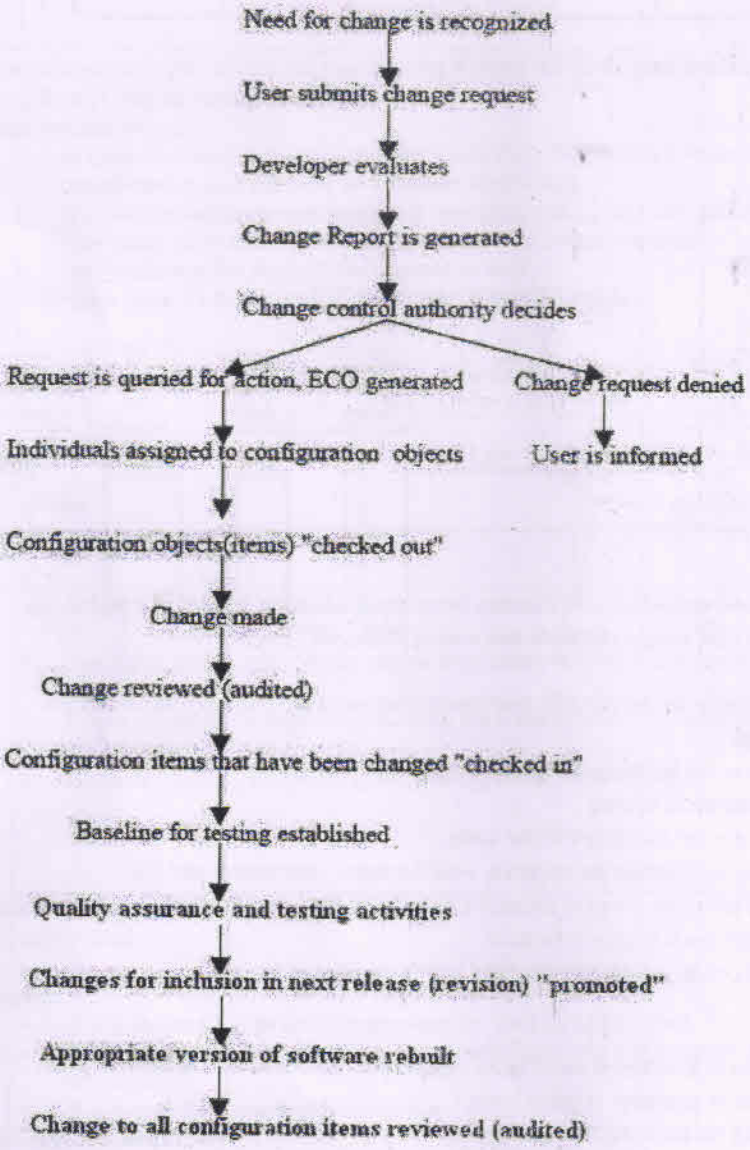
- Limiting work in progress (WIP) means setting a limit on the number of work items that can be in a particular stage in the project's workflow.

2 mark

			
13.b	(ii)	Outline and define briefly the principles of XP **(Any 8)** - Humanity - Economics - Mutual benefit - Self-Similarity - Improvement - Diversity - Reflection - Flow - Opportunity - Redundancy - Failure - Quality - Accepted responsibility - Baby steps	8 mark
14.a	(i)	**Use Your Own words** and describe the difference between verification and validation testing. Do both make use of test case design methods and testing strategies? Explain. **Verification:** - Verification refers to the set of tasks that ensure that software correctly implements a specific function. - Verification: "Are we building the product right?" - Verification is the static testing. - It does not include the execution of the code. - Methods used in verification are reviews, walkthroughs, inspections and desk-checking. - Quality assurance team does verification. - It consists of checking of documents/files and is performed by human.	4 mark
		Validation: - Validation refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements. - Validation: "Are we building the right product?" - It includes testing and validating the actual product. - Validation is the dynamic testing - It includes the execution of the code - Methods include Black box, White box and non-functional testing - Software Testing team perform validation - It consists of execution of program and is performed by computer.	4 mark
		Verification involves verification of the document which contains test case design and testing strategies Validation make use of test case designs and various testing strategies to develop the fault free software	2 mark

14.a (ii) Draw a neat diagram that depicts the change control process and summarize it

4 mark


Need for change is recognized
User submits change request
Developer evaluates
Change Report is generated
Change control authority decides
Request is queried for action, ECO generated
Change request denied
Individuals assigned to configuration objects
User is informed
Configuration objects(items) "checked out"
Change made
Change reviewed (audited)
Configuration items that have been changed "checked in"
Baseline for testing established
Quality assurance and testing activities
Changes for inclusion in next release (revision) "promoted"
Appropriate version of software rebuilt
Change to all configuration items reviewed (audited)

Steps in change control Process

2 mark

- 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.
- The ECO describes the change to be made, the constraints that must be respected, and the criteria for review and audit.
- A version control system updates the original file once the change has been made.

14.b (i) Show the Procedure to compute the cyclomatic complexity in testing with suitable example

Cyclomatic Complexity:

4 mark

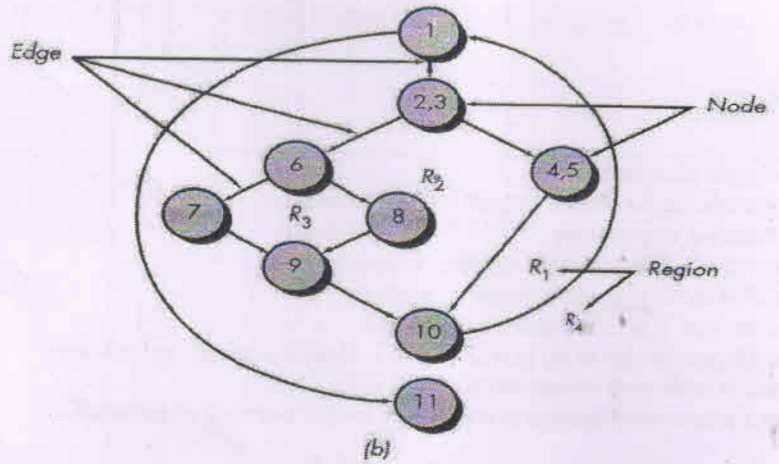
- Cyclomatic complexity is a software metric that provides a quantitative measure of the logical complexity of a program.
- Cyclomatic complexity has a foundation in graph theory and provides you with an extremely useful software metric.
- The number of regions of the flow graph corresponds to the cyclomatic complexity.
- Cyclomatic complexity **V (G)** for a flow graph G is defined as

  V (G)=E-N+2

  where E is the number of flow graph edges and
  N is the number of flow graph nodes.

Example: **(Own Example)**

4 mark



(b)

The cyclomatic complexity can be computed using each of the algorithms just noted:

1. The flow graph has **four regions**.
2. V (G)=E-N+2 **11 edges - 9 nodes +2 = 4.**

14.b (ii) Illustrate multiclass testing with banking application **(Any 10 Test Cases)**

8 mark

1. Test Cases for New Branch

- Generate a new branch with data from the valid and invalid tests
- Generate a new branch without data
- Generate a new branch with existing data
- Double-check the reset and cancel options
- Add branch details with valid and invalid test data
- Update branch details with existing test data
- Verify whether the new branch has been added
- Check if the cancelation option is working
- Check the branch deletion with and without dependencies
- Check if the branch search option is working

2. Test Cases for New Role

- Generate a new role with data from the valid and invalid tests
- Generate a new role without data
- Check if a new role can be created with existing test data
- Check the role description and role type
- Check whether the cancelation and reset option is working
- Check the role deletion process with and without dependency
- Double-check the links on the role detail page
- Check the admin login without test data
- Double-check all home links for the admin role
- Check if the admin can change the password with valid and invalid test data
- Check if the admin can log out successfully

3. Test Cases for Customers and Bankers

- Check if all visitor and customer links are working properly
- Double-check the customer's login with valid and invalid test data
- Check the customer's login without any data
- Check the banker login without any data
- Check the banker's login with valid or invalid test data
- Check whether the customer or banker was able to log out successfully

4. Test Cases for New Users

- Check if the new user can be created with valid and invalid test data
- Generate a new user with existing branch test data
- Check whether the cancel and reset option is working properly
- Add user details with valid and invalid test data
- Check the deletion of the new user
- Check whether the new user can be verified
- Check mandatory input parameters

- Check optional input parameters
- Check whether a user can be created without optional parameters

5. Test Cases for Net Banking Application

- Check whether the user is able to open the bank website
- Double-check if all the links on the website are working
- Check whether the user is able to create a new account
- Verify whether the user is able to log in with a valid or invalid username and password
- Check if the user is allowed to change the password
- Check whether a proper error message is shown if an invalid username or password is entered
- Make sure that after repeated attempts to log in with an incorrect password, the user should be shown an error message and blocked
- Verify whether the user is able to perform some basic transactions
- Make sure that the user is able to add a beneficiary with valid and invalid details
- Check whether the user can delete the beneficiary
- Make sure that the user is able to make transactions to the newly added beneficiary
- Verify whether the user is able to enter the amount in a decimal number
- Check whether the user is not able to enter negative numbers in the amount field
- Check whether the user is allowed to make transactions with or without a minimum balance
- Check whether the user can add a new RD
- Make sure that the correct message is showing in case of a transaction done with an insufficient balance
- Check whether the user is asked for confirmation before any transaction is made
- Check whether acknowledgment receipts are provided on each successful transaction
- Check if the user is able to transfer money to multiple accounts
- Check whether the user can cancel the transaction
- Make sure that the account details reflect the financial transactions done
- Check whether the timeout feature has been implemented
- Make sure that in case of session time out, a user should log in again
- Make sure that the proper session time out is done in case of any inactivity
- Make sure that while doing the transaction, the user is taken to secure mode
- Check whether the user was able to log out successfully
- Double-check the search and reset options

Solu:

Task	planned Value	Earned value	Actual cost
1	11,10,000	10,00,000	12,50,000
2	6,00,000	7,50,000	5,00,000
3	25,00,000	-	-
4	8,00,000	-	-

BCWS = 11,10,000 + 6,00,000

= 17,10,000

BAC = 11,10,000 + 6,00,000 + 25,00,000 + 8,00,000

= 50,10,000

BCWP = 10,00,000 + 7,50,000

= 17,50,000

ACWP = 12,50,000 + 5,00,000

= 17,50,000

Schedule performance Index : [3 Mark]

$$\boxed{SPI = \frac{BCWP}{BCWS}}$$

$$= \frac{17,50,000}{17,10,000}$$

$$= 1.02$$

Schedule variance (SV): [3 Mark]

$$\boxed{SV = BCWP - BCWS}$$

$= 17,50,000 - 17,10,000$

$SV = 40,000$

cost variance [3 mark]

$$CV = \boxed{BCWP - ACWP}$$

$= 17,50,000 - 17,50,000$

$= 0$

Cost performance Index: [3 mark]

$$\boxed{CPI = \frac{BCWP}{ACWP}}$$

$= \frac{17,50,000}{17,50,000}$

$= 1$

i)Since SV is +ve and SPI > 0, [2 Mark]

i) the above project is on schedule

Since, CV is 0 and the CPI is 1,

(ii) the project is on budget [2 mark]

15) b) i) Function point value :

[10 mark]

Solu:

Step -1 :

AS complexity adjustment factor is average (give in question), Scale = 3

[1 Mark]

$$\boxed{F = 14 \times 3 = 42}$$

Step - 2 :

$CAF = 0.65 + (0.01 * 42)$

$= 1.07$.

[2 mark]

Step -3 :

UFP :

$= (32 \times 4) + (60 \times 5) + (24 \times 4) + (8 \times 10) +$

(2×7)

$= 128 + 300 + 96 + 80 + 14$

$= 618$.

[4 mark]

Total count = 618

Step -5

[3 mark]

Function point = count * CAF

$= 618 * 1.07$

$= 661.26$

$\simeq 661$

Function pointvalue

i)

ii) compute cost per function point : [2 Mark]

$$\text{cost per FP} = \frac{\text{Average Labor Rate}}{\text{Average productivity}}$$

Assume, [own value]

Average Labor Rate = $8000 per month.
Average productivity = 6.5 FP/pm

$$\text{Cost per FP} = \frac{8000}{6.5}$$

$$\simeq 1230.76$$

$$\simeq 1230$$

cost per FP approximately $1230

iii) Total Estimated project cost : [2 Mark]

= cost per FP × count Total

= $1230 × 618

= $760,140

iv) Total Estimated project Effort : [2 Mark]

$$= \frac{\text{FP Estimated}}{\text{Average productivity}}$$

$$= \frac{661}{6.5}$$

$$= 101.69$$

= 102 persons - month

Note : Based on the assumption of average Labor Rate & [illegible] answer for (ii, iii, iv). will be