

BE Degree Examination November 2024

Fifth Semester

Computer Science and Engineering

22CSC51 - AGILE METHODOLOGIES

(Regulations 2022)

Time: Three hours

Maximum: 100 marks

Answer all Questions

Part - A ($10 \times 2 = 20$ marks)

1. List any two characteristics of the software. [CO1,K2]
2. Specify the procedures that are identified during requirement elicitation process. [CO1,K2]
3. State why the product owner is so important to scrum teams? [CO2,K2]
4. Many teams find that their daily scrum meetings are more effective. Justify with your views. [CO2,K2]
5. Why are informative workspaces important in XP? [CO3,K2]
6. If I am not using scrum, XP, Lean or Kanban does that mean my team isn't agile? [CO3,K2]
7. What guidelines lead to a successful software testing strategy? [CO4,K2]
8. Distinguish between white box testing and black box testing. [CO4,K2]
9. Mention the significance and five levels of capability maturity model (CMM). [CO5,K2]
10. Specify the need of software configuration management system. [CO5,K2]

Part - B ($5 \times 16 = 80$ marks)

11. a. Elaborate the procedure in collecting the requirements. Explain how the (16) [CO1,K2] gathered requirements are validated and how will you analyse whether requirements are sufficient?
(OR)
- b. Identify and brief the need of evolutionary approach for software development. (16) [CO1,K2] Explain in brief about the common models used in evolutionary approach.
12. a. i) Mention the rules of scrum that describe the basic pattern for a scrum (12) [CO2,K2] project in detail.
ii) Write the steps how to plan and run effective scrum print. (4) [CO2,K2]
(OR)

- b. i) How do teams estimate tasks? How do you handle global teams? (12) [CO2,K2]
 ii) Daily scrum keeps people working on the right tasks. But even well-meaning developers can get caught up doing things that aren't really the best use of their time. Can't scrum teams still get side-tracked? (4) [CO2,K2]

13. a. i) State all the primary practice of Extreme Programming (XP) and elaborate any two in detail. (12) [CO3,K2]
 ii) Why teams resist changes and how the practices help in XP? (4) [CO3,K2]

(OR)

- b. State the principles of kanban. How to use kanban board to visualize the workflow and explain with an example. (16) [CO3,K2]

14. a. i) With a suitable example, demonstrate the basis path testing and control structure testing. (8) [CO4,K3]
 ii) Is system testing, white-box testing or Black-box testing? Elaborate on the working principle of system testing. (8) [CO4,K2]

(OR)

- b. i) Elaborate on the various test strategies handled for conventional and object-oriented software with a suitable example. (8) [CO4,K3]
 ii) State the differences between testing and debugging. What comes first, testing or debugging? How to do debugging? Demonstrate with suitable example. (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

- 1) Project is on schedule and
- 2) Project is on budget.

Year	Planned value	Earned value	Actual cost
1	11,50,000	10,20,000	12,80,000
2	6,60,000	7,90,000	5,50,000
3	27,00,000		
4	8,50,000		

(OR)

- b. i) Elaborate the features of SCM process and discuss the various elements and its applications in software configuration management. (10) [CO5,K3]
 ii) Illustrate the COCOMO model with its advantages and challenges. (6) [CO5,K3]

Bloom's Taxonomy Level	Remembering (K1)	Understanding (K2)	Applying (K3)	Analysing (K4)	Evaluating (K5)	Creating (K6)
Percentage	-	74	26	-	-	-

Kongu Engineering College, Perundurai

BE Degree Examination Nov 2024

Fifth Semester

Computer Science and Engineering

20CSC51 - Agile Methodologies

(Regulation 2022)

Time: 3 hrs

ANSWER KEY

Max: 100 marks

PART-A ($10 \times 2 = 20$ marks)

1. List any two characteristics of the software.

- Software is developed or engineered; it is not manufactured
- Software doesn't wear out
- Although the industry is moving toward component based construction, most software continues to be custom built.

[Note:- Any two points]

2. Specify the procedures that are identified during requirement elicitation process.

- Identify stakeholders
- Recognize multiple viewpoints
- Work towards collaboration

3. State why the product owner is so important to Scrum teams?

The product owner is crucial to Scrum teams because they act as the voice of the customer, ensuring the team delivers maximum value. They prioritize the product backlog, provide clarity on requirements, and maintain alignment between

Stakeholders and the development team. This enables the team to stay focused on delivering the most important features efficiently.

4. Many teams find that their daily scrum meetings are more effective. Justify with your views.

Daily Scrum meetings are effective because they foster team alignment and collaboration by allowing members to share updates, identify roadblocks and plan the day's work efficiently. This promotes transparency, keeps everyone focussed on the sprint goals and enables the team to quickly adapt to any challenges by ensuring continuous progress.

5. Why are informative workspaces important in XP?

Informative workspaces are important in XP because they provide real-time visibility into the project's progress and status which ensures that all team members and stakeholders are aligned. Visual aids like board and charts promote transparency, facilitate collaboration and enable quick decision-making.

6. If I am not using Scrum, XP, Lean or Kanban does that mean my team isn't agile?

No, the team can still be agile without using Scrum, XP, Lean or Kanban. Agility is about adhering to the Agile principles, such as delivering value incrementally, responding to change and fostering collaboration, regardless of the framework or methodology used.

7. What guidelines lead to a successful software testing strategy?

- Specify product requirements
- State testing objectives explicitly
- Understand the user of the SW and develop a profile for each user category
- Develop testing plan (rapid cycle testing)
- Build robust SW that is designed to test itself
- Use effective technical reviews to assess the test strategy
- Develop a continuous improvement approach for the testing process.

8. Distinguish between white box and Black box testing

White Box	Black box
<ul style="list-style-type: none">- Testing method without having knowledge about the actual code and internal structure of the application	<ul style="list-style-type: none">- Testing method without having knowledge about the actual code or internal structure of the application.

White Box

- performed at lower level of testing such as unit testing, integration testing.
- It concentrates on the actual code and its syntax.
- Requires design documents with data flow diagrams, flowcharts etc.
- Done by developer or tester with programming knowledge.

Black Box

- performed at higher level testing such as functional testing.
- concentrates on the functionality of the system under test.
- Requires Requirement Specification to test.
- Done by the tester.

9. Mention the significance and five levels of capability maturity model (CMM).

CMM is a framework that helps organizations to improve their software development process.

five levels of CMM

1. Initial
2. Repeatable
3. Defined
4. Managed
5. Optimizing

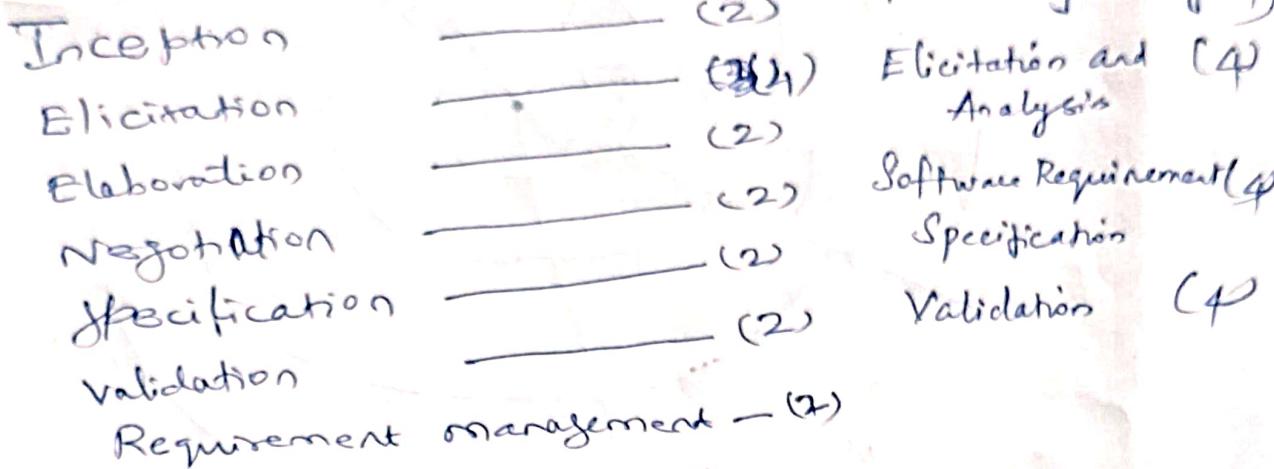
10 Specify the need of software configuration management system.

Software configuration management is a set of activities that have been developed to manage change throughout the life cycle of computer software and is essential to track and control changes and also to maintain consistency.

NOTE:- OUT OF syllabus. mark may be awarded if attended.

PART-B (15x16 = 80 marks)

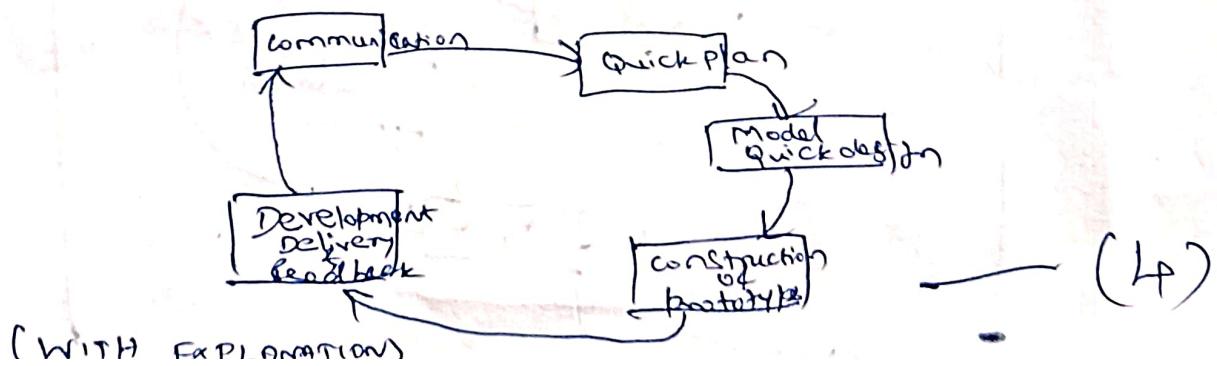
11(a) Elaborate new procedure in collecting the requirements. Explain how the gathered requirements are validated and how will you analyze whether requirements are sufficient?

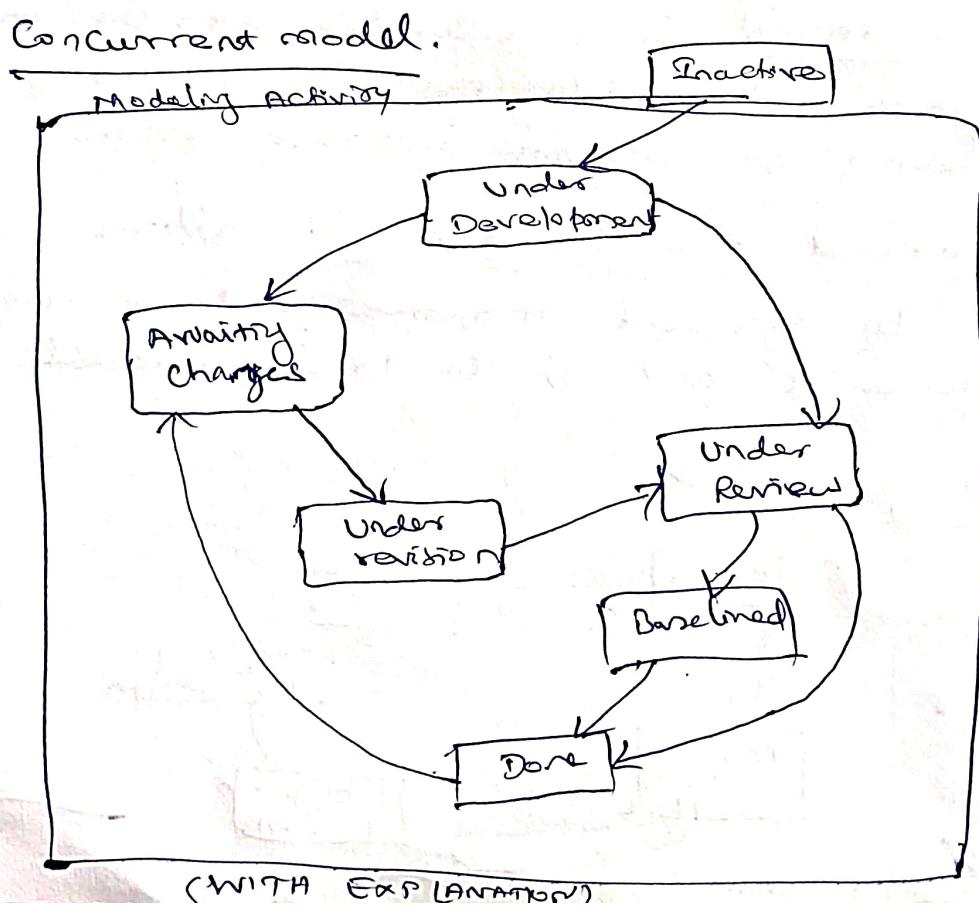
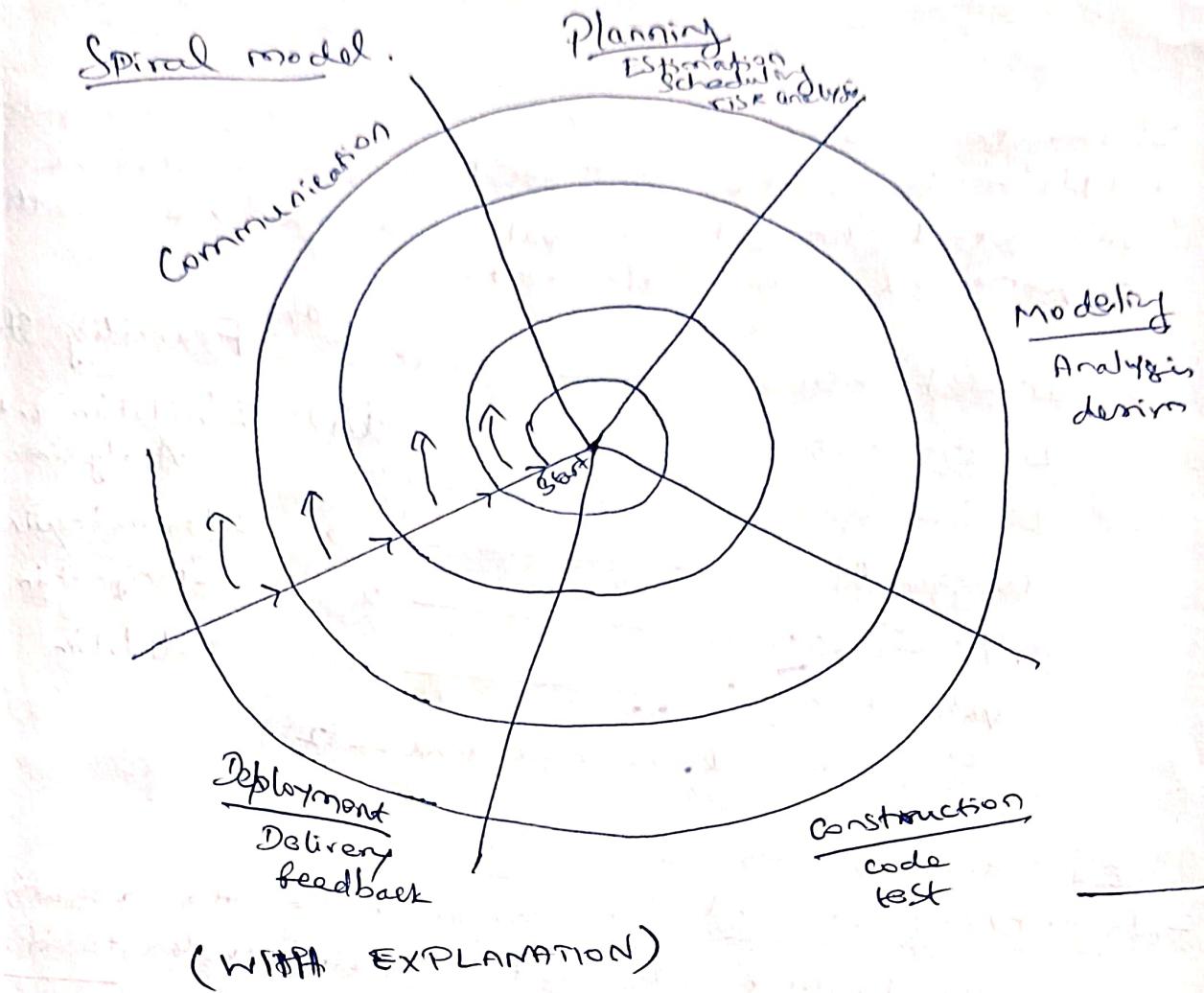


Evolutionary process model:-

- Prototyping model is used when the requirements of the product are not clearly understood. — (1)
- Spiral model is a risk-driven process model that is used to guide multi-stakeholder concurrent engineering of software intensive systems. — (1)
- Concurrent development model:- allows a software team to represent iterative and concurrent elements of any of the process models. — (1)

Prototyping model:





12

a)
i)

Scrum is a framework for managing and completing complex projects

certain rules that define the basic pattern for a Scrum project are:

→ Scrum (3)

- Scrum Roles

- Product owner
- Scrum master
- Development team

- Scrum Events (5)

- Sprint
- Sprint planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective

- Scrum Artifacts (3)

- Product Backlog
- Sprint Backlog
- Increment

- Transparency, Inspection and adaptation (1)

- Time Boxed (1)

- Definition of Done (1)

12
a) i)

- Sprint planning — (2)
 - Define the sprint goal
 - Select Backlog Items
 - Plan tasks
- Conduct the Sprint — (1)
 - Daily Scrums
 - focus on collaboration
- Deliver Increment
- Review and Retrospect } — (1)

12
b) i)

Teams Estimate tasks:

Scrum team uses the techniques are

- Planning Poker
- Generally accepted Scrum Practices (GRASP) } — (1)

Planning poker — (4)

- Each Team member uses a deck of cards with numbers like Fibonacci series numbers.
- After discussion about the task, members simultaneously reveal their estimates and the team discusses differences to reach a consensus.

GRASP - commonly practiced and widely accepted techniques and principles in Scrum that enhance its implementation and success.

- clear definition of Done (DOD)
- Regular refinement of product Backlog
- Timeboxing

} — (3)

Handling global teams.

(6)

- fosters clear communication
- schedule overlapping work hours
- use asynchronous communication
- collaboration tools
- Emphasize cultural sensitivity
- Establish strong division marker
- track progress transparently
- Promote team bonding.

(or) User Stories, Story Points, Velocity & Burndown Charts (3)

12)
b)(ii)

- Yes, Scrum team can still get side-tracked despite the structure provided by the Daily Scrum.
- This can happen due to unclear sprint goals.
 - Ineffective daily scrums, misaligned priorities.
 - What did WE achieve yesterday on priority 1?
 - What was OUR contribution on priority 1?
 - What is our plan for completing priority 1 today?
 - What if anything, is blocking US or has the potential to slow US down today?

The team works together to answer these questions.

4

13

a))

Primary practices of Extreme Programming (XP)

- Business or Planning Practices. — (4)
 - Planning tea frame
 - Small release
 - metaphor
 - On site customer
- Programming practices — (4)
 - Simple design
 - Test Driven Development
 - Pair programming
- Team work practices. — (4)
 - collective ownership
 - Coding standard
 - Refactoring
 - 40 hours a week
- Integration practices. — (4)
 - Continuous Integration
 - 10 minute Build.

Elaborate any two practices 4+4 — (8)

13

ii)

why team resist changes

— (2)

- fear of the unknown
- comfort with status quo
- lack of understanding
- fear of failure

How XP practices help. — (2)

- Planning
- Team
- Integration
- Programming

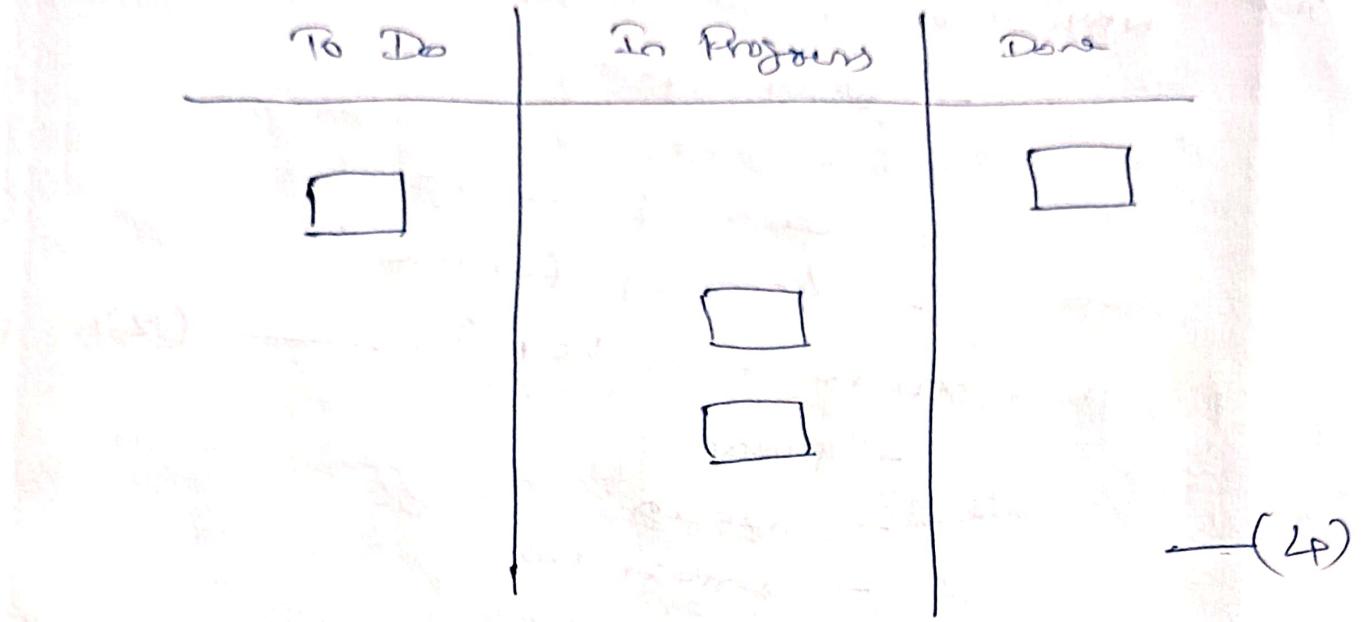
Kanban Principles. — (4)

- Start with what you do know
- Agree to pursue Incremental, Evolutionary change
- Respect the current process, Roles, Responsibilities and titles.
- Encourage acts of leadership at all levels in your organization.

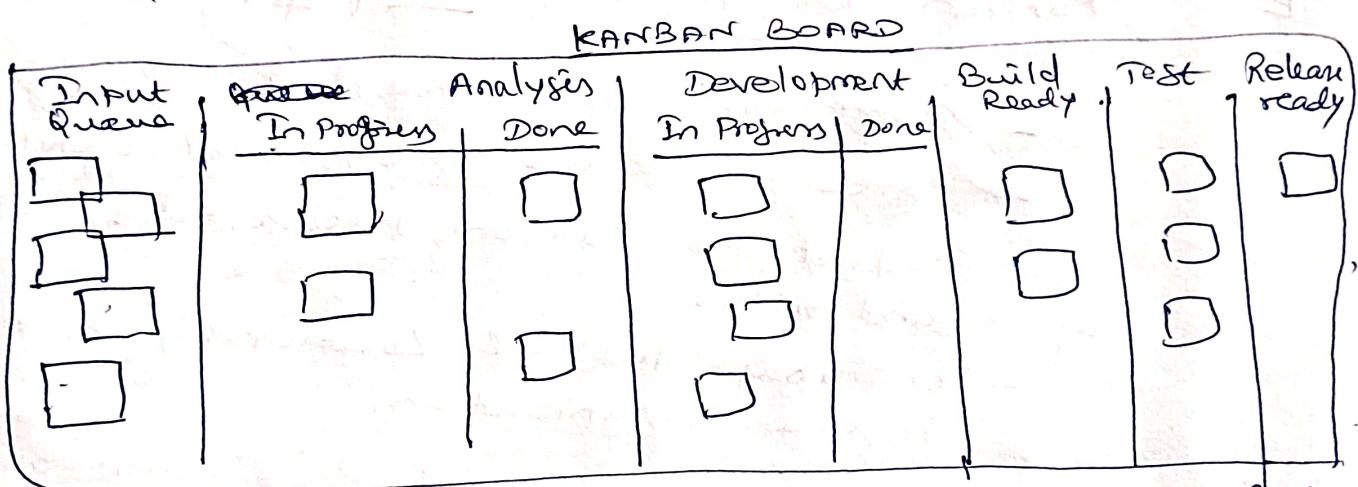
Kanban board to visualize the workflow

A Kanban board is a tool that teams use to visualize their workflow.

A basic kanban board has three-step workflow.



for example Consider a kanban board with Input Queue, Analysis, Development, Build ready, Test and Release as shown



With ~~explanations~~ with explanation → (8)

Note:- Students may write any realtime examples for creating kanban Board

14
a
v

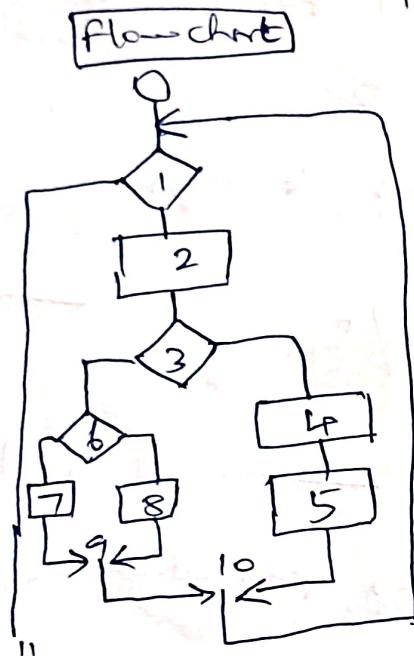
Basis Path testing

- Flow graph notation
- Independent program paths
- Deriving test cases
- Graph matrices.

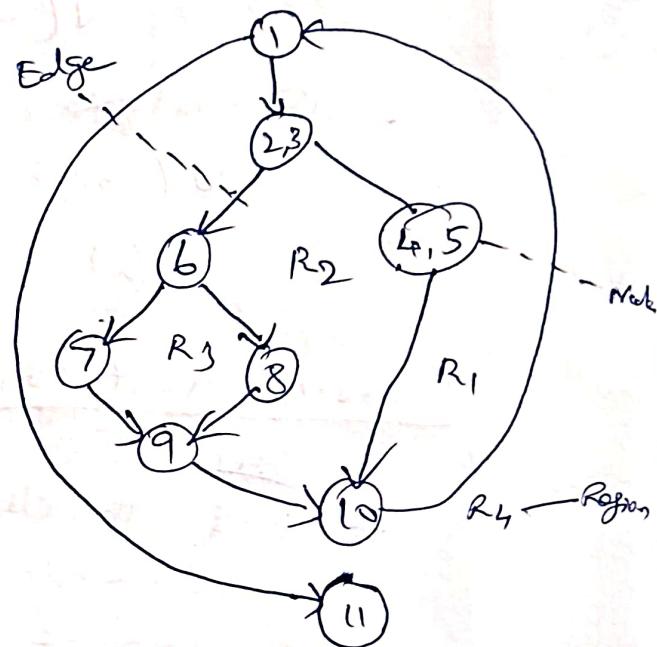
Flow graph notation — (2)

Have simple notations for the representation of control flow such as sequence, if, while, until and case.

Example: Consider the flowchart is used to abstract program control structure



Corresponding flow graph.



Independent Program paths is any path through the program that introduces atleast one new set of processing statements or a new condition.

A set of independent paths for the flow graph.

Path 1 : 1 - 11

Path 2 : 1 - 2 - 3 - 4 - 5 - 10 - 1 - 11

Path 3 : 1 - 2 - 3 - 6 - 8 - 9 - 10 - 1 - 11

Path 4 : 1 - 2 - 3 - 6 - 7 - 9 - 10 - 1 - 11 — (1)

Cyclomatic complexity is a software metric that provides a quantitative measure of the logical complexity of a program.

- Number of regions of flow graph corresponds to the cyclomatic complexity
- Cyclomatic complexity $v(G)$ for a graph G is

$$v(G) = E - N + 2$$

- The above flow graph has 4 regions

$$v(G) = E - N + 2$$

$$v(G) = 11 - 9 + 2 = 4$$

(1)

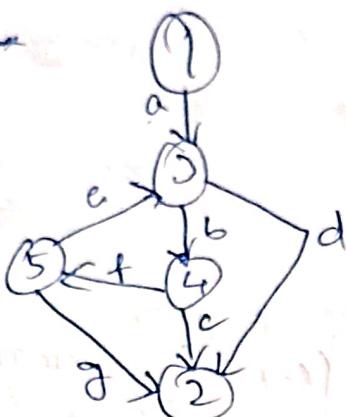
Deriving Test Cases

Steps

- Using the design or code as foundation, draw a corresponding flow graph.
- Determine the cyclomatic complexity
- Determine a basis set of linearly independent paths
- Determine a basis set of linearly independent paths
- Prepare test cases that will force execution of each path in the basis set.

Graph matrices is a square matrix whose size is equal to the number of nodes on the flow graph.

- Each row and column corresponds to an identified node and matrix entries correspond to connection between nodes longest to node



flow graph

Node	1	2	3	4	5
1			a		
2					b
3			d		f
4		c			
5	e	g			

Graph matrix
— (1)

control structure testing

- Conditional testing
- Data flow testing
- Loop testing.

Conditional testing is a test case design method that exercises the logical conditions contained in a program module.

Data flow testing selects test paths of a program according to the locations of definitions and uses of variables in the program.

Loop testing is a white box testing that focuses exclusively on the ~~is~~ validity of loop conditions

[Basis path Testing (any 2 methods) - ~~2m~~]
[Control structure Testing (any 2 methods) - ~~2m~~]

14
aii)

System testing is a Black box testing technique. The entire system is integrated and then tested to ensure that it meets the specified requirements, and functions as expected. It focuses on validating the system from an end-user perspective without requiring any knowledge of the internal working or code structure of the system.

- focuses on functional
- Test Planning
- Test design
- Test execution
- Defect reporting

requirements
Performance Testing
Stress Testing
Security Testing
Recovery Testing
Deployment Testing

(4)

14
bii)

Test strategies for conventional software

- ~~Unit testing~~
~~Integration testing~~
- Unit testing → ②
 - unit test - consideration
 - unit test procedures.
- Integration testing → ②
 - Top down integration
 - Bottom up integration
 - Regression testing
 - Smoke testing

Test Strategies for object-oriented software.

Unit Testing in the oo context. — (2)

Integration testing in the oo context. — (2)
with object oriented sl/w example.

15
Q)

14 B ii) in last page

Budget at completion as per given Data is 53,60,000

$$BCWS = 18,10,000$$

$$BCWP = 18,10,000$$

$$ACWP = 18,30,000$$

$$SPI = \frac{BCWP}{BCWS} = \frac{18,10,000}{18,10,000} = 1$$

$$SPI = 1$$
 — (2)

$$SV = BCWP - BCWS$$

$$= 18,10,000 - 18,10,000 = 0$$

$$SV = 0$$
 — (2)

$$CPI = \frac{BCWP}{ACWP} = \frac{18,10,000}{18,30,000} = 0.98$$

$$CV = BCWP - ACWP$$

$$= 18,10,000 - 18,30,000$$

$$CV = -20,000$$
 — (3)

Here SV = 0 and SPI = 1 > 0

∴ Project is on schedule. — (2)

CV = -ve and CPI < 1

∴ Project is over Budget — (2)

15
b)

Features of SCM process and its Elements.

- to identify all items that collectively define the software configuration
 - to manage changes to one or more items
 - to facilitate the construction of different versions of an application
 - to ensure that SW quality is maintained as the configuration evolves over time.
- (3)

Scm Process Elements/Tasks — (7)

Identification
Version control
Change control
Configuration auditing
Reporting.

(with explanation)

[NOTE:- THIS QUESTION IS OUT OF SYLLABUS. SO
IF STUDENTS ATTEMPT THIS QUESTION, MAY
BE AWARDED FULL MARKS.]

15

b) i)

COCOMO model constructive cost model is a software cost estimation model helps to estimate the effort, cost and schedule of software projects based on their size and complexity. — (2)

Illustration

$$\text{Effort}(E) = a \times (\text{kLOC})^b$$

$$\text{Development Time}(D) = c \times (\text{Effort})^d$$

where a, b, c, d are constants based on the project type such as organic, semidetached and Embedded.

(1)

Advantages

- Transparency
- Adaptability
- Wide applicability
- Refinement
- Ease of use.

(1)

Challenger

- Dependency on KLOC
- Obsolescence.

14b
ii)

Testing

- Process of identifying defects or issues in software
- Ensure SW works as intended
- Testers or quality assurance team will perform
- Output in test cases, test results and defect reports
- performed before debugging

Debugging

- Process of finding, analyzing and fixing identified defects.
- correct errors found during testing.
- performed by developers.
- A working version of the SW with fixed defects
- performed after testing reveals issues.

(4)

- Testing comes first which identifies defects and then resolved during debugging. — (1)

How to do debugging

- understand the problem
 - reproduce the issue
 - Analyze the code
 - modify the code
 - verify the fix

(Explanation with Sample code.)

— (3)

(18) Brute force

Backtracking

Cause Elimination with diagram (6)

Testing (2)