Roll No.

PRESIDENCY UNIVERSITY
BENGALURU

SCHOOL OF ENGINEERING

TEST – I

Sem AY: Odd Sem 2019-20	**Date:** 28.9.2019
Course Code: CSE 227	**Time:** 9.30AM to 10.30AM
Course Name: SOFTWARE ENGINEERING & PROJECT MANAGEMENT	**Max Marks:** 40
Program & Sem: B.Tech (CSE) & V	**Weightage:** 20%

Instructions:

(i) Read the question properly and answer accordingly.
(ii) Question paper consists of 3 parts.

Part A [Memory Recall Question]

Answer all the Questions. Each Question carries four marks. **(3Qx4M =12M)**

1. Enlist and explain any 4 reasons for incorporating changes to a software. (C.O.NO.1)[Knowledge]
2. Explain the Spiral model of Software development. What are the limitations of such a model? (C.O.NO.1)[Knowledge]
3. Explain the terms: entity class, boundary class and control class with examples. (C.O.NO.1)[Knowledge]

Part B [Thought Provoking Questions]

Answer all the Questions. Each Question carries eight marks **(2Qx8M=16M)**

4. (a) Provide two examples for interaction based system where use case approach would be amenable. Be specific (C.O.NO.1)[Comprehension] [2M]

(b) Describe the XP concepts of user stories, refactoring and pair programming in your own words. (C.O.NO.1][Knowledge] [6M]

5. Select the appropriate process model based on characteristics of requirements which are as follows. Substantiate your answer.

(a) Requirements are easily understandable and well defined. [2M]

(b) Unclear user requirement. [2M]

(c) A complex system needs to be built. [2M]

(d) Unfamiliar technology. [2M]

(C.O.NO.1)[Comprehension]

Part C

Answer all the Questions. Each Question carries twelve marks. (1Qx12M=12M)

6. Consider the scenario given below: In a library management system,

Any library member should be able to search books by their title, author, subject category as well by the publication date. Each book will have a unique identification number and other details including a rack number which will help to physically locate the book. There could be more than one copy of a book, and library members should be able to check-out and reserve any copy. We will call each copy of a book, a book item. The system should be able to retrieve information like who took a particular book or what are the books checked-out by a specific library member. There should be a maximum limit (5) on how many books a member can check-out. There should be a maximum limit (10) on how many days a member can keep a book. The system should be able to collect fines for books returned after the due date. Members should be able to reserve books that are not currently available. The system should be able to send notifications whenever the reserved books become available, as well as when the book is not returned within the due date. Each book and member card will have a unique barcode. The system will be able to read barcodes from books and members' library cards.

(a) Identify the actor(s) for the system. [C.O.NO.1][Application][4M]

(b) Write the use cases with brief description. [C.O.NO.1][Application][4M]

(c) Draw a Use Case Diagram for the Library system. [C.O.NO.1][Application][4M]

SCHOOL OF ENGINEERING

Semester: V

Course Code: CSE227

Course Name: Software Engineering and Project Management

Date: 28 September 2019

Time: 9.30AM to 10.30 AM

Max Marks: 40

Weightage: 20%

Extract of question distribution [outcome wise & level wise]

Q.NO.	C.O.NO	Unit/Module Number/Unit /Module Title	Memory recall type [Marks allotted] Bloom's Levels			Thought provoking type [Marks allotted] Bloom's Levels			Problem Solving type [Marks allotted]			Total Marks
			K			C			A			
1	CO1	M1/ Introduction to Software Engineering & Process Models	4									4
2	CO1	M1/ Introduction to Software Engineering & Process Models	4									4
3	CO2	M2/ Software Requirements and Design	4									4
4.a	CO1	M1/ Introduction to Software Engineering & Process Models				2						2
4.b	CO1	M1/ Introduction to Software	6									6

		Engineering & Process Models										
5	CO1	M1/ Introduction to Software Engineering & Process Models				8						8
6 a	CO1	M1/ Introduction to Software Engineering & Process Models								4		4
6b	CO1	M1/ Introduction to Software Engineering & Process Models								4		4
6c	CO1	M1/ Introduction to Software Engineering & Process Models								4		4
Total			18			10				12		40

K =Knowledge Level   C = Comprehension Level, A = Application Level

Note: While setting all types of questions the general guideline is that about 60%

Of the questions must be such that even a below average students must be able to attempt, About 20% of the questions must be such that only above average students must be able to attempt and finally 20% of the questions must be such that only the bright students must be able to attempt.

[I hereby certify that All the questions are set as per the above guide lines. Dr. Jeno Lovesum]

Reviewers' Comments

SCHOOL OF ENGINEERING

SOLUTION

Semester: V

Course Code: CSE227

Course Name: Software Engineering& Project Management

Date: 28 September 2019

Time: 9.30am to 10.30 am

Max Marks: 40

Weightage: 20%

Part A

(3 Q x 4M = 12 Marks)

Q No	Solution	Scheme of Marking	Max. Time requirec for each Question
1	• Software must be adapted to meet the needs of new computing environments or technology. • Software must be enhanced to implement new business requirements. • Software must be extended to make it interoperable with other more modern systems or databases. • Software must be re-architected to make it viable within a network environment.	4 marks for any 4 reason explanation. 1 mark for each.	5 mins
2	Spiral model is an evolutionary software process model that couples the iterative nature of prototyping with the controlled and systematic aspects of the waterfall model. Spiral includes framework activities: Communication, Planning (Risk analysis), Modeling, Construction and Deployment. Limitations: Difficulty in controlling the evolutionary approach, over relying on expertise in risk assessment, difficulty in project planning etc.	3 marks for explanation of spiral model + 1 mark for any one limitation	5 mins

3	Entity Class: Also called as "Model" or "Business classes • Terms that are domain-specific in use cases • Recurring nouns • Real-world entities and activities tracked by system • They correspond to "Database Table" Examples of entity classes are: • Employee • Student • Field Officer Boundary Class: • Identify general user interface controls that initiate a use case • Identify forms or windows for entering data into a system • Identify messages used by system to respond to a user Examples of boundary classes are: • Student Attendance Report • Employee Details Capture Form Controller Class: • Creation or update of entity objects • Instantiation of boundary objects as they obtain information from entity objects • Complex communication between sets of objects • Validation of data communicated between objects or between the user and the application. Examples of controller classes are: • Product Controller in a Product automation system • Home Controller in a Home automation system	2 marks for explanation + 2 marks for examples.	5 mins

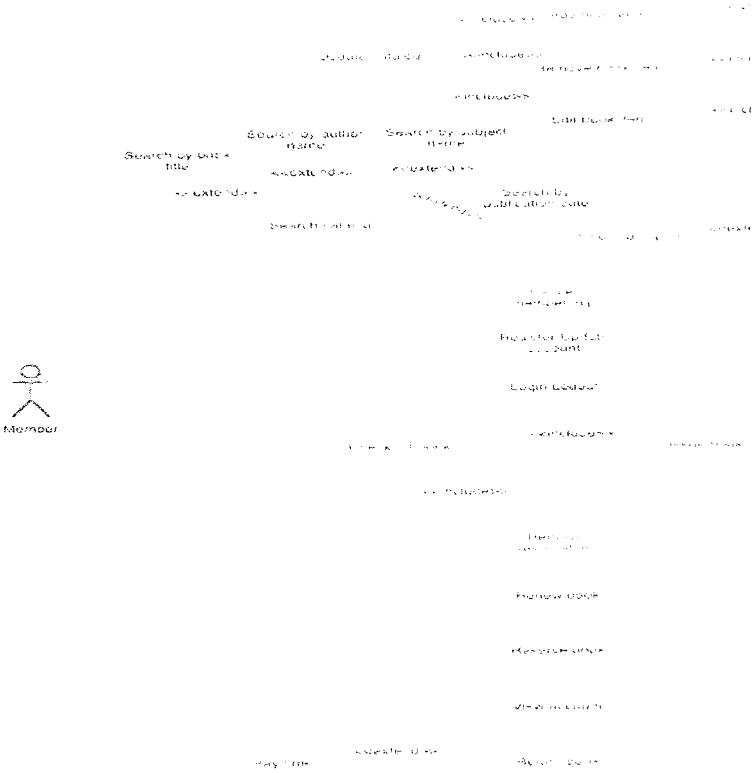
Part B (2 Q x 8M = 16 Marks)

Q No	Solution	Scheme of Marking	Max. Time require for each Question
4a.	Web Application Mobile Application	2 marks for two examples	3 mins
4b .	In XP. User stories describe the features and functionality of the system to be developed. Each story is assigned a "value" (priority). Refactoring is an activity to improve the design. after the code is written. It minimizes the chances of introducing bugs. Pair programming involves	6 marks. 2 marks each for user stories. refactoring and	10 mins

	two developers coding on the user stories on the same machine. One might look into the unit testing, while other refines the design, while coding activity is on.	pair programming.	
5	(a) Waterfall model since requirements are well defined: all the framework activities can be performed sequentially. (b) Agile or Spiral since first few cycles (spiral) or user stories (agile) can be spent in requirement gathering and prototyping, if required. (c) Iterative / Incremental development where the base set of requirements are implemented first; and complexity is added incrementally. (d) Evolutionary model – Prototyping or Spiral.	(One mark for identification of software model + one mark for substantiation) * 4 questions	8 mins

Part C (1Q x 12M = 12 Marks)

Q No	Solution	Scheme of Marking	Max. Time requirec for each Question
6a.	a. **Librarian:** Mainly responsible for adding and modifying books, book items, and users. The Librarian can also issue, reserve, and return book items. b. **Member:** All members can search the catalog, as well as check-out, reserve, renew, and return a book. c. **System:** Mainly responsible for sending notifications for overdue books, canceled reservations, etc.	4 mark for Actor identification	6 mins
6b.	Use Cases: 1. **Add/Remove/Edit book:** To add, remove or modify a book or book item. 2. **Search catalog:** To search books by title, author, subject or publication date. 3. **Register new account/cancel membership:** To add a new member or cancel the membership of an existing member. 4. **Check-out book:** To borrow a book from the library. 5. **Reserve book:** To reserve a book which is not currently available. 6. **Renew a book:** To reborrow an already checked-out book.	4 marks for identifying any 4 Use Cases with brief description. 1 mark for each.	6 mins

	7. **Return a book:** To return a book to the library which was issued to a member.		
6c.	Use Case Diagram:  Search by author name Search by subject name Search by book title <<extend>> <<extend>> <<extend>> Search by publication date <<include>> Register/Update account Login/Logout Member <<include>> <<include>> Renew book <<extend>> Pay fine	1 mark for SUD 1 mark for Actors 1 mark for Use cases 1 mark for relationships between the Actors and Use Cases	12 mins

Roll No.													

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY
BENGALURU

SCHOOL OF ENGINEERING

TEST – 2

Sem & AY: Odd Sem 2019-20

Course Code: CSE 227

Course Name: SOFTWARE ENGINEERING AND PROJECT MANAGEMENT

Program & Sem: B.Tech (CSE) & V Sem

Date: 20.11.2019

Time: 9.30 AM to 10.30 AM

Max Marks: 40

Weightage: 20%

Instructions:

(i) Answer all the questions.

Part A [Memory Recall Questions]

Answer all the Questions. Each question carries four marks. (4Qx4M=16M)

1. What are the core steps of Six Sigma methodology for software engineering? (C.O.NO.4)[Knowledge]
2. Considering the cost of quality, which do you think is the most expensive and why? (C.O.NO.4)[Comprehension]
3. Using your own words, describe the difference between verification and validation? (C.O.NO.3)[Comprehension]
4. List out various architectural styles with example applications? (C.O.NO.2)[Knowledge]

Part B [Thought Provoking Questions]

Answer both the Questions. Each question carries six marks. (2Qx6M=12M)

5. Explain Integration testing with an example? (C.O.NO.3)[Comprehension]
6. List out the golden rules of a user interface design and explain them. (C.O.NO.2)[Knowledge]

Part C [Problem Solving Questions]

Answer the Question. The question carry twelve marks. (1Qx12M=12M)

7. Demonstrate basis path testing for the given program. (C.O.NO.3)[Application]

```
Function delete_element (int value, int array_size, int array[])
{
        int i;
        location = array_size + 1;

        for i = 1 to array_size
                if ( array[i] == value )
                        location = i;
                end if;
        end for;

        for i = location to array_size
                array[i] = array[i+1];
        end for;

        array_size --;
}
```

SCHOOL OF ENGINEERING

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Date: 20/11/2019

Time: 9.30 to 10.30 a.m

Max Marks: 40

Weightage: 20%

Semester: V

Course Code: CS227

Course Name: Software Engineering and Project Management

Extract of question distribution [outcome wise & level wise]

Q.NO	C.O.NO	Unit/Module Number/Unit /Module Title	Memory recall type [Marks allotted] Bloom's Levels			Thought provoking type [Marks allotted] Bloom's Levels			Problem Solving type [Marks allotted]			Total Marks
			K			C			A			
1	CO4	4	4									4
2	CO4	4				4						4
3	CO3	3				4						4
4	CO2	2	4									4
5	CO3	3				6						6
6	CO2	2	6									6
7	CO3	3							12			12
	Total Marks		14			14			12			40

K =Knowledge Level C = Comprehension Level, A = Application Level

Note: While setting all types of questions the general guideline is that about 60%

Of the questions must be such that even a below average students must be able to attempt; About 20% of the questions must be such that only above average students must be able to attempt and finally 20% of the questions must be such that only the bright students must be able to attempt.

Annexure- II: Format of Answer Scheme

SCHOOL OF ENGINEERING

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

SOLUTION

Date: 20/11/2019

Semester: V

Time: 9.30 to 10.30 a.m.

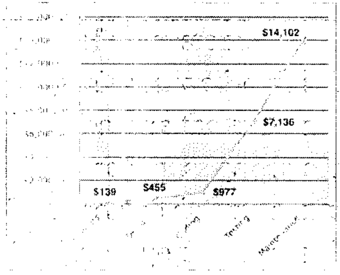
Course Code: CS227

Max Marks: 40

Course Name: Software Engineering and Project Management

Weightage: 20%

Part A (4 x 4 = 16)

Q No	**Solution**	**Scheme of Marking**	**Max. Time required for each Question**
1	1. **Define** customer requirements and deliverables and project goals via well-defined methods of customer communication. 2. **Measure** the existing process and its output to determine current quality performance (collect defect metrics). 3. **Analyze** defect metrics and determine the vital few causes. 4. **Improve** the process by eliminating the root causes of defects. 5. **Control** the process to ensure that future work does not reintroduce the causes of defects.	For each point :**0.8 M**	**5 Mins**
2	The relative costs to find and repair an error or defect increase dramatically as we go from prevention to detection to internal failure to **external failure costs**.  	External Failure Costs: **2M** Explanation with example: **2M**	**5 Mins**
3	***Verification*** refers to the set of tasks that ensure that software correctly implements a specific function.	Verification :**2M**	5 Mins

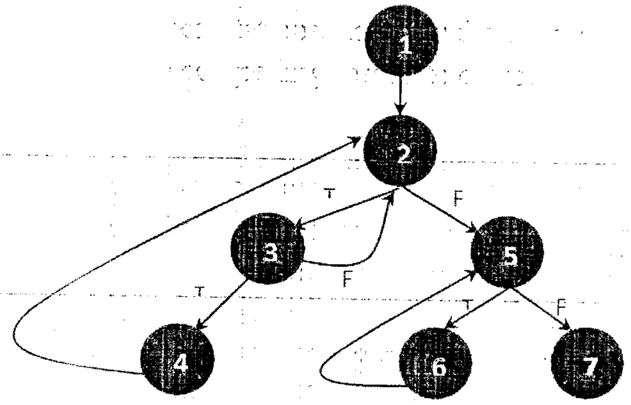
	Validation refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements. ***Verification:*** "Are we building the product right?" ***Validation:*** "Are we building the right product?"	Validation :**2M**	
4	• Data-centered architectures • Data flow architectures • Call and return architectures • Object-oriented architectures • Layered architectures	Architectures Styles : **3M** Example:**1M**	**5 mins**

Part B (2 x 6 = 12)

Q No	Solution	Scheme of Marking	Max. Time required for each Question
5	**Integration testing** - testing done before, during and after integration of a new module into the main software package. This involves testing of each individual code module. One piece of software can contain several modules which are often created by several different programmers. It is crucial to test each module's effect on the entire program model. Performed by **professional testing team.** 1. Big Bang approach 2. Incremental Construction strategy a. Top Down Integration b. Bottom-Up Integration c. Sandwich Testing	Definition: **2M** Listing types of Integration Testing: **2M** Example: **2M**	10 mins
6	1. **Place the User in Control** a. Define interaction modes in a way that does not force a user into unnecessary or undesired actions. (Ex: Spell – Check) b. Provide for flexible interaction. (Ex: Keyboard, Mouse, Voice) c. Allow user interaction to be interruptible and undoable. d. Streamline interaction as skill levels advance and allow the interaction to be customized. e. Hide technical internals from the casual user. f. Design for direct interaction with objects that appear on the screen. (Ex: Stretch the screen) 2. **Reduce the User's Memory Load** a. Reduce demand on short-term memory. b. Establish meaningful defaults.	Golden Rules Only: **2M** Explanation: **4M**	10 mins

	c. Define shortcuts that are intuitive. d. The visual layout of the interface should be based on a real world metaphor. e. Disclose information in a progressive fashion **3. Make the Interface Consistent** a. Allow the user to put the current task into a meaningful context. b. Maintain consistency across a family of applications. c. If past interactive models have created user expectations, do not make changes unless there is a compelling reason to do so.		

Part C (1 x 12 = 12)

Q No	Solution	Scheme of Marking	Max. Time required for each Question
7	(a). Control Flow Graph:  1 2 T F 3 F 5 T T F 4 6 7 (b). Cyclometric complexity: 4 (c). Independent Paths: **Path 1: 1 - 2 - 5 - 7** **Path 2: 1 - 2 - 5 - 6 -5- 7** **Path 3: 1 - 2 - 3 - 2 - 5 - 6 - 5- 7** **Path 4: 1 - 2 - 3 - 4 - 2 - 5 - 6 - 5- 7** (d) Test Cases: Path 1: value=2, array_size=0, array={} Path 2: Unable to do Independently Path 3. Unable to do Independently Path 4: value=2, array_size=4,array{2,3,4,5}	(a) CFG : **4M** (b) Cyclometric complexity **:4M** (c) Independent Paths/ Basis Test Path sets : **4M** (d) Designing Test Cases (for two paths): **4M** (2M+2M)	20 Mins

END TERM FINAL EXAMINATION

Semester: Odd Semester: 2019 - 20 **Date**: 30 Dec 2019

Course Code: CSE 227 **Time**: 9:30 AM to 12:30 PM

Course Name: SOFTWARE ENGINEERING AND PROJECT MANAGEMENT **Max Marks**: 80

Program & Sem: B.Tech (CSE) & V **Weightage**: 40 %

Instructions:

(i) Read the all questions carefully and answer accordingly.

(ii) Question paper consists of 3 parts.

Part A [Memory Recall Questions]

Answer all the Questions. Each Question carries 2 marks. (10Qx4M=40M)

1. What is Software Process? List Software Process Framework Activities (C.O.No.1) [Knowledge]
2. What is the Agile Manifesto? With the help of a neat diagram, illustrate Scrum method. (C.O.No.1) [Knowledge]
3. With neat diagram discuss the spiral model of software process. Illustrate how both the waterfall model of the software process and the prototyping model can be accommodated in the spiral process model. (C.O.No.1) [Comprehension]
4. With a neat diagram, explain Analysis to Design Model transition. (C.O.No.2) [Knowledge]
5. List and explain Web App Interface Design Principles (C.O.No.2) [Knowledge]
6. What is Software Architecture? With neat diagram explain any two Architectural Styles. (C.O.No.2) [Knowledge]
7. With a neat diagram illustrate Testing Strategies for software development (C.O.No.3) [Comprehension]
8. Illustrate the followings (C.O.No.3) [Comprehension]
   i. Regression Testing
   ii. Smoke Testing
9. List and explain Quality Dimensions (C.O.No.4) [Knowledge]
10. List and explain 4Ps of Software Project Management. (C.O.No.5) [Knowledge]

Part B [Thought Provoking Questions]

Part C [Problem Solving Questions]

Answer both the Questions. Each Question carries 11 marks. **(2Qx11M=22M)**

14. Project manager handling 3 different software project classes such as Organic mode project, Semi-detached mode project, embedded mode project and suppose all the projects size is 55 KLOC. using Basic COCOMO model, kindly estimates the effort in person-months (E), development time in chronological months (D) and number of people required to each project based on the below given data. (C.O.No.4) [Application]

Project class	ab	bb	cb	db
Organic	2.4	1.05	2.5	0.38
semidetached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

15. If you are a Project manager in an organization. COE is allotted one software project and they give the Estimated Activity Duration of a Project. So you need to find the Critical Path of the project using Network Planning Model. (C.O.No.5) [Application]

Estimated Activity Duration of a Project:

Activity		Duration (weeks)	Precedents
A	Hardware selection	10	
B	Software design	8	
C	Install hardware	7	A
D	Code & test software	8	B
E	File take-on	7	B
F	Write user manuals	14	
G	User training	7	E,F
H	Install & test system	6	C,D

END TERM FINAL EXAMINATION

Extract of question distribution [outcome wise & level wise]

Q.NO.	C.O.NO (% age of CO)	Unit/Module Number/Unit /Module Title	Memory recall type [Marks allotted] Bloom's Levels	Thought provoking type [Marks allotted] Bloom's Levels	Problem Solving type [Marks allotted]	Total Marks
			K	C	A	
1	1	1	4			4
2	1	1	4			4
3	1	1		4		4
4	2	2	4			4
5	2	2	4			4
6	2	2	4			4
7	3	3		4		4
8	3	3		4		4
9	4	4	4			4
10	5	5	4			4
11	3	3			6	6
12	4	4		6		6
13	5	5		6		6
14	4	4			11	11
15	5	5			11	11
	Total Marks		28	24	28	80

I hereby certify that all the questions are set as per the above guidelines.

Faculty Signature: (Dr. Nagaraja. S R).

Reviewer Commend:

Format of Answer Scheme

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

SCHOOL OF ENGINEERING

SOLUTION

Semester: Odd Semester: 2019-20

Course Code: CSE 227

Course Name: SOFTWARE ENGINEERING AND PROJECT MANAGEMENT

Program & Sem: B.Tech (CSE) & V

Date: 30 December 2019

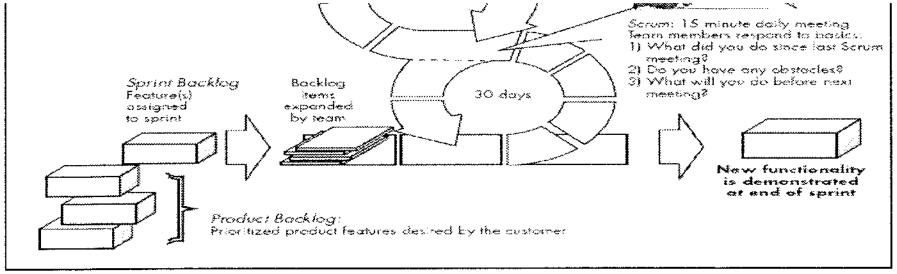
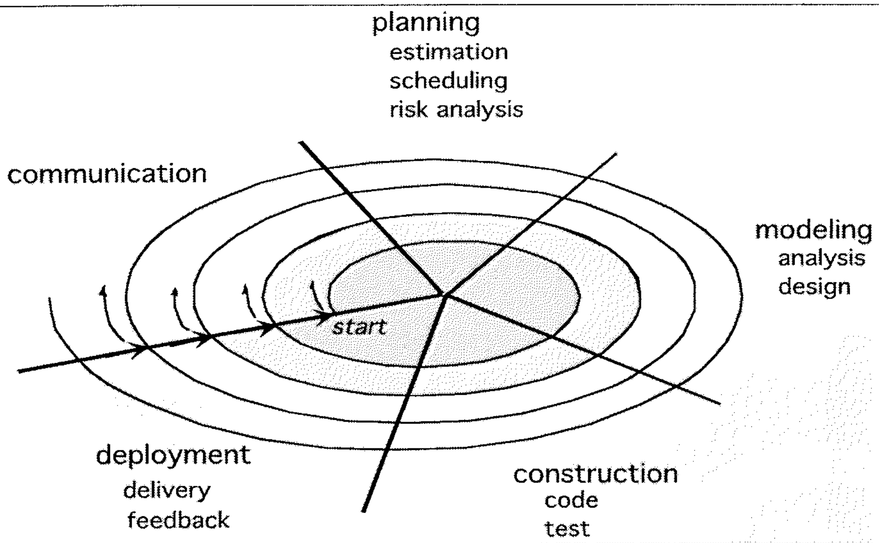
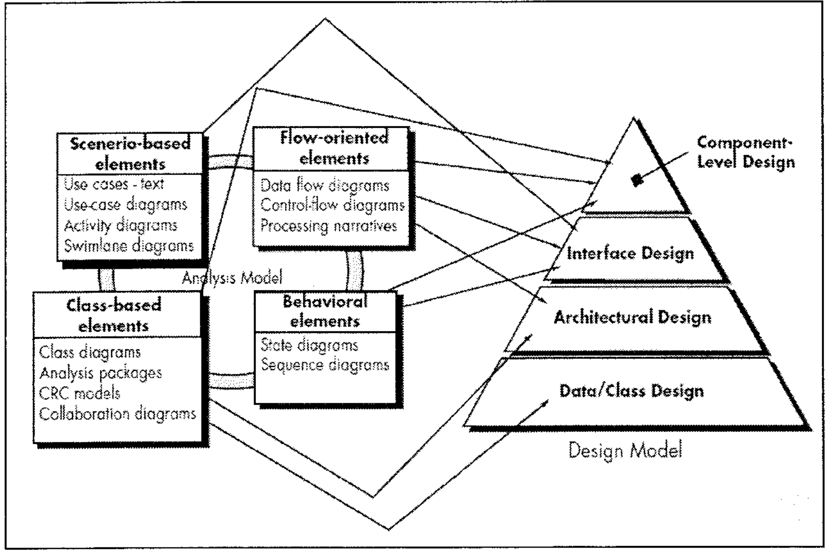
Time: 9:30 AM to 12: 30 PM

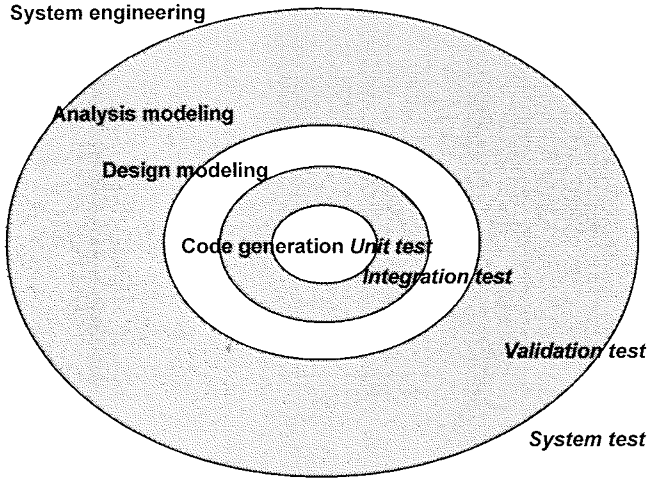
Max Marks: 80

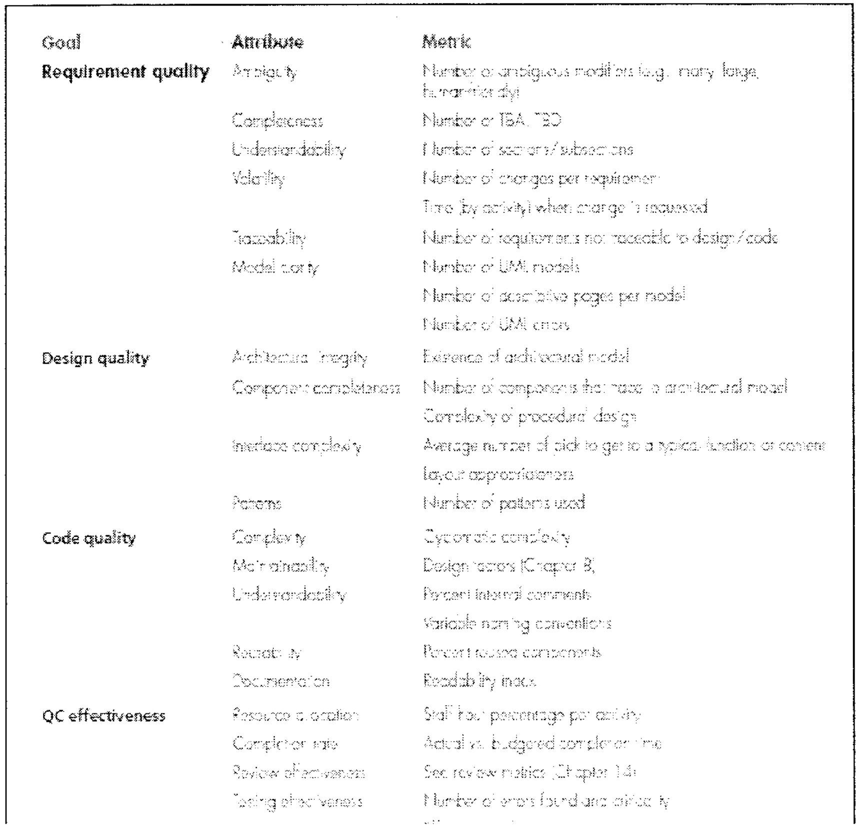
Weightage: 40%

Part A (10Q x 04M = 40Marks)

Q No	Solution	Scheme of Marking	Max. Time required for each Question
1	Software Process (SP) is a **framework** for the activities, actions, and tasks that are required to build high-quality software. Framework Activities: ■ Communication ■ Planning ■ Modeling ■ Analysis of requirements ■ Design ■ Construction ■ Code generation ■ Testing ■ Deployment	**Definition 2 marks and framework activity 2 marks**	**6 min**

	 Scrum: 15 minute daily meeting Team members respond to basics: 1) What did you do since last Scrum meeting? 2) Do you have any obstacles? 3) What will you do before next meeting? Sprint Backlog Feature(s) assigned to sprint Backlog items expanded by team 30 days New functionality is demonstrated at end of sprint Product Backlog: Prioritized product features desired by the customer		
3	 planning estimation scheduling risk analysis communication modeling analysis design start deployment delivery feedback construction code test	3 marks for diagram, 3 marks for Illustrate	**8 min**
4	 Scenerio-based elements Use cases - text Use-case diagrams Activity diagrams Swimlane diagrams Flow-oriented elements Data flow diagrams Control-flow diagrams Processing narratives Analysis Model Class-based elements Class diagrams Analysis packages CRC models Collaboration diagrams Behavioral elements State diagrams Sequence diagrams Component-Level Design Interface Design Architectural Design Data/Class Design Design Model	2 marks for diagram, 2 marks for explanation	**8 min**
5	List WebApp Interface Design Principles. Anticipation Communication Consistency	**4 marks for any 8,principles**	**8 min**

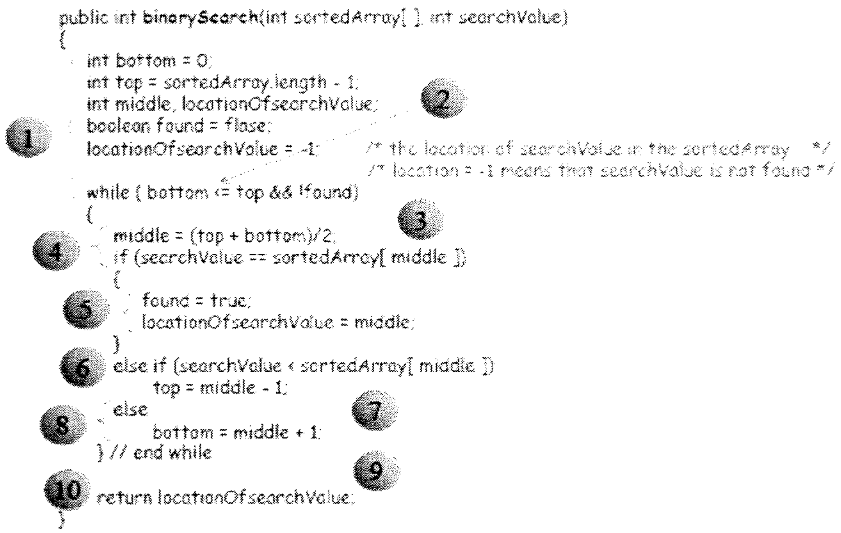
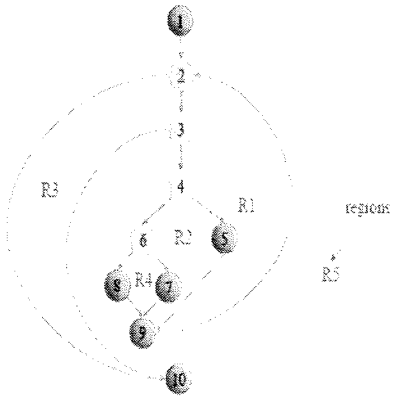
	Track state Visible navigation (any 8 principles)		
6	The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them. Architectural Styles ➢ Data-centered architectures ➢ Data flow architectures ➢ Call and return architectures ➢ Object-oriented architectures ➢ Layered architectures With neat diagram explain any two Architectural Styles	**1 marks for definition , 3 marks for any two** Architectural Styles	**8 min**
7	Testing Strategies for software development  System engineering Analysis modeling Design modeling Code generation Unit test Integration test Validation test System test explanation	**2 marks for** diagram And 2 marks for explanation	**7 min**
8	i. Regression Testing *Regression testing* is the re-execution of some subset of tests that have already been conducted to ensure that changes have not propagated unintended side effects Whenever software is corrected, some aspect of the software configuration (the program, its documentation, or the data that support it) is changed. Regression testing helps to ensure that changes (due to testing or for other reasons) do not introduce unintended behavior or additional errors	2 marks each	**8 min**

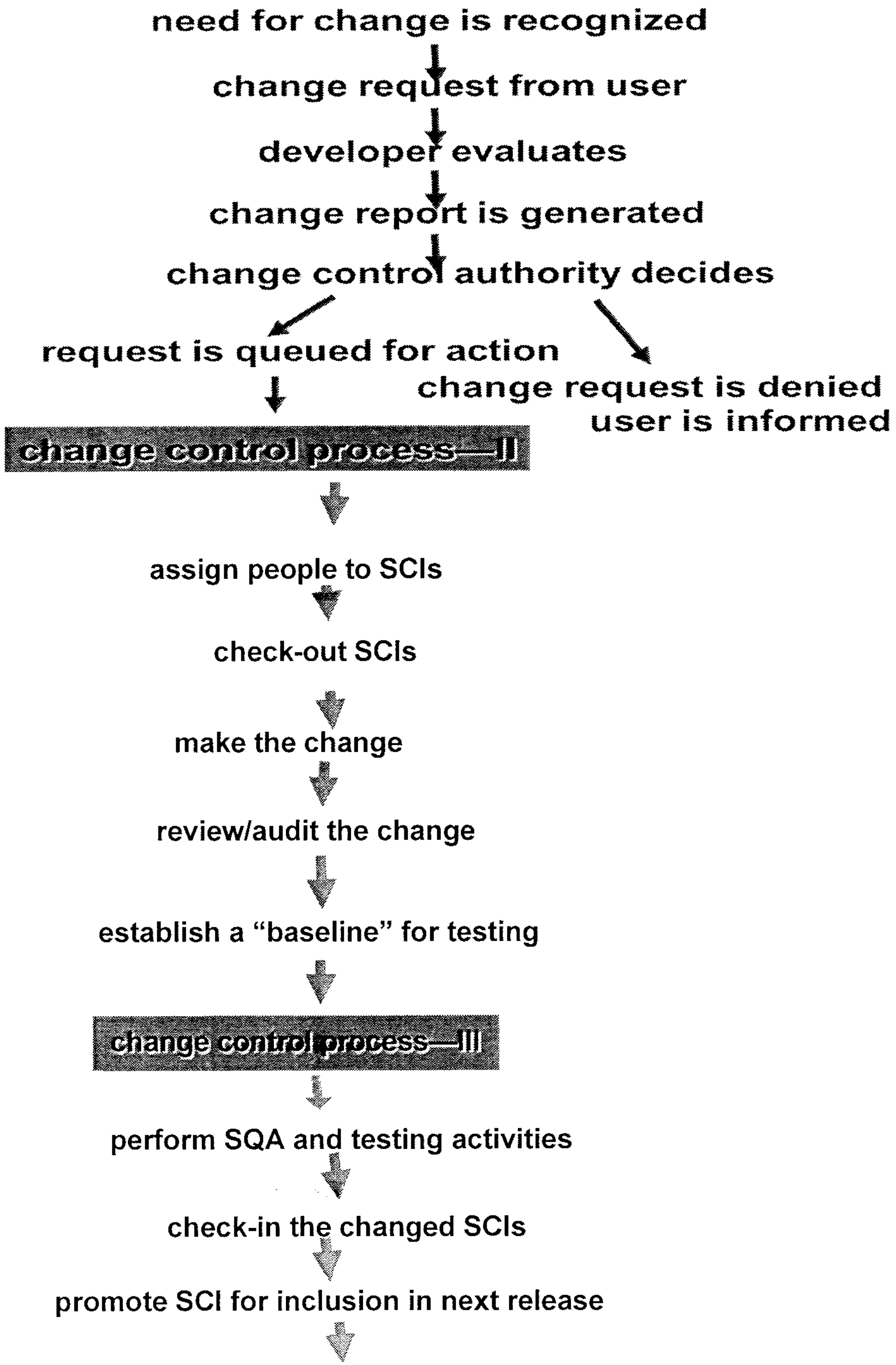
	Smoke testing steps: Software components that have been translated into code are integrated into a “build.” A build includes all data files, libraries, reusable modules, and engineered components that are required to implement one or more product functions. A series of tests is designed to expose errors that will keep the build from properly performing its function. The intent should be to uncover “show stopper” errors that have the highest likelihood of throwing the software project behind schedule. The build is integrated with other builds and the entire product (in its current form) is smoke tested daily. The integration approach may be top down or bottom up.		
9		4 marks	**8 min**

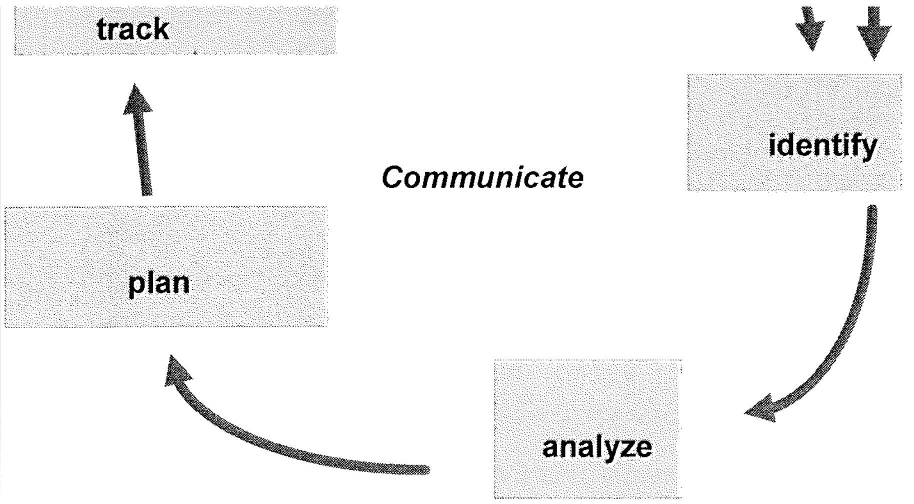
Goal	Attribute	Metric
Requirement quality	Ambiguity	Number of ambiguous modifiers (e.g., many, large, human-friendly)
	Completeness	Number of TBA, TBD
	Understandability	Number of sections/subsections
	Volatility	Number of changes per requirement
		Time (by activity) when change is requested
	Traceability	Number of requirements not traceable to design/code
	Model clarity	Number of UML models
		Number of descriptive pages per model
		Number of UML errors
Design quality	Architectural integrity	Existence of architectural model
	Component completeness	Number of components that trace to architectural model
		Complexity of procedural design
	Interface complexity	Average number of pick to get to a typical function or content
		Layout appropriateness
	Patterns	Number of patterns used
Code quality	Complexity	Cyclomatic complexity
	Maintainability	Design factors (Chapter 8)
	Understandability	Percent internal comments
		Variable naming conventions
	Reusability	Percent reused components
	Documentation	Readability index
QC effectiveness	Resource allocation	Staff hour percentage per activity
	Completion rate	Actual vs. budgeted completion time
	Review effectiveness	See review metrics (Chapter 14)
	Testing effectiveness	Number of errors found and criticality

Part B (03Q x 06M = 18 Marks)

Q No	Solution	Scheme of Marking	Max. Time required for each Question

11	
	public int binarySearch(int sortedArray[ ], int searchValue)
	{
	int bottom = 0;
	int top = sortedArray.length - 1;
	int middle, locationOfsearchValue;
	boolean found = flase;
	locationOfsearchValue = -1; /* the location of searchValue in the sortedArray */
	/* location = -1 means that searchValue is not found */
	while (bottom <= top && !found)
	{
	middle = (top + bottom)/2;
	if (searchValue == sortedArray[middle])
	{
	found = true;
	locationOfsearchValue = middle;
	}
	else if (searchValue < sortedArray[middle])
	top = middle - 1;
	else
	bottom = middle + 1;
	} // end while
	return locationOfsearchValue;
	}
	
	draw the CFG
	Cyclomatic Complexity of Function binarySearch()
	
	 1
	2
	3
	4
	R3
	R1
	regions
	6
	R2
	5
	R5
	8
	R4
	7
	9
	10
	Determine the cyclomatic complexity of the flow graph.
	• V(G) = 5 regions
		2 marks CFG
		2 marks for Compute the cyclomatic complexity number,
		1 mark for finds the Basis Paths Set 1 mark for Design test cases
			15 min

	• Path 5: 1-2-3-4-6-8-9-2-... 4. Prepare test cases that force the execution of each path in the basis set • Path 1 test case: – Inputs: sortedArray = { }, searchValue = 2 – Expected results: locationOfSearchValue = -1		
12	 need for change is recognized change request from user developer evaluates change report is generated change control authority decides request is queued for action change request is denied user is informed change control process—II assign people to SCIs check-out SCIs make the change review/audit the change establish a "baseline" for testing change control process—III perform SQA and testing activities check-in the changed SCIs promote SCI for inclusion in next release	6 marks	15 min

	 Identify: Search for the risks before they create a major problem Analyze: understand the nature , kind of risk and gather information about the risk. Plan: convert them into actions and implement them. Track: we need to monitor the necessary actions. Control: Correct the deviation and make any necessary amendments. Communicate: Discuss about the emerging risks and the current risks and the plans to be undertaken.	**3 marks for** diagram And 3 marks for explanation	

Part C (2Q x 11M = 22 Marks)

Q No	Solution	Scheme of Marking	Max. Time required for each Question
14	all the projects size is 55 KLOC The Basic COCOMO equations have the following form:	Equations 2 marks And each project class computation 3 marks	32 min

where: E is effort in person-months

D is development time in chronological months

the coefficients are given in the table below:

Project class	ab	bb	cb	db
Organic	2.4	1.05	2.5	0.38
semidetached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

$$E = a_b * (\text{ KLOC })\text{exp}(b_b)$$

$$= 3.0 * (55)\text{exp}(1.12)$$

$$= 3.0 * 55^{1.12}$$

$$= 266.887 \text{ [person-months]}$$

$$D = c_b *(E)\text{exp}(d_b)$$

$$= 2.5(266.887)\text{exp}(0.35)$$

$$= 2.5*266.887^{0.35}$$

$$= 17.6 \text{ [months]}$$

The recommended number of people for semidetached project is:

$$\text{N} = \text{E} / \text{D}$$

	N=9
	
	Embedded project class:
	E=441.3
	D=17.54
	N=25.15		

15	**The Forward Pass** Calculation of Earliest Start Date – I Activities A, B and F may start immediately The earliest date for their start is zero Activity A will take 10 weeks The earliest it can finish is week 10 Activity F will take 14 weeks The earliest it can finish is week 14 **The Forward Pass** Calculation of Earliest Start Date – II Activity C can start as soon as A has finished Its earliest start date is week 10 It will take 7 weeks, so the earliest it can finish is week 17 Activities D and E can start as soon as B is complete The earliest they can each start is week 8 Activity D will take 8 weeks, so the earliest it can finish is week 16 Activity E will take 7 weeks, so the earliest it can finish is week 15 **The Forward Pass** Calculation of Earliest Start Date – III Activity G cannot start until both E and F have been completed It cannot start until week 15 - the later of weeks 15 (activity E) and 14 (for activity F) It takes 7 weeks and finishes in week 22 Similarly, activity H cannot start until week 23 – the later of the two earliest finished dates for the preceding activities C and D **The Forward Pass** Calculation of Earliest Start Date – IV The project will be complete when both activities H and G have been completed The earliest project completion date will the later of weeks 22 and 23 – that is, week 23 **The Backward Pass** The Latest Activity Dates Calculation-I The latest completion date for activities G and H is assumed to be week 23 Activity H must therefore start at week 17 at the latest (23-6) and the	**Labelling Convention** Network diagram 5 marks, forward pass 2 marks, backward pass 2 marks and finding Critical Path 2 marks	33 min

The latest start is week 1 (9-8)

The Backward Pass

The Latest Activity Dates Calculation-III

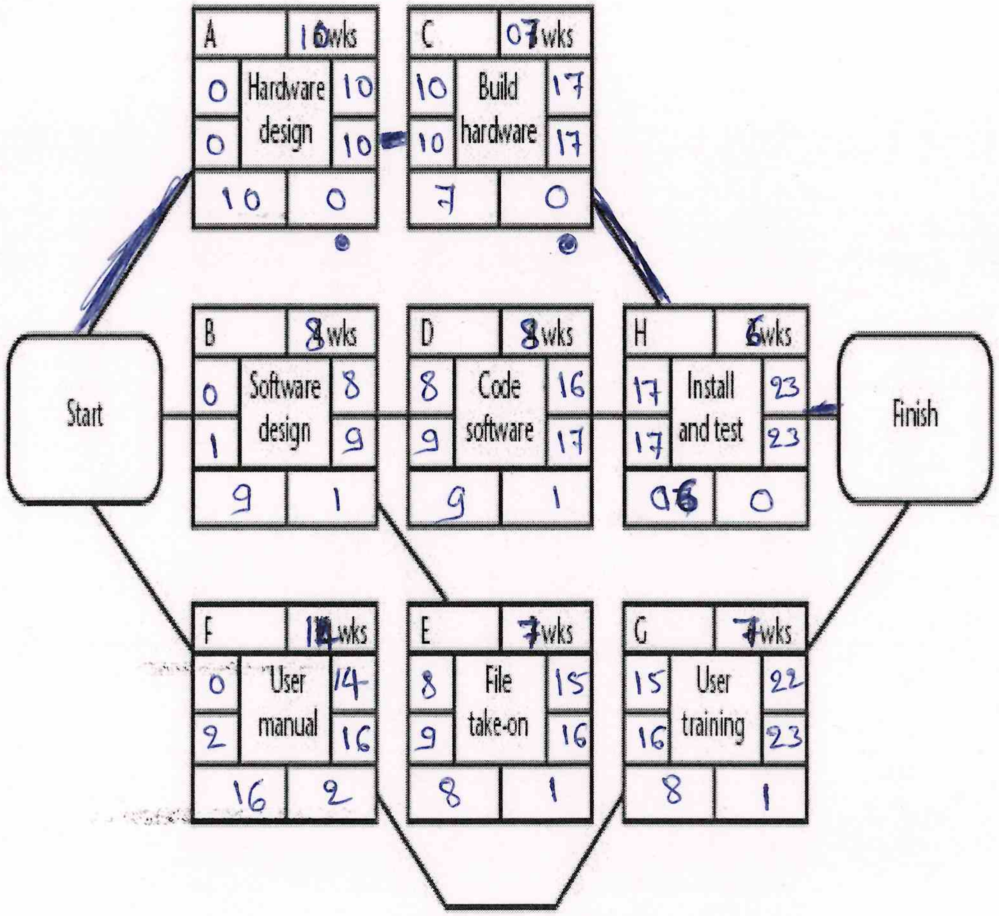
Activity A must be completed by week 10 (the latest start date for activity C)

Its latest start is week 0 (10-10)

The latest start date for the project start is the earliest of the latest start dates for activities A, B and F

This week is week zero

It tells us that if the project does not start on time it won't finish on time



Start → A → C — H → finish