



**St. Joseph College of Engineering**  
**Sriperumbudur, Chennai – 602 117**  
**EVEN SEMESTER (2017 - 2018)**

**Name of the Department : INFORMATION TECHNOLOGY**

**Name of the Faculty : MRS.A.THAVA VINU**

**Year /Semester : II / IV**

**Lesson Plan & Schedule (CS6403 – SOFTWARE ENGINEERING)**

Unit	Task	No of Hours Required	Actual hours Taken, Reason for the deviation	Teaching Method BB/PP/ Video	Book	Dependencies (page no.)	Remarks	No of Students Present
UNIT I (9 hours)	Introduction to Software Engineering, Software Process	1		BB	T1	53-54		
	Waterfall Model, Incremental Model	1		BB	T1	78-82		
	Spiral Model, Prototyping, RAD	1		BB	T1	83-88		
	Formal Methods Model	1		BB	T1	91-94		
	Unified Process Phases	1		BB	T1	95-96		
	LOC and FP Based Estimation	1		BB	R2	41-45		
	COCOMO Model	1		BB	R2	48-53		
	Project Scheduling ,Earned Value Analysis	1		BB	R1	123-127	Assignment-I	
	Risk Management	1		BB	R1	128-135		
	Tutorial (Test)	1	Problems in COCOMO Model					



**St. Joseph College of Engineering**  
**Sriperumbudur, Chennai – 602 117**  
**EVEN SEMESTER (2017 - 2018)**

**Name of the Department : INFORMATION TECHNOLOGY**

**Name of the Faculty : MRS.A.THAVA VINU**

**Year /Semester : II / IV**

**Lesson Plan & Schedule (CS6403 – SOFTWARE ENGINEERING)**

Unit	Task	No of Hours Required	Actual hours Taken, Reason for the deviation	Teaching Method BB/PP/ Video	Book	Dependencies (page no.)	Remarks	No of Students Present
UNIT II (9 hours)	Functional and Non-Functional requirements,	1		BB	R1	143-150		
	User requirements, System requirements	1		BB	R1	151-158		
	Software Requirements Document	1		BB	R1	160-163		
	Requirement Engineering Process: Feasibility Studies	1		BB	T1	176-190		
	Elicitation, Management, Validation	1		BB	T1	191-203		
	Structured system Analysis: Introduction	1		BB	R5	335-339	Assignment-II	
	Requirements of Structured system models	1		BB	R5	340-342		
	Petri Nets	1		BB	R5	353-356		
	Data Dictionary	1		BB	WS1	*		



**St. Joseph College of Engineering**  
**Sriperumbudur, Chennai – 602 117**  
**EVEN SEMESTER (2017 - 2018)**

**Name of the Department : INFORMATION TECHNOLOGY**

**Name of the Faculty : MRS.A.THAVA VINU**

**Year /Semester : II / IV**

**Lesson Plan & Schedule (CS6403 – SOFTWARE ENGINEERING)**

Unit	Task	No of Hours Required	Actual hours Taken, Reason for the deviation	Teaching Method BB/PP/ Video	Book	Dependencies (page no.)	Remarks	No of Students Present
	Tutorial (Test)	1						
					Requirement Engineering Process			
Unit III (9 Hours)	Design process	1		BB	T1	261-264		
	Design Concepts	1		BB	T1	265-271		
	Design Model	1		BB	T1	274-279		
	Design Heuristic	1		BB	WS2	*		
	Architectural Design	1		BB	T1	298-303	Assignment-III	
	Architectural styles	1		BB	T1	291-297		
	Architectural Mapping using Data Flow	1		BB	T1	307-320		
	User Interface Design	1		BB	T1	357-381		



**St. Joseph College of Engineering**  
**Sriperumbudur, Chennai – 602 117**  
**EVEN SEMESTER (2017 - 2018)**

**Name of the Department : INFORMATION TECHNOLOGY**

**Name of the Faculty : MRS.A.THAVA VINU**

**Year /Semester : II / IV**

**Lesson Plan & Schedule (CS6403 – SOFTWARE ENGINEERING)**

Unit	Task	No of Hours Required	Actual hours Taken, Reason for the deviation	Teaching Method BB/PP/ Video	Book	Dependencies (page no.)	Remarks	No of Students Present
	Designing Class based components, traditional Components.	1		BB	WS3	*		
	Tutorial (Test)	1			Architectural styles			
Unit IV (9 Hours)	Software testing fundamentals-Internal and external views of Testing	1		BB	T1	421-423		
	White box testing	1		BB	T1	424-426		
	Basis path testing	1		BB	T1	427-431		
	Control structure testing	1		BB	T1	432-433	Assignment-IV	
	Black box testing	1		BB	T1	434-439		
	Regression Testing ,Unit Testing, Integration Testing	1		BB	T1	386-397		
	System Testing, Validation Testing, Debugging	1		BB	T1	398-416		
	Coding practices	1		BB	WS4	*		
	Refactoring	1		BB	WS5	*		



**St. Joseph College of Engineering**  
**Sriperumbudur, Chennai – 602 117**  
**EVEN SEMESTER (2017 - 2018)**

**Name of the Department : INFORMATION TECHNOLOGY**

**Name of the Faculty : MRS.A.THAVA VINU**

**Year /Semester : II / IV**

**Lesson Plan & Schedule (CS6403 – SOFTWARE ENGINEERING)**

Unit	Task	No of Hours Required	Actual hours Taken, Reason for the deviation	Teaching Method BB/PP/ Video	Book	Dependencies (page no.)	Remarks	No of Students Present
	Tutorial (Test)	1				White box testing- Basis path testing		
Unit V (9 Hours)	LOC and FP Based Estimation	1		BB	R2	41-45		
	Make/Buy Decision	1		BB	WS6	*		
	COCOMO II Model	1		BB	R2	48-50		
	Examples of COCOMO II Model	1		BB	R2	51-53		
	Project Planning	1		BB	R1	120-123		
	RFP Risk Management	1		BB	R1	128-135	Assignment-V	
	Scheduling and Tracking, EVA	1		BB	R1	123-127		
	Process Metrics.	1		BB	T1	650-659		



**St. Joseph College of Engineering**  
**Sriperumbudur, Chennai – 602 117**  
**EVEN SEMESTER (2017 - 2018)**

**Name of the Department : INFORMATION TECHNOLOGY**

**Name of the Faculty : MRS.A.THAVA VINU**

**Year /Semester : II / IV**

**Lesson Plan & Schedule (CS6403 – SOFTWARE ENGINEERING)**

Project Metrics	1		<b>BB</b>	TI	660-668		
Tutorial (Test)	1						Scheduling and Tracking

**(Note: BB= Black Board, PP=Power Point Presentation, VIDEO = Video Lecture, WS=Web Sources)**

**Text Book:**

1. Roger S. Pressman, “Software Engineering – A Practitioner’s Approach”, Seventh Edition, Mc Graw-Hill International Edition, 2010.

**Reference Book:**

1. Ian Sommerville, “Software Engineering”, 9th Edition, Pearson Education Asia, 2011.
2. Rajib Mall, “Fundamentals of Software Engineering”, Third Edition, PHI Learning Private Limited, 2009.
3. Pankaj Jalote, “Software Engineering, A Precise Approach”, Wiley India, 2010.
4. Kelkar S.A., “Software Engineering”, Prentice Hall of India Pvt Ltd, 2007.
5. Stephen R.Schach, “Software Engineering”, Tata McGraw-Hill Publishing Company Limited, 2007.
6. <http://nptel.ac.in/>.



**St. Joseph College of Engineering**  
**Sriperumbudur, Chennai – 602 117**  
**EVEN SEMESTER (2017 - 2018)**

**Name of the Department : INFORMATION TECHNOLOGY**

**Name of the Faculty : MRS.A.THAVA VINU**

**Year /Semester : II / IV**

**Lesson Plan & Schedule (CS6403 – SOFTWARE ENGINEERING)**

**Web sources:**

1. <https://www.tutorialcup.com/dbms/data-dictionary-functions.htm>
2. <https://www.designheuristics.com/>
3. [https://www.tutorialspoint.com/software\\_architecture\\_design/component\\_based\\_architecture.htm](https://www.tutorialspoint.com/software_architecture_design/component_based_architecture.htm)
4. [http://www.tutorialspoint.com/developers\\_best\\_practices/](http://www.tutorialspoint.com/developers_best_practices/)
5. [https://en.wikibooks.org/wiki/Introduction\\_to\\_Software\\_Engineering/Testing/Refactoring](https://en.wikibooks.org/wiki/Introduction_to_Software_Engineering/Testing/Refactoring)
6. [https://www.tutorialspoint.com/management\\_concepts/the\\_make\\_or\\_buy\\_decision.htm](https://www.tutorialspoint.com/management_concepts/the_make_or_buy_decision.htm)

**Signature of the Staff**

**Signature of the HOD**

**Signature of the Principal**



**ST JOSEPH COLLEGE OF ENGINEERING, SRIPERUMBUDUR**  
**DEPARTMENT OF INFORMATION TECHNOLOGY**  
**CS403- SOFTWARE ENGINEERING**  
**QUESTION BANK**

**UNIT I**

**SOFTWARE PROCESS AND PROJECT MANAGEMENT**

**PART A**

**1. What is software engineering?**

Software engineering is a discipline in which theories, methods and tools are applied to develop professional software.

**2. What is Software?**

Software is nothing but a collection of computer programs that are related documents that are indented to provide desired features, functionalities and better performance.

**3. Define software process.**

Software process is defined as the structured set of activities that are required to develop the software system.

**4. What are the fundamental activities of a software process?**

- \* Specification
- \* Design and implementation
- \* Validation
- \* Evolution

**5. What are the umbrella activities of a software process?**

- \* Software project tracking and control.
- \* Risk management.
- \* Software Quality Assurance.
- \* Formal Technical Reviews.
- \* Software Configuration Management.
- \* Work product preparation and production.
- \* Reusability management.
- \* Measurement.

**6. What are the merits of incremental model?**

- i) The incremental model can be adopted when there is less number of people involved in the project.
- ii) Technical risks can be managed with each increment.
- iii) For a very small time span, at least core product can be delivered to the customer.

**7. List the task regions in the Spiral model.**

- \* Customer communication - it is suggested to establish customer communication.
- \* Planning – All planning activities are carried out
- \* Risk analysis – The tasks required to calculate technical and management risks.
- \* Engineering – tasks required to build one or more representations of applications
- \* Construct and release – tasks required to construct, test, install the applications
- \* Customer evaluation - tasks are performed and implemented at installation stage based on the customer evaluation.



**8. What are the drawbacks of spiral model?**

- i) It is based on customer communication. If the communication is not proper then the software product that gets developed will not be the up to the mark.
- ii) It demands considerable risk assessment. If the risk assessment is done properly then only the successful product can be obtained.

**9. What is System Engineering?**

System Engineering means designing, implementing, deploying and operating systems which include hardware, software and people.

**10. List the process maturity levels in SEIs CMM.**

Level 1: Initial - Few processes are defined and individual efforts are taken.

Level 2: Repeatable – To track cost schedule and functionality basic project management processes are established.

Level 3: Defined – The process is standardized, documented and followed.

Level 4: Managed – Both the software process and product are quantitatively understood and controlled using detailed measures.

Level 5: Optimizing – Establish mechanisms to plan and implement change.

**11. What does Verification represent?**

Verification represents the set of activities that are carried out to confirm that the software correctly implements the specific functionality.

**12. What does Validation represent?**

Validation represents the set of activities that ensure that the software that has been built is satisfying the customer requirements.

**13. What are the steps followed in testing?**

- 1) Unit testing - The individual components are tested in this type of testing.
- 2) Module testing – Related collection of independent components are tested.
- 3) Sub-system testing – Various modules are integrated into a subsystem and the whole subsystem is tested.
- 4) System testing – The whole system is tested in this system.
- 5) Acceptance testing – This type of testing involves testing of the system with customer data.

**14. What is the use of CMM?**

Capability Maturity Model is used in assessing how well an organization's processes allow to complete and manage new software projects.

**15. Name the Evolutionary process Models.**

- i. Incremental model
- ii. Spiral model
- iii. WIN-WIN spiral model
- iv. Concurrent Development

**16. What is meant by Software engineering paradigm?**

The development strategy that encompasses the process, methods and tools and generic phases is often referred to as a process model or software engineering paradigm.

**17. What are the various elements for computer based system?**

- |                  |                |
|------------------|----------------|
| 1. Software      | 2. Hardware    |
| 3. People        | 4. Database    |
| 5. Documentation | 5. Procedures. |

**18. Define dynamic verification?**

Dynamic verification is performed during the execution of software and dynamically checks its behavior.

**19. Define static verification?**

Static verification is a process to check some requirements of software doing a physical inspection of it. Example: software metric calculation.

**20. What are the challenges in software?**

- \* Copying with legacy systems.
- \* Heterogeneity challenge
- \* Delivery times challenge.

**21. Define the computer based system.**

The computer based system can be defined as “a set or an arrangement of elements that are organized to accomplish some predefined goal by processing information”.

**PART B**

1. Explain iterative waterfall and spiral model for software life cycle and discuss various activities in each phase. (16)
2. List several software process paradigms. Explain how both waterfall model and prototyping model can be accommodated in the spiral process model. (16)
3. Explain in detail Boehm's spiral model for software life cycle and discuss various activities in each phase. (16)
- 4.1) Which is more important-the product or process? Justify your answer. (4)
- 2 Identify the umbrella activities in software engineering process. (4)
- 3 With suitable illustration explain SPIRAL model evolutionary software development. (8)
5. Explain iterative waterfall and Prototyping model for software life cycle and various activities in each phase. (16)
6. Explain about the generic process model with neat diagram. (16)
7. Explain in detail about the software process paradigms. (16)
8. Explain Spiral model and win-win spiral model in detail? (16)
9. Explain in detail about incremental and Rapid application development Model (RAD) and mention its advantages and disadvantages. (16)
10. Distinguish between verification and validation process. (8)
11. Draw the system engineering hierarchy diagram and explain in detail. (8)
12. Explain the concept of Business process engineering with neat sketch. (16)
13. Explain the concept of Business process engineering with neat sketch. (8)
14. Discuss about the layers of software engineering. (8)

**UNIT – II****REQUIREMENTS ANALYSIS AND SPECIFICATION****PART A****1. What is requirement engineering?**

Requirement engineering is the process of establishing the services that the customer requires from

the system and the constraints under which it operates and is developed.

## 2. What are the various types of traceability in software engineering?

- i. Source traceability – These are basically the links from requirement to stakeholders
- ii. Requirements traceability – These are links between dependant requirements.
- iii. Design traceability – These are links from requirements to design.

## 3. Define software prototyping.

Software prototyping is defined as a rapid software development for validating the requirements.

## 4. What are the benefits of prototyping?

- i. Prototype serves as a basis for deriving system specification.
- ii. Design quality can be improved.
- iii. System can be maintained easily.
- iv. Development efforts may get reduced.
- v. System usability can be improved.

## 5. What are the prototyping approaches in software process?

- i. Evolutionary prototyping – the initial prototype is prepared and it is then refined through number of stages to final stage.
- ii. Throw-away prototyping – a rough practical implementation of the system is produced. The requirement problems can be identified from this implementation.

## 6. What are the advantages of evolutionary prototyping?

- i. Fast delivery of the working system.
- ii. User is involved while developing the system.
- iii. More useful system can be delivered.
- iv. Specification, design and implementation work in co-ordinate manner.

## 7. What are the various Rapid prototyping techniques?

- i. Dynamic high level language development.
- ii. Database programming.
- iii. Component and application assembly.

## 8. What is the use of User Interface prototyping?

This prototyping is used to pre-specify the look and feel of user interface in an effective way.

## 9. What are the characteristics of SRS?

- i. Correct – The SRS should be made up to date when appropriate requirements are identified.
- ii. Unambiguous – When the requirements are correctly understood then only it is possible to write an unambiguous software.
- iii. Complete – To make SRS complete, it should be specified what a software designer wants to create software.
- iv. Consistent – It should be consistent with reference to the functionalities identified.
- v. Specific – The requirements should be mentioned specifically.
- vi. Traceable – What is the need for mentioned requirement?

## 10. What is data modeling?

Data modeling is the basic step in the analysis modeling. In data modeling the data objects are examined independently of processing. The data model represents how data are related with one another.

**11. What is a data object?**

Data object is a collection of attributes that act as an aspect, characteristic, quality, or descriptor of the object.

**12. What are attributes?**

Attributes are the one, which defines the properties of data object.

**13. What is cardinality in data modeling?**

Cardinality in data modeling, cardinality specifies how the number of occurrences of one object is related to the number of occurrences of another object.

**14. What does modality in data modeling indicates?**

Modality indicates whether or not a particular data object must participate in the relationship.

**15. What is ERD?**

Entity Relationship Diagram is the graphical representation of the object relationship pair. It is mainly used in database applications.

**16. What is DFD?**

Data Flow Diagram depicts the information flow and the transforms that are applied on the data as it moves from input to output.

**17. What does Level0 DFD represent?**

Level 0 DFD is called as „fundamental system model“ or „context model“. In the context model the entire software system is represented by a single bubble with input and output indicated by incoming and outgoing arrows.

**18. What is a state transition diagram?**

State transition diagram is basically a collection of states and events. The events cause the system to change its state. It also represents what actions are to be taken on the occurrence of particular event.

**19. Define Data Dictionary.**

The data dictionary can be defined as an organized collection of all the data elements of the system with precise and rigorous definitions so that user and system analyst will have a common understanding of inputs, outputs, components of stores and intermediate calculations.

**20. What are the elements of Analysis model?**

- i. Data Dictionary
- ii. Entity Relationship Diagram
- iii. Data Flow Diagram
- iv. State Transition Diagram
- v. Control Specification
- vi. Process specification.

**21. What are functional requirements?**

Functional requirements are” statements of services the system should provide how the system should react to particular input and how the system should behave in particular situation.

**22. What are non functional requirements?**

Non functional requirements are constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.....

**PART B**

1. Explain in detail about Functional Modeling. (16)
2. Explain in detail about Structural Modeling and data modeling. (16)
3. Explain in detail about Rapid prototyping techniques . (16)
4. Explain the prototyping model. How do we select appropriate Prototyping approach? (16)
5. Explain the prototyping approaches in software process. (16)
6. Explain briefly about functional and non functional requirements . (16)
7. Explain the concept of Requirements engineering process. (16)
8. Discuss about a) Elicitation (4) b) Validation (4) c) Management (4) d) Documents. (4)
9. How do you prepare the Software documents for the following applications a. Bank management systems. (8) b. Railway Ticket Reservation. (8)

**UNIT – III**  
**SOFTWARE DESIGN**  
**PART A**

**1. What are the elements of design model?**

- i. Data design
- ii. Architectural design
- iii. Interface design
- iv. Component-level design

**2. Define design process.**

Design process is a sequence of steps carried through which the requirements are translated into a system or software model.

**3. List the principles of a software design.**

- i. The design process should not suffer from “tunnel vision”
- ii. The design should be traceable to the analysis model.
- iii. The design should exhibit uniformity and integration.
- iv. Design is not coding.
- v. The design should not reinvent the wheel.

**4. What is the benefit of modular design?**

Changes made during testing and maintenance becomes manageable and they do not affect other modules.

**5. What is a cohesive module?**

A cohesive module performs only “one task” in software procedure with little interaction with other modules. In other words cohesive module performs only one thing.

**6. What are the different types of Cohesion?**

- i. Coincidentally cohesive - The modules in which the set of tasks are related with each other loosely.
- ii. Logically cohesive – A module that performs the tasks that are logically related with each other.
- iii. Temporal cohesion – The module in which the tasks need to be executed in some specific time span.
- iv. Procedural cohesion – When processing elements of a module are related with one another and must be executed in some specific order.

v. Communicational cohesion – When the processing elements of a module share the data then such module is called communicational cohesive.

### **7. What is coupling?**

Coupling is the measure of interconnection among modules in a program structure. It depends on the interface complexity between modules.

### **8. What are the various types of coupling?**

i. Data coupling – The data coupling is possible by parameter passing or data interaction.

ii. Control coupling – The modules share related control data in control coupling.

iii. Common coupling – The common data or a global data is shared among modules.

iv. Content coupling – Content coupling occurs when one module makes use of data or control information maintained in another module.

### **9. What are the common activities in design process?**

System structuring – The system is subdivided into principle subsystems components and communications between these subsystems are identified.

Control modeling – A model of control relationships between different parts of the system is established.

Modular decomposition – The identified subsystems are decomposed into modules.

### **10. What are the benefits of horizontal partitioning?**

Software that is easy to test.

Software that is easier to maintain.

Propagation of fewer side effects.

Software that is easier to extend.

### **11. What is vertical partitioning?**

Vertical partitioning often called factoring suggests that the control and work should be distributed top-down in program structure.

### **12. What are the advantages of vertical partitioning?**

i. These are easy to maintain changes.

ii. They reduce the change impact and error propagation.

### **13. What are the various elements of data design?**

i. Data object – The data objects are identified and relationship among various data objects can be represented using ERD or data dictionaries.

ii. Databases – Using software design model, the data models are translated into data structures and data bases at the application level.

iii. Data warehouses – At the business level useful information is identified from various databases and the data warehouses are created.

### **14. List the guidelines for data design.**

i. Apply systematic analysis on data.

ii. Identify data structures and related operations.

iii. Establish data dictionary.

iv. Use information hiding in the design of data structure.

v. Apply a library of useful data structures and operations.

### **15. Name the commonly used architectural styles.**

- i. Data centered architecture.
- ii. Data flow architecture.
- iii. Call and return architecture.
- iv. Object-oriented architecture.
  
- v. Layered architecture.

**16. What is Transform mapping?**

The transform mapping is a set of design steps applied on the DFD in order to map the transformed flow characteristics into specific architectural style.

**17. What is a Real time system?**

Real time system is a software system in which the correct functionalities of the system are dependent upon results produced by the system and the time at which these results are produced.

**18. What are the objectives of Analysis modeling?**

- i. To describe what the customer requires.
- ii. To establish a basis for the creation of software design.
- iii. To devise a set of valid requirements after which the software can be built.

**19. What is an Architectural design?**

The architectural design defines the relationship between major structural elements of the software, the “design patterns” that can be used to achieve the requirements that have been defined for the system.

**20. What is data design?**

The data design transforms the information domain model created during analysis into the data structures that will be required to implement the software.

**21. What is interface design?**

The interface design describes how the software communicates within itself, with systems that interoperate with it, and with humans who use it.

**22. What is component level design?**

The component level design transforms structural elements of the software architecture into a procedural description of software components.

**23. What is software design?**

Software design is an iterative process through which the requirements are translated into a “blueprint” for constructing the software.

**24. What is user interface design?**

User interface design creates an effective communication medium between a human and a computer.

**25. What is system design?**

System design process involves deciding which system capabilities are to be implemented in software and which in hardware.

**PART A**

- 1 a. Explain in detail about the user interface design activities. (8)
- b. Explain data, architectural and procedural design principles for the software. (8)

3. Describe the design procedures for the data acquisition system. (16)
4. Explain in detail about the real time systems. (16)
5. Explain in detail about Software Configuration Management (SCM). (16)
6. What are the different types of architectural styles exist in software and Explain any one software architecture in detail. (16)
7. a. Explain how the interrupts handled in real time systems? (8)
- b. Explain the interface design activities. What steps do we perform to accomplish Interface design? (8)
7. Describe Transform and Transactional mapping by applying design steps to an example system. (16)
8. a. Explain the two qualitative criteria – Coupling and Cohesion. (8)
- b. Explain what are the design considerations for real time systems (8)
9. Explain about the component level design .
10. Describe about the designing class based components & traditional components.
11. Write elaborates about architectural design.

### UNIT – IV

### TESTING AND IMPLEMENTATION

### PART A

#### 1. Define software testing?

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and coding.

#### 2. What are the objectives of testing?

Testing is a process of executing a program with the intend of finding an error.

A good test case is one that has high probability of finding an undiscovered error.

A successful test is one that uncovers as an-yet undiscovered error.

#### 3. What are the testing principles the software engineer must apply while performing the software testing?

1. All tests should be traceable to customer requirements.
2. Tests should be planned long before testing begins.
3. The pareto principle can be applied to software testing-80% of all errors uncovered during testing will likely be traceable to 20% of all program modules.
4. Testing should begin “in the small” and progress toward testing “in the large”.
5. Exhaustive testing is not possible.
6. To be most effective, an independent third party should conduct testing.

#### 4. What are the two levels of testing?

Component testing - Individual components are tested. Tests are derived from developer’s experience.

System Testing - The group of components are integrated to create a system or sub-system is done. These tests are based on the system specification.



**5. What are the various testing activities?**

Test planning  
 Test case design  
 Test execution  
 Data collection  
 Effective evaluation

**6. Write short note on black box testing.**

The black box testing is also called as behavioral testing. This method fully focus on the functional requirements of the software. Tests are derived that fully exercise all functional requirements.

**7. What is equivalence partitioning?**

Equivalence partitioning is a black box technique that divides the input domain into classes of data. From this data test cases can be derived. Equivalence class represents a set of valid or invalid states for input conditions.

**8. What is a boundary value analysis?**

A boundary value analysis is a testing technique in which the elements at the edge of the domain are selected and tested. It is a test case design technique that complements equivalence partitioning technique.

**9. What are the reasons behind to perform white box testing?**

There are three main reasons behind performing the white box testing.

1. Programmers may have some incorrect assumptions while designing or implementing some functions.
2. Certain assumptions on flow of control and data may lead programmer to make design errors. To uncover the errors on logical path, white box testing is must.
3. There may be certain typographical errors that remain undetected even after syntax and type checking mechanisms. Such errors can be uncovered during white box testing.

**12. What is cyclomatic complexity?**

Cyclomatic complexity is software metric that gives the quantitative Measure of logical complexity of the program.

**11. How to compute the cyclomatic complexity?**

The cyclomatic complexity can be computed by any one of the following ways. 1. The numbers of regions of the flow graph correspond to the cyclomatic complexity.

2. Cyclomatic complexity (G), for the flow graph G, is defined as:  $V(G)=E-N+2$ , E -- number of flow graph edges, N -- number of flow graph nodes
3.  $V(G) = P+1$  Where P is the number of predicate nodes contained in the flow graph.

**12. Distinguish between verification and validation.**

Verification refers to the set of activities that ensure that software correctly implements a specific function.

Validation refers to a different set of activities that ensure that the software that has been built is traceable to the customer requirements.

**13. What are the various testing strategies for conventional software?**

- i. Unit testing
- ii. Integration testing.

- iii. Validation testing.
- iv. System testing.

#### 14. Write about drivers and stubs.

Drivers and stub software need to be developed to test incompatible software.

The “driver” is a program that accepts the test data and prints the relevant results.

The “stub” is a subprogram that uses the module interfaces and performs the minimal data manipulation if required.

#### 15. What are the approaches of integration testing?

The integration testing can be carried out using two approaches.

1. The non-incremental testing.
2. Incremental testing.

#### 16. What are the advantages and disadvantages of big-bang?

Advantage:

This approach is simple.

Disadvantages:

It is hard to debug.

It is not easy to isolate errors while testing.

In this approach it is not easy to validate test results.

#### 17. What are the benefits of smoke testing?

- \* Integration risk is minimized.
- \* The quality of the end-product is improved.
- \* Error diagnosis and correction are simplified.
- \* Assessment of program is easy.

#### 18. What are the conditions exists after performing validation testing?

- \* The function or performance characteristics are according to the specifications and are accepted.
- \* The requirement specifications are derived and the deficiency list is created.

#### 19. Distinguish between alpha and beta testing

Alpha and beta testing are the types of acceptance testing.

Alpha test : The alpha testing is attesting in which the version of complete software is tested by the customer under the supervision of developer. This testing is performed at developer’s site.

Beta test : The beta testing is a testing in which the version of the software is tested by the customer without the developer being present. This testing is performed at customer’s site.

#### 20. What are the various types of system testing?

1. Recovery testing – is intended to check the system’s ability to recover from failures.
2. Security testing – verifies that system protection mechanism prevent improper penetration or data alteration.
3. Stress testing – Determines breakpoint of a system to establish maximum service level.
4. Performance testing – evaluates the run time performance of the software, especially real-time software.

#### 21. Define debugging.

Debugging is defined as the process of removal of defect. It occurs as a consequence of successful testing.

**PART -B**

- 1.Explain the various types of software testing strategies. (16)
- 2.Explain in detail about Black box testing and white box testing. (16)
- 3.Explain the Concept of unit testing testing strategy in detail. (16)
4. Explain in detail about Integration testing with neat diagram. (8) b. Explain in detail about system testing. (8)
- 5.How to derive the test cases for the given project? Explain in detail (16)
- 6.Why testing is important? Narrate the path testing procedure in detail with a sample code. (16)
- 7.What are all the formulas available for calculating Cyclomatic complexity?
- 8.Calculate cyclomatic complexity for greatest of three numbers. (16)
9. Explain in detail about the different types of Software Measures. (8)
- 10.Explain in detail about the test coverage criteria based on the data flow mechanisms. (8)
- 11.Explain about regression testing with real time example.
- 12.Discuss about software implementation technique & coding practices.

**UNIT – V****PROJECT MANAGEMENT****PART A****1. Define measure.**

Measure is defined as a quantitative indication of the extent, amount, dimension, or size of some attribute of a product or process.

**2. Define metrics.**

Metrics is defined as the degree to which a system component, or process possesses a given attribute.

**3. What are the types of metrics?**

Direct metrics – It refers to immediately measurable attributes.

Example – Lines of code, execution speed.

Indirect metrics – It refers to the aspects that are not immediately quantifiable or measurable.

Example – functionality of a program.

**4. Write short note on the various estimation techniques.**

Algorithmic cost modeling – the cost estimation is based on the size of the software.

Expert judgment – The experts from software development and the application domain.

Estimation by analogy – The cost of a project is computed by comparing the project to a similar project in the same application domain.

Parkinson's Law – The cost is determined by available resources rather than by objective assessment.

Pricing to win – The project costs whatever the customer ready to spend it.

**5. What is COCOMO model?**

Constructive COSt MOdel is a cost model, which gives the estimate of number of man-months it will take to develop the software product.

**6. Give the procedure of the Delphi method.**

1. The co-coordinator presents a specification and estimation form to each expert.

2. Co-coordinator calls a group meeting in which the experts discuss estimation issues with the coordinator and each other.
3. Experts fill out forms anonymously.
4. Co-coordinator prepares and distributes a summary of the estimates.
5. The Co-coordinator then calls a group meeting.

### **7. What is the purpose of timeline chart?**

The purpose of the timeline chart is to emphasize the scope of the individual task. Hence set of tasks are given as input to the timeline chart.

### **8. What is EVA?**

Earned Value Analysis is a technique of performing quantitative analysis of the software project. It provides a common value scale for every task of software project. It acts as a measure for software project progress.

### **9. What are the metrics computed during error tracking activity?**

Errors per requirement specification page.  
Errors per component-design level  
Errors per component-code level  
DRE-requirement analysis  
DRE-architectural analysis  
DRE-component level design  
DRE-coding.

### **10. What is software maintenance?**

Software maintenance is an activity in which program is modified after it has been put into use.

### **11. Define maintenance.**

Maintenance is defined as the process in which changes are implemented By either modifying the existing system's architecture or by adding new components to the system.

### **12. What are the types of software maintenance?**

Corrective maintenance – Means the maintenance for correcting the software faults.

Adaptive maintenance – Means maintenance for adapting the change in environment. Perfective

maintenance – Means modifying or enhancing the system to meet the new requirements.

Preventive maintenance – Means changes made to improve future maintainability.

### **13. How the CASE tools are classified?**

CASE tools can be classified by a. By function or use

b. By user type (e.g. manager, tester), or

c. By stage in software engineering process (e.g. requirements, test).

### **14. What are the types of static testing tools?**

There are three types of static testing tools.

Code based testing tools – These tools take source code as input and generate test cases.

Specialized testing tools – Using this language the detailed test specification can be written for each test case.

Requirement-based testing tools – These tools help in designing the test cases as per user

requirements.

**15. What is meant by Software project management?**

Software project management is an activity of organizing, planning and scheduling software projects.

**16. What is meant by software measurement?**

Software measurement means deriving a numeric value for an attribute of a software product or process.

**17. What is meant by software cost estimation?**

The software cost estimation is the process of predicting the resources required for software development process.

**18. What is meant by CASE tools?**

The computer aided software engineering tools automatic the project management activities, manage all the work products. The CASE tools assist to perform various activities such as analysis, design, coding and testing.

**19. What is meant by Delphi method?**

The Delphi technique is an estimation technique intended to active a common agreement for estimation efforts.

**20. What is meant by software evolution?**

Software evolution is a process of managing the changes in the software.

**21. What is software configuration management (SCM)?**

Software configuration management is the art of identifying, organizing, and controlling modifications to the software being built by a programming team.

**PART -B**

1. Explain the need for software measures and describe various metrics. (16)
2. Explain in detail about the various levels of Capability Maturity Model(CMM) (16)
3. Explain the salient features of COCOMO model for cost estimation .Bring out Different variant of this model. (16)
4. Explain in detail about Delphi Method. (16)
- 5.a. What are the different types of software Maintenance activities envisaged for an installed software . (8)
- b. Describe in detail about the standardization procedures of benchmarking (8) 6. Explain the features of a popular CASE tools that u are aware of. (16)
- 7.a. Give a detail note on SPICE . (8)
- b. Write Detail notes on IS9000 series of quality management standards. (8) 8.a. What is an activity network? Explain. (8)
- 8.Explain the scheduling of a software project.How are PERT and CPM useful in project scheduling? (
9. Discuss about the relationship between people and effort ,task set network. 10.Explain about the EVA (Earned Value Analysis) and project metrics.

# SUCCE