MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

Estd. 2000

IQAC

MUST KNOW CONCEPTS

MKC

CSE

2020-21

Course Code & Course Name : 19CSC12/SOFTWARE ENGINEERING

Year/Sem/Sec : II/IV/A & B

S.No.	Term	Notation (Symbol)	Concept / Definition / Meaning / Units / Equation / Expression	Units
UNIT I : SOFTWARE PROCESS AND PROJECT MANAGEMENT				
1.	**Software**		Software is a program that enables a computer to perform a specific task, as opposed to the physical components of the system	
2.	**Software Engineering**		Software engineering is the systematic application of engineering approaches to the development of software.	
3.	**Software Process**		Software process (also known as software methodology) is a set of related activities that leads to the production of the software.	
4.	**Waterfall Model**		The waterfall model is a sequential approach, where each fundamental activity of a process represented as a separate phase, arranged in linear order.	
5.	**Prototype**		A prototype is useful when a customer or developer is not sure of the requirements, or of algorithms, efficiency, business rules, response time, etc.	
6.	**Spiral Model**		The spiral model is a risk-driven where the process is represented as spiral rather than a sequence of activities.	
7.	**Framework Activities**		1. Communication 2. Planning 3. Modelling 4. Construction 5. Deployment	
8.	**Types of prescriptive process models**		1. The Waterfall Model 2. Incremental Process model 3. Evolutionary Process Models 4. Concurrent Models	
9.	**Incremental Process Model**		The incremental model combines the elements of waterfall model and they are applied in an iterative fashion.	
10.	**Evolutionary Model**		The Evolutionary development model divides the development cycle into smaller, incremental waterfall models in which users are able to get access to the product at the end of each cycle.	
11.	**Concurrent Process Model**		The concurrent process model defines a series of events that will trigger transitions from state to state for each of the software engineering activities.	
12.	**Special process models**		Special process models take many features from one or more conventional models.	
13.	**Component based development**		The component based development model incorporates many of the characteristics of the spiral model.	

14.	**Aspect oriented Software Development**		Aspect Oriented Software Development (AOSD) often referred to as aspect oriented programming (AOP), a relatively new paradigm that provides process and methodology for defining, specifying designing and constructing aspects.	
15.	**Software Project Management**		Software Project Management (SPM) is a proper way of planning and leading software projects. It is a part of project management in which software projects are planned, implemented, monitored and controlled.	
16.	**Management Spectrum**		Effective software project management focuses on the four P's: people, product, process, and project.	
17.	**Project estimation Types**		Software size estimation Effort estimation Time estimation Cost estimation	
18.	**Project size estimation techniques**		Lines of Code Number of entities in ER diagram Total number of processes in detailed data flow diagram Function points	
19.	**Lines of Code (LOC):**		As the name suggest, LOC count the total number of lines of source code in a project.	
20.	**units of LOC**		KLOC- Thousand lines of code NLOC- Non comment lines of code KDSI- Thousands of delivered source instruction	
21.	**Function Point Analysis:**		In this method, the number and type of functions supported by the software are utilized to find FPC(function point count).	
22.	**COCOMO Model**		Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality.	
23.	**COCOMO Model key parameters**		Effort & Schedule	
24.	**Types of COCOMO Models**		Basic COCOMO Model Intermediate COCOMO Model Detailed COCOMO Model	
25.	**Project-task scheduling**		Project-task scheduling is a significant project planning activity. It comprises deciding which functions would be taken up when.	
UNIT II : REQUIREMENTS ANALYSIS AND SPECIFICATION				
26.	**Requirement Engineering**		The process to gather the software requirements from client, analyze and document them is known as requirement engineering.	
27.	**Software requirements**		Software requirements is a field within software engineering that deals with establishing the needs of stakeholders that are to be solved by software	
28.	**User requirements**		User requirements, often referred to as user needs, describe what the user does with the system, such as what activities that users must be able to perform.	
29.	**System requirements**		System requirements are the configuration that a system must have in order for a hardware or software application to run smoothly and efficiently.	
30.	**Hardware system requirements**		Hardware system requirements often specify the operating system version, processor type, memory size, available disk space and additional peripherals, if any, needed.	

31.	**Software system requirements**		Software system requirements, in addition to the aforementioned requirements, may also specify additional software dependencies (e.g., libraries, driver version, framework version).	
32.	**System Requirement Document (SRD)**		The System Requirement Document (SRD) defines system level functional and performance requirements for a system. It should include a system level description of all software elements required by the preferred system concept.	
33.	**Requirement Engineering Process**		Feasibility Study Requirement Elicitation and Analysis Software Requirement Specification Software Requirement Validation Software Requirement Management	
34.	**Types of Feasibility**		Technical Feasibility Operational Feasibility Economic Feasibility	
35.	**Problems of Elicitation and Analysis**		Getting all, and only, the right people involved. Stakeholders often don't know what they want Stakeholders express requirements in their terms. Stakeholders may have conflicting requirements.	
36.	**Software requirement specification**		Software requirement specification is a kind of document which is created by a software analyst after the requirements collected from the various sources - the requirement received by the customer written in ordinary language.	
37.	**Software Requirement Management**		Requirement management is the process of managing changing requirements during the requirements engineering process and system development.	
38.	**Classical Analysis**		The evaluation of an activity to identify its desired objectives and determine procedures for efficiently attaining them.	
39.	**Structured Analysis Tools**		Data Flow Diagrams Data Dictionary Decision Trees Decision Tables Structured English Pseudocode	
40.	**Data Flow Diagrams**		DFD is easy to understand and quite effective when the required design is not clear and the user wants a notational language for communication.	
41.	**Context Diagram**		A context diagram helps in understanding the entire system by one DFD which gives the overview of a system.	
42.	**Data Dictionary**		A data dictionary is a structured repository of data elements in the system. It stores the descriptions of all DFD data elements that is, details and definitions of data flows, data stores, data stored in data stores, and the processes.	
43.	**Decision Trees**		Decision trees are a method for defining complex relationships by describing decisions and avoiding the problems in communication.	
44.	**Decision Tables**		Decision tables are a method of describing the complex logical relationship in a precise manner which is easily understandable.	
45.	**Structured English**		Structure English is derived from structured programming language which gives more understandable and precise description of process.	

46.	**Pseudocode**		A pseudocode does not conform to any programming language and expresses logic in plain English.	
47.	**Technical Feasibility**		Technical feasibility evaluates the current technologies, which are needed to accomplish customer requirements within the time and budget.	
48.	**Functional Requirement**		A functional requirement defines a system or its component. It describes the functions software must perform.	
49.	**Non-Functional Requirement**		A non-functional requirement is essential to ensure the usability and effectiveness of the entire software system. Failing to meet non-functional requirements can result in systems that fail to satisfy user needs.	
50.	**E-R diagram**		It is a detailed logical representation of the data for the organization and uses three main constructs i.e. data entities, relationships, and their associated attributes.	
UNIT III : SOFTWARE DESIGN				
51.	**Software design**		Software design is a process to transform user requirements into some suitable form, which helps the programmer in software coding and implementation.	
52.	**Objectives of Software Design**		Correctness: Efficiency: Understandability Completeness: Maintainability	
53.	**Levels Of Phases Of Design**		Interface Design Architectural Design Detailed Design	
54.	**Interface design**		Interface design is the specification of the interaction between a system and its environment. This phase proceeds at a high level of abstraction with respect to the inner workings of the system.	
55.	**Architectural design**		Architectural design is the specification of the major components of a system, their responsibilities, properties, interfaces, and the relationships and interactions between them.	
56.	**Issues in architectural design**		Gross decomposition of the systems into major components. Allocation of functional responsibilities to components. Component Interfaces	
57.	**Detailed Design**		Design is the specification of the internal elements of all major system components, their properties, relationships, processing, and often their algorithms and the data structures.	
58.	**Software Design Concepts**		The software design concept simply means the idea or principle behind the design. It describes how you plan to solve the problem of designing software, the logic, or thinking behind how you will design software.	
59.	**Design Heuristic**		Heuristics refers to a non-optimal solution for experience-based techniques to solve problems, learning, and discovery. The main goal of heuristic evaluations is to identify any problems associated with the design of user interfaces.	
60.	**Architectural design**		Architectural design is a process for identifying the sub-systems making up a system and the framework for sub-system control and communication. The output of this design process is a description of the software architecture.	

61.	**Architectural styles**		The software that is built for computer-based systems can exhibit one of these many architectural styles.	
62.	**Data flow architectures**		This kind of architecture is used when input data to be transformed into output data through a series of computational manipulative components.	
63.	**User Interface Design**		User interface is the front-end application view to which user interacts in order to use the software.	
64.	**Command Line Interface**		Command Line Interface provides a command prompt, where the user types the command and feeds to the system.	
65.	**Graphical User Interface**		Graphical User Interface provides the simple interactive interface to interact with the system.	
66.	**Abstraction**		Abstraction simply means to hide the details to reduce complexity and increases efficiency or quality.	
67.	**Modularity**		Modularity in design means to subdivide a system into smaller parts so that these parts can be created independently and then use these parts in different systems to perform different functions.	
68.	**Refinement**		Refinement simply means to refine something to remove any impurities if present and increase the quality.	
69.	**Pattern**		The pattern simply means a repeated form or design in which the same shape is repeated several times to form a pattern.	
70.	**Refactoring**		Refactoring simply means to reconstruct something in such a way that it does not affect the behaviour or any other features.	
71.	**Two levels of abstraction**		Architecture in the small Architecture in the large	
72.	**User Interface Golden rules**		Strive for consistency - Consistent sequences of actions should be required in similar situations. Identical terminology should be used in prompts, menus, and help screens. Consistent commands should be employed throughout.	
73.	**Traditional Components**		Traditional components are designed based on different constructs like. Sequence implements processing steps that are essential in the specification of any algorithm.	
74.	**Interface Validation**		This phase focuses on testing the interface. The interface should be in such a way that it should be able to perform tasks correctly and it should be able to handle a variety of tasks.	
75.	**Information Hiding**		Information hiding simply means to hide the information so that it cannot be accessed by an unwanted party.	
Unit-IV : TESTING AND IMPLEMENTATION				
76.	**Software Testing**		Software Testing is vital for any software development life cycle, it is fundamental to ensure the software quality and to have a workable functional software at the end of the project.	
77.	**White-box testing**		It is conducted to test program and its implementation, in order to improve code efficiency or structure. It is also known as 'Structural' testing.	
78.	**White-box testing techniques**		Control-flow testing Data-flow testing	
79.	**Basic Path Testing**		Path Testing is a method that is used to design the test cases. In path testing method, the control flow graph of a program is designed to find a set of linearly independent paths of execution.	

80.	**Advantages of Path Testing**		Path testing method reduces the redundant tests. Path testing focuses on the logic of the programs. Path testing is used in test case design.	
81.	**Control structure testing**		Control structure testing is used to increase the coverage area by testing various control structures present in the program.	
82.	**Condition Testing**		Condition testing is a test cased design method, which ensures that the logical condition and decision statements are free from errors.	
83.	**Loop Testing**		Loop testing is actually a white box testing technique. It specifically focuses on the validity of loop construction.	
84.	**Concatenated Loops**		If loops are not dependent on each other, contact loops can be tested using the approach used in simple loops. if the loops are interdependent, the steps are followed in nested loops	
85.	**Black box testing**		Black box testing is a type of software testing in which the functionality of the software is not known. The testing is done without the internal knowledge of the products.	
86.	**Regression Testing**		Regression Testing is the process of testing the modified parts of the code and the parts that might get affected due to the modifications to ensure that no new errors have been introduced in the software after the modifications have been made.	
87.	**Advantages of Regression Testing**		It ensures that no new bugs have been introduced after adding new functionalities to the system.	
88.	**Disadvantages of Regression Testing**		It can be time and resource consuming if automated tools are not used. It is required even after very small changes in the code.	
89.	**Unit testing**		Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules.	
90.	**Integration testing**		Integration testing is the process of testing the interface between two software units or module. It's focus on determining the correctness of the interface.	
91.	**Bottom-Up Integration Testing**		In bottom-up testing, each module at lower levels is tested with higher modules until all modules are tested.	
92.	**Top-down integration testing**		Top-down integration testing technique used in order to simulate the behaviour of the lower-level modules that are not yet integrated.	
93.	**System Testing**		System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements.	
94.	**Performance Testing**		Performance Testing is a type of software testing that is carried out to test the speed, scalability, stability and reliability of the software product or application.	
95.	**Top-Down Integration Testing Advantages**		Separately debugged module. Few or no drivers needed. It is more stable and accurate at the aggregate level	
96.	**Top-Down Integration Testing Disadvantages**		Needs many Stubs. Modules at lower level are tested inadequately.	

97.	**Mixed Integration Testing Disadvantages**		For mixed integration testing, require very high cost because one part has Top-down approach while another part has bottom-up approach.	
98.	**Load Testing**		Load Testing is a type of software Testing which is carried out to determine the behavior of a system or software product under extreme load.	
99.	**Stress Testing**		Stress Testing is a type of software testing performed to check the robustness of the system under the varying loads.	
100.	**Scalability Testing**		Scalability Testing is a type of software testing which is carried out to check the performance of a software application or system in terms of its capability to scale up or scale down the number of user request load.	
Unit-V : PROJECT MANAGEMENT				
101.	**Project management**		Project management is the application of processes, methods, skills, knowledge and experience to achieve specific project objectives according to the project acceptance criteria within agreed parameters.	
102.	**Make or buy decision**		Make or buy decision is always a valid concept in business. No organization should attempt to make something by their own, when they stand the opportunity to buy the same for much less price.	
103.	**Reasons for Making**		Cost concerns Desire to expand the manufacturing focus Need of direct control over the product	
104.	**COCOMO II Model**		COCOMO-II is the revised version of the original Cocomo (Constructive Cost Model) and is developed at University of Southern California. It is the model that allows one to estimate the cost, effort and schedule when planning a new software development activity.	
105.	**Project planning**		Project planning is an organized and integrated management process, which focuses on activities required for successful completion of the project.	
106.	**Project Planning Process**		The project planning process involves a set of interrelated activities followed in an orderly manner to implement user requirements in software and includes the description of a series of project planning activities and individual(s) responsible for performing these activities.	
107.	**Objectives and scope of the project**		Techniques used to perform project planning Effort (in time) of individuals involved in project Project schedule and milestones Resources required for the project Risks associated with the project.	
108.	**Project Plan**		A project plan helps a project manager to understand, monitor, and control the development of software project. This plan is used as a means of communication between the users and project management team.	

109.	**RFP**		An RFP stands for "request for proposal" and is generated as part of the bidding procedure for a product or service. The purpose of an RFP is to provide a structured way for companies to learn about doing business with software development teams.	
110.	**Risk management**		Risk management is a management specialism aiming to reduce different risks related to a preselected domain to the level accepted by society. It may refer to numerous types of threats caused by environment, technology, humans, organizations and politics.	
111.	**Task**		**Task** is part of a set of actions which accomplish a job, problem or assignment.	
112.	**Management process**		Management process is a process of planning and controlling the performance or execution of any type of activity.	
113.	**Process**		Process is an ongoing collection of activities, with an inputs, outputs and the energy required to transform inputs to outputs.	
114.	**Task analysis**		Task analysis is the analysis or a breakdown of exactly how a task is accomplished, such as what sub-tasks are required	
115.	**Project manager**		Professional in the field of project management. Project managers can have the responsibility of the planning, execution, and closing of any project, typically relating to construction industry, architecture, computer networking, telecommunications or software development.	
116.	**Resources**		Resources are what is required to carry out a project's tasks. They can be people, equipment, facilities, funding, or anything else capable of definition (usually other than labor) required for the completion of a project activity.	
117.	**Allocation**		Allocation is the assignment of available resources in an economic way.	
118.	**Project network**		Project network is a graph (flow chart) depicting the sequence in which a project's terminal elements are to be completed by showing terminal elements and their dependencies.	
119.	**Quality, Cost, Delivery (QCD)**		Quality, Cost, Delivery (QCD) as used in lean manufacturing measures a business's activity and develops Key performance indicators. QCD analysis often forms a part of continuous improvement programs	
120.	**Scope**		Scope of a project in project management is the sum total of all of its products and their requirements or features.	
121.	**Six Sigma**		Six Sigma is a business management strategy, originally developed by Motorola, that today enjoys widespread application in many sectors of industry.	
122.	**Case study**		Case study is a research method which involves an in-depth, longitudinal examination of a single instance or event: a case. They provide a systematic way of looking at events, collecting data, analyzing information, and reporting the results.	
123.	**Portfolio**		Portfolio in finance is an appropriate mix of or collection of investments held by an institution or a private individual.	
124.	**Project**		Project : A temporary endeavor undertaken to create a unique product, service, or result.	
Placement Questions				
125.	**What is the average of first five multiples of 10?**		Average = $10*(1+2+3+4+5) * \frac{1}{5}$	

			$= 10 * 15 * \frac{1}{5}$ $= 10 * 3 = 30$	
126.	**What is the difference in the place value of 5 in the numeral 754853?**		The digit 5 has two place values in the numeral, $5 * 10^5 = 50{,}000$ and $5 * 10^1 = 50$. ∴Required difference = 50000 - 50 = 49950	
127.	**A number added to 1459 so that it is exactly divisible by 12.**		On dividing 1459 by 12, the remainder is 7. ∴The number to be added would be = 12 - 7 = 5	
128.	**In the given expression (1.05)2 *x = 44.1, find the value of x.**		(1.05)2 *x=44.1 Or, x = 44.1/ (1.05)2 = 44.1/ (1.05 * 1.05) Hence, x = 40.00	
129.	**If January 1, 1996, was Monday, what day of the week was January 1, 1997?**		The year 1996 is divisible by 4, so it is a leap year with 2 odd days. As per the question, the first day of the year 1996 was Monday, so the first day of the year 1997 must be two days after Monday. So, it was Wednesday.	
130.	**A: B: C is in the ratio of 3: 2: 5. How much money will C get out of Rs 1280?**		C's share = [C's ratio/ sum of ratios] * total amount C's share = (5/10) * 1280 C's share = 640	
131.	**Today is Wednesday, after 69 days, it will be**		Each day of a week is repeated after 7 days, so after 70 days, it will be Wednesday. Therefore, after 69 days, it will be Tuesday.	
132.	**A Number times the hands of a clock coincide in a day**		The hands of a clock coincide only once between 11 O' clock and 1 O' clock, so in every 12 hours, the hands of a clock will coincide for 11 times. ∴ In a day or 24 hours, the hands of a clock will coincide for 22 (11+11) times.	
133.	**The area of a triangle with base 10 meters and height 20 meters.**		Area of a triangle = ½ * base * height So, the area = ½* 10 * 20 =100 square meters	
134.	**A: B: C:D is in the ratio of 3: 2: 5:2. Calculate C's share out of 1260.**		C's share = [C's ratio/ sum of ratios] * total amount C's share = (5/10) * 1260 C's share = 630	
135.	**CKDL, EKFL, GKHL, _, KKLL,**		The second and fourth letters (K and L) in the series are static. The first and third letters are in alphabetical order starting with the letter C. So, the missing letters are IKJL.	
136.	**RQP, ONM, _, IHG, FED,**		The series consists of letters in reverse alphabetical order. Therefore, the missing letters are LKJ.	
137.	**GAH, IBJ, KCL, MDN**		The middle letters in this series follow the order ABCDE. The first and third letters are in alphabetical order starting with the letter G.	
138.	**E3FG, _, E5FG, E6FG, E7FG**		The letters are the same in the series; they differ only in numbers. So, focus on the number series which is a simple series of numbers; 3, 4,5,6,7. Therefore, the missing letters are E4FG.	
139.	**BKK, DMM, FOO, _, JSS**		The first letters of the series are in an alphabetical order in which a letter is skipped between each two letters; B, D, F, H, J. The second and third letters are repeated in each segment, and they are also in alphabetical order with a skipped letter; K, M, O, Q, S. So, the missing letters are HQQ.	
140.	**4, 7, 12, 19, _, 39**		In this series, the difference between the consecutive	

			numbers increases by 2; 7 - 4 = 3 12 - 7 = 5 19 - 12 = 7 , Therefore, the next number would be 19 + 9 = 28	
141.	**15, 20, 24, 15, 28, 32 15, _, _, 15**		This is a simple addition series in which the number "15" is interpolated as every third number. And, except 15, four is added to each number to arrive at the next number	
142.	**77, 70, 63, 56, 49, _,**		This is a simple subtraction series in which each number is 7 less than the previous number.	
143.	**12, 24, 14, 28, 18, 36, __,**		This is an alternating multiplication and subtraction series; first multiply by 2 then subtract 10. Therefore, 26 (36 - 10) should come next.	
144.	**72, 36, 18, ...**		On dividing 72 by 2, we get 36 On dividing 36 by 2, we get 18 So, on dividing 18 by 2, we will get 9	
145.	**40 % of 200 =**		x % of a given number 'n' $=\frac{x}{100} * n$ x = 40 and n = 200 $\therefore$ 40 % of 200 $=\frac{40}{100} * 200 = 80$	
146.	**GAH, IBJ, KCL, MDN, _.**		The middle letters in this series follow the order ABCDE. The first and third letters are in alphabetical order starting with the letter G.	
147.	**U, O, _, E, A**		The series contains vowels in reverse order, U, O, I, E, A. So, the missing letter is I.	
148.	**467X4 is divisible by 9**		The number is divisible by 9 so the sum of its digits would be divisible by 9. $\therefore$4 + 6 + 7 + X + 4 = 21+ X, must be divisible by 9.X = 6, fulfills our requirement so the required digit is 6.	
149.	**A shopkeeper sold an article for Rs. 3500. If the cost price of the article is 2000, find the profit percent.**		C.P. = Rs. 2000 S.P. = Rs. 3500 Profit or Gain = S.P. -C.P. = 3500 - 2000 = 1500 **Apply formula:** Profit % $=\frac{\text{Profit}}{\text{C.P.}} * 100$ $=\frac{1500}{2000} * 100 = 75$ %	
150.	**KDLOC**		E_i=a*(KDLOC)b	

Signatures

Faculty Prepared By

1. **V.Karuppuchamy**
2. **S.R.Sridhar**

HoD