III/IV B.Tech (Regular) DEGREE EXAMINATION

SCHEME OF EVALUATION

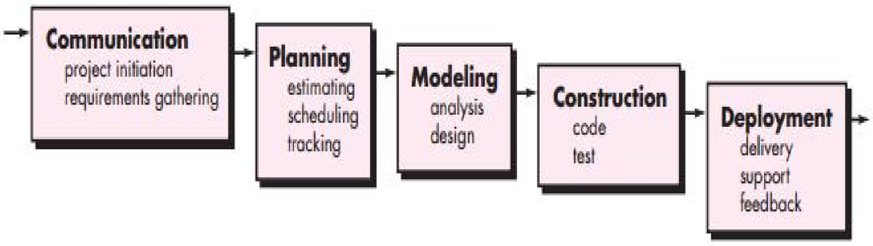
February,2023 **Common to CB,CS,DS & IT Branches**

Fifth Semester **(20IT503)Software Engineering**

Time: Three Hours **Maximum :** 70 Marks

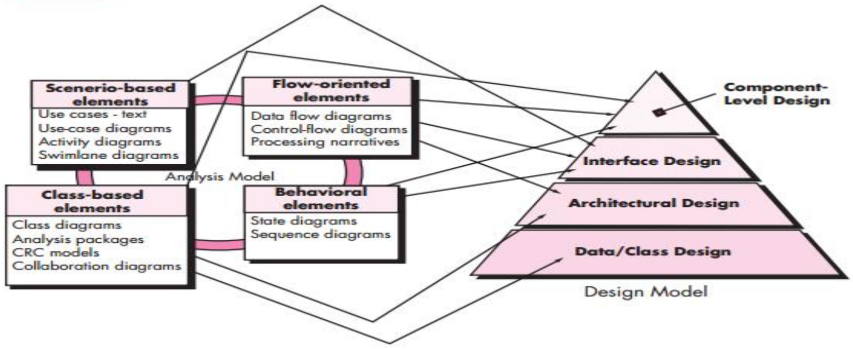
Q.No.	QUESTION & ANSWER	MARKS
1 A)	**What are the characteristics of the software? L2 CLO1 1M** **ANS)**Characteristics of the software are A)Software is developed or engineered; it is not manufactured in the classical sense. B) Software doesn’t “wear out.” C) Although the industry is moving toward component-based construction, most software continues to be custom built.	1 M
1 B)	**What are the umbrella activities of a software process? L2 CLO1 1M** **ANS)**Umbrella activities are applied throughout a software project and help a software team manage and control progress, quality, change, and risk. Typical umbrella activities include • Software project tracking and control • Risk management • Software quality assurance • Technical reviews • Measurement • Software configuration management • Reusability management • Work product preparation and production	1 M
1 C)	**Define Software Engineering.. L1 CLO1 1M** **ANS)**Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines. (OR) Software engineering is systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software.	1 M
1 D)	**Define use case. Give an example. L1 CLO1 1M** **ANS)**A use case is a description of a set of sequences of actions, including variants, that a system performs to yield an observable result of value to an actor. Graphically, a use case is rendered as an ellipse.	1 M
1 E)	**What is agility? L1 CLO2 1M**	1 M

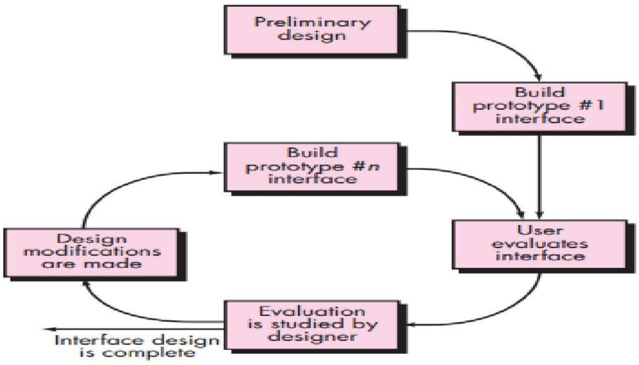
	ANS) Agility is effective response to change, it is more than an effective response to change. It encourages team structures and attitudes that make communication more facile. It emphasizes rapid delivery of operational software and de-emphasizes the importance of intermediate work products. It adopts the customer as a part of the development team and works to eliminate the "us and them" attitude that continues to pervade many software projects; It recognizes that planning in an uncertain world has its limits and that a project plan must be flexible.	
1 F)	**What is Refactoring? L2 CLO2 1M** **ANS)**Refactoring is a reorganization technique that simplifies the design (or code) of acomponent without changing its function or behaviour. (or) Refactoring is the process of changing a software system in such a way that it does not alter the external behaviour of the code [design] yet improves its internal structure.	**1 M**
1 G)	**Define Behavioural Modelling. L2 CLO2 1M** **ANS)**Behavioural Modelling is used to visualize, specify, construct, and document the dynamic aspects of a system**.**	**1 M**
1 H)	**Define Requirements Engineering. L1 CLO2 1M** **ANS)**The broad spectrum of tasks and techniques that lead to an understanding of requirements is called requirements engineering.	**1 M**
1 I)	**List commonly used architectural styles. L1 CLO3 1M** **ANS)** Commonly used architectural styles are • Data-centered architectures. • Data-flow architectures • Call and return architectures. • Object-oriented architectures • Layered architectures	**1 M**
1 J)	**Define Component. L2 CLO3 1M** **ANS)**A component as a modular, deployable, and replaceable part of a system that encapsulates implementation and exposes a set of interfaces.	**1 M**
1 K)	**What is software Architecture? L1 CLO3 1M** **ANS)**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.	**1 M**
1 L)	**Define software Quality. L2 CLO4 1M** **ANS)** An effective software process applied in a manner that creates a useful product that provides measurable value for those who produce it and those who use it.	**1 M**

1 M)	**Define software testing.** **L1 CLO4 1M** **ANS)** Software testing is a road map that describes the steps to be conducted as part of testing. (OR) Software Testing is a set of activities that can be planned in advance and conducted systematically. For this reason a template for software testing—a set of steps into which you can place specific test case design techniques and testing methods—should be defined for the software process.	**1 M**
1 N)	**Define project metrics.** **L1 CLO4 1M** **ANS)** A quantitative measure of the degree to which a system, component, or process possesses a given attribute.	**1 M**
2 A	**Apply waterfall model for development of a telephone directory system. Indicate in detail each of the steps.** **L3 CLO1 7M** **ANS)** A Brief introduction to waterfall model or diagram.-------------  Communication project initiation requirements gathering Planning estimating scheduling tracking Modeling analysis design Construction code test Deployment delivery support feedback Introduction to telephone directory system.----------------------------- Detail explanation of developing telephone directory system using----waterfall model.	**2 M** **1 M** **4 M**
2 B	**Explain in detail about process assessment.** **L1 CLO1 7M** **ANS)** 1. Introduction to process assessment (or) Need of process assessment.--- 2. Detail explanation of four approaches to software process assessment.--- 1. Standard CMMI Assessment Method for Process Improvement (SCAMPI) 2. CMM-Based Appraisal for Internal Process Improvement 3. SPICE (ISO/IEC15504) **4.** ISO 9001:2000 for Software	**1 M** **6 M**
3 A	**Differentiate Personal and Team Process Models** **L2 CLO1 7M** **ANS)** 1. Detail explanation of Personal Software Process (PSP).------------- 1.1. Framework activities of PSP. 2. Detail explanation of Team Software Process (TSP)----------------- 2.1. Framework activities of TSP	**4 M** **3 M**

3 B	**Mention the important phases of spiral model of software development. Compare the relative merits and demerits of spiral model with the prototype model. L2 CLO1 7M** **ANS)** **1.** A detailed explanation of Spiral model.----------------------------- 2. A diagram which shows process of Spiral model.------------------ 3. Comparison between Spiral and Prototype model:----------------- Any three relevant comparison statements	**3 M** **1 M** **3 M**
4 A	**Explain briefly about scrum process model? L1 CLO2 7M** **ANS)** 1.A brief introduction to scrum process model. -------------------------- 2.A Diagram which represent scrum process flow. ---------------------- 3.Explanation about each --- • Backlog • Sprints • Scrum meetings • Demos	**1 M** **2 M** **4 M**
4 B	**What is the purpose of requirements elicitation? who are the different stakeholders involved in requirements elicitation? L2 CLO2 7M** **ANS:-** **1. Purpose of requirements elicitation:----------------------------------** Requirements elicitation (also called requirements gathering) In order to encourage a collaborative, team-oriented approach to requirements gathering, stakeholders work together to identify the problem, propose elements of the solution, negotiate different approaches and specify a preliminary set of solution requirements. **1.1. Collaborative Requirements Gathering:** **1.2. Quality Function Deployment:** QFD identifies three types of requirements : • Normal requirements. • Expected requirements • Exciting requirements **1.3. Elicitation Work Products:** **2. Different stakeholders involved in requirements elicitation: -------** Customers, Users, Developer, Marketing person, SQA Team, Software Engineer, Software team member, Software engineering manager, Product engineering, Software Testing team member..etc.,	**5 M** **2 M**
5 A	**Explain requirement engineering tasks? L1 CLO2 7M** **ANS:-** Requirements engineering provides the appropriate mechanism for understanding what the customer wants, analyzing need, assessing feasibility, negotiating a reasonable solution, specifying the solution	

	unambiguously, validating the specification, and managing the requirements as they are transformed into an operational system. It encompasses seven distinct tasks: inception, elicitation, elaboration, negotiation, specification, validation, and management. **Explanation of each task:** Inception Elicitation Elaboration Negotiation Specification Validation Management	 **1 M** **1 M** **1 M** **1 M** **1 M** **1 M** **1 M**
5 B	**Explain flow oriented modelling. L1 CLO2 7M** **ANS)** **1. Data Flow Model:- ---** • A brief introduction and guidelines to draw DFD. • Diagrams of Level 0 DFD, Level 1 DFD of any system **2. Control Flow Model:- ---** A large class of app lications are "driven" by events rather than data, produce control information rather than reports or displays, and process information with heavy concern for time and performance. Such applications require the use of control flow modelingin addition to data flow modeling. **1) The Control Specification** A control specification(CSPEC) represents the behaviour of the system (at the level from which it has been referenced) in two different ways. The CSPEC contains a state diagram that is a sequential specification of behaviour. It can also contain a program activation table—a combinatorial specification of behaviour. **1.1) State diagrams:-** **1.2) Program activation table:-** **2) The Process Specification**: The process specification (PSPEC) is used to describe all flow model processes that appear at the final level of refinement. The content of the process specification can include narrative text, a program design language (PDL) description of the process algorithm, or UML activity diagrams. & Example of PSPEC	 **3 M** **4 M**
6 A	**Briefly explain taxonomy of Architectural Styles. L1 CLO3 7M** **ANS)** **1. List of all five Architectural Styles:-----------------------------------** **2. Explanation of any 3 architectural styles with diagrams---------** • **Data-centered architectures** • **Data-flow architectures** • **Call and return architectures** A number of sub styles exist within this category: •Main program/subprogram architectures:- •Remote procedure call architectures:- • **Object-oriented architectures**	 **2 M** **5 M**

	• **Layered architectures**	
6 B	**Describe the process of translating requirements in to design model with a neat diagram. L2 CLO3 7M** **ANS)** **1. Diagram ---**  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. Explanation of each design in design model:--------------------------- • **Data/class design** • **Architectural design** • **Interface design** • **Component-level design**	**3 M** **4 M**
7 A	**Describe the characteristics of good design. L2 CLO3 7M** **ANS)** **A brief explanation of any 7 characteristics of good gesign: -----------** 1. A design should exhibit an architecture that (A) has been created using recognizable architectural styles or patterns, (B) is composed of components that exhibit good design characteristics and (C) can be implemented in an evolutionary fashion, thereby facilitating implementation and testing. 2. A design should be modular 3. A design should contain distinct representations of data, architecture, interfaces, and components. 4. A design should lead to appropriate data structures 5. A design should lead to components that exhibit independent functional characteristics. 6. A design should lead to interfaces 7. A design should be derived using a repeatable method 8. A design should be represented using a notation that effectively communicates its meaning.	**7 M**
7 B	**Write about the interface design evaluation. L1 CLO3 7M** **ANS)** A brief introduction to interface design evaluation/ purpose of design evaluation.--- Explanation of process or methodology to evaluate interface design with diagram.---	**2 M** **5 M**


Preliminary design
Build prototype #1 interface
Build prototype #n interface
User evaluates interface
Design modifications are made
Evaluation is studied by designer
Interface design is complete

8 A **Write about size-oriented and function-oriented metrics.** **L1 CLO4 7M**

ANS)

1. Size-Oriented Metrics:- -- **2 M**

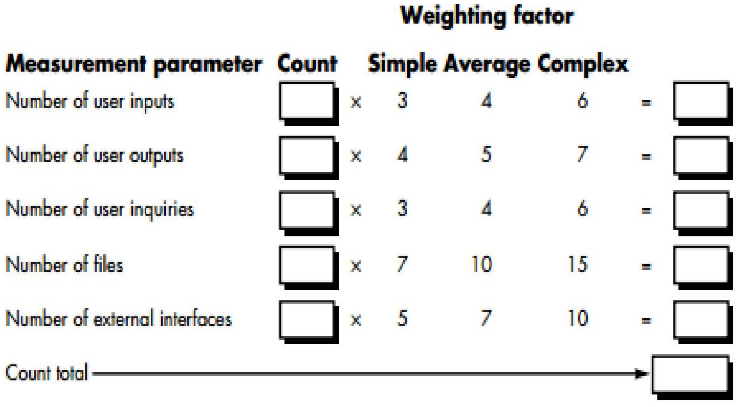
Size-oriented software metrics are derived by normalizing quality and/or productivity measures by considering the sizeof the software that has been produced.

A set of simple size-oriented metrics are

- Errors per KLOC (thousand lines of code).
- Defect per KLOC.
- $ per LOC.
- Page of documentation per KLOC.

2. Function-Oriented Metrics:--- **5 M**

- Function point:
  Function points are computed by completing the table shown



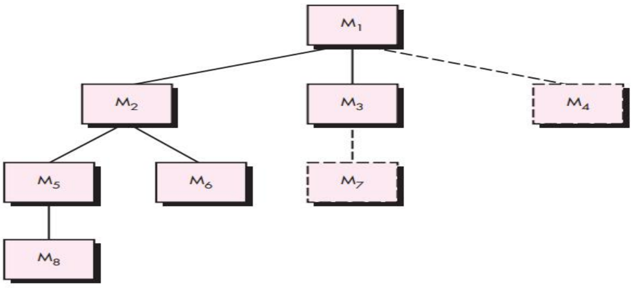
			Weighting factor				
Measurement parameter	**Count**		**Simple**	**Average**	**Complex**		
Number of user inputs	□	×	3	4	6	=	□
Number of user outputs	□	×	4	5	7	=	□
Number of user inquiries	□	×	3	4	6	=	□
Number of files	□	×	7	10	15	=	□
Number of external interfaces	□	×	5	7	10	=	□
Count total →							□

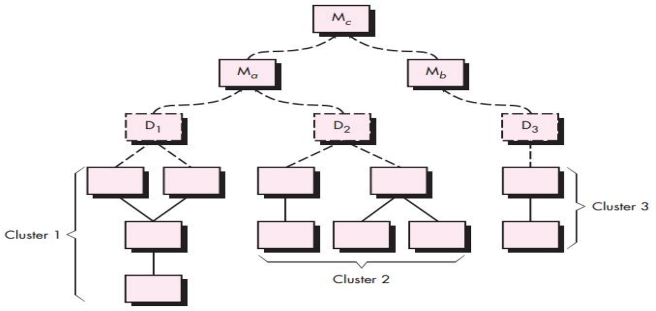
To compute function points (FP), the following relationship is used:

$$\text{FP} = \text{count total} \times [0.65 + 0.01 \times \Sigma(F_i)]$$

The *Fi* (i = 1 to 14) are "complexity adjustment values" based on responses to the set of questions.

Once function points have been calculated, they are used in a manner analogous to LOC as a way to normalize measures for software productivity, quality, and other attributes:

	• Errors per FP. • Defects per FP. • $ per FP. • Pages of documentation per FP. • FP per person-month.	
8 B	**Discuss about ISO 9000 Quality Standards and SQA Plan. L1 CLO4 7M** **ANS)** **1. ISO 9000: ---** Fundamentals of ISO 9000 **2. SQA Plan:---** The SQA Plan provides a road map for instituting software quality assurance. Developed by the SQA group, the plan serves as a template for SQA activities that are instituted for each software project. **2.1. Initial sections** **2.2. Management section** **2.3. Documentation section** describes (by reference) each of the work products produced as part of the software process. These include • project documents (e.g., project plan) • models (e.g., ERDs, class hierarchies) • technical documents (e.g., specifications, test plans) • user documents (e.g., help files) **2.4. Standards, practices, and conventions section** **2.5. Reviews and audits section** **2.6. Test section**. The remainder of the SQA Plan identifies the tools and methods that support SQA activities and tasks;	**3 M** **4 M**
9 A	**Explain about integration testing strategies for conventional software system. L1 CLO4 7M** **ANS)** A brief introduction to Integration testing strategy ------------------- Types of Integration testing: --- 1. **Non Incremental integration** 2. **Incremental integration** A number of different incremental integration strategies are **2.1 Top-down integration**  **2.1.1 Depth-first integration** **2.1.2 Breadth-first integration** **2.2 Bottom-up integration**	**1 M** **4 M**

	 M_c M_a M_b D_1 D_2 D_3 Cluster 1 Cluster 2 Cluster 3 **3. Regression testing:---**	**2 M**
9 B	**Explain about formal technical reviews in detail.** **L2 CLO4 7M** ANS) 1. **Objectives of an FTR ---** **2. Review Meeting Constraints --** **3. Review Reporting and Record Keeping --------------------------** 4. **Review Guidelines for FTR ---**	**1M** **2M** **1M** **3M**