Government of Karnataka
Department of Technical Education
Bengaluru

	Course Title: Software Engineering		
	Scheme (L:T:P) : **4:0:0**	Total Contact Hours: **52**	Course Code: **15CS51T**
	Type of Course: **Lectures, Self Study & Student Activity.**	Credit **:04**	Core/ Elective: **Core**
CIE- 25 Marks			SEE- 100 Marks

Prerequisites

Basic knowledge about the development of programs.

Course Objectives

1. Discuss factors like cost, schedule, quality that drive a software project and to Learn the role of software process and a process model in a project.
2. Understand the role of SRS in a project and to know how requirements are validated.
3. Understand software architectural views and learn planning and estimation of a software project.
4. Describe the key design concepts of software engineering and articulate in Projects.
5. Illustrate principles of structured programming, testing modules using unit test frameworks and code inspection to improve quality of code.
6. Identify evolving trend in software Engineering with DevOps.

Course Outcome

On successful completion of the course, the students will be able to attain below Course Outcome (CO):

Course outcome		**CL**	**Linked PO**	**Teaching Hours**
CO1	Identify the factors that drive software project and summarize the role of process model in a project	***R,U***	**1,2,9,10**	**10**
CO2	Appraise SRS and to develop use cases for a project	***R,U,A***	**1,2,3,8,9,10**	**08**
CO3	Evaluate system architecture and develop detailed task schedule from the overall estimates and planning	***U,A***	**1,2,3,8,9,10**	**12**
CO4	Summarize structured design Methodologies	***U,A***	**1,2,3,4,8,9,10**	**08**
CO5	Illustrate different coding principles with unit test process	***R,U,A***	**1,2,3,7,8,9,10**	**10**
CO6	Identify the need for DevOps.	***R,U***	**1,2,4,9,10**	**04**
			Total	**52**

Legends: R = Remember U= Understand; A= Apply and above levels (Bloom's revised taxonomy)

Course-PO Attainment Matrix

Course	Programme Outcomes									
	1	2	3	4	5	6	7	8	9	10
Software Engineering	**3**	**3**	**3**	**2**	**-**	**-**	**1**	**3**	**3**	**3**

Level 3- Highly Addressed, Level 2-Moderately Addressed, Level 1-Low Addressed.

Method is to relate the level of PO with the number of hours devoted to the COs which address the given PO.
If ≥40% of classroom sessions addressing a particular PO, it is considered that PO is addressed at Level 3
If 25 to 40% of classroom sessions addressing a particular PO, it is considered that PO is addressed at Level 2
If 5 to 25% of classroom sessions addressing a particular PO, it is considered that PO is addressed at Level 1
If < 5% of classroom sessions addressing a particular PO, it is considered that PO is considered not-addressed.

Course Content and Blue Print of Marks for SEE

Unit No	Unit Name	Hour	Questions to be set for SEE			Marks Weightage	Marks Weightage (%)
			R	U	A	A	
I	The software problem & Software process	**10**	05	15	05	25	17
II	Software requirements analysis and specifications.	**08**	05	10	10	25	17
III	Software architecture & planning a software project	**12**	--	20	15	35	25
IV	Design	**08**	05	10	10	25	17
V	Coding and Testing	**10**	--	15	10	25	17
VI	Devops	**04**	--	05	05	10	7
	Total	**52**	**15**	**75**	**55**	**145**	**100**

UNIT I : The software problem & Software process 10 Hrs

Software Problem - Cost, Schedule and quality, Scale and Change.
Software Process - Process and Project, Component Software Processes, Software development Process Models, Waterfall, Prototyping, Iterative development, Rational Unified Process, Time boxing Model, Extreme programming and Agile processes, Using process model in a project, Project Management Process.

UNIT II: Software requirements analysis and specifications 08 Hrs

Value of good SRS, Requirement process, Requirement Specification, Desirable Characteristics of an SRS, Components of an SRS, Structure of a Requirements Documents, Functional Specification with use cases, Basics, Examples, Extensions, Developing use cases, Other Approaches for Analysis, Data Flow Diagram, Validation.

UNIT III: Software architecture & Planning a software project 12 Hrs

Software Architecture - Role of Software Architecture, Architecture Views, Component and Connector view, Components, Connectors, Example, Architecture styles for C&C View, Pipe and Filter, Shared data Style, Client Server style, Some other styles, Documenting Architecture Design, Evaluating Architectures.

Planning a software project - Effort Estimation, Top-Down Estimation Approach, Bottom-Up Estimation Approach, Project Schedule and Staffing, Quality Planning, Risk Management Planning, Risk Management Concepts, Risk Assessment, Risk Control, A Practical Risk Management Approach, Project Monitoring Plan, Measurements, Project Monitoring and tracking**,** Detailed Scheduling.

UNIT IV: Design 08 Hrs

Design Concepts, Coupling, Cohesion, the Open-Closed Principle, Function Oriented Design, Structure Charts, Structured Design Methodology, Example, Object Oriented Design, OO Concepts Unified Modelling Language (UML), A Design Methodology, Examples, Detailed Design, Logic/Algorithm Design, State Modelling of Classes, Verification.

UNIT V: Coding & Testing 10 Hrs

Coding - Programming Principles and Guidelines, Structured Programming, Information Hiding, Some Programming Practices, Coding Standards, Incrementally Developing Code, An incremental coding process, Test Driven development, Pair Programming, Managing Evolving Code, Source Code Control and Build, Refactoring, Unit Testing, Testing procedural units, Unit testing of Classes, Code Inspection, Planning, Self review, Group review meeting.

Testing - Testing Concept, Error, Fault and Failure, Test Case, Test Suite and Test Harness, Psychology of Testing, Levels of Testing, Testing Process, Test Plan, Test Case Design, Test Case Execution.

UNIT VI:Devops 04Hrs

Devops, Introduction-Definition, Devops Tool chain, Why Devops?, Goals, Benefits, Relationship to Agile and Devops (continuous delivery), Devop Tools.

Text books

1. Software Engineering –A Precise Approach, Pankaj Jalote, edition 2010, Wiley India, ISBN: 9788126523115.
2. https://en.wikipedia.org/wiki/DevOps
3. http://www.informationweek.com/devops/agile-vs-devops-10-ways-theyre-different/d/d-id/1326121

References

1. Software Engineering A Practitioners Approach, Roger S Pressman, 2010, Tata McGraw Hill Publishing Co Ltd, ISBN 9780070701137
2. Software Engineering, Sajan Mathew, 2009 Reprint, S Chand publications.
3. Software Engineering Principles and Practices, Rajesh Narang.Mc Graw Hill Education.
4. Software Engineering, Subramanian Dutt, 2015 Pearson Publications

Suggested list of student activities

Note: the following activities or similar activities for assessing CIE (IA) for 5 marks (Any one)

Student activity like mini-project, surveys, quizzes, etc. should be done in group of 3-5 students.

1. Each student should do any one of the following type activity or any other similar activity related to the course and before conduction, get it approved from concerned course coordinator and programme coordinator.
2. Each student should conduct different activity and no repeating should occur
   1. A survey on Software Development Life Cycle(SDLC).
   2. Presentation on Software Architecture.
   3. A report on testing the software products.
   4. Quiz.

Course Delivery

The course will be delivered through lectures and Power point presentations/ Video

Course Assessment and Evaluation Scheme

Method	What		To whom	When/Where (Frequency in the course)	Max Marks	Evidence collected	Course outcomes
Direct Assessment	CIE	IA	Students	Three IA tests (Average of three tests will be computed)	20	Blue books	1,2,3,4
				Student activities	05	Report	1,2,3,4
				Total	**25**		
	SEE	End Exam		**End of the course**	**100**	Answer scripts at BTE	1,2,3,4,5,6
Indirect Assessment	Student Feedback on course		Students	Middle of the course		Feedback forms	1 & 2 Delivery of course
	End of Course Survey			End of the course		Questionnaires	1,2,3,4,5,6 Effectiveness of Delivery of instructions & Assessment Methods

***CIE** – Continuous Internal Evaluation *SEE – Semester End Examination

Note: I.A. test shall be conducted for 20 marks. Average marks of three tests shall be rounded off to the next higher digit.

***Note to IA verifier**: The following documents to be verified by CIE verifier at the end of semester*

1. Blue books (20 marks)
2. Student suggested activities report for 5 marks evaluated through appropriate rubrics.
3. Student feedback on course regarding Effectiveness of Delivery of instructions & Assessment Methods.

Questions for CIE and SEE will be designed to evaluate the various educational components (Bloom's taxonomy) such as:

Sl. No	Bloom's Category	%
1	Remembrance	10

2	Understanding	50
3	Application	40

***Note to IA verifier**: The following documents to be verified by CIE verifier at the end of semester*

4. Blue books (20 marks)
5. Student suggested activities report for 5 marks
6. Student feedback on course regarding Effectiveness of Delivery of instructions & Assessment Methods.

FORMAT OF I A TEST QUESTION PAPER (CIE)

Test/Date and Time	Semester/year	Course/Course Code	Max Marks
Ex: I test/6 th week of sem 10-11 AM	V SEM		20
	Year:		

Name of Course coordinator :
Units:__ CO's:____

Question no	Question	MARKS	CL	CO	PO
1					
2					
3					
4					

Note: Internal choice may be given in each CO at the same cognitive level (CL).

MODEL QUESTION PAPER (CIE)

Test/Date and Time	Semester/year	Course/Course Code	Max Marks
Ex: I test/6 th week of sem 10-11 AM	V SEM	Software Engineering	20
	Year: 2017-18	Course code:**15CS51T**	

Name of Course coordinator :
Units:1,2 Co: 1,2

Note: Answer all questions

Question no	Question	CL	CO	PO
1	Explain components of software processes.(5) **OR** Discuss the desirable characteristics of an SRS.(5)	U	1,2	1,2
2	Discuss the various notations used in ER-diagram.(5) **OR** Compare scale and change in a software problem.(5)	U	1,2	1,2
3	Explain Waterfall model with a neat diagram. (10) **OR** Explain Prototyping model.(10)	U	1	1,2

Format for Student Activity Assessment

DIMENSION	Unsatisfactory 1	Developing 2	Satisfactory 3	Good 4	Exemplary 5	Score

Collection of data	Does not collect any information relating to the topic	Collects very limited information; some relate to the topic	Collects some basic information; refer to the topic	Collects relevant information; concerned to the topic	Collects a great deal of information; all refer to the topic	3
Fulfill team's roles & duties	Does not perform any duties assigned to the team role	Performs very little duties	Performs nearly all duties	Performs all duties	Performs all duties of assigned team roles with presentation	4
Shares work equally	Always relies on others to do the work	Rarely does the assigned work; often needs reminding	Usually does the assigned work; rarely needs reminding	Does the assigned job without having to be reminded.	Always does the assigned work without having to be reminded and on given time frame	3
Listen to other Team mates	Is always talking; never allows anyone else to speak	Usually does most of the talking; rarely allows others to speak	Listens, but sometimes talk too much	Listens and contributes to the relevant topic	Listens and contributes precisely to the relevant topic and exhibit leadership qualities	3
					TOTAL	**13/4=3.25=4**

Note: This is only an example. Appropriate rubrics/criteria may be devised by the concerned course co-ordinator for assessing the given activity.

MODEL QUESTION PAPER Code: **15CS51T**

Diploma in Computer Science & Engineering

V- Semester

Course Title: Software Engineering

Time: **3 Hours** Max Marks**: 100**

PART-A

Answer any SIX questions. Each carries 5 marks. **5X6=30 Marks**

1. Explain cost, schedule and quality.
2. Explain components of software processes.
3. Explain desirable characteristics of an SRS.
4. Explain Data Flow Diagram.
5. Discuss the role software architecture.
6. Explain Bottom-Up estimation approach.
7. Explain Structure charts.
8. Discuss the various OO concepts.
9. Define use case. List out the terms used in use cases.
10. Explain different attributes of software quality.
11. Define the term Error, Fault, Failure, Test Case and Test Suite.
12. Define Devops. Discuss the goals and benefits of Devops.
13. Explain the life cycle of defect.

PART-B

Answer any SEVEN full questions each carries 10 marks. **10X7=70 Marks**

1. Explain the Waterfall model with a neat diagram.
2. Explain Time boxing model with a neat diagram.
3. Explain components and connectors with an example.
4. Explain different architectural styles.
5. Explain the effort estimation of a software project.
6. Explain the Risk management concepts.
7. Explain various design concepts.
8. Explain structured design methodology.
9. Explain Test Driven development.
10. Compare Agile and Devops.

MODEL QUESTION BANK

Diploma in Computer Science & Engineering

V Semester

Course Title: Software Engineering

CO	Question	CL	Marks

I	Define software Engineering. Differentiate between process and project	R	05
	Discuss the main differences between student software and industrial software.	U	
	Explain cost, schedule and quality.	U	
	Define scale and change in a software problem.	U	
	Define different attributes of software quality.	U	
	Explain software process and software project.	A	
	Classify the software process components.	U	
	Explain software development process model. List the various software development process models.	A	
	Explain project management process.	A	
	Explain Extreme programming.	U	
	Explain Agile processes.	U	
	Explain the temporal relationship between development and management process.	A	
	Discuss the software process and software project.	A	10
	Explain components of software processes.	A	
	Explain Waterfall model with a neat diagram.	A	
	Explain Time boxing model with a neat diagram.	U	
	Explain Prototyping model.	U	
	Explain iterative development model.	U	
	Explain Rational Unified Process(RUP) model.	A	
	Explain Extreme programming and Agile processes.	A	
	Explain the phases of project management process.	U	
II	Mention the values of good SRS.	R	05
	Explain requirement process.	A	
	Discuss the desirable characteristics of an SRS.	U	
	Explain the general structure of SRS.	A	
	Define use case. List out the terms used in use cases.	R	
	Explain the various abstractions in each level while developing use cases.	A	
	Write the steps for analysis when employing use cases.	A	
	Explain the concept of state and projection .List the different approaches for problem analysis.	U	
	Discuss the various notations used in ER-diagram.	U	
	Explain the validation of SRS.	A	
	Explain the components of SRS.	U	10
	Explain the Data Flow Diagram with a neat diagram.	A	
	Discuss use case scenario with an example.	A	
III	Define software architecture. Explain the important uses of software architecture.	R	05
	Explain the role of software architecture.	A	
	Explain Components view.	U	
	Explain Connectors view.	U	
	Define architectural styles. List the architecture styles.	U	

	Explain Peer-to-Peer & Publish-Subscribe styles.	**U**	
	Discuss how to evaluate a proposed architecture.	**A**	
	Describe the role of effort estimation in a project.	**A**	
	Explain Top-Down estimation approach.	**U**	
	Explain Bottom-Up estimation approach.	**U**	
	Explain project scheduling and staffing with an example.	**A**	
	Define quality plan. Explain quality control activities.	**R**	
	Explain Risk Control.	**U**	
	Explain practical approach for Risk management planning.	**A**	
	Explain measurements in project monitoring plan.	**U**	
	Explain project monitoring and tracking plan.	**U**	
	Explain detailed scheduling.	**A**	
	Explain the architectural view.	**U**	**10**
	Explain architectural view with an example.	**U**	
	Explain components and connectors with a neat diagram.	**U**	
	Explain different architectural styles.	**U**	
	Explain Pipe and Filter Style.	**U**	
	Explain Shared-Data Style.	**U**	
	Explain Client-Server Style.	**U**	
	What should an architecture document for a system contain?	**A**	
	Explain COCOMO Model.	**A**	
	Explain Bottom-Up approach.	**U**	
	Explain Quality Planning.	**U**	
	Explain Risk management concepts.(any one)	**U**	
	Explain the goal of Risk Assessment to prioritize the risk.	**A**	
IV	Identify the goal of Design and mention the main criteria for evaluating the design.	U,A	**05**
	Define coupling. List the different types of it.	**R**	
	Define cohesion. List the different levels of cohesion.	**R**	
	Describe the cohesion in OO system.	**U,A**	
	Explain open closed principle.	**U**	
	Explain Structure Charts for creating function oriented design.	**U**	
	Explain OO design methodology.	**U**	
	Define UML. List the different types of UML modelling.	**R**	
	With an example explain word counting problem for OO design.	**U,A**	
	Write a note on Logic/Algorithm Design.	**A**	
	Write a note on State Modelling of Classes.	**A**	
	Explain the approaches for design verification.	**A**	
	Explain the various Design concepts.	**U**	**10**
	Explain the different types of coupling between the modules.	**U**	
	Explain the different levels of cohesion.	**A**	
	Explain Structure Design Methodology for function oriented design.	**U**	
	Explain the word counting problem through DFD and factoring.	**A**	
	Explain main Object Oriented Concepts.	**U**	

	Explain Class diagram with an example.	U	
	Explain Sequence diagram with an example.	U	
	Explain Collaboration diagram with an example.	U	
	Explain state diagram and activity diagram.	U	
	Explain the OO Design methodology.	U	
V	Define coding. Explain the basic goal of coding.	R,U	05
	Predict the necessity of coding standards explain any one	U,A	
	Write a note on Source code control.	U	
	Write a note on Bad smells of refactoring.	A	
	Define Unit Testing. Write a note on Testing Procedural Units.	R,U	
	Define Code inspection and summarize the report of an inspection.	R,A	
	Define testing. Discuss the goal of testing.	U	
	Define error, fault, failure, Test case, Test suite and test harness.	R	
	Write a note on Test plan.	U	
	Explain the test case specification in the test case design.	A	
	Explain the life cycle of defect.	U,A	10
	Explain the necessary Programming Practices for coding	U,A	
	Explain Coding standards.	U	
	Explain Incremental Coding process with a neat flow chart.	U	
	Explain Test Driven Development (TDD) with a neat flow chart.	U	
	Explain Structured Programming.	A	
	Explain Information Hiding.	U	
	Explain Pair Programming.	U	
	Explain Refactoring.	A	
	Explain Unit testing of Classes with example.	U,A	
	Explain Code Inspection.	U	
	Explain different levels of testing and the goals of each level.	U,A	
	Explain the Testing process.	U,A	
VI	Define DevOps. Discuss the goal of DevOps	R,U	5
	Explain the benefits of DevOps.	U	
	Explain DevOps tool chain.	U	
	Explain the DevOps Tools.	U	10
	Explain the necessity of DevOps in detail	U,A	
	Compare Agile and DevOps.	A	