T.E. COMP/S.E /20-21(II)

COURSE FILE – SOFTWARE ENGINEERING

SUBJECT INCHARGE – Dr. B.S. DAGA

Course Code	Course/Subject Name	Credits
CSC601	Software Engineering	05

University Syllabus:

Module	Detailed Contents	Hrs
1.	**Introduction To Software Engineering and Process Models** 1.1 Nature of Software, Software Engineering, Software Process, Capability Maturity Model (CMM) 1.2 Generic Process Model, **Prescriptive Process Models:** The Waterfall Model, V-model, Incremental Process Models, Evolutionary Process Models, Concurrent Models, Agile process, Agility Principles, Extreme Programming (XP), Scrum, Kanban model	08
2.	**Requirements Analysis and Modelling** 2.1 Requirement Elicitation, Software requirement specification (SRS), Developing Use Cases (UML) 2.2 Requirement Model – Scenario-based model, Class-based model, Behavioural model.	08
3.	**Project Scheduling and Tracking** 3.1 Management Spectrum, 3Ps (people, product and process) 3.2 Process and Project metrics 3.3 **Software Project Estimation:** LOC, FP, Empirical Estimation Models - COCOMO II Model, Specialized Estimation Techniques 3.4 **Project scheduling:** Defining a Task Set for the Software Project, Timeline charts, Tracking the Schedule, Earned Value Analysis	08
4.	**Software Design** 4.1 Design Principles, Design Concepts, Effective Modular Design – Cohesion and Coupling.	10

	4.2 Architectural Design 4.3 Component-level design 4.4 User Interface Design	
5.	**Software Risk, Configuration Management & Quality Assurance** 5.1 Risk Identification, Risk Assessment, Risk Projection, RMMM 5.2 Software Configuration management, SCM repositories, SCM process 5.3 Software Quality Assurance Task and Plan, Metrics, Software Reliability, Formal Technical Review (FTR), Walkthrough.	08
6.	**Software Testing and Maintenance** 6.1 Strategic Approach to Software Testing, Unit testing, Integration testingVerification, Validation Testing, System Testing 6.2 Software Testing Fundamentals, White-Box Testing , Basis Path Testing,Control Structure Testing, Black-Box Testing 6.3 Software maintenance and its types, Software Re-engineering, ReverseEngineering	10

Course Objectives:

The main objective is to introduce to the students about the product that is to be engineered andthe process that provides a framework for the engineering technology.

1. To provide knowledge of software engineering discipline.
2. To analyze risk in software design and quality.
3. To introduce the concept of advance software methodology.

Course Outcomes

Upon completion of this course students will have:

No	Course Outcome	Blooms Taxonomy	Explanation
CSC601.1	To compare different types of software development models and choose one based on types of application	Comprehension (explains, gives examples, shows relationship of)	Demonstrate/understanding of Software Development Models
CSC601.2	To Gather the	Analysis	Students will be able to

	requirements and model them using UML principles.		identify problem and analyse the requirements and document them
CSC601.3	To Design Software System effectively	Application (applies, solves , uses, demonstrates)	They will apply software design principles on to make it effective and modular (Flexible)
CSC601.4	To Apply Software Testing and Maintenance Principles on given problem	Application (applies, solves , uses, demonstrates)	Apply different testing types and testing strategies for making bug free software
CSC601.5	To apply umbrella activities on software projects	Application (applies, solves , uses, demonstrates)	Applying Umbrella Activities in different phases of core software development process to help making quality software

CO Assessment Tools:

CSC601.1 To compare different types of software development models and choose one based on types of application

Direct Method Tools *(dm)*	**1**
Test ***(UT)***	0.2
Assignment***(Assign)***	0.2
Practicals***(lab)***	0.2
End Sem Marks(THEORY)***(utTH)***	0.2
End Sem Marks(ORAL)***(utOral)***	0.2
Indirect Method Tools*(idm)*	**1**
Course Exit Survey ***(C01idm)***	**0.2**
CSC601.1 = 0.8*CO1dm + 0.2* CO1idm	

CSC601.2. To Gather the requirements and model them using UML principles.

Direct Method Tools *(dm)*	**1**
Test *(UT)*	0.2
Assignment *(Assign)*	0.2
Practicals*(lab)*	0.2
End Sem Marks(THEORY)*(utTH)*	0.2
End Sem Marks(ORAL)*(utOral)*	0.2
Indirect Method Tools*(idm)*	**1**
Course Exit Survey *(C02idm)*	**0.2**
CSC601.2 = 0.8*CO2dm + 0.2* CO2idm	

CSC601.3. To Design Software System effectively.

Direct Method Tools *(dm)*	**1**
Test *(UT)*	0.2
Assignment *(Assign)*	0.2
Practicals*(lab)*	0.2
End Sem Marks(THEORY)*(utTH)*	0.2
End Sem Marks(ORAL)*(utOral)*	0.2
Indirect Method Tools*(idm)*	**1**
Course Exit Survey *(C03idm)*	**0.2**
CSC601.3 = 0.8*CO3dm + 0.2* CO3idm	

CSC601.4. To Apply Software Testing and Maintenance Principles on given problem.

Direct Method Tools *(dm)*	**1**
Test *(UT)*	0.2
Assignment *(Assign)*	0.2
Practicals*(lab)*	0.2
End Sem Marks(THEORY)*(utTH)*	0.2
End Sem Marks(ORAL)*(utOral)*	0.2
Indirect Method Tools*(idm)*	**1**
Course Exit Survey *(C04idm)*	**0.2**
CSC601.4 = 0.8*CO4dm + 0.2* CO4idm	

CSC601.5. To apply umbrella activities on software projects.

Direct Method Tools *(dm)*	**1**
Assignment *(Assign)*	0.3
Practicals*(lab)*	0.3
End Sem Marks(THEORY)*(utTH)*	0.2
End Sem Marks(ORAL)*(utOral)*	0.2
Indirect Method Tools*(idm)*	**1**
Course Exit Survey *(C05idm)*	
CSC601.5 = 0.8*CO5dm + 0.2* CO5idm	

Weak Student Coverage by Retest and coaching

Course Outcomes Target:

Target	Attainment 20-21	Attainment 19-20	Attainment 18-19 (By Prof. Sunil Chaudhari)	Attainment 17-18 (By Dr. Sunil Surve)
CSC601.1 – Target: 2.6			**2.8**	**2.84**
CSC601.2 – Target: 2.6			**2.48**	**2.36**
CSC601.3 – Target: 2.6			**2.8**	**2.68**
CSC601.4 – Target: 2.6			**2.48**	**2.84**
CSC601.5 – Target: 2.6			**2.64**	**2.3**

Mapping of CO and PO/PSO

Relationship of course outcomes with program outcomes: Indicate 1 (low importance), 2 (Moderate Importance) or 3 (High Importance) in respective mapping cell.

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CSC601.1	2	3											3	3
CSC601.2	2	3			3				2			3	3	3
CSC601.3	2	2	3		3				2			3	3	3
CSC601.4	2	2	3		3				2			3	3	3
CSC601.5	2	3			3				2			3	3	3

Justification

PO1: This subject all COs are mapped to PO1 because engineering graduates will be able to apply the Software Engineering knowledge to solve business problems

PO2: All COs are in general related to PO2 because students analyze the different scenarios of business to decide the solution for the problem

C03: mapped to PO2 because students can analyze the information flow in the organization.

CSC4: mapped to PO2 because student perform review of literature of real world problem to develop business application

PO3: C03 & CO4 is mapped to PO3 because students design an application for domain.

PO5: C0s2,3,4,5 are mapped to PO5 because the students use the tools and defined processes for development.

PO9: C0 2,3,4,5 is mapped to PO9 because the students work in a team

PO12: C0 2,3,4,5

PSO1: All COs are mapped to PSO1 because the graduates will be able to apply fundamental knowledge of software Engineering to solve the real world problems.
PSO2: All COs are mapped to PSO2 because students design and implement the solution to meet specific requirement.

Rubrics for the Assignments:

Indicator	Average	Good	Excellent	Marks
Organization (2)	Readable with some mistakes and structured (1)	Readable with some mistakes and structured (1)	Very well written and structured without any mistakes (2)	
Level of content(4)	All major topics are covered, the information is accurate (2)	Most major and some minor criteria are included. Information is accurate (3)	All major and minor criteria are covered and are accurate (4)	
Depth and breadth of discussion and representation(4)	Minor points/information maybe missing and representation isminimal (1)	Discussion focused on some ofthe points and covers themadequately (2)	Information is presented indepth and is accurate (4)	
Total				

Assignment 1

Chapter 1-2: SDLC Models

Q.1) Explain the Spiral Model of software development with advantages, disadvantages and diagram.

Q.2) Explain the Agile Software Development Methodology with advantages, disadvantages.

Q.3) CMM.

Q.4) Explain Project and Process metrics

Chapter 3: Scheduling, Control & Monitoring

Q.1) Explain Function Point Analysis (FPA) with help of an example

Q.2) Short Note on COCOMO Model

Q.3) Distinguish between CPM and PERT (min. 5 points)

Q.4) Explain how to do qualitative cost estimation using Earned Value Analysis(EVA)

Assignment 2

1) Explain the relevance of Cohesion & Coupling in software design
2) How Service Oriented Software Architecture are implemented?
3) Explain process of Risk Management
4) Explain RMM plan details of your problem statement
5) Short Note on Risk Exposure
6) Short Note on SCM
7) Explain Change Control Process
8) Explain Version Control Process
9) Write short note on White box & Back box testing.

Chapter:

Unit Test I

1. Explain the various types of models with examples which are used in software Engineering? –

(10 MARKS)

2. Do cost estimation in terms of Effort for the given problem by using Basic COCOMO Model .Use Function Point as size estimator by assuming development language.

(10 MARKS)

Problem Statement: The spell Checker Accepts as input a document file & optional personal dictionary file .The checker lists all words not contained in either of these files. The user can query the number of words processed & number of spelling errors found at any stage during processing.

Mode	**a**	**b**
Organic	2.4	1.05
Semi-detached	3.0	1.12
Embedded	3.6	1.20

Unit Test II :

Q1. Explain software design principle based on types of cohesion and coupling with example. **(10 MARKS)**

Q2. Compute cyclomatic complexity for given code. Also compute basis set & determine test data for each basis set. **(10 MARKS)**

```
 1  intfunctionY(void)
 2  {
 3     int x = 0;
 4     int y = 19;

 5  A: x++;
 6     if (x > 999)
 7        goto D;
 8     if (x % 11 == 0)
 9        goto B;
10     else goto A;
11  B: if (x % y == 0)
12        goto C;
13     else goto A;
14  C: printf("%d\n", x);
15     goto A;
16  D: printf("End of list\n");
17     return 0;
18  }
```

List of Practical's:

#	Experiment Name	Aim	CO map

1	SOFTWARE REQUIREMENTS SPECIFICATION	To prepare software requirements specification document for a selected case study in IEEE format	CO1
2	FUNCTION POINT CALCULATION	To calculate function point for a selected case study	CO2
3	COST ESTIMATION USING COCOMO MODEL	To calculate function point for a selected case study	CO2
4	PROJECT SCHEDULING USING PROJECT MANAGEMENT TOOL	Use project management tool to schedule project plan for a selected case study	CO2
5	RISK MANAGEMENT MONITORING AND MITIGATION PLAN	Develop a risk table for a selected case study	CO5
6	General Test-Driven Development	General test driven development for a selected case study	CO4
7	BLACK BOX TESTING	To design test cases for performing black box testing (equivalence partitioning and boundary value analysis) for a selected case study	CO4
8	WHITE BOX TESTING	To design test cases for performing white box testing for a selected case study	CO4
9	COHESION AND COUPLING	Draw Architecture diagram and incorporate Cohesion and Coupling for each module of <type your selected case study	CO3
10	DESIGN PATTERNS	Application of at least two types of design patterns in selected case study	CO3

Practical Submission Date (Batch wise)

#	Experiment Name	Date (A Batch)	Date (C Batch)	Date (D Batch)

1	SOFTWARE REQUIREMENTS SPECIFICATION			
2	FUNCTION POINT CALCULATION			
3	COST ESTIMATION USING COCOMO MODEL			
4	PROJECT SCHEDULING USING PROJECT MANAGEMENT TOOL			
5	RISK MANAGEMENT MONITORING AND MITIGATION PLAN			
6	General Test-Driven Development			
7	BLACK BOX TESTING			
8	WHITE BOX TESTING			
9	COHESION AND COUPLING			
10	DESIGN PATTERNS			

Exit Survey For Software Engineering – Sem VI, Academic Term II (20-21)

Student Name:_____ Roll No: _____

		Very Well	Well	Neutral	Average
1.	How well you can apply software engineering principles/processes for software project				
2.	How well you have understood Project Scheduling and Tracking				
3.	How well you understood software Design Principles				
4.	How well you have understood principles of software testing				
5.	How well you have understood advance software Engineering principles like (RMMM,SCM,SQA) to create high quality software Applications				

Lecture/Lesson Plan:

Modes of Content Delivery:

I	**Class Room/Lab Teaching**	v	**Self Learning Online Resources**	Ix	Industry Visit
Ii	Tutorial	vi	**Slides**	X	**Group Discussion**
Iii	**Remedial Coaching**	vii	Simulations/Demonstrations	xi	**Seminar**
Iv	**Lab Experiment**	viii	Expert Lecture	xii	**Case Study**

Date wise Lesson Plan –

Module	Detailed Contents	Text book and references	Content Delivery Mode	Hrs	Planned Dates
1.	**Introduction To Software Engineering and Process Models** 1.1 Nature of Software, Software Engineering, Software Process, Capability Maturity Model (CMM) 1.2 Generic Process Model, **Prescriptive Process Models:** The Waterfall Model, V-model, Incremental Process Models, Evolutionary Process Models, Concurrent Models, Agile process, Agility Principles, Extreme Programming (XP), Scrum, Kanban model	T1, R1		08	25-Jan-21 28-Jan-21 29-Jan-21 1-Feb-21 2-Feb-21 4-Feb-21 5-Feb-21 8-Feb-21
2.	**Requirements Analysis and Modelling** 2.1 Requirement Elicitation, Software requirement specification (SRS), Developing Use Cases (UML) 2.2 Requirement Model – Scenario-based model, Class-based model, Behavioural model.	T1, R2		08	9-Feb-21 11-Feb-21 12-Feb-21 15-Feb-21 16-Feb-21 18-Feb-21 22-Feb-21 23-Feb-21
3.	**Project Scheduling and Tracking** 3.1 Management Spectrum, 3Ps (people, product and process) 3.2 Process and Project metrics 3.3 **Software Project Estimation:** LOC,	T2, R2. R1		08	25-Feb-21 26-Feb-21 1-Mar-21 2-Mar-21 4-Mar-21

	FP, Empirical Estimation Models - COCOMO II Model, Specialized Estimation Techniques 3.4 **Project scheduling:** Defining a Task Set for the Software Project, Timeline charts, Tracking the Schedule, Earned Value Analysis				5-Mar-21 8-Mar-21 9-Mar-21
4.	**Software Design** 4.1 Design Principles, Design Concepts, Effective Modular Design – Cohesion and Coupling. 4.2 Architectural Design 4.3 Component-level design 4.4 User Interface Design	T2, R1, R2		10	12-Mar-21 15-Mar-21 16-Mar-21 18-Mar-21 19-Mar-21 22-Mar-21 23-Mar-21 25-Mar-21 26-Mar-21 30-Mar-21
5.	**Software Risk, Configuration Management & Quality Assurance** 5.1 Risk Identification, Risk Assessment, Risk Projection, RMMM 5.2 Software Configuration management, SCM repositories, SCM process 5.3 Software Quality Assurance Task and Plan, Metrics, Software Reliability, Formal Technical Review (FTR), Walkthrough.	T2, R1, R3		08	1-Apr-21 5-Apr-21 6-Apr-21 8-Apr-21 9-Apr-21 12-Apr-21 15-Apr-21 16-Apr-21
6.	**Software Testing and Maintenance** 6.1 Strategic Approach to Software Testing, Unit testing, Integration testing Verification, Validation Testing, System Testing 6.2 Software Testing Fundamentals, White-Box Testing , Basis Path Testing, Control Structure Testing, Black-Box Testing 6.3 Software maintenance and its types, Software Re-engineering, Reverse Engineering	T2, R1, R3		10	19-Apr-21 20-Apr-21 22-Apr-21 23-Apr-21 26-Apr-21 27-Apr-21 29-Apr-21 30-Apr-21 3-May-21 4-May-21

Text Books:

T1. Roger Pressman, Software Engineering: A Practitioners Approach, (6th Edition), McGraw Hill, 2010

T2. Ian Somerville, Software Engineering, 9th edition, Addison Wesley, 2011

Reference Books:

R1. Eric J. Braude and Micheal E. Bernstein, Software Engineering Modern Approach, 2nd edition, Wiley, 2011.
R2. Ali Behforooz Fredrick Hudson, Software Engineering Fundamentals, Oxford UniversityPress, 2006.
R3. PankajJalote, " Integrated Software Engineering", Wiley.

Reference Web sites: Study Material On Moodle, tutorialpoint.com,mkyong.com

Term Work consists of 25 Marks (15 Marks for Practicals and Presentation +5 Marks for Attendance+ 5 Marks for Assignment)

Total Software Engineering Marks

TW +ORAL+THEORY=25+25+100=150