Program Name : **Computer Engineering Program Group**

Program Code : **CO/CM/IF/CW**

Semester : **Fourth**

Course Title : **Software Engineering**

Course Code : **22413**

1. RATIONALE

Software Engineering is the foundation for professional processes to be followed involving principles, techniques, and practices for software development. The course provides a framework for software professionals for building quality assured software products. It enables students to blend the domain specific knowledge with the programming skills to create quality software products.

2. COMPETENCY

The aim of this course is to help the student to attain the following industry identified competency through various teaching learning experiences:

- **Use relevant software process model for developing software products.**

3. COURSE OUTCOMES (COs)

The theory, practical experiences and relevant soft skills associated with this course are to be taught and implemented, so that the student demonstrates the following ***industry oriented*** COs associated with the above-mentioned competency:

a. Select suitable Software Process model for software development.
b. Prepare software requirement specifications.
c. Use Software modeling to create data designs.
d. Estimate size and cost of software product.
e. Apply project management and quality assurance principles in software development.

4. TEACHING AND EXAMINATION SCHEME

Teaching Scheme			Credit (L+T+P)	Examination Scheme												
				Theory							Practical					
L	T	P		Paper Hrs.	ESE		PA		Total		ESE		PA		Total	
					Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min
3	-	2	5	3	70	28	30*	00	100	40	25@	10	25	10	50	20

(*): Under the theory PA; Out of 30 marks, 10 marks of theory PA are for micro-project assessment to facilitate integration of COs and the remaining 20 marks is the average of 2 tests to be taken during the semester for the assessment of the UOs required for the attainment of the COs.

***Legends:** L-Lecture; **T** – Tutorial/Teacher Guided Theory Practice; **P** - Practical; **C** – Credit, **ESE** - End Semester Examination; **PA** - Progressive Assessment*

5. COURSE MAP (with sample COs, PrOs, UOs, ADOs and topics)

This course map illustrates an overview of the flow and linkages of the topics at various levels of outcomes (details in subsequent sections) to be attained by the student by the end of the course, in all domains of learning in terms of the industry/employer identified competency depicted at the centre of this map.

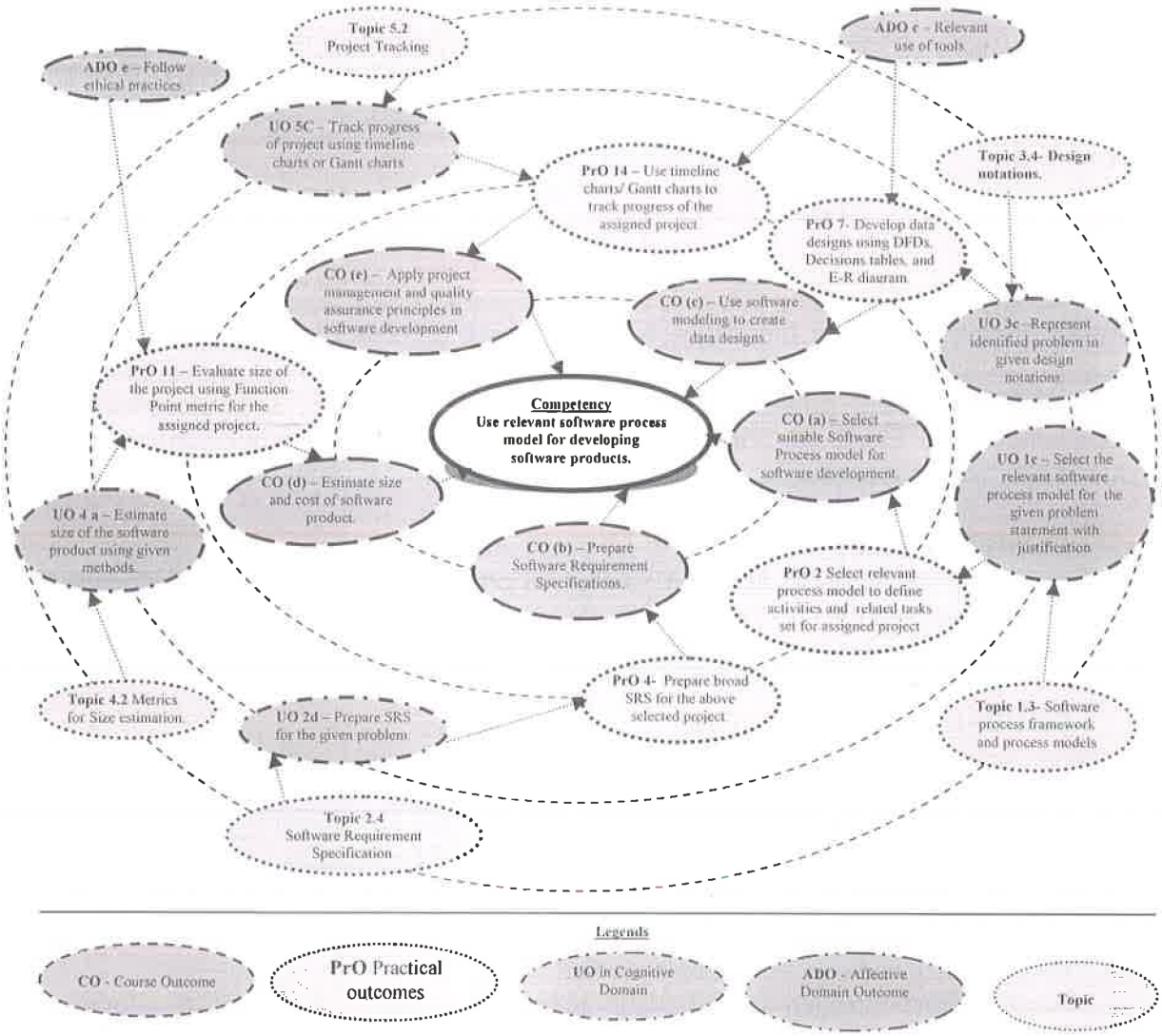

Figure 1 - Course Map

6. SUGGESTED PRACTICALS/ EXERCISES

The practicals in this section are PrOs (i.e. sub-components of the COs) to be developed and assessed in the student for the attainment of the competency.

S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
1	Write problem statement to define the project title with bounded scope of the project.	I	02*
2	Select relevant process model to define activities and related tasks set for assigned project.	I	02*
3	Gather application specific requirements for assimilate into RE (Requirements engineering) model.	II	02*
4	Prepare broad SRS (software requirement software) for the above selected project.	II	02*
5	Prepare use-cases and draw use – case diagram using Software Modeling Tool.	II	02*

S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
6	Develop the activity diagram to represent flow from one activity to another for software development.	II	02
7	Develop data designs using DFDs (data flow diagram), Decision tables and E-R (entity-relationship) diagram.	III	02*
8	Draw class diagram, Sequence diagram, Collaboration diagram, State Transition Diagram for the assigned project.	III	02
9	Write test cases to validate requirements of assigned project from SRS document.	III	02*
10	Identify risks involved in the project and prepare RMMM (RMMM-Risk Management, Mitigation and Monitoring) plan.	IV	02
11	Evaluate size of the project using Function point metric for the assigned project.	IV	02*
12	Estimate cost of the project using COCOMO (Constructive Cost Model) / COCOMO II approach for the assigned project.	IV	02*
13	Use CPM (Critical Path Method) / PERT (Programme Evaluation and Review Technique) for scheduling the assigned project.	V	02*
14	Use Timeline charts/ Gantt charts to track progress of the assigned project.	V	02
15	Prepare SQA plan that facilitates various attributes of quality of process.	V	02*
16	Prepare SQA plan that facilitates various attributes of quality of product.	V	02*
	Total		**32**

Note

i. *To carry out above listed practical /tasks, relevant* software tool may be chosen (preferably open-source based).

ii. *A suggestive list of PrOs is given in the above table. More such PrOs can be added to attain the COs and competency. A judicial mix of minimum 12 or more practical need to be performed, out of which, the practicals marked as '*' are compulsory, so that the student reaches the 'Application Level' of Bloom's Taxonomy' as generally required by the industry.*

iii. *The 'Process' and 'Product' related skills associated with each PrO is to be assessed according to a suggested sample given below:*

i.

S.No.	Performance Indicators	Weightage in %
1	Problem selection and its feasibility study	20
2	Logical thinking to decompose problem into modules	30
3	Ability to Estimate size and cost of a software	30
4	Presentation and technical documentation skills	10
5	Submission of reports within time.	10
	Total	**100**

The above PrOs also comprise of the following social skills/attitudes which are Affective Domain Outcomes (ADOs) that are best developed through the laboratory/field based experiences:

a. Follow safety practices.
b. Practice good housekeeping.
c. Demonstrate working as a leader/a team member.

d. Follow ethical practices.

The ADOs are not specific to any one PrO, but are embedded in many PrOs. Hence, the acquisition of the ADOs takes place gradually in the student when s/he undertakes a series of practical experiences over a period of time. Moreover, the level of achievement of the ADOs according to Krathwohl's 'Affective Domain Taxonomy' should gradually increase as planned below:

- 'Valuing Level' in 1st year
- 'Organising Level' in 2nd year and
- 'Characterising Level' in 3rd year.

7. MAJOR EQUIPMENT/ INSTRUMENTS REQUIRED

The major equipment with broad specification mentioned here will usher in uniformity in conduct of experiments, as well as aid to procure equipment by authorities concerned.

S. No.	Equipment Name with Broad Specifications	Pro. S. No.
1	Hardware: Personal computer, (i3-i5 preferable), RAM minimum 2 GB	For all Experiments
2	Operating system: Windows 7/Windows 8/Windows 10/LINUX or any other .	
3	Software tools: Any UML tool	

8. UNDERPINNING THEORY COMPONENTS

The following topics are to be taught and assessed in order to develop the sample UOs given below for achieving the COs to attain the identified competency. More UOs could be added.

Unit	Unit Outcomes (UOs) (in cognitive domain)	Topics and Sub-topics
Unit – I Software Development Process	1a. Suggest the attributes that match with standards for the given software application. 1b. Recommend the relevant software solution for the given problem with justification. 1c. Select the relevant software process model for the given problem statement with justification. 1d. Suggest the relevant activities in Agile Development Process in the given situation with justification	1.1 Software, Software Engineering as layered approach and its characteristics, Types of software. 1.2 Software development framework. 1.3 Software Process Framework, Process models: Perspective Process Models, Specialized Process Models. 1.4 Agile Software development: Agile Process and its importance, Extreme Programming, Adaptive Software Development, Scrum, Dynamic Systems Development Method (DSDM), Crystal. 1.5 Selection criteria for software process model.
Unit– II Software Requirement Engineering	2a. Apply the principles of software engineering for the given problem. 2b. Choose the relevant	2.1 Software Engineering Practices and its importance, Core principles. 2.2 Communication Practices, Planning Practices, Modelling practices.

Unit	Unit Outcomes (UOs) (in cognitive domain)	Topics and Sub-topics
	'requirement engineering' steps in the given problem. 2c. Represent the 'requirement engineering' model in the given problem. 2d. Prepare SRS for the given problem.	construction practices, software deployment (Statement and meaning of each principle for each practice). 2.3 Requirement Engineering: Requirement Gathering and Analysis, Types of requirements (Functional, Product, organizational, External Requirements), Eliciting Requirements, Developing Use-cases, Building requirement models, Requirement Negotiation, Validation. 2.4 Software Requirement Specification: Need of SRS, Format, and its Characteristics.
Unit– III Software Modelling and Design	3a. Identify the elements of analysis model for the given software requirements. 3b. Apply the specified design feature for software requirements modeling. 3c. Represent the specified problem in the given design notation. 3d. Explain the given characteristics of software testing. 3e. Prepare test cases for the given module.	3.1 Translating Requirement model into design model: Data Modelling. 3.2 Analysis Modelling: Elements of Analysis model. 3.3 Design modelling: Fundamental Design Concepts (Abstraction, Information hiding, Structure, Modularity, Concurrency, Verification, Aesthetics). 3.4 Design notations: Data Flow Diagram (DFD), Structured Flowcharts, Decision Tables. 3.5 Testing – Meaning and purpose, testing methods - Black-box and White-box, Level of testing – Unit testing. 3.6 Test Documentation – Test Case Template, test plan, Introduction to defect report, test summary report.
Unit-IV Software Project Estimation	4a. Estimate the size of the software product using the given method. 4b. Estimate the cost of the software product using the given empirical method. 4c. Evaluate the size of the given software using CoCoMo model. 4d. Apply the RMMM strategy in Identified risks for the given software development problem.	4.1 The management spectrum – 4P's 4.2 Metrics for Size Estimation: Line of Code(LoC), Function Points(FP). 4.3 Project Cost Estimation Approaches: Overview of Heuristic, Analytical, and Empirical Estimation. 4.4 COCOMO (Constructive Cost Model), COCOMO II. 4.5 Risk Management: Risk Identification, Risk Assessment, Risk Containment, RMMM strategy.
Unit –V Software	5a. Use the given scheduling technique for the	5.1 Project Scheduling: Basic principles, Work breakdown structure, Activity

Unit	Unit Outcomes (UOs) (in cognitive domain)	Topics and Sub-topics
quality assurance and Security	identified project. 5b. Draw the activity network for the given task. 5c. Prepare the timeline chart/ Gantt chart to track progress of the given project. 5d. Describe the given Software Quality Assurance (SQA) activity. 5e. Describe features of the given software quality evaluation standard.	network and critical path Method, Scheduling techniques (CPM, PERT). 5.2 Project Tracking: Timeline charts, Earned Value Analysis, Gantt Charts 5.3 Software Quality Management vs. Software Quality Assurance. Phases of Software Quality Assurance: Planning, Activities, audit, and review 5.4 Quality Evaluation standards: Six Sigma, ISO for software, CMMI: Levels, Process areas. 5.5 Software Security,Introduction to DevOps,Secure software engineering

***Note**: To attain the COs and competency, above listed UOs need to be undertaken to achieve the 'Application Level' of Bloom's 'Cognitive Domain Taxonomy'.*

9. SUGGESTED SPECIFICATION TABLE FOR QUESTION PAPER DESIGN

Unit No.	Unit Title	Teaching Hours	Distribution of Theory Marks			
			R Level	U Level	A Level	Total Marks
I	Software development process	08	04	04	04	12
II	Software Requirement Engineering	10	02	04	08	14
III	Software Modelling and Design	10	-	04	10	14
IV	Software Project Estimation	10	04	04	08	16
V	Software Project Management and quality assurance	10	04	04	06	14
Total		**48**	**14**	**20**	**36**	**70**

***Legends:** R=Remember, U=Understand, A=Apply and above (Bloom's Revised taxonomy)*
***Note**: This specification table provides general guidelines to assist student for their learning and to teachers to teach and assess students with respect to attainment of UOs. The actual distribution of marks at different taxonomy levels (of R, U and A) in the question paper may vary from above table.*
This specification table also provides a general guideline for teachers to frame internal end semester practical theory exam paper which students have to undertake.

10. SUGGESTED STUDENT ACTIVITIES

Other than the classroom and laboratory learning, following are the suggested student-related ***co-curricular*** activities which can be undertaken to accelerate the attainment of the various outcomes in this course: Students should conduct following activities in group and prepare reports of about 5 pages for each activity, also collect/record physical evidences for their (student's) portfolio which will be useful for their placement interviews:

a. Prepare journals based on practical performed in laboratory.
b. Give seminar on relevant topics.
c. Study and analyze college website from perspective of software application.
d. Study and analyze any available application software from perspective of software engineering.

11. SUGGESTED SPECIAL INSTRUCTIONAL STRATEGIES (if any)

These are sample strategies, which the teacher can use to accelerate the attainment of the various outcomes in this course:

a. Massive open online courses (***MOOCs***) may be used to teach various topics/sub topics.
b. ***'L' in item No. 4*** does not mean only the traditional lecture method, but different types of teaching methods and media that are to be employed to develop the outcomes.
c. About ***15-20% of the topics/sub-topics*** which is relatively simpler or descriptive in nature is to be given to the students for ***self-directed learning*** and assess the development of the COs through classroom presentations (see implementation guideline for details).
d. With respect to item No.10, teachers need to ensure to create opportunities and provisions for ***co-curricular activities***.
e. Guide student(s) in undertaking micro-projects.
f. Demonstrate students thoroughly before they start doing the practice.
g. Encourage students to refer different websites to have deeper understanding of the subject.
h. Observe continuously and monitor the performance of students.

12. SUGGESTED MICRO-PROJECTS

Only one micro-project is planned to be undertaken by a student that needs to be assigned to him/her in the beginning of the semester. In the first four semesters, the micro-project are group-based. However, in the fifth and sixth semesters, it should be preferably be ***individually*** undertaken to build up the skill and confidence in every student to become problem solver so that s/he contributes to the projects of the industry. In special situations where groups have to be formed for micro-projects, the number of students in the group should ***not exceed three.***

The micro-project could be industry application based, internet-based, workshop-based, laboratory-based or field-based. Each micro-project should encompass two or more COs which are in fact, an integration of PrOs, UOs and ADOs. Each student will have to maintain dated work diary consisting of individual contribution in the project work and give a seminar presentation of it before submission. The total duration of the micro-project should not be less than ***16 (sixteen) student engagement hours*** during the course. The student ought to submit micro-project by the end of the semester to develop the industry oriented COs.

A suggestive list of micro-projects are given here. Similar micro-projects could be added by the concerned faculty:

a. Study and analyze given software and write the characteristics and functions of the same.
b. Case study of application specific software product for requirement engineering
    i. Identify the problem statement
    ii. Perform feasibility analysis
    iii. Identify application specific requirements by following RE steps
    iv. Prepare SRS
c. Choose any problem statement and use data models to represent the solution
    i. Search and utilize different UML tools to represent models
d. Choose a problem, create activity network and use different project scheduling and tracking tools for the same.

13. SUGGESTED LEARNING RESOURCES

S. No.	Title of Book	Author	Publication
1	Software Engineering: A practitioner's approach	Pressman, Roger S.	McGraw Hill Higher Education, New Delhi, (Seventh Edition)ISBN 978–0–07–337597–7
2	Software Engineering Concepts	Fairly, Richard	McGraw Hill Education New Delhi – 2001, ISBN-13: 9780074631218
3	Software Engineering: Principles and practices	Jain, Deepak	Oxford University Press, New Delhi ISBN 9780195694840

14. SUGGESTED SOFTWARE/LEARNING WEBSITES

a. http://www.rspa.com/spi/
b. www.tutorialspoint.com//software_engineering/
c. www.versionone.com/agile-101/
d. www.sei.cmu.edu
e. www.nptel.ac.in/courses/
f. https://techbeacon.com/secure-devops