GUJARAT TECHNOLOGICAL UNIVERSITY (GTU)

Competency-focused Outcome-based Green Curriculum-2021 (COGC-2021)
Semester-IV

Course Title: Introduction to Software Engineering
(Course Code: 4340702)

Diploma programme in which this course is offered	Semester in which offered
Computer Engineering	4th Semester

1. RATIONALE

The Software engineering provides platform to develop Application or software in a systematic way. After studying the subject the students will be able to develop and design the software according to given requirements. It involves various steps in analysis and design of the system. It includes the knowledge of preparing project systematically. This course helps to know about various aspects of the software engineering so that the students will be able to understand the responsibilities while designing and implementing the project.

2. COMPETENCY

The course content should be taught and implemented with the aim to develop various types of related skills leading to the achievement of the following competency

- Identify and analyze problems in the field of Software development.

3. COURSE OUTCOMES (COs)

The practical exercises, the underpinning knowledge and the relevant soft skills associated with this competency are to be developed in the student to display the following COs:

The 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) Compare various software development process models.
b) Prepare software analysis and design using SRS, DFD and object oriented UML diagrams.
c) Prepare software development plan using project scheduling.
d) Prepare test-cases to test software functionalities.

4. TEACHING AND EXAMINATION SCHEME

Teaching Scheme (In Hours)			Total Credits (CI+T/2+P/2)	Examination Scheme				
				Theory Marks		Practical Marks		Total Marks
CI	T	P	C	CA	ESE	CA	ESE	
3	0	2	4	30	70	25	25	150

Out of 30 marks under the theory CA, 10 marks are for assessment of the micro-project to facilitate integration of COs and the remaining 20 marks is the average of 2 tests to be taken during the semester for the assessing the attainment of the cognitive domain UOs

required for the attainment of the COs.
Legends: CI-ClassRoom Instructions; T – Tutorial/Teacher Guided Theory Practice; P - Practical; C – Credit, CA - Continuous Assessment; ESE - End Semester Examination.

5. SUGGESTED PRACTICAL EXERCISES

The following practical outcomes (PrOs) that are the sub-components of the COs. *Some of the **PrOs** marked **'*'** are compulsory, as they are crucial for that particular CO. These PrOs need to be attained at least at the 'Precision Level' of Dave's Taxonomy related to 'Psychomotor Domain'.*

Sr.No	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Req
1	Describe various software development models with appropriate diagram	I	02
2	Write problem statement to define the project title with bounded scope of the project.	I	01
3	Select relevant process model to define activities and related tasks set for assigned project	I	01
4	Gather application specific requirements- Requirement gathering	II	02
5	Prepare broad SRS (software requirement software) for the above selected project	II	02
6	Develop data designs using DFDs (data flow diagram) and E-R (entity-relationship) diagram.	II	04
7	Prepare use-cases and draw use case diagram	II	02
8	Develop a class diagram for selected project	II	02
9	Develop Sequence diagram for selected project	II	02
10	Develop the activity diagram to represent flow from one activity to another for software development.	II	02
11	Evaluate size of the project using Function point metric for the assigned project.	III	02
12	Estimate cost of the project using COCOMO (Constructive Cost Model) / COCOMO II approach for the assigned project.	III	02
13	Use flow chart and Gantt charts to track progress of the assigned project. (Use Sprint burn down chart if agile model is selected).	III	02
14	Prepare various test case for selected project.	IV	02
	Total		28

Note

i. *Faculty should ensure that students select different problem statement in a group for practical 2 to 14. Size of group should not be more than 3 students.*

ii. *More **Practical Exercises** can be designed and offered by the respective course teacher to develop the industry relevant skills/outcomes to match the COs. The above table is only a suggestive list.*

iii. *The following are some **sample** 'Process' and 'Product' related skills (more may be added/deleted depending on the course) that occur in the above listed **Practical Exercises** of this course required which are embedded in the COs and ultimately the competency..*

S. No.	Sample Performance Indicators for the PrOs	Weightage in %
1	Problem selection and its feasibility study	30
2	Decompose problem into modules	20
3	Ability to estimate size and cost of the software	30
4	Presentation and Documentation Skills	10
5	Submission of reports within time	10
Total		**100**

6. MAJOR EQUIPMENT/ INSTRUMENTS AND SOFTWARE REQUIRED

These major equipment/instruments and Software required to develop PrOs are given below with broad specifications to facilitate procurement of them by the administrators/management of the institutes. This will ensure conduction of practical in all institutions across the state in proper way so that the desired skills are developed in students.

S. No.	Equipment Name with Broad Specifications	PrO. No.
1	Computer with latest configuration of windows or UNIX OS	All
2	Software tool : Any UML tool	All

7. AFFECTIVE DOMAIN OUTCOMES

The following ***sample*** Affective Domain Outcomes (ADOs) are embedded in many of the above-mentioned COs and PrOs. More could be added to fulfil the development of this competency.

a) Work as a leader/a team member.
b) Follow ethical practices.

The ADOs are best developed through the laboratory/field-based exercises. Moreover, the level of achievement of the ADOs according to Krathwohl's 'Affective Domain Taxonomy' should gradually increase as planned below:

i. 'Valuing Level' in 1st year
ii. 'Organization Level' in 2nd year.
iii. 'Characterization Level' in 3rd year.

8. UNDERPINNING THEORY

The major Underpinning Theory is formulated as given below and only higher level UOs of *Revised Bloom's taxonomy* are mentioned for development of the COs and competency in the students by the teachers. (Higher level UOs automatically includes lower level UOs in them).

If required, more such higher level UOs could be included by the course teacher to focus on attainment of COs and competency.

Unit	Unit Outcomes (UOs)	Topics and Sub-topics
Unit –1: Software Process Models	1a. Define Software Engineering. 1b. Recommend the relevant software solution for the given problem. 1c. Describe Generic Framework Activity 1d. Select the relevant software process model for the given problem statement with justification. 1e.. Suggest the relevant activities in Agile Development Process in the given situation with justification	1.1 Defining software 1.2 Software Application Domain i. System Software ii. Application Software iii. Embedded Software iv. Web Application v. Artificial Intelligence Software 1.3 Software Engineering – A layered Approach 1.4 Generic Process Model 1.5 Generic Framework Activity, Umbrella activity 1.6 Software Development Models i. Waterfall Model ii. Incremental Process Model iii. Prototype Model iv. Spiral Model 1.7 Agile Development Model i. Agility Principles ii. Agile Model vs Iterative Waterfall Model 1.8 Types of widely used Agile Models i. Extreme Programming(XP) ii. Scrum
Unit– 2: Software Requirement Analysis and Design	2a. Identify Software requirements for the given problem 2b. Prepare SRS from the requirement analysis 2c. Represent the specified problem in the given design notation – DFD 2d. Draw the relevant UML diagrams for the given problem	2.1 Requirement Gathering and Analysis 2.2 Software Requirement Specification (SRS) i. Characteristic ii. Customer requirement iii. Functional Requirement 2.3 Software Requirement Specification(SRS) 2.4 Software Design i. Characteristics of good software design ii. Analysis v/s design 2.5 Cohesion & Coupling i. Classification of cohesion ii. Classification of coupling **Function Oriented Software Design** 2.6 Data Flow Diagram(DFD) i. Context Diagram ii. Level 1 DFD

		Object Modeling with UML 2.7 Use case Diagram 2.8 Class Diagram 2.9 Sequence Diagram 2.10 Activity Diagram
Unit– 3: Software Project Estimation & Scheduling	3a. Estimate the size of the software product using the given method 3b. Evaluate the size of the given software using COCOMO model 3c.Prepare the Flow chart/ Gantt chart/ Sprint burn down chart to track progress of the given project.	3.1 Responsibility of software project Manager 3.2 Metrics for Size Estimation i. Line of Code ii. Function Points 3.2 Project Estimation Techniques using COCOMO model 3.3 Project Scheduling i. Gantt Chart ii. Flow Chart iii. Sprint burn down chart for agile model 3.3 Risk Management i. Risk Identification ii. Risk Assessment iii. Risk Control
Unit–4: Software Coding and Testing	4c. Describe different code review techniques 4d. Prepare test cases for the given module.	**Coding** 4.1 Coding standards and guidelines 4.2 Code review i. Code Work through ii. Code Inspection 4.2 Software Documentation i. Internal Documentation ii. External Documentation **Testing** 4.3 Testing Fundamentals 4.4 Functional Testing – Black box testing 4.5 Structural Testing – White box testing 4.6 Overview of Alpha & Beta Testing 4.7 Overview of Unit testing & Integration testing 4.8 Test Documentation – test case templates

***Note**: The UOs need to be formulated at the 'Application Level' and above of Revised Bloom's Taxonomy' to accelerate the attainment of the COs and the competency.*

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 Process Models	10	06	08	04	18
II	Software Requirement Analysis and Design	14	04	08	08	20
III	Software Project Estimation & Scheduling	10	06	08	04	18
IV	Software Coding and testing	08	04	06	04	14
Total		**42**	**22**	**32**	**16**	**70**

Legends: *R=Remember, U=Understand, A=Apply and above (Revised Bloom's taxonomy)*

Note*: This specification table provides general guidelines to assist student for their learning and to teachers to teach and question paper designers/setters to formulate test items/questions assess the attainment of the UOs. The actual distribution of marks at different taxonomy levels (of R, U and A) in the question paper may vary from above table.*

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 small reports (of 1 to 5 pages for each activity). For micro project report should be as per suggested format, for other activities students and teachers together can decide the format of the report. Students should also collect/record physical evidences such as photographs/videos of the activities for their (student's) portfolio which will be useful for their placement interviews:

a) Undertake micro-projects in teams.

b) Prepare charts for various models, SDLC life cycles, UML notations etc.

c) Prepare SRS documents based on case study.

d) Discuss various case studies available on internet.

e) An hour of problem solving for various case study topics may be organized and students are encouraged to participate

f) Students are encouraged to register themselves in various MOOCs such as: Swayam, edx, Coursera, Udemy etc to further enhance their learning.

g) Encourage students to interact with the industry person to discuss and gather information of current trends, models, documentation, testing methods and different tools used in industry.

11. SUGGESTED SPECIAL INSTRUCTIONAL STRATEGIES

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) Guide student(s) in undertaking micro-projects.

c) Managing Learning Environment

d) Diagnosing Essential Missed Learning concepts that will help for students.

e) Guide Students to do Personalized learning so that students can understand the course material at his or her pace.

f) Encourage students to do Group learning by sharing so that teaching can easily be enhanced.

g) ***'CI" in section No. 4***means different types of teaching methods that are to be employed by teachers to develop the outcomes.

h) About ***20% of the topics/sub-topics*** which are relatively simpler or descriptive in nature is to be given to the students for ***self-learning***, but to be assessed using different assessment methods.

i) With respect to ***section No.10***, teachers need to ensure to create opportunities and provisions for ***co-curricular activities***.

j) Guide students on how to address issues on environment and sustainability using the knowledge of this course

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 (group of 3 to 5). However, **in the fifth and sixth semesters**, 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 work load on each student due to the micro-project should be about ***16 (sixteen) student engagement hours*** (i.e., about one hour per week) during the course. The students ought to submit micro-project by the end of the semester (so that they develop the industry-oriented COs).

A suggestive list of micro-projects is given here. This should relate highly with competency of the course and the COs. Similar micro-projects could be added by the concerned course teacher:

a) Case study on student management system
b) Case study on library management system
c) Case study on hospital management system
d) Case study on banking system
e) Case study on collage management system
f) Case study on movie ticket booking system
g) Case study on online food ordering application
h) Case study on online shopping
i) Case study on hotel management system
j) Case study on bus ticket reservation system
k) Case study on railway ticket reservation system
l) Case study on flight ticket reservation system
m) Case study on leave management system in large organization
n) Case study on stock management system in mall
o) Case study on attendance management system
p) Case study on vehicle rental system
q) Case study on hospital appointment booking
r) Case study on gym management system
s) Case study on Tours and Travel Management
t) Case study on hostel management system
u) Case study on employee management system

13. SUGGESTED LEARNING RESOURCES

Sr. No.	Title of Book	Author	Publication with place, year and ISBN
1	Software Engineering: A Practitioner's Approach	Roger S. Pressman	Tata McGraw Hill, 2010, ISBN :978-007-126782-3
2	Fundamentals of Software Engineering	Rajib Mall	PHI,2018, ISBN:978-93-88028-02-8
3	Object Oriented Modeling and design with UML	Michael R Blaha and James R Rambaugh	Pearson Prentice Hall, 2009 ISBN:978-81-317-1106-4

14. SUGGESTED LEARNING WEBSITES

a) https://www.javatpoint.com/
b) https://www.geeksforgeeks.org/
c) https://www.tutorialspoint.com/
d) www.w3schools.com
e) https://www.techtarget.com/searchsoftwarequality/definition/agile-software-development

15. PO-COMPETENCY-CO MAPPING

Semester IV	Introduction to Software Engineering (Course Code: 4340702)									
	POs and PSOs									
Competency & Course Outcomes	**PO 1 Basic & Discipline specific knowledge**	**PO 2 Problem Analysis**	**PO 3 Design/ development of solutions**	**PO 4 Engineering Tools, Experimentation &Testing**	**PO 5 Engineering practices for society, sustainability & environment**	**PO 6 Project Management**	**PO 7 Life-long learning**	**PSO 1**	**PSO 2**	**PSO 3** (If needed)
Competency **Identify and analyze problems in the field of Software development**										
Course Outcomes CO a) Compare various software development process models.	**2**	-	-	-	-	-	-			
CO b) Prepare software analysis and design using SRS, DFD and object oriented UML diagrams.	**2**	**3**	**2**	**2**	-	**2**	**2**			
CO c) Prepare software development plan using project scheduling.	**2**	**1**	**2**	**2**	-	**3**	**2**			
CO d) Prepare test-cases to test software functionalities	**2**	-	-	-	-	-	-			

Legend: '**3'** for high, '**2**' for medium, **'1'** for low or **'-'** for the relevant correlation of each competency, CO, with PO/ PSO

16. COURSE CURRICULUM DEVELOPMENT COMMITTEE

GTU Resource Persons

S. No.	Name and Designation	Institute	Contact No.	Email
1	Shri P. P. Kotak Principal	S S Gandhi Polytechnic, Surat	9825469617	Kotakp2003@yahoo.com
2	Smt. M. P. Mehta HOD, Computer	Government Polytechnic, Himmatnagar	9879578273	manishamehtain@gmail.com
3	Smt. Avani S Galathiya	R C Technical Institute, Ahmedabad	9904126016	asgalathiya@gmail.com
4	Smt. Soniya S Dadhania	R C Technical Institute, Ahmedabad	9974006746	soniyasdadhania@gmail.com