Course Syllabus
Gyanmanjari Diploma Engineering College
Semester-3 (Diploma)

Subject: Fundamental of Software Engineering–DETCE13206

Type of course: Major (Core)

Prerequisite: Basic Knowledge of Programming Language

Rationale:

This course is designed to provide students with a comprehensive understanding of the principles, techniques, methodologies, and tools necessary for effective software development. It covers software development lifecycle models, requirements engineering, design principles. It teaches techniques for gathering, analyzing, and managing software requirements. It involves various steps in analysis and design of the system. This syllabus aids students in honing skills to develop, design and analyze software projects throughout advanced semesters of the diploma program.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks					Total Marks
				Theory Marks		Practical Marks		CA	
CI	T	P	C	ESE	MSE	V	P	ALA	
3	0	2	4	60	30	10	20	30	150

Legends: CI-ClassRoom Instructions; T – Tutorial; P - Practical; C – Credit; ESE - End Semester Examination; MSE- Mid Semester Examination; V – Viva; CA - Continuous Assessment; ALA- Active Learning Activities.

Course Content:

Sr. No	Course Content	Hrs.	% Weightage
1	**Introduction to Software and Software Engineering** Define Software And Software Engineering, Software Application Domain, Software Myths, Software Engineering as a Layered Technology, Process Framework, Common Process Framework , Umbrella Activities, Need for Software Process Model, Software Process Model/Software Life Cycle Model, Waterfall Model, Iterative Waterfall Model, Incremental Model, Prototype Model, Spiral Model, Product and Process.	17	35
2	**Agile Development** What is Agile and Agility?, Concept of agile Process and characteristics of agile process, Agile Principles/Agile Manifesto, Agile Models, Extreme Programming(XP), Scrum, Adaptive Software development(ASD).	08	20
3	**Requirement Analysis and Design** What is Requirement? , Requirement Gathering Activities, Requirement Analysis, Functional and Non functional Requirements, Software Requirement Specification, Importance of SRS, Users of SRS, Characteristics of Good SRS, Example of SRS, Software Design, Characteristics of Software Design, Analysis v/s Design, Cohesion & Coupling, Classification of Cohesion, Classification or Coupling, Difference between Cohesion and Coupling	12	25
4	**Function Oriented Software Design** Data Flow Diagram (DFD), Context Diagram, Level 1 DFD Object Modeling With UML, Use Case Diagram, Class Diagram, Sequence Diagram, Activity Diagram	08	20

Continuous Assessment:

Sr. No	Active Learning Activities	Marks
1	**Process Model Comparison Matrix** Students will create a comparison matrix to analyze and compare different software process models. They will research and identify key characteristics, such as development phases, feedback mechanisms, and flexibility, for each model. Then, students will collaboratively fill in the matrix, highlighting the differences and similarities between the process model and upload it on GMIU web portal.	10

2	**Requirements Prioritization Game** Students will engage in an activity where they are preparing report with a list of requirements for a software project. Each requirement will have different attributes such as importance, urgency, and feasibility and upload it on GMIU web portal.	10
3	**Use Case Puzzle** Students will be given a use case diagram with missing elements, such as actors, use cases, or relationships. Working individually, they will analyze the diagram and fill in the missing parts based on their understanding of use case and upload it on GMIU web portal.	10
	Total	30

Suggested Specification table with Marks (Theory):60

Distribution of Theory Marks **(Revised Bloom's Taxonomy)**						
Level	Remembrance (R)	Understanding (U)	Application (A)	Analyze (N)	Evaluate (E)	Create (C)
Weightage %	30%	40%	20%	10%	-	-

Course Outcome:

After learning the course the students should be able to:	
CO1	Understand the fundamental concepts of software engineering and different Models.
CO2	Understand modern Agile Development.
CO3	Prepare software analysis and design using SRS, DFD and object-oriented UML diagrams.
CO4	Demonstrate Software design and its implementation.

List of Practical

Sr. No	Description	Unit No	Hrs.
1	Draw SDLC model and explain its various phases properly.	1	2

2	Describe various software development models with appropriate diagram.	1	4
3	Write problem statement to define the project title with bounded scope of the project	1	2
4	Select relevant process model to define activities and related tasks set for assigned project	1	4
5	Gather application specific requirements-Requirement gathering	3	2
6	Prepare broad SRS (software requirement software) for the above selected project	3	4
7	Draw use case diagram for given project.	4	2
8	Develop a class diagram for selected project	4	2
9	Develop Sequence diagram for selected project	4	2
10	Develop the activity diagram to represent flow from one activity to another for software development	4	2
11	Develop data designs using DFDs (data flow diagram)	4	4
		Total	30

Instructional Method:

The course delivery method will depend upon the requirement of content and need of students. The teacher in addition to conventional teaching method by black board, may also use any of tools such as demonstration, role play, Quiz, brainstorming, MOOCs etc.

From the content 10% topics are suggested for flipped mode instruction.

Students will use supplementary resources such as online videos, NPTEL/SWAYAM videos, e-courses, Virtual Laboratory.

The internal evaluation will be done on the basis of Active Learning Assignment.

Practical/Viva examination will be conducted at the end of semester for evaluation of performance of students in laboratory.

Reference Books:

[1] Software Engineering, Technical publications, A.APuntambekar
[2] Roger S.Pressman, Software Engineering- A practitioner's Approach, McGraw-Hill International Edition
[3] Ian Sommerville, Software engineering, Pearson education Asia
[4] Pankaj Jalote, Software Engineering – A Precise Approach Wiley
[5] Behhforoz & Frederick Hudson, Software Engineering Fundamentals, OXFORD