Gyanmanjari
Innovative University

Course Syllabus
Gyanmanjari Institute of Technology
Semester-5 (B.Tech.)

Subject: Software Engineering-BETCE15322

Type of course: Professional Core and Professional Elective Courses

Prerequisite: Object Oriented Programming fundamental, UML.

Rationale:

This course provides an in-depth understanding of Software Development Life Cycle (SDLC), Agile development models, software testing methodologies, and software process improvement. It focuses on industry-relevant concepts, testing tools, and quality assurance techniques to ensure students are prepared for modern software engineering roles.

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks					Total Marks
CI	T	P	C	Theory Marks		Practical Marks		CA	
				ESE	MSE	V	P	ALA	
2	0	2	3	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 Engineering:** Role of Software in Industry, Software Myths & Software Engineering as a Layered Technology, Software Process Models (Waterfall, Prototyping, RAD, Agile, Evolutionary Process Model, Component-Based Development), Product vs. Process.	04	15%

2	**Agile Software Development:** Agility and Agile Process Model, Scrum, Extreme Programming (XP), Kanban, Lean Software Development, Agile Project Management Tools (JIRA, Trello).	03	10%
3	**Managing Software Project:** Software Metrics (Process, Product, and Project Metrics), Software Project Estimation and Scheduling and Tracking, Risk Analysis & Management (Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation).	03	10%
4	**Requirement Analysis and Specification:** Requirement Engineering Process, Functional and Non-Functional Requirements, Software Requirement Specification (SRS) Document.	03	10%
5	**Software Design and architecture:** Design Concepts and Design Principal, Architectural Design, Component-Level Design, Function-Oriented vs. Object-Oriented Design, User Interface & Web Application Design.	04	15%
6	**Software Development and Testing:** Coding Standard and coding Guidelines, Code Review, Software Documentation, Testing Strategies, Testing techniques and Test Case, Test Suites Design, Testing Conventional Applications, Testing Object Oriented Applications, Testing Web and Mobile Applications, Testing Tools (Win runner, Load runner), Bug Report.	04	15%
7	**Quality Assurance and Management:** Software Quality Assurance (SQA), Formal Technical Reviews Software Reliability, Quality Standards: ISO 9000, CMM, Six Sigma, SQA Planning.	02	05%
8	**Software Maintenance and Configuration Management:** Types of Software Maintenance, Reengineering, Reverse Engineering, Forward Engineering, Software Configuration Management (SCM), Version & Change Control. **Devops:** What is Devops? DevOps Impotance and Benefits, DevOps Principles and Practices, 7 Cs of DevOps Lifecyvle for Business Agility, DevOps and Contiguous Testing, How to Choose Right DevOps Tools, Challenges with DevOps Implementation, Must Do Things for DevOps, Mapping My App to DevOps.	07	20%

Continuous Assessment:

Sr. No	Active Learning Activities	Marks
1	**UML Blueprint: Visualizing System Design:** In this activity, each student individually is required to select a real-world problem and gather detailed system requirements. Based on their analysis, they must identify key actors and processes involved in the system. Using this information, students will design at least two types of UML diagrams, such as a Use Case Diagram and a Class Diagram, utilizing tools like paper, MS Visio, or Lucid chart. Upon completion, the diagrams must be uploaded to the GMIU web portal.	10
2	**Project Sizing with COCOMO: From Estimates to Execution:** In this activity, each student will individually utilize COCOMO model to estimate the effort, development time, and team size required for a software project (real world problem) using the Constructive Cost Model (COCOMO). They will classify their project as Organic, Semi-Detached, or Embedded and apply mathematical formulas for precise calculations. The final report, including project classification and estimation results must be uploaded to the GMIU web portal.	10
3	**Hunt the Broken Links Using Xenu Tool:** Each student will individually analyze a publicly accessible website or their own blog using the Xenu Link Sleuth Tool, which is designed to detect broken or problematic links within a website. Each student will run a full website scan, identify issues such as broken internal or external links, redirect loops, and missing media files, and interpret the resulting status codes (e.g., 404 Not Found, 301 Moved Permanently). Based on their findings, students will prepare a concise Link Audit Report detailing the errors found, affected URLs. Upon completion student must upload their final report to the 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%	35%	15%	10%	05%	05%

Course Outcome:

After learning the course, the students should be able to:	
CO1	Understand and apply SDLC and Agile methodologies.
CO2	Develop a Software Requirements Specification (SRS).
CO3	Apply project management techniques and software design principles using both function-oriented and object-oriented approaches.
CO4	Implement software quality standards and testing strategies.
CO5	Utilize software maintenance techniques and tool like Devops for efficient software deployment and management.

List of Practical:

Sr. No	Description	Unit No	Hrs.
1	Study the complete Software Development Life Cycle (SDLC) and analyze various activities conducted as a part of various phases.	1	2
2	Develop requirements specification for a given problem.	4	4
3	Develop UML Use case diagram for a problem.	5	2
4	Develop UML Activity and Swim Lane diagram for a problem.	5	2
5	Develop UML Class diagram for a problem.	5	2
6	Develop UML Sequence diagram for a problem.	5	2
7	Develop UML Data Flow diagram for a problem.	5	2
8	Develop UML ER diagram for a problem.	5	2
9	Estimation of Test Coverage and structural complexity.	6	2
10	Design and develop test suites for a given software module by applying various testing techniques.	6	2
11	Test the login, Registration (Sign Up), Sign-in functionality of an OAuth module.	6	2
12	Use the Blisk tool to test the layout and responsiveness of your website or personal project.	6	2
13	Generate a bug report of any given definition.	6	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: A Practitioner's Approach – Roger S. Pressman, McGraw-Hill International Editions.
[2] Fundamentals of Software Engineering – Rajib Mall, Prentice Hall of India.
[3] Software engineering- Ian Sommerville, Pearson education Asia.
[4] Software Engineering – A Precise Approach- Pankaj Jalote, Wiley.
[5] Software Engineering Fundamentals, Behhforoz & Frederick Hudson, OXFORD.