



**Subject:** Software Engineering–MSCIT12507

**Type of course:** Major (Core)

**Prerequisite:** Basic knowledge of Software Development process with involves designing, Developing, and testing new technologies to solve real-world problems.

**Rationale:**

Software Engineering is the important lessons of software engineering only apply to large scale software development the skills needed to develop such systems have very little to do with the type of programming.

In fact, software engineering has to do with programming at all, Software Engineering is about the discipline needed to develop high quality software that can be understood, maintained and adapted over long periods of time by many different people.

An understanding of what software quality really means is central to understanding what software by attempting to understand, assess, and modify one another's programs, students will gain insight into the nature of software quality, and why ability to program is not sufficient for the construction of high-quality software.

**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	
4	1	0	5	60	30	10	-	50	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.*





**CourseContent:**

Sr. No	Course content	Hrs	% Weightage
1	<b>Introduction to Software and Software Engineering</b> Introduction to Software Engineering, Software Engineering Approach (Work), Software Process and It's Characteristics, Software Development Life Cycle (SDLC), Software Development Process Model, Waterfall Model, Prototyping Model, Iterative Enhancement Model (Incremental model), Spiral Model.	15	30%
2	<b>Software Design</b> Importance of Design: UML- Component of UML – Class Diagram, Object Diagram, Use Case Diagram, Activity Diagram Case study –Library management system, ticket reservation system, hospital management system and Online Customer service system.	10	10%
3	<b>Coding &amp; Testing:</b> Programming practice, way of Testing, Testing Fundamentals (errors, fault & failure), level of testing, types of testing, black box testing, gray box testing.	10	10%
4	<b>Agile Development</b> Agile Development Agility and Agile Process model, Agile Principles: 12 principles of Agile software, Scrum, crystal, FDD used in agile development model.	15	30%
5	<b>Software Requirement Specifications(SRS):</b> Introduction to Software Requirement Specification, Need of SRS/Role of SRS Requirement Specification, Component of SRS, Data Flow Diagrams, Entity Relationship Diagrams, Decision Tables, SRS Document.	10	20%

**Continuous Assessment:**

Sr. No	Active Learning Activities	Marks
1	<b>Requirements Crafting:</b> Students have to build an SRS document of the project and upload it on GMIU web portal.	10
2	<b>The Art of Effective Presenting:</b> Create a presentation on given topic, prepare its video and upload it on GMIU web portal.	10
3	<b>Diagram Creation:</b> Student will create Advance UML Class diagram in chart paper on given	10





	topic and upload it on GMIU web portal.	
4	<b>System life cycle:</b> Students have to create Spiral and waterfall Model in the Designing tools or give definition and upload on GMIU web portal.	10
5	<b>A Case Study:</b> Students will perform case study using modern software engineering like Agile Software Development and upload its documentation on GMIU web portal.	10
Total		50

### 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%	0	0

**Note:** This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

### Course Outcome:

After learning the course the students should be able to:	
CO1	Understand the concepts of software engineering, software process model, requirements of software engineering and how to manage Client's requirement.
CO2	Know design concepts and user interface
CO3	Know coding concepts and designing interface
CO4	Understand agile methodology and scrum development processing.
CO5	Build an SRS document of the project.

### Tutorial:

Sr. No.	Tutorial	Hrs.
1	Preparing Software Requirements Specifications:SRS	4
2	Identifying Domain Classes from the Problem Statements	1
3	Modeling UML Class Diagrams and Sequence diagrams	1
4	Modeling UML Use Case Diagrams and Capturing Use Case Scenarios	1
5	E-R Modeling	1
6	State chart and Activity Modeling	1





7	Modeling Data Flow Diagrams	1
8	Estimation of Project Metrics	1
9	Estimation of Test Coverage Metrics and Structural Complexity	2
10	Designing Test Suites	2

**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] Waman S. Jawadekar: Software Engineering– Principles and Practices: TMGH Publication
- [2] Roger S Pressman : Software Engineering-Software Engineering: A Practitioner's Approach
- [3] Ian Sommerville: Software Engineering- Software Engineering: Seventh Edition
- [4] A. A. Puntambekar: Software Engineering
- [5] Pankaj Jalote's: Software Engineering: A Precise Approach

