AMITY UNIVERSITY

UTTAR PRADESH

Course Title: Software Engineering

Credit Units:

L	T	P/S	SW/FW	TOTAL CREDIT UNITS
3	1	2	-	5

Course Level: UG
Course Code: IT301

Course Objectives:

1. To make the students to develop skills that will enable them to construct software of high quality – software that is reliable, and that is reasonably easy to understand, modify and maintain
2. To make student learn how to use available resources to develop software, reduce cost of software and how to maintain quality of software

Pre-requisites: Student should have knowledge of development languages of software

Course Contents/Syllabus:

	Weightage (%)
Module I Introduction	**20**
▪ Software life cycle models: Waterfall, Prototype, Evolutionary and Spiral models ▪ Agile Methodology ▪ Overview of Quality Standards like ISO 9001, SEI-CMM	
Module II Software Metrics and Project Planning	**20**
▪ Size Metrics like LOC, Token Count, Function Count ▪ Design Metrics ▪ Data Structure Metrics ▪ Information Flow Metrics ▪ Cost estimation, static, Single and multivariate models, COCOMO model, Putnam Resource Allocation Model ▪ Risk management	

Module III Software Requirement Analysis, design and coding	**20**
▪ Problem Analysis ▪ Software Requirement and Specifications ▪ Behavioural and non-behavioural requirements ▪ Software Prototyping ▪ Cohesion & Coupling ▪ Classification of Cohesiveness & Coupling ▪ Function Oriented Design, Object Oriented Design, User Interface Design ▪ Top-down and bottom-up Structured programming, Information hiding	
Module IV Software Reliability, Testing and Maintenance	**25**
▪ Failure and Faults ▪ Reliability Models: Basic Model, Logarithmic Poisson Model ▪ Software process ▪ Functional testing: Boundary value analysis, Equivalence class testing, Decision table testing, Cause effect graphing ▪ Structural testing: path testing ▪ Data flow and mutation testing, unit testing, integration and system testing, Debugging, Testing Tools, & Standards. ▪ Management of maintenance, Maintenance Process, Maintenance Models, Reverse Engineering, Software RE-engineering	
Module V UML	**15**
▪ Introduction to UML ▪ Introduction to Rational Rose Environment ▪ Class Diagram in UML ▪ Use Case Diagram in UML ▪ State Diagram in UML ▪ Object Diagram in UML ▪ Activity Diagram in UML ▪ Sequence Diagram in UML ▪ Collaboration Diagram in UML ▪ Componant Diagram in UML ▪ Deployment Diagram in UML	

Student Learning Outcomes:

1. Understand the software life cycle models;
2. Understand the importance of the software development process;

3. Design and develop correct and robust software products,
4. Understand business requirements pertaining to software development

Pedagogy for Course Delivery:

The course would be covered under theory and laboratory. In addition to assigning project–based learning, early exposure to hands-on design to enhance the motivation among the students. It incorporates designing of problems, analysis of solutions submitted by the students groups and how learning objectives were achieved. Continuous evaluation of the students would be covered under quiz, viva etc.

Lab/ Practical's details, if applicable:

List of Experiments:

- Class Diagram in UML
- Use Case Diagram in UML
- State Diagram in UML
- Object Diagram in UML
- Activity Diagram in UML
- Sequence Diagram in UML
- Collaboration Diagram in UML
- Componant Diagram in UML
- Deployment Diagram in UML

Assessment/ Examination Scheme:

Theory L/T (%)	Lab/Practical/Studio (%)	Total
80%	20%	100%

Theory Assessment (L&T):

Continuous Assessment/Internal Assessment					End Term Examination
Components (Drop down)	**Mid Term Exam**	**HA**	**Viva**	**Attendance**	
Weightage (%)	10	8	7	5	70

Lab/ Practical/ Studio Assessment:

	Continuous Assessment/Internal Assessment **30**				**End Term Examination**		
Components (Drop down	Lab Performance	Lab File	Viva	Attendance			
Weightage (%)	10	10	5	5		**70**	

Text Reading:

1. K. K. Aggarwal & Yogesh Singh, “Software Engineering”, 2nd Ed, New Age International, 2005.
2. R. S. Pressman, “Software Engineering – A practitioner’s approach”, 5th Ed., McGraw Hill Int. Ed., 2001.
3. Rajib Mall, Fundamentals of Software Engineering, Prentice Hall India.
4. Ian Summerville, Software Engineering, Addison-Wesley.

References:

1. R. Fairley, “Software Engineering Concepts”, Tata McGraw Hill, 1997.
2. P. Jalote, “An Integrated approach to Software Engineering”, Narosa, 1991.