Annexure 'CD – 01'

AMITY UNIVERSITY KOLKATA

FORMAT FOR COURSE CURRICULUM

Course Title: **Software Engineering**

Course Code: **IT301**

Credit Units:

L	T	P/S	SW/FW	No. of PSDA	TOTAL CREDIT UNITS
3	-	2	2	2	5

Course Level:UG

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	**15**
Descriptors/Topics : • 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	**25**
Descriptors/Topics : • Size Metrics like LOC, Token Count, Function Count • Design Metrics • Data Structure Metrics • Information Flow Metrics • Overview of Project Planning	

• Cost estimation, static, Single and multivariate models, COCOMO model, Putnam Resource Allocation Model • Risk management	
Module III : Software Requirement Analysis, design and coding **Descriptors/Topics ;** • 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	**20**
Module IV : Software Reliability, Testing and Maintenance **Descriptors/Topics :** • 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 Reengineering	**25**
Module V: UML and DevOps **Descriptors/Topics :** • 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 • Component Diagram in UML	**15**

• Deployment Diagram in UML • DevOps: Introduction, Life cycle, tools, Difference between Agile and DevOps	

Course Learning Outcomes:

Students will be able to:

- Understand software life cycle models and quality standards
- Interpret the importance of the software planning using project metrics
- Identify software requirements and develop software design
- Design software test cases and understand maintenance process
- Illustrate entire software designing with various diagrams

Pedagogy for Course Delivery:

The class will be taught using remote teaching methodology. Students' learning and assessment will be on the basis of four quadrants and flipped class method. E-content will be also provided to the students for better learning. The class will be taught using theory, practical and case-based method.

Lab/ Practical 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
- Component Diagram in UML
- Deployment Diagram in UML

List of Professional Skill Development Activities (PSDA)

I. Quiz

II. Case Study/ Group Presentation

Assessment/ Examination Scheme:

Theory L/T (%)	Lab/Practical/Studio (%)	End Term Examination
80	**20**	**100**

Theory Assessment (L&T):

Continuous Assessment/Internal Assessment							End Term Examination
Components (Drop down)	**Minor Exam**	**Home Assignment**	**Viva**	**Attendance**	**Quiz**	**Case Study/ Group Presentation**	
Weightage (%)	10	8	7	5	5	5	60

Lab/ Practical/ Studio Assessment:

Continuous Assessment/Internal Assessment					End Term Examination
Components(Dropdown	**Performance**	**Lab Record**	**Viva**	**Attendance**	**EE**
Weightage(%)	15	10	10	5	60

Text:

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.