CSEN1131	SOFTWARE ENGINEERING	L	T	P	S	J	C
		3	0	2	0	0	4
Pre-requisite	None						
Co-requisite	None						
Preferable exposure	None						

Course Description:

The purpose of this course is to impart knowledge on the basic principles of software engineering and enabling the learner to understand software lifecycle stages. Systematic development of software products or solutions is emphasized throughout the course to enable the student ensure quality of development activities.

Course Educational Objectives:

- Provide Introduction to Software Engineering and process of Software production
- Enable understanding of widely varying nature of software solutions and domain and technology aspects of software
- Deconstruct different stages of Software products' life cycle and software Evolution, life cycle processes
- Facilitate Analysis of requirements for software solution development
- Summarize architecture, design, and implementation considerations of software solution
- Exposure to quality aspects across the stages, planning aspects

UNIT 1 Introduction to Software Engineering 9 hours

The story of Software development and issues faced, Need for Systematic process for addressing issues,Products, custom solutions, services, domains, Technologies, Software life cycle, software development lifecycle, software release process, source control, versioning, maintenance of software. DevOps.
Software Development Processes: Waterfall, Iterative, Spiral.

UNIT 2 Software development phases and processes 9 hours

Software development and processes – RAD, RUP, Agile: Scrum, Prototyping
Development phases of Software in relation to Processes
What to develop? – Requirements gathering and Analysis, Types- functional, non-functional, system, User Interface, quality requirements and putting together– UML use cases, scenarios.

UNIT 3 Requirements, design and solution 9 hours

Considerations for architecture, design, Data, modules, interfaces – application architectures.

System design: modular design – cohesion and coupling, Structural Design -Top down, Bottom up approaches; data models, User Interface guidelines; UML Activity, Sequence, Component, Collaboration, Deployment diagrams

UNIT 4 Implementation and Ensuring Quality of software and Metrics 9 hours

How to implement – practices to follow for development – language, platform choices, coding practices, cost of bugs through life cycle

Quality from requirements to release and across versions: Faults and Fixes, Reliability models: Logarithmic Poisson Model

Testing mechanisms across life cycle: Functional, system integration, user testing, testing on different platforms.Testing Tools

Quality across versions and metrics for quality; Quality Models: ISO, CMM, Boehm, McCall; Metrics: Process and Product metrics - LOC, Function Points, Token Count

UNIT 5 9 hours

Software Management - Planning for software development

Estimation of time, resources, the cost for software development: COCOMO, Function Point, Putnam Resource Allocation Models

Planning activities and re-planning, Risk Analysis

Release mechanisms, Configuration Management, Licensing methods and Maintenance

Software Life Cycle Management - planning, tracking, communication, negotiation, delivery, quality aspects.

Lab Activities Suggested:

1. Implement weather modeling* using the quadratic solution in stages: hard-coding variables keyboard input, read from a file, for a single set of input, multiple sets of inputs.
   a. save all versions, debug, fix problems, create a Github account

2. Develop weather modeling using the quadratic model in teams of 5 using Waterfall, Iterative, Agile modes

3. a. Teams of 5 to work on gathering requirements for different simple projects related to University and student activities

b. Represent requirements in terms of lists, use cases, scenarios (UML)

c. try simple architecture and design of modules. Represent in activity, sequence, collaboration diagrams(UML)

4.a. Testing quadratic modeling of weather modeling example

b. Testing using open source testing tools: Selenium, Jmeter

5.a. Understand cost drivers using the COCOMO site for team projects

b. Create a project plan in Jira

Lab Infrastructure:

1. Eclipse, Visual Studio, SQL Server, MS Access, Oracle
2. StarUML /RationalPro, jira

TextBooks:

1. Ian Sommerville, "Software Engineering", 10th Edition, Pearson Education, 2015 (overall – Part I: 1 to 9, Part IV: parts of 23 to 25)

2. Klaus Pohl, Chris Rupp, "Requirements Engineering Fundamentals" 2nd Edition, RockyNook, 2015.

3. Rajib Mall, Fundamentals of Software Engineering, 4/e, PHI, 2009. (for metrics))

4. K. K. Aggarwal & Yogesh Singh, "Software Engineering", 3rd Edition, New Age International, 2008. (for metrics)

5. Steve Mcconnell, "Code complete", 2nd Edition, Microsoft Press, 2004, Print 2015 (for design)

6. Frederic P. Brooks, "The Mythical Man-Month: Essays on Software Engineering", Addison-Wesley,1995, print 2010 (for project management)

References:

1. Michael R Blaha, James R Rumbaugh, "Object-Oriented Modeling and Design with UML", 2nd Edition,Pearson Education, 2005
2. Axel van Lamsweerde, "Requirements Engineering" Wiley Publications, 2009
3. http://vlabs.iitkgp.ernet.in/se/
4. http://softwarecost.org/tools/COCOMO/

Course Outcomes:

After successful completion of the course the student will be able to:

1. Demonstrate understanding of the process of Software Development

2. Determine Suitability of processes for varying software applications development

3. Differentiate Development phases through the life cycle of software

4. Reflect on design choices and development standards

5. Check and verify software quality from requirements to release of software and across versions

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3					1							2		
CO2			3	2					2					2	
CO3		2	2												1
CO4		3											2		
CO5			3		2							2		2	

Note: 1 - Low Correlation 2 - Medium Correlation 3 - High Correlation

APPROVED IN:
BOS : 06-09-2021 **ACADEMIC COUNCIL: 01-04-2022**

SDG No. & Statement: 4 and 8

SDG Justification:

SDG 8 “Economic growth”: Software engineering helps build computer applications that help connect businesses with customers and other businesses, leading to economic growth.

SDG 4 “Education”: Software engineering helps build computer applications that can help enable online learning, taking free and good education to the doorstep of those who cannot access or afford it.