



बिरसा मुंडा ट्रायबल युनिवर्सिटी Birsa Munda Tribal University

राजपिपला, जि० नर्मदा Rajpipla, Dist. Narmda

Established by Tribal Development Department, Govt. of Gujarat

School of Computer Science

M. Sc. (Information Technology) Programme

Subject Code & Name: MITMC204 - Software Engineering

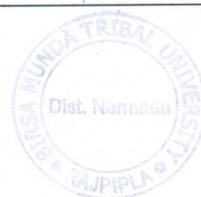
Teaching And Evaluation Scheme:

Teaching Scheme				Examination Scheme			
Credits				Component Weightage			
				CCE		SEE	
L	T	P	Total	TH	PWE	TH	PWE
2	-	2	4	25%	25%	25%	25%

Programme Name	M. Sc. (Information Technology)
Semester	2
Course Code	MITMC204
Course Title	Software Engineering
Course Content Type (Th. / Pr.)	Th.+ Pr.
Course Credit	2+2
Sessions + Lab Per Week	2+4
Total Teaching/ Lab Hours	30+60
* 2 Lab = 1 session	

Learning Objectives

Upon successful completion of this course, students will be able to:



- Understand the fundamentals of software engineering, software process models, and various SDLC approaches used in software development.
- Learn requirement gathering techniques, project planning methods, estimation models, and software design strategies for software system development.
- Understand software testing methods, maintenance activities, and configuration management techniques to improve software quality.
- Learn software quality measurement techniques and project documentation standards using software engineering tools.

Learning Outcomes

After successful completion of the course, students will be able to:

- Explain software engineering concepts and select appropriate SDLC models for different software projects.
- Prepare requirement specifications, software design documents, project plans, and risk management reports using standard software engineering practices
- Perform software testing, maintenance, verification, validation, and configuration management activities using appropriate testing approaches
- Develop project documentation including SRS, ER diagrams, DFDs, and software metrics reports for software applications.

CLO – PLO Mapping Matrix

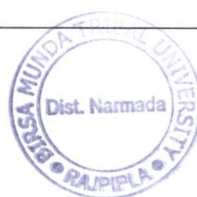
CLO/ PLO	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	3	3	2	2	1	2	1	1	2
CLO2	2	3	3	3	2	3	1	2	2
CLO3	3	3	3	3	1	2	2	1	2
CLO4	2	3	3	3	2	2	1	3	2

Detailed Contents



Unit	Topic/Sub Topic	Teaching Hours
I Introduction to Software Engineering and SDLC	Introduction to Software Engineering and SDLC • Definition and Importance of Software Engineering • Characteristics of Software • Software Process Framework • Role of Software Engineer • Software Development Life Cycle ○ Waterfall Model ○ Prototype Model ○ Spiral Model ○ Incremental Model ○ Agile Model ○ DevOps Basics	15
II Software Requirement analysis and specification, software Project Planning and designing	Software Requirement analysis and specification • Requirement engineering • Types of requirement • Requirement elicitation ○ Interviews ○ Brainstorming session ○ The use case approach • Requirement analysis ○ Data flow diagram ○ ER diagram ○ Software prototyping • Requirements documentation	15

	○ Software Requirement Specification (SRS) • Software project planning • Size estimation • Cost estimation ○ COCOMO Model • Software risk management • Software design • Conceptual and technical design • Modularity ○ Coupling ○ Cohesion • Strategy of design ○ Bottom-up design ○ Top-down design ○ Hybrid design • Overview of function oriented and Object Oriented design	
III Software testing and software maintenance	Software Testing Techniques • Verification vs Validation • Testing Levels: ○ Unit Testing, ○ Integration Testing, ○ System Testing, ○ Acceptance Testing • Functional Testing • Non-functional Testing • Regression Testing	15



	• Black Box Testing • White Box Testing • Boundary Value Analysis • Equivalence Partitioning • Decision Table Testing • Path Testing • Introduction to Testing automation Tools • Introduction to Software Maintenance • categories of software maintenance ○ Corrective Maintenance ○ Patching ○ Adaptive maintenance ○ Perfective maintenance ○ Preventive maintenance • Software Re-engineering ○ Source code translation ○ Program reconstruction • Configuration Management ○ Configuration management activities ○ Software versions ○ Change control process	
IV project Documentation	Software Metrics and Project Documentation • Software Metrics ○ Size-function points ○ Quality metrics ○ Information flow metrics ○ Cyclomatic complexity	15



	• Project Documentation ○ How to create ER diagram using online tools ○ Practice of creating DFD ○ Drafting SRS	
Reference Books		
1. Aggarwal, K. K. (2005). Software engineering. New Age International. 2. Pfleeger, S. L., & Atlee, J. M. (2010). Software engineering: Theory and practice. Pearson Education India. 3. Jalote, P. (1997). An integrated approach to software engineering (2nd ed.). Narosa Publishing House.		
Web Resources		
1. https://www.tutorialspoint.com/software_engineering/index.htm 2. https://www.geeksforgeeks.org/software-engineering/software-engineering/		

L:: Lecture, **T::** Tutorial, **P::** Practical

CCE:: Continuous & Comprehensive Evaluation

(CCE theory includes Mid Semester Examination, Assignment, MCQ Quizzes, Seminar, Reflective Notes, Class Participation, Case Analysis and Presentation, Slip Tests (Announced/ Surprised), Attendance etc. or any combination of these)

PWE: Practical Work Examination

(PWE includes Laboratory Practical Work, Project Work, Viva simulation exercise work etc.)

SEE:: Semester End Evaluation

Sample Practical Exercises:

1. Select a software project and identify stakeholders, objectives, scope, and system requirements.
2. Develop a Software Requirement Specification (SRS) document for a selected software project.



3. Create Use Case Diagrams and Use Case Descriptions using UML modeling tools.
4. Design Context-Level and Level-1 Data Flow Diagrams (DFD) for a software application.
5. Create an Entity Relationship (ER) Diagram for a selected project using online modelling tools.
6. Prepare a complete Software Project Documentation Portfolio including SRS, DFD, ER Diagram, UML Diagrams, Test Cases, Risk Analysis, and Project Plan.

