

Course code: Course Title	Course Structure			Pre-Requisite
SE302: Empirical Software Engineering	L	T	P	Software Engineering
	3	0	2	

**Course Objective:** The objective is to study the collection and analysis of data and experience that can be used to characterize, evaluate and reveal relationships between software development deliverables, practices, and technologies.

S. NO	Course Outcomes (CO)
CO1	Demonstrate deep understanding of fundamentals of empirical study.
CO2	Apply software metrics and experimental design principles to measure software quality and conduct empirical studies.
CO3	Extract and analyze data collected from software repositories to derive insights using historical analysis techniques.
CO4	Develop predictive models, evaluate their performance and validate models using statistical testing.
CO5	Identify and mitigate threats to validity, report findings ethically, and utilize empirical tools.

S. NO	Contents	Contact Hours
UNIT 1	<b>Introduction:</b> What Is Empirical Software Engineering?; Overview of Empirical Studies; Types of Empirical Studies; Empirical Study Process; Ethics of Empirical Research; Importance of Empirical Research; Basic Elements of Empirical Research; Some Terminologies.  <b>Systematic Literature Reviews:</b> Basic Concepts; Case Study; Planning the Review; Methods for Presenting Results; Conducting the Review; Reporting the Review.	6
UNIT 2	<b>Software Metrics:</b> Introduction; Measurement Basics; Measuring Size; Measuring Software Quality; Object-Oriented Metrics; Dynamic Software Metrics; System Evolution and Evolutionary Metrics; Validation of Metrics; Practical Relevance and Use of Software Metrics in Research; Industrial Relevance of Software Metrics.  <b>Experimental Design:</b> Overview of Experimental Design; Case Study: Fault Prediction Systems; Research Questions; Reviewing the Literature; Research Variables; Terminology Used in Study Types; Hypothesis Formulation; Data Collection; Selection of Data Analysis Methods.	8
UNIT 3	<b>Mining Data from Software Repositories:</b> Configuration Management Systems; Importance of Mining Software Repositories; Common Types of Software Repositories; Version Control Systems; Bug Tracking Systems; Extracting Data from Software Repositories; Static Source Code Analysis; Software Historical Analysis; Software Engineering Repositories and Open Research Data Sets; Case Study: Defect Collection and Reporting System for Git Repository.	6
UNIT 4	<b>Data Analysis and Statistical Testing:</b> Analyzing the Metric Data; Attribute Reduction Methods; Hypothesis Testing; Statistical Testing; Example—Univariate Analysis Results for Fault Prediction System.  <b>Model Development and Interpretation:</b> Model Development; Statistical Multiple Regression Techniques; Machine Learning Techniques; Concerns in Model Prediction; Performance Measures for Categorical Dependent Variable; Performance Measures for Continuous Dependent Variable; Cross-Validation;	6

	Model Comparison Tests; Interpreting the Results; Example—Comparing ML Techniques for Fault Prediction.	
UNIT 5	<b>Validity Threats:</b> Categories of Threats to Validity; Example—Threats to Validity in Fault Prediction System; Threats and Their Countermeasures. <b>Reporting Results:</b> Reporting and Presenting Results; Guidelines for Masters and Doctoral Students; Research Ethics and Misconduct.	8
UNIT 6	<b>Mining Unstructured Data: Introduction;</b> Steps in Text Mining; Applications of Text Mining in Software Engineering; Example—Automated Severity Assessment of Software Defect Reports. <b>Case Study &amp; Tools:</b> Demonstrating Empirical Procedures; WEKA; KEEL; SPSS; MATLAB; R; Comparison of Tools.	8
	<b>TOTAL</b>	<b>42</b>

<b>REFERENCES</b>		
<b>S.No.</b>	<b>Name of Books/Authors/Publishers</b>	<b>Year of Publication / Reprint</b>
1	Ruchika Malhotra, “Empirical Research in Software Engineering: Concepts, Analysis & Applications”, CRC press, 1 <sup>st</sup> Edition.	2016
2	B. Boehm, H. D. Rombach, M. V. Zelkowitz, “Foundations of Empirical Software Engineering: The Legacy of Victor R. Basili”, Springer.	2010