

Course : 504 : System Analysis and Design

Course Code	504
Course Title	System Analysis and Design
Credit	4
Minimum weeks per semester	4 Hrs
Minimum weeks per semester	15
Last Review/Revision	June 2018
Purpose of the course	To enable students to understand modeling ,designing, analyzing and testing a system
Course objective	To give coherent view of design and analysis intended for software development process. The course provides comprehensive concepts of software engineering practice that involves various models, analysis models, design concepts, various testing methods and introduction to UML diagrams.
Prerequisites	Must have practical knowledge of at least one language for better understanding and relevance
Course Outcome	After finishing with the course, student will be able to choose the right model for a new to be developed system, perform requirement elicitation and prepare documentation for design, analysis and testing of system.
Course Content	<p>Unit : 1 : Fundamentals of SE</p> <p>1.1 Software Process</p> <p> 1.1.1 Software, software characteristics</p> <p> 1.1.2 Software types</p> <p> 1.1.3 Management, Customers’ and practitioners’ myths</p> <p>1.2 Structured Analysis and design</p> <p> 1.2.1 Roles and responsibilities</p> <p> 1.2.2 Process , Methods and tools</p> <p> 1.2.3 Generic phases of SE</p> <p>1.3 Process Models</p> <p> 1.3.1 Waterfall Model</p> <p> 1.3.2 Incremental Model</p> <p> 1.3.3 RAD Model</p> <p> 1.3.4 Prototyping model</p> <p> 1.3.5 Spiral Model</p> <p> 1.3.6 Concurrent Development Model</p> <p> 1.3.7 Component Based Models</p> <p> 1.3.8 Formal Methods model</p> <p> 1.3.9 Aspect – Oriented Software Development</p> <p> 1.3.10 Unified Process</p> <p>Unit : 2 : Requirement Gathering and Analysis</p> <p>2.1 Requirement Engineering</p> <p> 2.1.1 Inception, elicitation specification, validation, elaboration, negotiation</p> <p> 2.1.2 Quality Function Deployment</p> <p> 2.1.3 Functional and Non- Functional requirements</p> <p> 2.1.4 FAST</p> <p> 2.1.5 SRS- descriptive and scenario based</p> <p>2.2 Analysis Modeling</p> <p> 2.2.1 Analysis Modeling Approaches</p> <p> 2.2.2 Elements of Analysis Modeling</p> <p> 2.2.3 Data Modeling</p> <p> 2.2.4 Data Object, types, attributes and relationships</p> <p> 2.2.5 E-R D</p> <p> 2.2.6 Introduction to DFD</p> <p> 2.2.7 Data Dictionary</p> <p> 2.2.8 Process specification</p> <p> 2.2.9 Structures English</p> <p> 2.2.10 Decision tables</p> <p> 2.2.11 Decision trees</p>

	<p>Unit : 3 : Designing</p> <p>3.1 Design Concepts and Principles</p> <ul style="list-style-type: none"> 3.1.1 Relation between analysis and design 3.1.2 Design model 3.1.3 design Principles 3.1.4 Design concepts <ul style="list-style-type: none"> 3.1.4.1 Abstraction 3.1.4.2 Refinement 3.1.4.3 Modularity 3.1.4.4 Structural diagrams 3.1.4.5 Structural patterns <p>3.2 Cohesion and Coupling</p> <ul style="list-style-type: none"> 3.2.1 Information hiding, Functional independence 3.2.2 Coupling 3.2.3 Cohesion <p>Unit : 4 : Object Oriented Model</p> <p>4.1 Agile View of Process</p> <ul style="list-style-type: none"> 4.1.1 Extreme Programming Adaptive Software Development 4.1.2 Dynamic systems Development Method 4.1.3 Scrum 4.1.4 Crystal 4.1.5 Feature Driven Development <p>4.2 Unified Modeling model</p> <ul style="list-style-type: none"> 4.2.1 UML Diagrams 4.2.2 Use Case Diagrams 4.2.3 Class Diagrams 4.2.4 Interaction Diagrams <ul style="list-style-type: none"> 4.2.4.1 Sequence Diagrams 4.2.4.2 Collaboration Diagrams 4.2.5 Activity diagrams <ul style="list-style-type: none"> 4.2.5.1 Activity Diagram 4.2.5.2 Swim lane activity diagrams 4.2.6 State chart diagrams <p>Unit : 5 : Testing</p> <p>5.1 Principles of testing</p> <ul style="list-style-type: none"> 5.1.1 Equivalence Partitioning testing 5.1.2 Boundary Value Analysis 5.1.3 Comparison testing 5.1.4 Orthogonal testing 5.1.5 Integration Testing- big bang, top bottom, bottom up, mixed 5.1.6 System testing – Alpha , Beta, Acceptance 5.1.7 Performance testing- Stess testing, regression, Smoke <p>5.2 Debugging</p> <ul style="list-style-type: none"> 5.2.1 Debugging Approaches
Reference Books	<ol style="list-style-type: none"> 1. Software Engineering Practitioner’s Approach ,Roger Pressman, 6th edition, TMH 2. Fundamentals of software engineering ,Rajib mall, Third edition, PHI 3. An Integrated Approach to Software Engineering, Pankaj Jalote, Narosa 4. Software engineering concepts, Fairley RE, TMH
Teaching Methodology	Black Board Teaching, power point presentation for theory, practical shown in projector for various documents on various topics ,Assignments
Evaluation Method	30% Internal Exam 70% External Exam