

**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR**

**Course structure and syllabus for M.Tech. Software Engineering  
for affiliated Engineering Colleges 2009-10**

**I YEAR I SEMESTER**

Subject	Hours/ Week
Advanced Data Structures and Algorithms	4
Advanced Software Engineering	4
Software Requirements and Estimation	4
Software Metrics	4
Middleware Technologies	4
Elective-I a. Object Oriented Modeling b. Software Project Management c. Software Process Management	4
Software Lab- 1 (Covering the experiments: Data structures & Algorithms and Middleware Technologies)	4

**I YEAR II SEMESTER**

Subject	Hours/ Week
Software Architecture and Design Patterns	4
Software Quality Assurance and Testing	4
Service Oriented Architecture	4
Software Reliability	4
Software Reengineering	4
Elective-II a. Secure Software Engineering b. Soft Computing c. Model Driven Software Development	4
Software Lab- 2 (Covering the experiments: Software Testing and Design Patterns)	4

**II YEAR (III & IV Semesters)**

SUBJECTS
Seminar
Project work

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

## M.Tech (Software Engineering)

### I Year I Semester

#### ADVANCED DATA STRUCTURES AND ALGORITHMS

##### UNIT I : Overview of Data Structures

Review of Arrays, Stacks, Queues, linked lists , Linked stacks and Linked queues, Applications

##### UNIT II: Algorithm Analysis

Efficiency of algorithms, Apriori Analysis, Asymptotic Notations, Time complexity of an algorithm using O notation, Polynomial Vs Exponential Algorithms, Average, Best, and Worst Case Complexities, Analyzing Recursive Programs.

##### UNIT III: Trees and Graphs

Introduction, Definition and Basic terminologies of trees and binary trees, Representation of trees and Binary trees, Binary tree Traversals, Threaded binary trees, Graphs-basic concepts, representation and traversals.

##### UNIT IV: Binary Search Trees, AVL Trees and B Trees

Introduction, Binary Search Trees: Definition, Operations and applications. AVL Trees: Definition, Operations and applications. B Trees: Definition, Operations and applications.

##### UNIT V: Red – Black Trees, Splay Trees and Hash Tables

Red – Black Trees, Splay Trees and its applications. Hash Tables: Introduction, Hash Tables, Hash Functions and its applications.

##### UNIT VI: Divide – and – Conquer & Greedy Method

General Method, Binary Search, Finding Maximum and Minimum, Quick Sort, Merge sort, Strassen's Matrix Multiplication, Greedy Method- General Method, Minimum Cost Spanning Trees, Single Source Shortest Path.

##### UNIT VII: Dynamic Programming

General Method, All Pairs Shortest Path, Single Source Shortest Path, 0 / 1 Knapsack problem, Reliability Design, Traveling Sales Person's Problem.

##### UNIT VIII: Back Tracking and Branch – and – Bound

General Method, 8 – Queen's Problem, Graph Coloring. Branch – and – Bound: The Method, LC Search, Control Abstraction, Bounding, 0 / 1 Knapsack Problem.

##### TEXT BOOKS:

1. Data Structures and Algorithms by G.A.V. Pai, 2009, TMH.
2. Fundamentals of Computer Algorithms by Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, 2<sup>nd</sup> edition, University Press.

##### REFERENCE BOOKS:

1. Classic Data Structures by D. Samanta, 2005, PHI
2. Design and Analysis of Computer Algorithms by Aho, Hopcraft, Ullman 1998, PEA.
3. Introduction to the Design and Analysis of Algorithms by Goodman, Hedetniemi, TMG.
4. Design and Analysis of Algorithms by E. Horowitz, S. Sahani, 3<sup>rd</sup> Edition, Galgotia.
5. Data Structures and Algorithms in C++ by Drozdek 2<sup>nd</sup> Edition, Thomson.

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

**M.Tech (Software Engineering)**

**I Year I Semester**

## **ADVANCED SOFTWARE ENGINEERING**

### **UNIT-I**

#### **Software and Software Engineering**

The Nature of Software, Software characteristics, The Unique Nature of WebApps, Software Engineering, The Software Process, Software Engineering Practice, Software Myths.

#### **Software Paradigms**

Prespective Process Models, Specialized Process Models.

### **UNIT- II**

#### **Unified Process Model**

The Unified Process, Personal and Team Process Models, Process Technology, Product and Process.

#### **Agile Development**

What is Agility? Agility and the Cost of Change, What is an Agile Process? Extreme Programming (XP), Other Agile Process Models, A Tool set for the Agile Process.

### **UNIT-III**

#### **Critical Systems**

A simple safety-critical system, System dependability, Availability and reliability, Safety, Security.

#### **Critical systems specification**

Risk-driven specification, Safety specification, Security specification Software reliability specification.

#### **Formal Specification**

Formal specification in the software process, Sub-system interface specification, Behavioural specification.

### **UNIT- IV**

#### **Software Reuse**

The reuse landscape, Design patterns, Generator-based reuse, Application frameworks, Application system reuse.

#### **Component-based Software Engineering**

Components and component models, The CBSE process, Component composition.

### **UNIT-V**

#### **Software Testing**

System testing, Component testing, Test case design, Test automation.

#### **Software Evolution**

Program evolution dynamics, Software maintenance, Evolution Processes, Legacy system evolution.

## UNIT-VI

**Aspect oriented software engineering:** The separation of concerns, Aspects, Join points and pointcuts, Software engineering with aspect, Using AOSD to streamline complex systems development without sacrificing flexibility or scalability.

**Service oriented software engineering:** Service-based concepts, modeling and documentation, Service discovery and composition, Service-oriented architecture, Services as reusable components, Software development with services.

## UNIT-VII

### Quality Management

Process and product quality, Quality assurance and standards, Quality Planning, Quality control, Software Measurement and metrics.

### Process Improvement

Process and product quality, process classification, Process measurement, Process analysis and modeling, Process change, The CMMI process improvement framework.

## UNIT-VIII

### Model Driven Software Development (MDSD)

Goals of MDSD, The MDSD approach, An overview of MDA concepts, Common MDSD concepts and terminology, Model driven architecture, Domain specific modeling, What is meta modeling?, MOF and UML, Software architecture in the context of MDSD.

## TEXTBOOKS:

1. Software Engineering, by Ian Sommerville, Addison-Wesley, 8<sup>th</sup> Edition, 2006.
2. Software Engineering, A Practitioner's Approach, by Roger S. Pressman, 7<sup>th</sup> Edition, 2009.
3. Model-Driven Software Development: Technology, Engineering, Management , by Thomas Stahl , Markus Voelter , Krzysztof Czarnecki , Wiley; 1st edition , 2006

## REFERENCE BOOKS:

1. Using UML: Software Engineering with Objects and Components, by Perdita Stevens, Rob Pooley, Addison-Wesley, 2<sup>nd</sup> edition, 2006.
2. The Mythical Man-Month : Essays on Software Engineering, by Frederick P., Jr. Brooks, Frederick P. Brooks Jr, Addison-Wesley, 1995.
3. The Future of Software Engineering, edited by Anthony Finkelstein, ACM Press, 2000.
4. Aspect-Oriented Software Development, by [Robert E. Filman](#), [Tzilla Elrad](#), [Siobh  n Clarke](#), [Mehmet Aksit](#), Addison-Wesley, 1st edition, 2004.
5. Service-Oriented Software System Engineering : Challenges and Practices, by [Zoran Stojanovic](#), [Ajantha Dahanayake](#), IGI Global, 2005.
6. Software Reuse, by I. Jacobson, M. Griss, and P. Jonsson, ACM Press, 1997

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

## M.Tech (Software Engineering)

### I Year I Semester

#### SOFTWARE REQUIREMENTS AND ESTIMATION

##### UNIT I:

##### **Software Requirements: What And Why**

Essential Software requirement, Good practices for requirements engineering, Improving requirements processes, Software requirements and risk management.

##### UNIT II:

##### **Software Requirements Engineering**

Requirements elicitation, requirements analysis documentation, review, elicitation techniques, analysis models, Software quality attributes, risk reduction through prototyping, setting requirements priorities, verifying requirements quality.

##### UNIT-III:

##### **Software Requirements Modeling-**

Use Case Modeling, Analysis Models, Dataflow diagram, state transition diagram, class diagrams, Object analysis, Problem Frames

##### UNIT IV:

##### **Software Requirements Management**

Requirements management Principles and practices, Requirements attributes, Change Management Process, Requirements Traceability Matrix, Links in requirements chain.

##### UNIT-V:

##### **Requirements Management Tools:**

Benefits of using a requirements management tool, commercial requirements management tool, Rational Requisite pro, Caliber – RM, implementing requirements management automation.

##### UNIT VI:

##### **Software Estimation**

Components of Software Estimations, Estimation methods, Problems associated with estimation, Key project factors that influence estimation. **Size Estimation**-Two views of sizing, Function Point Analysis, Mark II FPA, Full Function Points, LOC Estimation, Conversion between size measures.

##### Unit VII:

##### **Effort, Schedule And Cost Estimation**

What is Productivity? Estimation Factors, Approaches to Effort and Schedule Estimation, COCOMO II, Putnam Estimation Model, Algorithmic models, Cost Estimation.

##### UNIT-VIII:

##### **Software Estimation Tools:**

Desirable features in software estimation tools, IFPUG, USC's COCOMO II, SLIM (Software Life Cycle Management) Tools

TEXT BOOKS:

1. Software Requirements and Estimation by *Rajesh Naik and Swapna Kishore*, Tata Mc Graw Hill

REFERENCES:

1. Software Requirements by Karl E. Weigers, Microsoft Press.
2. Managing Software Requirements, Dean Leffingwell & Don Widrig, Pearson Education, 2003.
3. Mastering the requirements process, second edition, Suzanne Robertson & James Robertson, Pearson Education, 2006.
4. Estimating Software Costs, Second edition, Capers Jones, Tata McGraw-Hill, 2007.
5. Practical Software Estimation, M.A. Parthasarathy, Pearson Education, 2007.
6. Measuring the software process, William A. Florac & Anita D. Carleton, Pearson Education, 1999.

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

**M.Tech (Software Engineering)**

**I Year I Semester**

## **SOFTWARE METRICS**

### **UNIT-I: MEASUREMENT**

Measurement in Everyday Life, Measurement in Software Engineering, Scope of Software Metrics.

### **UNIT-II: BASICS OF MEASUREMENT**

Representational Theory of Measurement, Measurement and Models, Measurement Scales and Scale Types.

### **UNIT-III: FRAME WORK FOR SOFTWARE MEASUREMENT**

Classifying Software Measures, Applying Frame Work, Software Measurement Validation.

### **UNIT-IV: SOFTWARE METHODS IN DATA COLLECTION**

Good Data, Definition of Data, Collecting, Storing and Extracting Data.

### **UNIT-V: MEASURING INTERNAL PRODUCT ATTRIBUTES**

Measuring Size and Structure.

### **UNIT –VI: MEASURING EXTERNAL PRODUCT ATTRIBUTES**

Modeling Software Quality, Measuring Aspects of Quality.

### **UNIT-VII: MEASUREMENT AND MANAGEMENT**

Planning a Measurement Program, Measurement in Practice.

### **UNIT-VIII: CUSTOMER SATISFACTION**

Empirical Research in Software Engineering, Measuring and Analyzing Customer Satisfaction: Customer Satisfaction Surveys, Analyzing Satisfaction Data, Satisfaction with Company.

### **TEXT BOOKS:**

1. Fenton, Pfleeger, "Software Metrics: A Rigorous and Practical Approach", Thomson.
2. Stephen H. Kan: "Metrics & Models in Software Quality Engineering", PEA.

### **REFERENCE BOOKS:**

1. Sheppard, "Software Engineering Metrics", 1992, MCG.
2. Pertis et al, "Software Metrics: An Analysis and Evaluation", 1981, MIT Press.

# **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR**

**M.Tech (Software Engineering)**

**I Year I Semester**

## **MIDDLEWARE TECHNOLOGIES**

### **UNIT-I:**

#### **CLIENT/SERVER COMPUTING**

Building blocks-types of servers-types of Clients-types of middleware-aspects of client/server systems-sizing-scalability-tiered architecture-client/server models-requirements of client/server systems-Distributed objects-benefits-drawbacks-from distributed objects to components

### **UNIT-II:**

#### **COMPONENT TECHNOLOGY**

Components- definitions-properties-benefits-components and interfaces- direct and indirect interfaces- versions-interfaces as contracts- callbacks- forms of design levels reuse- connection oriented programming – connectable objects.

### **UNIT-III:**

#### **COMPONENT ARCHITECTURE**

Component architecture- component frameworks- composition- data driven, contextual, aspect oriented programming, subject oriented programming, XML components-component development- assembly.

### **UNIT-IV:**

#### **THE MICROSOFT WAY**

Component object model- from COM, COM+, DCOM to .NET framework-evolution- web services technologies- XML, WSDL, UDDI, SOAP.

### **UNIT-V :**

#### **COMMON LANGUAGE INFRASTRUCTURE**

Common language infrastructure- common language Runtime - <NET framework class library- ADO.NET, ASP.NET- enterprise services.

### **UNIT-VI:**

#### **THE SUN WAY**

Component variety- applets, servlets, java beans, enterprise beans- EJB architecture- types of beans- characteristics- Building and deploying distributed applications using EJB-java and web services-JXTA and jinni.

### **UNIT-VII:**

#### **THE OMG WAY**

System object model- CORBA timeline- CORBA architecture- ORB- services- facilities-business objects-IIop- transport mechanisms- IDL- Drawbacks of CORBA.

### **UNIT-VIII:**

#### **CORBA COMPONENT MODEL**

CORBA Component model- POA- CCM components- CCM containers- Meta Object Facility. Comparison of CORBA, .NET and EJB.

**TEXT BOOKS:**

1. Clemens szyperski, Dominik Gruntz and Stephan Murer, “Component Software Beyond object oriented Programming” third edition, Pearson education, 2004.
2. Robert Orfali, Dan Harkey, Jeri Edwards, “Client/Server Survival Guide” third edition, John Wiley Inc, 2003.

**REFERENCES:**

1. David Chappell, “Understanding .NET”, Pearson education Inc, 2002.
2. Bill Burke, Richard Monson-Haefel, “Enterprise JavaBeans”, Fifth Edition, O’Reily, 2001.
3. Dan harkey, Robertt Orfali, “ Client/Server programming with JAVA and CORBA”, second edition, wiley& sons Inc, 1999.

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

**M.Tech (Software Engineering)**

**I Year I Semester**

## **OBJECT ORIENTED MODELING ELECTIVE-1**

### **UNIT I**

**Introduction to UML:** The meaning of Object Orientation, object identity, Encapsulation, information hiding, polymorphism, generosity, importance of modeling, principles of modeling, object oriented modeling, conceptual model of the UML, Architecture.

### **UNIT II**

**Basic Structural Modeling:** Classes, Relationships, common Mechanisms, and diagrams.

**Class & Object Diagrams:** Terms, concepts, modeling techniques for Class & Object Diagrams.

### **UNIT III**

**Collaboration Diagrams:** Terms, Concepts, depicting a message, polymorphism in collaboration diagrams, iterated messages, use of self in messages.

**Sequence Diagrams:** Terms, concepts, depicting asynchronous messages with/without priority, callback mechanism, broadcast messages.

### **UNIT IV**

**Basic Behavioral Modeling:** Use cases, Use case Diagrams, Activity Diagrams.

**Advanced Behavioral Modeling:** Events and signals, state machines, processes and Threads, time and space, state chart diagrams.

**Architectural Modeling:** Component, Deployment, Component diagrams and Deployment diagrams.

### **UNIT V**

**The Unified process:** use case driven, architecture centric, iterative, and incremental

**The Four Ps:** people, project, product, and process

**Use case driven process:** why use case, capturing use cases, analysis, design, and implementation to realize the use cases, testing the use cases

**Architecture-centric process:** architecture in brief, why we need architecture, use cases and architecture, the steps to architecture, an architecture description.

### **UNIT VI**

**Iterative incremental process:** iterative incremental in brief, why iterative incremental development? The iterative approach is risk driven, the generic iteration.

**The Generic Iteration workflow:** phases are the first division workflow, planning proceeds doing, risks affect project planning, use case prioritization, resource needed, assess the iteration and phases

### **UNIT VII**

**Inception phase:** early in the inception phase, the archetypal inception iteration workflow, execute the core workflows, requirements to test.

**Elaboration Phase:** elaboration phase in brief, early in the elaboration phase, the architectural elaboration iteration workflow, execute the core workflows-Requirements to test.

### **UNIT VIII**

**Construction phase:** early in the construction phase, the archetypal construction iteration workflow, execute the core workflow.

**Transition phase:** early in the transition phase, activities in transition phase

**Case Studies:** Automation of a Library, Software Simulator application (2-floor elevator simulator)

**TEXT BOOKS :**

- 1 The Unified Modeling Language User Guide By Grady Booch, James Rumbaugh, Ivar Jacobson 2<sup>nd</sup> Edition, Pearson Education.
2. UML 2 Toolkit By Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado WILEY-Dreamtech India Pvt. Ltd.
3. The Unified Software Development Process By Ivar Jacobson, Grady Booch, James Rumbaugh, Pearson Education

**REFERENCE BOOKS :**

1. Fundamentals of Object Oriented Design in UML By Meilir Page-Jones, Pearson Education
2. Object Oriented Analysis & Design By Atul Kahate, The McGraw-Hill.
3. Practical Object-Oriented Design with UML By Mark Priestley, TATA McGrawHill
4. Object Oriented Analysis & Design By Brett D McLaughlin, Gary Pollice and David West, O'REILY .
5. Object-Oriented Analysis and Design using UML By Simon Bennet, Steve McRobb and Ray Farmer, 2<sup>nd</sup> Edition, TATA McGrawHill.
6. Object-Oriented Analysis and Design with the Unified Process By John W. Satzinger, Robert B Jackson and Stephen D Burd, THOMSON Course Technology.
7. UML and C++,R.C.Lee, and W.M.Tepfenhart,PHI.

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

## M.Tech (Software Engineering)

### I Year I Semester

#### SOFTWARE PROJECT MANAGEMENT ELECTIVE-1

##### UNIT-I:

##### **Conventional Software Management**

The Waterfall Model, Conventional software Management Performance. Evolution of Software Economics: Software Economics, Pragmatic Software Cost Estimation.

##### UNIT-II:

##### **Improving Software Economics**

Reducing Software Product Size, Improving software Processes, Improving Team Effectiveness, Improving Automation, Achieving Required Quality, Peer Inspections.

##### UNIT-III:

##### **Conventional And Modern Software Management**

Principles of Conventional Software Engineering, Principles of Modern Software Management, Transitioning to an Iterative Process. Life Cycle Phases: Engineering and Production Stages, Inception. Elaboration, Construction, Transition Phases.

##### UNIT-IV:

##### **Artifacts Of The Process**

The Artifact Sets. Management Artifacts, Engineering Artifacts, Programmatic Artifacts. Model Based Software Architectures: A Management Perspective and Technical Perspective.

##### UNIT-V:

##### **Flows Of The Process**

Software Process Workflows. Inter Trans Workflows. Checkpoints of the Process : Major Mile Stones, Minor Milestones, Periodic Status Assessments. Interactive Process Planning: Work Breakdown Structures, Planning Guidelines, Cost and Schedule Estimating. Interaction Planning Process. Pragmatic Planning.

##### UNIT-VI:

##### **Project Organizations And Responsibilities**

Line-of-Business Organizations, Project Organizations, and Evolution of Organizations. Process Automation: Automation Building Blocks, The Project Environment.

##### UNIT-VII:

##### **Project Control And Process Instrumentation**

Server Care Metrics, Management Indicators, Quality Indicators, Life Cycle Expectations Pragmatic Software Metrics, Metrics Automation. Tailoring the process: Process Discriminates, Example.

##### UNIT-VIII:

##### **Future Software Project Management**

Modern Project Profiles Next Generation Software economics, Modern Process Transitions. Case Study: The Command Center Processing and Display System –Replacement (CCPDS-R)

##### **TEXT BOOKS:**

1. Walker Rayce, “Software Project Management”, 1998, PEA.
2. Henry, “Software Project Management” Pearson.

##### **REFERENCE BOOKS:**

1. Richard H. Thayer: “Software Engineering Project Management”, 1997, IEEE Computer Society.
2. Shere K. D. : “” Software Engineering and Management”, 1998, PHI.
3. S. A. Kelkar, “ Software Project Management: A Concise Study”, PHI.

4. Hughes Cotterell, “ Software Project Management”, 2e, TMH.
5. Kaeron Conway, “ Software Project Management from Concept to Development”, Dream Tech.

## **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR**

### **M.Tech (Software Engineering)**

#### **I Year I Semester**

### **SOFTWARE PROCESS MANAGEMENT**

#### **ELECTIVE-1**

##### **UNIT-I:**

##### **Software Process Maturity**

Framework: Software Process Improvement, Need for Process Optimization.

Software Process Change: Perceptive, Principles, Misconceptions, Change Implementation.

Software Process Assessment: Overview, Phases, Principles, Process and Conduct.

##### **UNIT-II:**

##### **Repeatable Process**

Managing Software Organizations: Discipline, Management System, Establishment.

Project Plan: Principles, Concepts, Size Measure, Estimation, Productivity Factors, Scheduling,

Project Tracking, Development Plan, Planning Models, Final Considerations.

##### **UNIT-III:**

##### **Configurations Management & Quality Assurance**

SCM: Introduction, Software Product Nomenclature, Functions, Baselines, Configurations, Responsibilities, Need for Automated Tools.

SQA: Quality Management, The Role of SQA, Launching the SQA Program, The SQA Plan, SQA People, Independent Verification and Validation.

##### **UNIT-IV:**

##### **Standards And Inspections**

Standards: Definitions, necessity, benefits, Major Standards, Standards Vs Guidelines.

Software Inspections: Types of Reviews, Objectives, Principles, Conduct of Inspections, Training, Reports and Tracking, Other Considerations, Initiating an Inspection Program.

##### **UNIT-V:**

##### **Software Testing And Scm Design**

Software Testing: Principles, Software Tests, Planning, Development, Execution, Reporting, Tools and Methods, Real-Time Testing.

SCM Design Phase: Plan, Questions, Support Functions, Requirement Phase, Design Control, Implementation Phase, Operational data, Test Phases, Tools, Accounting, Audit.

##### **UNIT-VI:**

##### **Software Process**

Software Process: Standards, Definitions, Levels of models & Uses, Architecture, Critical Software Process issues, Process Architecture, Process Models and Views, Establishing and Using a Process Definition, Basic process Guidelines.

Software Engineering Process Group: Changing Software Process, Role, Standards, Process Database, Technology Insertion Focal Point, Education and Training Process Consultation, Process Consultation, Process Status and Assessment, Establishing the SEPG.

**UNIT-VII:****Managed Process**

Data Gathering and Analysis: Principles, Process, Software Measure, Data Analysis.

Managing Software Quality: Paradigm, Quality Motivation, measurement Criteria, A Software Quality Program, Estimation, Goals, Plans, Tracking and Controlling Software Quality.

**UNIT-VIII:****Optimizing Process**

Defect Prevention Principles, Process Changes, Considerations and Management's Role.

Automating The Software Process: Need, Importance, Dev. Environments, Organizational Plan, Tech. Transition, Special Considerations, Productivity, Justification Considerations.

**TEXT BOOKS:**

1. Watt S. Humphrey, "Managing the Software Process", 2003, PEA.

**REFERENCE BOOKS:**

1. Watts S. Humphrey, "A Discipline for Software Engineering", 2002, PEA
2. Roger S. Pressman, " Software Engineering, A Practitioners Approach", 2003, 5e, TMH.
3. W. S. Humphrey, " Introduction to the Personal Software Precess", 1997, PEA.
4. W. S. Humphrey, " Introduction to the Team Software Process", 2002, PEA.

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

## **M.Tech (Software Engineering)**

### **I Year II Semester**

#### **SOFTWARE ARCHITECTURE AND DESIGN PATTERNS**

##### **UNIT I**

###### **Envisioning Architecture**

The Architecture Business Cycle, What is Software Architecture, Architectural patterns, reference models, reference architectures, architectural structures and views.

##### **UNIT II**

###### **Creating an Architecture**

Quality Attributes, Achieving qualities, Architectural styles and patterns, designing the Architecture, Documenting software architectures, Reconstructing Software Architecture.

##### **UNIT III**

###### **Analyzing Architectures**

Architecture Evaluation, Architecture design decision making, ATAM, CBAM.

##### **UNIT IV**

###### **Moving from one system to many**

Software Product Lines, Building systems from off the shelf components, Software architecture in future.

##### **UNIT V**

###### **Patterns**

Pattern Description, Organizing catalogs, role in solving design problems ,Selection and usage.

##### **UNIT VI**

###### **Creational and Structural patterns**

Abstract factory, builder, factory method, prototype, singleton, adapter, bridge, composite, façade, flyweight, Proxy.

##### **UNIT VII**

###### **Behavioral patterns**

Chain of responsibility, command, Interpreter, iterator, mediator, memento, observer, state, strategy, template method, visitor.

##### **UNIT VIII**

###### **Case Studies**

A-7E – A case study in utilizing architectural structures, The World Wide Web - a case study in interoperability, Air Traffic Control – a case study in designing for high availability, Celsius Tech – a case study in product line development

**TEXT BOOKS:**

1. Software Architecture in Practice, second edition, Len Bass, Paul Clements & Rick Kazman, Pearson Education, 2003.
2. Design Patterns, Erich Gamma, Pearson Education, 1995.

**REFERENCE BOOKS:**

1. Beyond Software architecture, Luke Hohmann, Addison Wesley, 2003.
2. Software architecture, David M. Dikel, David Kane and James R. Wilson, Prentice Hall PTR, 2001
3. Pattern Oriented Software Architecture, F. Buschmann & others, John Wiley & Sons.
4. Head First Design patterns, Eric Freeman & Elisabeth Freeman, O'REILLY, 2007.
5. Design Patterns in Java, Steven John Metsker & William C. Wake, Pearson education, 2006
6. J2EE Patterns, Deepak Alur, John Crupi & Dan Malks, Pearson education, 2003.
7. Design Patterns in C#, Steven John metsker, Pearson education, 2004.
8. Software Design, David Budgen, second edition, Pearson education, 2003

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

**M.Tech (Software Engineering)**

**I Year II Semester**

## SOFTWARE QUALITY ASSURANCE AND TESTING

### UNIT I

**Software Quality Assurance Framework and Standards SQA Framework:** What is Quality? Software Quality Assurance, Components of Software Quality Assurance – **Software Quality Assurance Plan:** Steps to develop and implement a Software Quality Assurance Plan – **Quality Standards:** ISO 9000 and Companion ISO Standards, CMM, CMMI, PCMM, Malcom Balridge, 3 Sigma, 6 Sigma

### UNIT II

**Software Quality Assurance Metrics and Measurement Software Quality Metrics:** Product Quality metrics, In-Process Quality Metrics, Metrics for Software Maintenance, Examples of Metric Programs

### UNIT III

**Software Quality metrics methodology:** Establish quality requirements, Identify Software quality metrics, Implement the software quality metrics, analyze software metrics results, validate the software quality metrics – **Software quality indicators – Fundamentals in Measurement theory**

### UNIT IV

**Software Testing Strategy and Environment:** Establishing testing policy, structured approach to testing, test factors, Economics of System Development Life Cycle (SDLC) Testing

### UNIT V

#### **Software Testing Methodology**

Defects hard to find, verification and validation, functional and structural testing, workbench concept, eight considerations in developing testing methodologies, testing tactics checklist

### UNIT VI

#### **Software Testing Techniques**

Black-Box, Boundary value, Bottom-up, Branch coverage, Cause-Effect graphing, CRUD, Database, Exception, Gray-Box, Histograms, Inspections, JADs, Pareto Analysis, Prototyping, Random Testing, Risk-based Testing, Regression Testing, Structured Walkthroughs, Thread Testing, Performance Testing, White-Box Testing

## **UNIT VII**

### **Software Testing Tools**

Taxonomy of Testing tools, Methodology to evaluate automated testing tools, Load Runner, Win runner and Rational Testing Tools, Silk test, Java Testing Tools, JMetra, JUNIT and Cactus.

## **UNIT VIII**

### **Testing Process**

**Eleven Step Testing Process:** Assess Project Management Development Estimate and Status, Develop Test Plan, Requirements Phase Testing, Design Phase Testing, Program Phase Testing, Execute Test and Record Results, Acceptance Test, Report test results, testing software installation, Test software changes, Evaluate Test Effectiveness.

### **Testing Specialized Systems and Applications**

Testing Client/Server – Web applications, Testing off the Shelf Components, Testing Security, Testing a Data Warehouse

### **TEXT BOOKS:**

1. Effective Methods for Software Testing, 2nd Edition, William E. Perry , Second Edition, Wiley India, 2006.
2. Software Quality, Mordechai Ben-Menachem/Garry S. Marliss, Thomson Learning publication,1997.

### **REFERENCE BOOKS:**

1. Testing and Quality Assurance for Component-based Software, by Gao, Tsao and Wu, Artech House Publishers
2. Software Testing Techniques, by Bories Beizer, Second Edition, Dreamtech Press
3. Managing the Testing Process, by Rex Black, Wiley
4. Handbook of Software Quality Assurance, by G. Gordon Schulmeyer, James I.McManus, Second Edition, International Thomson Computer Press
5. Software Testing and continuous Quality Improvement, by William E.Lewis, Gunasekaran Veerapillai, Second Edition, Auerbach Publications
6. Metrics and Models for Software Quality Engineering, by Stephen H. Kan, by Pearson Education Publication
7. Software Testing Tools, K.V.K.K. Prasad, Dream tech press, 2008.
8. Practical Software Testing, Ilene Burnstein, Springer, 2003.
9. Software Testing, Srinivasan Desikan & Gopaldaswamy Ramesh, Pearson Education,2006.
10. Software testing techniques, Scott Loveland & Geoffrey Miller, Shroff Publishers, 2005.
11. Software Quality, Martin Wieczorek & Dirk Meyerhoff, Springer, 2001.

# **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR**

## **M.Tech (Software Engineering)**

### **I Year II Semester**

#### **SERVICE ORIENTED ARCHITECTURE**

##### **UNIT I**

SOA and Web Services Fundamentals

Introducing SOA- Fundamental SOA,Common Characteristics of Contemporary SOA ,Common tangible benefits of SOA,Common pitfalls of adopting SOA.The Evolution of SOA – An SOA timeline,The continuing evolution of SOA,The roots of SOA.Web Services and primitive SOA-The Web Services frame work,Services,Service descriptions,Messaging.

##### **UNIT II**

Web Services and Contemporary SOA(Part I-Activity management and Composition)

Message exchange patterns,Service Activity Coordination,Atomic transactions,Business Activities,Orchestration,Choreography.

##### **UNIT III**

Web Services and Contemporary SOA(Part-II-Advanced Messaging , Metadata , and Security)

Addressing , Reliable messaging,Correlation,Policies,Metadata exchange,Security,Notification and eventing.

##### **UNIT IV**

Principles of Service-Oriented

Service – Orientation and the enterprise,Anatomy of SOA,Common Principles of Service – Orientation,interrelation between Principles of Service-Oriented,Service Orientation and Object Orientation,Native Web Services support for Principles of Service-Oriented.

##### **UNIT V**

Service Layers

Service-Oriented and Contemporary SOA , Service Layer abstraction , Application Service Layer , Business Service Layer,Orchestration Service Layer,Agnostic Services,Service Layer Configuration Scenarios.

##### **UNIT VI**

Building SOA(Planning and Analysis)

SOA Delivery Strategies-SOA delivery lifecycle phases,The top-down strategy,The bottom-up strategy,The agile strategy.Service Oriented Analysis(Part I-Introduction)-Introduction to Service Oriented Analysis,Benefits of a Business Centric SOA,Deriving Business Services. Service Oriented Analysis(Part-II-Service Modelling)-Service Modelling,Service Modelling guidelines,Classifying Service model logic,Contrasting Service modelling approaches.

##### **UNIT VII**

Building SOA(Technology and Design)

Service Oriented Design(Part I-Introduction)-Introduction to Service-Oriented design,WSDL related XML Schema language basics,WSDL language basics,Service interface design tools. Service Oriented Design(Part II-SOA Composition Guidelines)-SOA Composing steps,Considerations for choosing service layers,Considerations for positioning core SOA standards,Considerations for choosing SOA extensions.

## **UNIT VIII**

Service Oriented Design(Part III- Service Design)

Service Design overview, Entity-centric business Service Design, Application Service Design, Task-centric business Service Design, Service Design guidelines. Service Oriented Design(Part IV-Business Process Design)-WS-BPEL language basics, WS- Coordination overview, Service Oriented Business process Design. Fundamental WS-\* Extensions-WS-Addressing language basics, WS-Reliable Messaging language basics, WS-Policy language basics, WS-Metadata Exchange language basics, WS-Security language basics. SOA Platforms-SOA platform basics, SOA support in J2EE and .NET, integration considerations.

### **TEXT BOOKS:**

1. Service-Oriented Architecture-Concepts, Technology, and Design, Thomas Erl, Pearson Education.
2. Understanding SOA with Web Services, Eric Newcomer, Greg Lomow, Pearson Education.

### **REFERENCE BOOKS:**

1. The Definitive guide to SOA, Jeff Davies & others, Apress, Dreamtech.
2. Java SOA Cook book, E.Hewitt, SPD.
3. SOA in Practice, N.M.Josuttis, SPD.
4. Applied SOA, M.Rosen and others, Wiley India pvt. Ltd.
5. Java Web Services Architecture, J.Mc Govern, and others, Morgan Kaufmann Publishers, Elsevier.
6. SOA for Enterprise Applications, Shankar.K, Wiley India Edition.
7. SOA-Based Enterprise Integration, W.Roshen, TMH.
8. SOA Security, K.Rama Rao, C.Prasad, dreamtech press.

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

**M.Tech (Software Engineering)**

**I Year II Semester**

## **SOFTWARE RELIABILITY**

**UNIT I:**

Introduction

**The Need for Reliable Software, Software Reliability Engineering Concepts, Basic definitions, Software practitioners biggest problem, software reliability engineering approach, software reliability engineering process, defining the product**

**UNIT II:**

The Operational Profile

**Reliability concepts, software reliability and hardware reliability, developing operational profiles, applying operational profiles, learning operations and run concepts.**

**UNIT III:**

Software Reliability Concepts

Defining failure for the product, common measure for all associated systems, setting system failure intensity objectives, determining develop software failure intensity objectives, software reliability strategies, failures, faults and errors, availability, system and component reliabilities and failure intensities, predicting basic failure intensity.

**UNIT IV:**

**Software Reliability Modeling Survey**

Introduction, Historical Perspective and Implementation, Exponential Failure Time Class of Models, Weibull and Gamma Failure Time Class of Models, Infinite Failure Category Models, Bayesian Models, Model Relationship, Software Reliability Prediction in Early Phases of the Life Cycle.

**UNIT V:**

**Software Metrics for Reliability Assessment**

Introduction, Static Program Complexity, Dynamic Program Complexity, Software Complexity and Software Quality, Software Reliability Modeling.

**UNIT VI:**

**Software Testing and Reliability**

Introduction, Overview of Software Testing, Operational profiles, Time/Structure Based Software Reliability Estimation.

**UNIT VII:****Best Practice of SRE**

Benefits and approaches of SRE, SRE during requirements phase, SRE during implementation phase, SRE during Maintenance phase.

**UNIT VIII:****Neural Networks for Software Reliability**

Introduction, Neural Networks, Neural Networks for software reliability, software reliability growth modeling.

**Text Books**

1. Handbook of Software Reliability Engineering Edited by Michael R. Lyu, published by IEEE Computer Society Press and McGraw-Hill Book Company.
2. Software Reliability Engineering John D. Musa, second edition Tata McGraw-Hill.

**Reference Books**

1. Practical Reliability Engineering, Patric D. T. O connor 4<sup>th</sup> Edition, John Wesley & Sons, 2003.
2. Fault tolerance principles and Practice, Anderson and PA Lee, PHI, 1981.
3. Fault tolerant computing-Theory and Techniques, Pradhan D K (Ed.): Vol 1 and Vol 2, Prentice hall, 1986.
4. Reliability Engineering E. Balagurusamy, Tata McGrawHill, 1994.

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

M.Tech (Software Engineering)

I Year II Semester

## SOFTWARE REENGINEERING

### Unit I:

#### **Software, Software evolution and Maintenance:**

Software, Legacy software, Well designed software, Software evolution challenges, Lehman's laws, Software deterioration curve. Software maintenance: Software change, Types of change encountered during the support phase, Maintenance costs, Why is software maintenance expensive?, Factors affecting maintenance, Maintenance process, Change and maintenance prediction.

### UNIT II:

**Software quality factors, Quality and Maintainability metrics:** Internal and external attributes, McCall's quality factors, ISO 9126 quality factors, Need and importance of quality and maintainability metrics, Metric for software correctness (Defects/KLOC), Metric for software integrity, Software reliability (MTBF), Metrics for maintainability (Mean-time-to-change (MTTC), Spoilage metric, Software maturity index, McCabe and Halstead metrics). Design maintainability: Cohesion, Coupling, Understandability and Adaptability.

### UNIT III:

#### **Legacy software structure, Software reengineering process model:**

Software change strategies include: Software maintenance, Architectural transformation, Software reengineering. Legacy software structure and distribution: Ideal structure, Real structure, Layered distribution model, Legacy software distribution, Architectural problems. Business process reengineering: Business processes, A BPR Model. Software reengineering and its importance, Goals of reengineering, A software reengineering process model, Software reengineering activities.

### UNIT IV:

#### **Design Extraction:**

Reverse Engineering: Goals of reverse engineering, Why design extraction is needed?, Reverse engineering process, Reverse engineering to understand processing, Code duplication detection, Reverse engineering to understand data, Reverse engineering user interfaces, Design extraction with UML, Heuristics to extract the design, Tools for reverse engineering.

### UNIT V:

**Restructuring (In Traditional context):** Code restructuring: Characteristics of unstructured code, Characteristics of structured code, Spaghetti logic, Structured control logic, Restructuring problems, Flow graph restructuring, Warnier's logical simplification techniques, Some basic code restructuring methods: Interchange, Transposition, Combination, Resolution, Substitution. Data restructuring (Data reengineering): Data

reengineering process, Data problems, Approaches: Data cleanup, Data extension, Data migration. Tools for restructuring.

#### **UNIT VI:**

**Refactoring (Restructuring in object oriented context):** What is refactoring?, Principles in refactoring: Why should you refactor?, When should you refactor?, Problems with refactoring, Refactoring and design, Refactoring and performance. Refactoring opportunities, Top ten of code bad smells, Different refactorings and their use, Refactoring tools.

#### **UNIT VII:**

**Forward Engineering:** What is forward engineering ? Goals of forward engineering, Forward engineering for client/server applications, Forward engineering for object oriented architectures, Forward engineering user interfaces, Tools for forward engineering.

#### **UNIT VIII:**

##### **Reengineering Metrics, Repositories, and Economics:**

Metrics in Reengineering: Why metrics in Reengineering?, Metrics as a reengineering tool, Which metrics to collect ?(Goal Question Metric (GQM) paradigm), Reengineering repositories: Why repositories?, Taxonomy (Functionality + Integration options), Issues. Reengineering economics.

#### **TEXT BOOKS:**

1. Software Reengineering, Ed. Robert S. Arnold, IEEE Computer Society, 1993.
2. Software Evolution, Tom Mens, Serge Demeyer, Springer publication company, 2008.

#### **REFERENCES**

1. Software Engineering, Ian Sommerville, Addison-Wesley, 8<sup>th</sup> Edition.
2. Software Engineering, A Practitioner's Approach, Roger S. Pressman, 7<sup>th</sup> Edition.
3. Refactoring: Improving the Design of Existing Code, Martin Fowler, K.Beck, J.Brant, W.Opdyke, D.Roberts, Addison- Wesley, NY, 1999.
4. Software Reengineering, Georg Abfalter, VDM Verlag, Germany, 2008.
5. Successful Software Reengineering, Salvatore Valenti, IRM Press, 2002.
6. Logical construction of programs, J.D.Warnier, Van Nostrand-Reinhold,1974.
7. Tutorial on Software Restructuring, Robert E.Arnold, IEEE Computer Society, 1986.

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

**M.Tech (Software Engineering)**

**I Year II Semester**

## **SECURE SOFTWARE ENGINEERING ELECTIVE-II**

### **UNIT I:**

#### **Why Is Security a Software Issue?**

Introduction, The problem, Software assurance and software security, Threats to software security, Sources of software insecurity, The benefits of detecting software security defects early, Managing secure software development.

### **UNIT II:**

#### **What Makes Software Secure?**

Defining properties of secure software, How to influence the security properties of software, How to assert and specify desired security properties.

### **UNIT III:**

#### **Requirements Engineering for Secure Software**

The SQUARE process model: Identifying security requirements using the security quality requirements engineering (SQUARE) method, SQUARE sample outputs, Requirements elicitation, Requirements prioritization.

### **UNIT IV:**

#### **Secure Software Architecture and Design**

Introduction, Software security practices for architecture and design: Architectural risk analysis. Software security knowledge for architecture and design: Security principles, Security guidelines, and Attack patterns.

### **UNIT V:**

#### **Considerations for Secure Coding and Testing**

Introduction, Code analysis, Coding practices, Software security testing, Security testing considerations throughout the SDLC.

### **UNIT VI:**

#### **Security and Complexity: System Assembly Challenges**

Introduction, Security failures, Functional and attacker perspectives for security analysis, System complexity drivers and security, Deep technical problem complexity.

### **UNIT VII:**

#### **Governance, and Managing for More Secure Software**

Governance and security, Adopting an enterprise software security framework, How much security is enough?, Security and project management, maturity of practice.

### **UNIT VIII:**

#### **Security metrics**

Defining security metrics, Diagnosing problems and measuring technical security, Analysis techniques, Organize, aggregate, and analyze data to bring out key insights.

## **TEXT BOOKS**

- 1. Software Security Engineering: A Guide for Project Managers,**  
by Julia H. Allen, Sean Barnum, Robert J. Ellison, Gary McGraw, Nancy R. Mead,  
Addison-Wesley , 1st edition, 2008.
- 2. Security Metrics: Replacing Fear, Uncertainty, and Doubt ,** by Andrew Jaquith, Addison-Wesley , 1st edition , 2007.

## **REFERENCES**

- 1. Integrating Security and Software Engineering: Advances and Future Vision,** by [Haralambos Mouratidis, Paolo Giorgini](#), IGI Global, 2006.
2. Software Security: Building Security In , by Gary McGraw , Addison-Wesley, 2006
- 3. The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities,** by Mark Dowd, John McDonald, Justin Schuh, Addison-Wesley, 1st edition, 2006
4. Building Secure Software: How to Avoid Security Problems the Right Way  
by John Viega, Gary McGraw, Addison-Wesley, 2001
5. Writing Secure Code, by M. Howard, D. LeBlanc, Microsoft Press, 2<sup>nd</sup> Edition, 2003.
6. Exploiting Software: How to break code, by G. Hoglund, G. McGraw, Addison Wesley, 2004.

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

**M.Tech (Software Engineering)**

**I Year II Semester**

## **SOFT COMPUTING ELECTIVE-II**

### **UNIT: I**

#### **Introduction to intelligent systems and soft computing**

Introduction, Intelligent systems, Knowledge-based systems, Knowledge representation and Processing, soft computing

### **UNIT-II**

#### **Fundamentals of Fuzzy Logic Systems**

Introduction, Background, Fuzzy sets, Fuzzy logic operations, Generalized fuzzy operations, Implication (if-then), Some definitions, Fuzziness and Fuzzy resolution, Fuzzy relations, Composition and Inference, considerations of fuzzy decision-making.

### **UNIT-III**

#### **Fuzzy Logic Control**

Introduction, Background, Basic of Fuzzy control, Defuzzification, Fuzzification, Fuzzy Control Surface, Extensions of Mamdani fuzzy control.

### **UNIT-IV**

#### **Fuzzy Control Architectures**

Fuzzy control architectures, Properties of fuzzy control, Robustness and Stability.

### **UNIT-V**

#### **Fundamentals of Artificial Neural Networks**

Introduction, Learning and acquisition of knowledge, Features of artificial Neural Networks, Fundamentals of Connectionist Modeling.

#### **Major Classes of Neural Networks**

Introduction, The Multilayer Perceptron.

### **UNIT-VI**

#### **Neuro-Fuzzy Systems**

Introduction, Background, Architectures of neuron-fuzzy systems, Construction of neuron-fuzzy systems.

### **UNIT-VII**

#### **Evolutionary Computing**

Introduction, Overview of evolutionary computing, Genetic algorithms and optimization, The schema theorem: the fundamental theorem of genetic algorithms, Genetic algorithm operators, Integration of genetic algorithms with neural networks, Integration of genetic algorithms with fuzzy logic, Known issues in GAs, Population-based incremental learning, Evolutionary strategies, ES applications.

### **UNIT-VIII**

#### **Applications**

Pattern Recognition, Image Processing, Information Retrieval Systems, Share Market Analysis, Soft Computing for Colour Recipe Prediction.

**TEXT BOOKS:**

1. "Soft Computing and Intelligent Systems Design", by Fakhreddine.O. Karray and Clarence De Silva, Pearson Education.
2. "Neuro-Fuzzy and Soft Computing", J.S.R. Jang, C.T. Sun and E.Mizutani, PHI, 2004, Pearson Education 2004.

**REFERENCE BOOKS:**

1. "Fuzzy Logic with Engineering Applications", by Timothy J. Ross, McGraw-Hill, 1997.
2. "Computational Intelligence Principles, Techniques and Applications", by A.Konar, Springer, 2005.
3. "Introduction to Pattern Recognition- Statistical, Structural, Neural and Fuzzy Logic Approaches", by M.Friedman and Abraham Kandal, World Scientific, 2005.

# JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

**M.Tech (Software Engineering)**

**I Year II Semester**

## **MODEL DRIVEN SOFTWARE DEVELOPMENT ELECTIVE-II**

### **UNIT-1**

#### **MDSO Basic Terminology**

Goals of MDSO, MDSO Approach, Overview of MDA concepts, Architecture-Centric MDSO, Common MDSO concepts and terminology, Model-Driven Architecture, Generative Programming, Software factories, Model-Integrated computing, Language-Oriented Programming, Domain specific modeling.

### **UNIT- 2**

#### **Metamodeling**

What is Metamodeling?, Metalevels vs. Level of Abstraction, MOF and UML, Extending UML, UML profiles, Metamodeling & OCL, Examples, Tool-supported Model validation, Metamodeling & Behavior, Pitfalls in Metamodeling, MDSO classification.

### **UNIT- 3**

#### **Model Transformation with QVT**

History, M2M language requirements, Overall Architecture, An Example Transformation, The OMG standardization Process and Tool Availability, Assessment.

### **UNIT- 4**

#### **MDSO Tools:Roles, Architecture, Selection Criteria, and Pointers**

Role of Tools in the Development Process, Tool Architecture and selection criteria, pointers.

**The MDA Standard:** Goals, Core concepts

### **UNIT- 5**

#### **MDSO Process Building Blocks and Best Practices**

Introduction, Separation between Application and domain Architecture Development, Two track Iterative Development, Target Architecture Development Process, Product-line Engineering.

### **UNIT- 6**

#### **Testing**

Test Types, Tests in Model-driven Application Development, Testing the Domain Architecture

#### **Versioning**

What is Versioned? Projects and Dependencies, The structure of Application Projects, The structure of Application Projects, Version management and Build Process for mixed files, Modeling in a team and versioning of partial models

### **UNIT- 7**

#### **Case study:** Embedded Component Infrastructures

Overview, Product-Line Engineering, Modeling, Implementation of Components, Generator Adaptation, Code Generation.

### **UNIT- 8**

**Quality :** Quality in Model Driven Engineering

**TEXT BOOKS:**

1. Model-Driven Software Development-Technology, Engineering, Management by Thomos Stahl, Markus Volter, jul 2006, John Wiley & Sons.
2. Model-Driven Software Development: Integrating Quality Assurance by Jorg Rech, Christian Bunse,2008,Information Science Publishing.(UNIT-8)

**REFERENCE BOOKS :**

1. Model-Driven Software Development by [Sami Beydeda](#) [Matthias Book](#) , [Volker Gruhn](#), Springer.
2. Model Driven Systems Development with Rational Products By Brian Nolan, Barclay Brown, Dr. Laurent Balmelli, Et Al Tim Bohn, 2008,IBM.
3. Model Driven Development with Executable UML by Dragan Milicev, 2009,Wilei India pvt Ltd.
4. Model Driven Software Development by Kevin Lano, Apr 2009, Ci Business Press.