M.Tech (Software Engineering)

Course Structure and Syllabus
(w.e.f 2010-2011)

DEPARTMENT OF INFORMATION TECHNOLOGY
# Department of Information Technology

## M.Tech (Software Engineering) (w.e.f. 2010-2011)

### M.Tech - I Semester

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Course</th>
<th>Credits</th>
<th>Max. Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PerIODS per Week</td>
<td>L</td>
<td>P</td>
</tr>
<tr>
<td>10MT10501</td>
<td>Advanced Data Structures and Algorithms</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>10MT10502</td>
<td>Advanced Software Engineering</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>10MT10503</td>
<td>Middleware Technologies</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>10MT10504</td>
<td>Software Metrics</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>10MT10505</td>
<td>Software Requirements and Estimation</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Elective-I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10MT10506</td>
<td>Software Process Management</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>10MT10507</td>
<td>Software Project Management</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10MT10508</td>
<td>Virtualization and Cloud Computing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10MT10509</td>
<td>Advanced Data Structures and Algorithms and Middleware Technologies</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td>24</td>
<td>3</td>
</tr>
</tbody>
</table>

Chairman, BOS (I.T)
# M.Tech (Software Engineering) (w.e.f. 2010-2011)

## II-Semester

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Course</th>
<th>Periods per week</th>
<th>Credits</th>
<th>Max. Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>L P</td>
<td></td>
<td>Internal</td>
</tr>
<tr>
<td>10MT20501</td>
<td>Service Oriented Architecture</td>
<td>4 -</td>
<td>4</td>
<td>40</td>
</tr>
<tr>
<td>10MT20502</td>
<td>Software Architecture and Design Patterns</td>
<td>4 -</td>
<td>4</td>
<td>40</td>
</tr>
<tr>
<td>10MT20503</td>
<td>Software Reengineering &amp; Reuse</td>
<td>4 -</td>
<td>4</td>
<td>40</td>
</tr>
<tr>
<td>10MT20504</td>
<td>Software Security Engineering</td>
<td>4 -</td>
<td>4</td>
<td>40</td>
</tr>
<tr>
<td>10MT20505</td>
<td>Software Quality Assurance and Testing</td>
<td>4 -</td>
<td>4</td>
<td>40</td>
</tr>
<tr>
<td>10MT20506</td>
<td>Information Retrieval Systems</td>
<td>4 -</td>
<td>4</td>
<td>40</td>
</tr>
<tr>
<td>10MT20507</td>
<td>Soft Computing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10MT20508</td>
<td>Software Reliability</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10MT20509</td>
<td>Industrial Visit/Seminar</td>
<td>- -</td>
<td>2</td>
<td>50</td>
</tr>
<tr>
<td>10MT20510</td>
<td>Software Testing and Design Patterns Lab</td>
<td>- 3</td>
<td>2</td>
<td>25</td>
</tr>
</tbody>
</table>

**Elective-II**

**Total** 24 3 28 315 410 725

## III-Semester

<table>
<thead>
<tr>
<th>Course</th>
<th>Periods per week</th>
<th>Credits</th>
<th>Max. Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L P</td>
<td></td>
<td>Internal</td>
</tr>
<tr>
<td>Project Phase-I</td>
<td>- -</td>
<td>4</td>
<td>50</td>
</tr>
</tbody>
</table>

## IV-Semester

<table>
<thead>
<tr>
<th>Course</th>
<th>Periods per week</th>
<th>Credits</th>
<th>Max. Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L P</td>
<td></td>
<td>Internal</td>
</tr>
<tr>
<td>Project Phase-II</td>
<td>- -</td>
<td>12</td>
<td>50</td>
</tr>
</tbody>
</table>

Approved in BOS (I.T) meeting held on 07.09.2010
UNIT I: Basic Data Structures
Review of Arrays, Stacks, Queues, Circular Queues, Singly Linked Lists, Doubly Linked Lists Circular Linked Lists, Linked stacks and Linked queues, Terminologies and 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 applications, DFS and BFS Traversals.

UNIT IV: Binary Search Trees, AVL Trees and Red – Black Trees

UNIT V: B- Trees, Splay Trees and Hash Tables

UNIT VI: Divide – and – Conquer & Greedy Method

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:

REFERENCE BOOKS:
UNIT-I
**Software and Software Engineering**

**Software Paradigms**

UNIT-II
**Unified Process Model**

**Agile Development**

UNIT-III
**Critical Systems**
A simple safety-critical system, System dependability, Availability and reliability, Safety, Security.

**Critical systems 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

UNIT-VIII
User Interface Design

TEXTBOOKS:

REFERENCE BOOKS:
UNIT-I:
CLIENT/SERVER COMPUTING
Building blocks-types of servers-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- An Overview of the library, Fundamental Namespaces

UNIT- VI:
ADO.NET- Using .NET Framework Data Providers, Using Connection and Command Objects, Accessing Data with Data Readers, Accessing Data with Data Sets, ASP.NET- enterprise services.

UNIT-VII:
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-VIII:
THE OMG WAY
System object model- CORBA timeline- CORBA architecture- ORB- services- facilities-business objects-IIop- transport mechanisms- IDL- Drawbacks of CORBA.
CORBA Component model- POA- CCM components- CCM containers- Meta Object Facility.
Comparison of CORBA, .NET and EJB.
TEXT BOOKS:

REFERENCES:
UNIT-I: Measurement
Measurement in Everyday Life, Measurement in Software Engineering, Scope of Software Metrics

UNIT-II: Basics Of Measurement

UNIT-III: Goal-Based Frame Work for Software Measurement
Classifying Software Measures, Applying Frame Work, Software Measurement Validation

UNIT-IV: Software Metrics Data Collection
Good Data, Definition of Data, Collecting, Storing and Extracting Data.

UNIT-V: Measuring Internal Product Attributes- Size:
Aspects of software size, Length, Reuse, Functionality, Complexity

UNIT-VI: Measuring Internal Product Attributes- Structure:
Types of structural measures, Control-flow structure, Modularity and information flow attributes, Object oriented metrics.

UNIT –VII: Measuring External Product Attributes
Modeling Software Quality, Measuring Aspects of Quality.

UNIT-VIII: Measurement and Management
Planning a Measurement Program: What is metrics plan, why and what, where and when, How, and who.
Measurement in Practice: Success criteria, Measurement in the small, Measurement in the large.

TEXT BOOKS:

REFERENCE BOOKS:
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 Management

UNIT IV:
Software Estimation
Components of Software Estimations, Estimation methods, Problems associated with estimation, Key project factors that influence estimation.

UNIT-V:
Size Estimation- Two views of sizing, Function Point Analysis, Mark II FPA, Full Function Points, LOC Estimation, Conversion between size measures.

UNIT VI:
Effort, Schedule And Cost Estimation

Unit VII:
Requirements Management Tools:
Benefits of using a requirements management tool, commercial requirements management tool, Rational Requisite pro, Caliber – RM, implementing requirements management automation.

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 by Karl E. Weigers, Microsoft Press.
2. Software Requirements and Estimation by Rajesh Naik and Swapna Kishore, Tata Mc Graw Hill

REFERENCES:
UNIT-I: Software Process Maturity

Software Process Maturity Framework: Software Process Improvement, Process Maturity Levels, People in the optimizing process, the need for the Optimizing Process


Software Process Assessment: Overview, Phases, Principles, Assessment Process and Conduct, Implementation Considerations

UNIT-II: Repeatable Process

Managing Software Organizations: Commitment Discipline, the Management System, Establishing a Project Management System

The Project Plan: Principles, Contents, Size Measures, Estimating, Productivity Factors, Scheduling, Project Tracking, the Development Plan, Planning Models, Final Considerations

UNIT-III: Software Configuration Management-I & Software Quality Assurance

SCM: The need for configuration Management, Software Product Nomenclature, Basic SCM Functions, Baselines, Configuration Management Responsibilities, The need for Automated Tools

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

UNIT-IV: Software Standards and Software Inspections


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

UNIT-V: Software Testing and SCM Design-II


SCM Design Phase: Plan, Questions, Support Functions, Requirement Phase, Design Control, The Implementation Phase, Operational data, The Test Phase, SCM for Tools, Configuration Accounting, Software Configuration Audit

UNIT-VI: Defining the Software Process


UNIT-VII: Managed Process  
**Data Gathering and Analysis**: Principles, Process, Software Measures, Data Analysis & considerations

**Managing Software Quality**: Paradigm, Quality Examples and Motivation, Measurement Criteria, Establishing a Software Quality Program, Estimation, Removal Efficiency, Quality Goals and Plans, Tracking and Controlling Software Quality.

UNIT-VIII: Optimizing Process  
**Defect Prevention**: Introduction, Principles, Process Changes for Defect Prevention, Considerations, Management’s Role.

**Automating the Software Process**: Need, What to Automate, Development Environments, Organizational Plans to Automate, Technology Transition, Special Considerations, Productivity, Justification Considerations

**TEXT BOOKS:**
1. Watt S. Humphrey,” Managing the Software Process”, 2003, PEA.

**REFERENCE BOOKS:**
UNIT - I
Conventional Software Management: The waterfall model, conventional software Management performance.

UNIT - II
The old way and the new: The principles of conventional software Engineering, principles of modern software management, transitioning to an iterative process.

UNIT - III
Life cycle phases: Engineering and production stages, inception, Elaboration, construction, transition phases.
Artifacts of the process: The artifact sets, Management artifacts, Engineering artifacts, programmatic artifacts.

UNIT - IV
Model based software architectures: A Management perspective and technical perspective.
Work Flows of the process: Software process workflows, Iteration workflows,

UNIT - V
Checkpoints of the process: Major mile stones, Minor Milestones, Periodic status assessments.
Iterative Process Planning: Work breakdown structures, planning guidelines, cost and schedule estimating, Iteration planning process, Pragmatic planning.

UNIT - VI

UNIT - VII
Project Control and Process instrumentation: The seven core Metrics, Management indicators, quality indicators, life cycle expectations, pragmatic Software Metrics, Metrics automation.
Tailoring the Process: Process discriminants.

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 BOOK :

REFERENCES :
2. Software Project Management, Joel Henry, Pearson Education.
UNIT- I
Introduction to virtualization - Objectives of virtualization, history of virtualization, benefits of virtualized technology, the virtual service desk, what can be virtualized, related forms of computing, cloud computing, software as a service – saas, grid computing, utility computing, virtualization processes, introduction review questions.

UNIT- II
Virtualization Technologies - ubuntu (server edition), altiris, windows, serve, software virtualization, vmware, intel virtualization, red hat virtualization, softgrid application, linux virtualization, desktop, virtualization, hardware virtualization, resource virtualization, processor virtualization, application virtualization.

UNIT- III
Virtualization Technologies - Storage virtualization, virtualization density, para-virtualization, os virtualization, virtualization software, data storage virtualization, intel virtualization technology, thinstall virtualization suite, net framework virtualization, windows virtualization on fedora, storage virtualization technologies, virtualization level, security monitoring and virtualization, oracle virtualization.

UNIT – IV
Virtualization and Storage Management - The heart of cloud computing - virtualization, defining virtualization, why virtualize, what can be virtualized, where does virtualization happen, how does virtualization happen, on the road to storage virtualization, improving availability using virtualization, improving performance through virtualization, improving capacity through virtualization, business value for virtualization.

UNIT – V

UNIT- VI

UNIT- VII

UNIT – VIII
Disaster Recovery- Disaster Recovery Planning, Disasters in the Cloud, Disaster Management. Scaling a Cloud Infrastructure- Capacity Planning, Cloud Scale.
Case Studies: Amazon S3, Google APP Engine, IBM Clouds.
TEXT BOOKS:
2. Cloud Application Architectures Building Applications and Infrastructure in the Cloud by George Reese, Publisher: O'Reilly Media, Released: April 2009, Press Release.

REFERENCE BOOKS:
2. Storage Virtualization: Technologies for Simplifying Data Storage and Management by Tom Clark.
4. Introduction to Cloud Computing: Business & Technology by Timothy Chou
SAEEE VIDYANIKEETHAN ENGINEERING COLLEGE (AUTONOMOUS)

M.Tech (Software Engineering) (w.e.f 2010-11)
I Year (I-Semester)

10MT10509: ADVANCED DATA STRUCTURES & ALGORITHMS AND MIDDLEWARE TECHNOLOGIES LAB

Data Structures Lab

Objectives:
- To make the student learn an object oriented way of solving problems.
- To make the student write ADTS for all data structures.
- Recommended Systems/Software Requirements:
  - Intel based desktop PC with minimum of 166 MHZ or faster processor with at least 64 MB RAM and 100 MB free disk space C++ compiler and STL Recommended.

Week 1: Write C++ programs to implement the following using an array.
   a) Stack ADT   b) Queue ADT

Week 2: Write C++ programs to implement the following using a singly linked list.
   a) Stack ADT   b) Queue ADT

Week 3: Write C++ programs to implement the deque (double ended queue) ADT using a doubly linked list and an Array.

Week 4: Write a C++ program to perform the following operations:
   a) Insert an element into a binary search tree.
   b) Delete an element from a binary search tree.
   c) Search for a key element in a binary search tree.

Week 5: Write C++ programs that use recursive functions to traverse the given binary tree in
   a) Preorder   b) inorder and   c) postorder

Week 6: Write C++ programs that use non-recursive functions to traverse the given binary tree in
   a) Preorder   b) Inorder and   c) Postorder.

Week 7: Write C++ programs for the implementation of bfs and dfs for a given graph.

Week 8: Write C++ programs for implementing the following sorting methods:
   a) Merge sort   b) Heap sort

Week 9: Write a C++ program to perform the following operations
   a) Insertion into a B-tree
   b) Deletion from a B-tree

Week 10: Write a C++ program to perform the following operation
   a) Insertion into an AVL - tree
   b) Deletion from an AVL - tree

Week 11: Write a C++ program to implement All Pairs Shortest Path.

Week 12: Write a C++ program for implementing 8 Queen’s problem.
1. Write an arithmetic operations calculator web application program using ASP.NET
2. Write a client/server arithmetic operations calculator program using java RMI
3. Write a client/server Stock market java program using java RMI
4. Write a client/server address book java program using java RMI.
5. Write a program to create various forms to store customer data, delete customer data, update customer data, retrieve customer data using ASP.NET and MS-ACCESS database.
6. Write a program to create various forms to maintain the Super market using ASP.NET and MS-ACCESS database.
7. Write an address book program using ASP.NET and MS-ACCESS database
8. Write a client/server arithmetic operations calculator program using CORBA.
9. Write a client/server Stock market java program using CORBA.
10. Write a client/server address book java program using CORBA.
UNIT I
SOA and Web Services Fundamentals

UNIT II

UNIT III

UNIT IV
Principles of Service-Orientation

UNIT V
Service Layers

UNIT VI
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)
UNIT VIII


TEXT BOOKS:
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.
7. SOA-Based Enterprise Integration, W.Roshen, TMH.
UNIT I
Envisioning Architecture

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
Introduction to Design Patterns
What is Design Pattern, 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, decorator, 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
Lexi Editor in Design Patterns
TEXT BOOKS:

REFERENCE BOOKS:
1. Software architecture, David M. Dikel, David Kane and James R. Wilson, Prentice Hall PTR, 2001
3. Design Patterns in Java, Steven John Metsker & William C. Wake, Pearson education, 2006
UNIT I:
Legacy software structure, Software reengineering process model:
Business process reengineering: Business processes, A BPR Model.

UNIT II:
Design Extraction:
Reverse Engineering: Goals of reverse engineering, Reverse engineering process, Reverse engineering to understand processing, Code duplication detection, Reverse engineering to understand data, Reverse engineering user interfaces, Tools for reverse engineering.
Software reengineering and its importance, Goals of reengineering, A software reengineering process model, Software reengineering activities.

UNIT III:

UNIT IV:

UNIT V:
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 VI:
Introducing the Reuse-Driven Software Engineering Business: Software Reuse Success Factors, Reuse Driven Software Engineering is a Business.
Architecture Style-I: Object-Oriented Software Engineering, Application and Component Systems

UNIT VII:
Architecture Style-II-Use case Components, Object Components, Layered Architecture
UNIT VIII:
Processes: Application Family Engineering, Component System Engineering, Application System Engineering

TEXT BOOKS:

REFERENCES
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

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

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

UNIT VIII:
Security metrics
Defining security metrics, Diagnosing problems and measuring technical security
TEXT BOOKS

1. Software Security Engineering: A Guide for Project Managers- 
by Julia H. Allen, Sean Barnum, Robert J. Ellison, Gary McGraw, and Nancy R. Mead, 
Pearson Education (India), 2009

2. Security Metrics: Replacing Fear, Uncertainty, and Doubt - by Andrew 

REFERENCES

2. The Art of Software Security Assessment: Identifying and Preventing Software 
Vulnerabilities, by Mark Dowd, John McDonald, Justin Schuh, Addison-Wesley, 
1st edition, 2006
by John Viega, Gary McGraw, Addison-Wesley, 2001
2004.

UNIT II: Software Quality Metrics Overview
Product Quality metrics, In-Process Quality Metrics, Metrics for Software Maintenance, Examples of Metrics Programs, Collecting Software Engineering Data

UNIT III: The Software Quality Metrics Methodology: Establish Software quality requirements, Identify Software quality metrics, implement the software quality metrics, analyze the software metrics results, validate the software quality metrics, Software quality indicators, Fundamentals of Measurement theory

UNIT IV: Building a Software Testing Strategy

UNIT V: Establishing a Software Testing Methodology
Testing Purpose, Defects hard to find, reduce the cost of Testing, 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 Software Testing Tools, Methodology to evaluate automated testing tools, Win runner, QTP, Load Runner, SQA Robot, Silk test, JMeter, JUNIT and Cactus

UNIT VIII: The 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 Systems, Testing Web-based systems, testing off-the-Shelf Software, Testing a Data Warehouse
TEXT BOOKS:

REFERENCE BOOKS:
UNIT-I
Introduction: Definition, Objectives, Functional Overview, Relationship to DBMS, Digital libraries and Data Warehouses.

UNIT-II
Information Retrieval System Capabilities: Search, Browse, Miscellaneous

UNIT-III

UNIT-IV

UNIT-V
Automatic Indexing: Classes of automatic indexing, Statistical indexing, Natural language, Concept indexing, Hypertext linkages

UNIT-VI
Document and Term Clustering: Introduction, Thesaurus generation, Item clustering, Hierarchy of clusters.

UNIT-VII
User Search Techniques: Search statements and binding, Similarity measures and ranking, Relevance feedback, Selective dissemination of information search, Weighted searches of Boolean systems, Searching the Internet and hypertext.
Information Visualization: Introduction, Cognition and perception, Information visualization technologies.

UNIT-VIII
Text Search Algorithms: Introduction, Software text search algorithms, Hardware text search systems.
Information System Evaluation: Introduction, Measures used in system evaluation, Measurement example – TREC results.

TEXTBOOKS :

REFERENCES :
2. Modern Information Retrival By Yates Pearson Education.
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

UNIT-VI
Neuro-Fuzzy Systems
Introduction, Background, Architectures of neuron-fuzzy systems, Construction of neuron-fuzzy systems.

UNIT-VII
Evolutionary Computing

UNIT-VIII
Applications
TEXT BOOKS:

REFERENCE BOOKS:
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

REFERENCE BOOKS
Design Patterns:

1. Write a C++ Program to perform Abstract Factory Design pattern.
2. Write a C++ Program to perform Singleton Design pattern.
3. Write a C++ Program to perform Adapter Design pattern.
4. Write a C++ Program to perform Composite Design pattern.
5. Write a C++ Program to perform Proxy Design pattern.
6. Write a C++ Program to perform Command Design pattern.
7. Write a C++ Program to perform Mediator Design pattern.
8. Write a C++ Program to perform State Design pattern.
9. Write a C++ Program to perform Strategy Design pattern
10. Write a C++ Program to perform Visitor Design pattern

Lab Experiments Using WinRunner

(I) Testing Flight Reservation Application

Functional specifications:

1. The width of the GUI object “Name” in the Flight Reservation window must be 150
2. The height of the GUI object “Order no” in the open order window must be 20
3. The flight schedule frame of Flight Reservation window should have the following properties
   a) Height - 283
   b) Width - 138
   c) X - 11
   d) Y - 84

4. The order information frame of Flight Reservation window should have the following properties
   a) Height - 283
   b) Width - 252
   c) X - 155
   d) Y - 84

5. It is sufficient to verify X or Y Axis, both properties are not necessary
(II) Testing standard Calculator Application

1) Test the inverse operation (inverse of 4,5 using 1/x button)

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 , 1/x</td>
<td>0.25</td>
</tr>
<tr>
<td>5 , 1/x</td>
<td>0.2</td>
</tr>
</tbody>
</table>

2) Test the operations “square root of -6 and square root of 4 ”

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>-6 , sqrt</td>
<td>Err: “Invalid Input for function”</td>
</tr>
<tr>
<td>4 , sqrt</td>
<td>2</td>
</tr>
</tbody>
</table>

3) Clear the display after performing some arithmetic operations

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>4+5=9 , C</td>
<td>Clears the display</td>
</tr>
<tr>
<td>5*3=15 , C</td>
<td>Clears the display</td>
</tr>
</tbody>
</table>

4) Test the multiplication of two numbers

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2 * 3</td>
<td>3.6</td>
</tr>
<tr>
<td>15 * 3</td>
<td>45</td>
</tr>
</tbody>
</table>

5) Test the Division of two numbers

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/2.0</td>
<td>2.5</td>
</tr>
<tr>
<td>4/2</td>
<td>2</td>
</tr>
</tbody>
</table>

6) Test the operations “7+8-9” and “15*3-10”

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>7+8-9</td>
<td>6</td>
</tr>
<tr>
<td>15*3-10</td>
<td>35</td>
</tr>
</tbody>
</table>

7) Test the operations “2% of 600” and “8% of 700”

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>600 * 2%</td>
<td>12</td>
</tr>
<tr>
<td>700 * 8%</td>
<td>56</td>
</tr>
</tbody>
</table>

8) Test the Buttons “MR” and “MS”

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 , MS , C , MR</td>
<td>2</td>
</tr>
<tr>
<td>5*3=15 , MS , C , MR</td>
<td>15</td>
</tr>
</tbody>
</table>

9) Test the Buttons “M+” and “MR”
<table>
<thead>
<tr>
<th>Test Case</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC,2,M+,3,M+,C,MR</td>
<td>5</td>
</tr>
<tr>
<td>MC,9,M+,5,M+,C,MR</td>
<td>14</td>
</tr>
</tbody>
</table>

10) Test the Buttons “MC” and “Backspace”

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>9,MS,C,MR,MC,C,MR</td>
<td>0</td>
</tr>
<tr>
<td>4*5=20,MS,C,MR,MC,C,MR</td>
<td>0</td>
</tr>
<tr>
<td>856,backspace(3 times)</td>
<td>0</td>
</tr>
</tbody>
</table>

(II) Testing Employee Database Application

1) Test the EMPDB Application with suitable operations

2) Implementation of Checkpoints

3) Implementation of Call Function

Lab Experiments Using Quick Test Professional

1) Testing Calculator, EMP DB, Flight Reservation Applications

2) Implementation of Checkpoints Using QTP

3) Implementation of Data Driven Testing in QTP

4) Testing a Web Application Using QTP

5) Synchronization of test cases in QTP