



SARVEPALLI RADHAKRISHNAN UNIVERSITY, BHOPAL

MSE-201 SOFTWARE PROJECT MANAGEMENT

Unit-I

Introduction to Software project Management: Software projects, Contract management and technical project management, Activities covered by software project management , key objectives of effective management , plans , methods & methodologies , problems associated with software projects management control .

Unit-II

Project Planning: Business Planning: determining objectives; forecasting demand for product proposal writing requirement analysis, legal issues; **Technical Planning:** Lifecycle models, types of plans, plan documentation methods: PERT & CPM, Gantt charts, work breakdown structures. Standards, planning for risk management and control Capacity planning.

Unit-III

Software Estimation Techniques, Expert judgement, estimating by analogy, Albrecht function point analysis, COSMIC Full Function Points, COCOMO-a parametric model.
Risk Identification, Risk Assessment, Risk Planning & Risk Management, Evaluating risks to the schedule, Critical chain concepts.

Unit-IV

Monitoring & Control: Creating the framework, collecting the data, Visualizing progress, Cost monitoring, Earned value analysis, Prioritizing monitoring, Change control. Managing people & organizing teams: Team organization, recruiting and staffing, Technical leadership, avoiding obsolescence training.

Unit-V

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)

References:

1. Bob Hughes and Mike Cotterell, Software Project Management, Tata McGraw-Hill Edition.
2. Thayer , Software Engineering Project Management ,2ed ,wiley India
3. Conway , Software Project Management ,Wiley India
4. Pankaj Jalote, Software Project Management in practice Pearson Education..



Unit-I

Software Quality Assurance Framework and Standards : Concept of Software quality Software, Software Quality Attributes ,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

Quality Assurance: Quality Assurance as dealing with defects, Defects Prevention Techniques: Education & training, Formal methods. Defect reduction: Inspection –Direct fault detection and removal, testing –Failure observation and risk identification. Defect Containment: Software fault tolerance and safety assurance and failure containment.

Unit-III

Software Quality Assurance Metrics and Measurement: Software Quality Metrics, Product Quality metrics, –Process Quality Metrics, Metrics for Software Maintenance, Examples of Metric Programs, Software Quality metrics methodology, Establish quality requirements, Identify Software quality metrics, Implement the software quality metrics, analyze software metrics results, and validate the software quality metrics, Software quality indicators, Fundamentals in Measurement theory.

Unit-IV

Software Testing: Functional vs, Structural testing, Test planning and preparation, Test executions, Result Checking and measurement, Test Automation.

Testing techniques: Adaptation, specialization and Integration, Case Study: Hierarchical web Testing. Process Improvement: Process Classification, Process Measurement, Process Analysis and Modelling Formal Verification & Specification, Fault tolerance and failure containment.

Unit-V

Quantifiable Quality Improvement : QA monitoring and measurement , Analysis and follow up actions , Implementations, Integration and tool support , Models for Quality Assessment ,Generalized and product specific models .Risk Identification for quantifiable quality improvement :Traditional statistical analysis techniques, New techniques for risk identification .Software Reliability Engineering :Reliability Analysis Using IDRM(Input Domain Reliability Model) & SRGMs(Software Reliability Growth Model) , TBRMs(Tree based reliability model) for reliability analysis and improvement .

References:

1. Tian , Software Quality Engineering ,Wiley IndiaPub
2. Limaye , Software Quality Assurance , TMH Pub.
3. Pressman , Software Engineering ,TMH Pub
4. Galin , Software Quality Assurance, Pearson Edu.
5. Musa , Software Reliability Engineering , TMH Pub
6. Wieczorek, Software Quality , Springer



SARVEPALLI RADHAKRISHNAN UNIVERSITY, BHOPAL

MSE-203 SOFTWARE ARCHITECTURE

UNIT - I

Software Architecture terms: Component, Relationship, View, Architectural Styles, Type of IT Architecture, Frameworks, Patterns, Methodologies, Processes, Functional and Non-functional Properties of Software Architectures .

UNIT - II

Designing Architecture: Quality Architecture, Architecture Life Style, Reconstructing Software Architecture, Service Oriented Architecture, Service Oriented Analysis and Design, Trends in SOA, Enterprise Wide SOA and Application.

UNIT – III

Architectural Styles: Pipes and Filters, Data Abstraction and Object-Oriented, Event-Based, Implicit Invocation, Layered Systems, Repositories, Interpreters, Process Control, Heterogeneous Architectures.

UNIT – IV

Formal Models and specifications: Formalizing the architecture of a Specific system ,formalizing an architectural style and architectural design space ,Requirements for Architecture –Description languages Tools for architectural design :Universal connector language ,automated support for architectural design , observations about environments for architectural design .

UNIT-V

Analyzing Architecture: Architecture Tradeoff Analysis Method(ATAM), Cost Benefit Analysis Methods(CBAM). Case Studies: Key word in Context, The World Wide Web a case study in interoperability, Instrumentation software, cruise control, Mobile Robotics .Three Vignettes in Mixed style.

References:

- 1.. Shankar Kambhampaty, Service Oriented Architecture, Wiley India
- 2 .Buschmann ,Pattern oriented Software Architecture Vol 1. ,Wiley India
3. Len Bass, Paul Clements, Rick Kazman, Software Architecture in Practice, Pearson Edu.
4. Shaw , Software Architecture , PHI



UNIT-1

Software Testing Fundamentals –Testing objectives, Testing lifecycles, Test cases, human error, testing and debugging, general principles of testing, test metrics, Agile methodology and Its Impact on Testing, Verification and Validation. Failure, Error, Fault, and Defect

UNIT-2

Testing Approaches - Static testing, structured group examination static analysis, Control flow and data flow Testing, Determining Metrics, Dynamic Testing, Black Box testing, equivalence Class Partitioning, Boundary Value Analysis, state transition test, cause effect graphing and decision table technique and used case testing and Advanced black box and white box techniques techniques Gray box testing, intuitive and Experience based

UNIT –3

Software Reliability-Reliability models, Reliability measures, verification and validation planning, Top down versus bottom up Testing Functional Vs Structured Testing, mutation testing, Test planning and Management, Testing process, Maturity Models.

UNIT –4

Types of Testing- Concept of Unit Testing, Domain testing, Concept of Integration Testing. System testing acceptance testing, Alpha &Beta testing, Installation Testing, Usability Testing, Regression testing, Performance testing, Load testing, Stress testing, Security testing, Gorilla testing, Syntax Based Testing .

UNIT-5

System Tests- Functionality Tests ,Robustness Tests, Interoperability Tests, Scalability Tests, Documentation Tests, Testing Tools-Automation of Test execution, Requirement Tracker, WinRunner, Load Runner, Test Director, Test Process , Test Plans ,

References:

1. Naik,Software Testing and Quality Assurance, Wiley India pub.
2. Limaye , Software Testing , TMH Pub
3. Ammann & Offutt , Introduction To Software Testing , Cambridge Univ Press
4. k.v.k.k.prasad, Software testing concepts Tools, Dreamtech press



SARVEPALLI RADHAKRISHNAN UNIVERSITY, BHOPAL

MSE-205 OBJECT ORIENTED ANALYSIS & DESIGN

Unit-I

Object Oriented Modeling, Characteristics Object Oriented Model, An Object Model, Benefits of OO Modeling, Introduction to OOAD tools, Object Oriented Analysis, Differences between Structured Analysis and Object Oriented Analysis. Analysis Techniques. UML: Introduction. Object Model Notations, Modeling using class diagrams, Interaction diagram, Use case modeling

Unit-II

System Design: Breaking into Subsystems, Concurrency Identification, Management of data store, Controlling events between Objects, Handling Boundary Conditions.

Object Design: Object Design Steps, Designing a Solution, Choosing Algorithms & data structures , Defining Classes and delegation of responsibilities to Methods . Inheritance Adjustment, Association, Object Representation, Design Optimization, Design Documentation

Unit-III

Object Modeling: Advance Modeling Concepts, Multiple Inheritance, Generalization as an Extension, Generalization as a Restriction, Metadata, Constraints.

Dynamic Modeling: Events, State and State Diagram , Advance Concepts in Dynamic Modeling, Concurrency.

Functional Modeling: Functional Models, Data Flow Diagrams, Features of a DFD, Design flaws in DFD, Relationship between Object, Dynamic, and Functional Models

Unit-IV

Implementation Modeling: Fine –tuning classes, Fine –tuning Generalization, Realizing Associations, Testing, OO languages , Implementaion structure ,Implementation Structure ,Implementing Functionality , Relational Database Schema for Object Modes, Object Classes to Database Tables, Mapping Associations to Tables, Mapping Generalizations to Tables, Interfacing to Database , Object Mapping with Databases .

Unit-V

Design of a object oriented system: Entity , interfaces, control, persistence, system classes. Design Relationships: Object Oriented design Process : Use case Model, Modeling Classes Interaction, UML Design and implementation : software Implementation, Component and deployment Diagram. Case Study: Warehouse Management System, Telecom, Managing Object-Oriented Software Engineering Object-Oriented Methods .

References:

1. Rumbaugh , Object Oriented Modeling and Design with UML , Pearson Edu.
2. Simon Bennett, Steve McRobb and Ray Farmer, Object Oriented system Analysis and Design Using UML, TMH
3. Docherty , Object Oriented Analysis & Design with UML , Wiley India
4. Ivar Jacobson, Patrik Jonsson: ,Object – Oriented Software Engineering , Pearson.Edu