Software Engineering Advanced (10 Days)

By Dr. Vishwanath Rao

1. SOFTWARE ENGINEERING FUNDAMENTALS

• Software Development Challenges
• Software Scope
• Software Engineering Discipline
• Software Methodologies and Related Process Models
• The Human Side of Software Development
• Introduction to Agile Software Engineering

2. SOFTWARE DEVELOPMENT LIFE CYCLES (SDLCs) – Part I
• Process Models and Solution Life Cycle Phases
• Traditional Life Cycle Models
o Waterfall
o V
o Phased
o Evolutionary
o Spiral
o CBSE

• Alternative Techniques
o UP
o RAD
o JAD
o PSP/TSP
o Prototyping

3. SOFTWARE DEVELOPMENT LIFE CYCLES (SDLCs) – Part II
• Agile Software Engineering Process Models
o Extreme Programming
o Agile Software Development
o DevOps
o Site Reliability Engineering (SRE)
• Roles and Types of Standards
• ISO 12207: Life Cycle Standard
• IEEE Standards for Software Engineering Processes and Specifications

4. SOFTWARE ENGINEERING TOOLS
• Requirements Management Tools (e.g., IBM Rational Doors)

• Design Tools (e.g., Sparx Enterprise Architect)
• Development Tools
o IDEs (e.g., Xcode, Eclipse, IntelliJ IDEA, NetBeans, Microsoft Visual Studio, Atom)
o Source Control Management (e.g.,GitHub)
o Release Orchestration (e.g., OpenMake)
o Collaboration (e.g., Jira, Trello, Slack)
• Operations Management Tools
o Database Automation (e.g., Datical)
o Deployment (e.g, ElasticBox)
o Configuration Management (e.g., Ansible, Chef, Puppet)
o Continuous Integration (e.g, Jenkins)
o Container Management (e.g., Docker, Kubernetes)

• Testing Tools and Frameworks
o Testing Tools (e.g., Junit, Selenium)
o PaaS (e.g., PythonAnywhere, AWS Code9, Heroku)
• Management and Monitoring Frameworks
o - AIOps (e.g, Splunk, Logstash)
o - Analytics (e.g., Dynatrace, ElasticSearch)
o - Monitoring (e.g., Nagios)

• Security Frameworks (e.g., Snort, BlackDuck)
• Cloud Platforms (e.g., AWS, Azure, GCP, IBM Cloud)
• Project Management (e.g., Scoro, Basecamp, Microsoft Project)
• Selecting Appropriate Tools

5. PLANNING AND MANAGING REQUIREMENTS
• Requirements Development Methodology
• Specifying Requirements
• Eliciting Accurate Requirements
• Documenting Business Requirements
• Defining User Requirements
• Validating Requirements
• Achieving Requirements Traceability
• Managing Changing Requirements
• Reviews, Walkthroughs, and Inspections
• Requirements Modeling
• Agile Requirements Engineering

6. INTRODUCTION TO SOFTWARE ANALYSIS AND DESIGN
• Roles of Analysis and Design
• Traditional Data and Process Modeling Approaches
• Performing Requirements Analysis
• Object-Oriented Modeling

• User Experience Design
• Design for Mobility
• Selecting and Combining Approaches
• Creating a Data Model

7. BUSINESS MODEL ENGINEERING
• Business Model Capture Tools
• Process Modeling
• Capturing the Organization and Location Aspects
• Developing a Process Model

8. FROM ANALYSIS AND DESIGN TO SOFTWARE ARCHITECTURES
• Building an Object Model using UML
• Architectural and Pattern-Based Design
• Model Driven Architectures
• Business Process Management
• Achieving Optimum-Quality Results
• Selecting Kits and Frameworks (e.g., Bootstrap, .Net)
• Using Open Source, free, freemium, paid, and Enterprise software components

9. BUILDING SOFTWARE
• Language and Platform Issues
• Component Infrastructures
• Pair Programming
• Refactoring
• Test Driven Development (TDD)
• Distributed Development and Agile Methods Scalability

10. SOFTWARE VERIFICATION AND VALIDATION
• Unit Testing
• Integration and System Testing
• Static Confirmation
• Dynamic Testing
• Traceability Matrices
• Automated Testing
• Other Specialized Testing

11. SOFTWARE QUALITY AND SECURITY
• Software Quality Concepts
• Software Configuration Management (CM)

• Software Quality Assurance (SQA)
• Software Quality and Agile Methods
o Automated and Manual Functional Testing
o Acceptance testing
o Mock objects
o User interface testing (HTTPUnit, Canoo)
o Performance testing
• Software Metrics and Analytics
• Quality and Process Standards and Guidelines
o ISO 9000
o SWEBOK
o ISO 15504
o SEI's Capability Maturity Model (CMM)
o CMM Integration (CMMI)
• Software Security Engineering

12. RISK MANAGEMENT IN SOFTWARE ENGINEERING PROJECTS
• Project Management Concepts
• Project Planning and Estimation
• Cooperative roles of software engineering and project management
• Developing risk response strategies
• Risk Management in Agile Processes
• Agile Project Planning
• Project Management Metrics
• Software Support Strategies

13. Advanced Topics
• Software Process Improvement
• Quantifying Software Specifications Using Formal Methods
o Using Set Theory and Logic
o Verifying Requirements Mathematically
• Emerging Trends in Software Engineering
• Data Science for Software Engineers
• Measuring User Satisfaction
• Software Engineering Ethics