RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL

New Scheme Based On AICTE Flexible Curricula

Computer Science and Engineering, VII-Semester

CS701 Software Architectures

Pre-Requisite: Software Engineering

Course Outcomes:
After completing the course student should be able to:

1. Describe the Fundamentals of software architecture, qualities and terminologies.
2. Understand the fundamental principles and guidelines for software architecture design, architectural styles, patterns, and frameworks.
3. Use implementation techniques of Software architecture for effective software development.
4. Apply core values and principles of software architectures for enterprise application development.

Course Contents:

Unit 1. Overview of Software development methodology and software quality model, different models of software development and their issues. Introduction to software architecture, evolution of software architecture, software components and connectors, common software architecture frameworks, Architecture business cycle – architectural patterns – reference model.

Unit 2. Software architecture models: structural models, framework models, dynamic models, process models. Architectures styles: dataflow architecture, pipes and filters architecture, call-and return architecture, data-centered architecture, layered architecture, agent based architecture, Micro-services architecture, Reactive Architecture, Representational state transfer architecture etc.

Unit 3. Software architecture implementation technologies: Software Architecture Description Languages (ADLs), Struts, Hibernate, Node JS, Angular JS, J2EE – JSP, Servlets, EJBs; middleware: JDBC, JNDI, JMS, RMI and CORBA etc. Role of UML in software architecture.

Unit 4. Software Architecture analysis and design: requirements for architecture and the life-cycle view of architecture design and analysis methods, architecture-based economic analysis: Cost Benefit Analysis Method (CBAM), Architecture Tradeoff Analysis Method (ATAM). Active Reviews for Intermediate Design (ARID), Attribute Driven Design method (ADD), architecture reuse, Domain –specific Software architecture.

Unit 5. Software Architecture documentation: principles of sound documentation, refinement, context diagrams, variability, software interfaces. Documenting the behavior of software elements and software systems, documentation package using a seven-part template.

Text Books

1. Bass, L., P. Clements, and R. Kazman, “Software Architecture in Practice”, Second Edition, Prentice-Hall.

2. Jim Keogh, “J2EE – Complete Reference”, Tata McGraw Hill.

3. Dikel, David, D. Kane, and J. Wilson, “Software Architecture: Organizational Principles and Practices”, Prentice-Hall.

Reference Books

1. Bennett, Douglas, “Designing Hard Software: The Essential Tasks”, Prentice-Hall, 1997.

2. Clements, Paul, R. Kazman, M. Klein, “Evaluating Software Architectures: Methods and Case Studies”, Addison Wesley, 2001.

3. Albin, S. “The Art of Software Architecture”, Indiana: Wiley, 2003.

4. Robert Mee, and Randy Stafford, “Patterns of Enterprise Application Architecture”, Addison-Wesley, 2002.

5. Witt, B., T. Baker and E. Meritt, “Software Architecture and Design: Principles, Models and Methods”, Nostrand Reinhold, 1994.