Architect Enterprise Applications with Java EE

Description

This Architect Enterprise Applications with Java EE training teaches you how to develop robust architectures for enterprise Java applications. Learn how to use Java Platform, Enterprise Edition (Java EE) technology. By enrolling in this course, you'll understand how Enterprise Java applications developed using the architecture as a guideline can accommodate rapid change and growth.

You'll also learn the strategies needed to create application blueprints that work well when implementing Java EE technologies. These strategies include effective decision-making through the use of non-functional qualities (such as scalability and flexibility), Java EE technology blueprints and design patterns.

Delegates will learn how to

- Define the Enterprise Architect's roles, responsibilities and deliverables.
- Identify non-functional requirements (NFRs) and describe common problems and solutions.
- Translate business requirements into an architecture.
- Weigh choices in architecting the client, web, business, integration and data tiers.
- Apply various evaluation criteria to choosing architectural elements and patterns, tools, servers and frameworks.

Introducing Enterprise Architecture

- An Architect’s Roles and Responsibilities
- What is Enterprise Architecture?

Introducing Fundamental Architectural Concepts

- Architectural Modeling Using UML
- What is an Enterprise Architecture Framework
- Architectural Deliverable Artifacts
- Architecture Workflow
- 4 + 1 View Model
- Architectural Patterns
- Distinguish between architecture and design

Developing a Security Architecture

- Analyzing the Impact of Security in Distributed Computing
- Understanding Web Services Security
- Examining Security in the Java EE Technology

Understanding Non-Functional Requirements

- Common Practices for Improving Qualities
- Prioritizing Quality-of-Service (QoS) Requirements
- Examining Non-Functional Requirements (NFRs)
- Inspecting QoS Requirements for Trade-offs

Defining Common Problems and Solutions: Risk Factors and System Flexibility

- Identifying Risk Factors
- Designing a Flexible Object Model

Defining Common Problems and Solutions: Network, Transaction and Capacity Planning

- Describing Network Communication Guidelines
- Justifying the Use of Transactions
- Planning System Capacity

Developing an Architecture for the Client Tier

- Discovering Reusability in the Client Tier

- Deployment Strategies for the User Interface
- Client Tier Development Roles
- Security Concerns in the Client Tier
- Selecting User Interface Devices and Technologies
- Information Architecture Client Concerns
- Testing

Developing an Architecture for the Web Tier

- Scaling the Web Tier
- Comparing Web Tier Frameworks
- Providing Security in the Web Tier
- Seperation of Concerns
- Responsibilities of the Web Tier

Developing an Architecture for the Business Tier

- Business Tier Technologies
- Development Best Practices
- Architecting the Domain Model

Developing an Architecture for the Integration and Resource Tiers

- Examining Service-Oriented Architecture (SOA)
- Reviewing Java Integration Technologies
- Applying Integration Patterns
- Examining Enterprise Information System Integration

Evaluating the Software Architecture

- Evaluating Software Architectures
- Evaluating Java EE Technologies
- Creating System Prototypes
- Selecting Servers and Frameworks