

**Solution Architect Practitioner's Guide (AWS Certification Prep)**

**Course Number:** SA-100WA
**Duration:** 4 days

**Overview**

This Solution Architect training course teaches attendees how to develop non-functional requirements, create architecture views, use patterns, and perform architecture reviews. This course also prepares participants for the AWS Certified Solutions Architect - Associate Certification exam.

**Prerequisites**

No prior experience is presumed.

**Materials**

All Solution Architect training students will receive comprehensive courseware.

**Software Needed on Each Student PC**

A modern web browser and an Internet connection.

**Objectives**

* Work with business use cases and requirements to identify architecturally significant requirements
* Define architecture to fulfill the requirements, ensuring that it is traceable, verifiable, and measurable
* Communicate the architecture to technical teams for implementation, use, and ongoing support
* Demonstrate the value of the architecture to the business
* Identify and apply appropriate techniques to build momentum in the rapid delivery of successful solutions

**Outline**

* Introduction
* Solution Architecture Overview
	+ Why is Solution Architecture Important?
	+ Communications Vehicle Among Stakeholders
	+ The Project is Organized Around Architectural Elements
	+ What is a System?
	+ Why Focus on Structure?
	+ Solution Architecture Context
	+ Solution Architecture & Domains
	+ SA Spans All Domains
	+ Relationship to EA Architecture Development Process
	+ Example: Solution Architecture Stakeholders
	+ Solution Architecture Deliverables
	+ EA Involvement in SA
	+ Architecturally Significant
	+ Resource: Software Engineering Institute (SEI)
	+ Resource: SWEBOK
	+ Resource: OpenUp
	+ Resource: Microsoft Library
	+ Group Discussion: Methodologies
* Core Solution Architecture Methods
	+ Shared Vision
	+ Example Shared Vision
	+ Draw the Boundary
	+ Well-defined Interface
	+ Example: Context Diagram
	+ Identify the External Interfaces
	+ Subsystems
	+ Subsystem Context Diagram
	+ Layers
	+ Example: Subsystems with Layers
	+ Components
	+ Decomposing the System
	+ Partitioning Patterns
	+ Example Partitioning Based on Patterns
	+ Example: Healthcare SOA Framework
	+ Requirements Allocation
	+ Group Discussion: Requirements Allocation
	+ Configuration Management Implications
	+ Release Management Implications
	+ Testing Implications
	+ Work Pattern & Skill Set Implications
	+ Work & Build Dependencies
	+ Increment/Sprint Planning
	+ Sizing Implications
	+ More Than Executable Architecture
	+ Development Architecture
	+ Operations Architecture
	+ Group Discussion: Integrating Operations Architecture
* Architecture Concepts
	+ Fundamental Architecture Concepts
	+ Abstraction
	+ Coupling
	+ Cohesion
	+ Decomposition & Modularization
	+ Encapsulation & Information Hiding
	+ Separation of Interface & Implementation
* Stakeholder Management
	+ When to Focus on Stakeholder Management
	+ Steps in the Stakeholder Management Process
	+ Identifying Stakeholders
	+ Points to Consider
	+ Example Stakeholders & Concerns
	+ Classifying Their Positions: The Stakeholder Matrix
	+ Determining the Stakeholder Management Approach and Tailoring the Deliverables: The Stakeholder Map
	+ Example: Stakeholder Map
	+ Template: Stakeholder Map Matrix Template
* Views & Viewpoints
	+ Example View: Claim Handling from a Process Viewpoint
	+ Example View: Claim Handling from a Data Viewpoint
	+ Example View: Claim Handling Project from a Financial Viewpoint
	+ Contents of Views and Viewpoints
	+ Example Formal Viewpoint: Security
	+ Software Architecture Viewpoints: 4+1
* Architecture Requirements
	+ Architecture Quality Attributes
	+ Quality of Service Requirement Categories
	+ Checklist: Quality Attribute (QA) Categories
	+ Trade-off Analysis
	+ Group Discussion: Trade-offs
	+ Technique: Requirement Patterns
	+ Tool: Non-Functional Requirement Patterns
	+ Checklist: Requirement Statement Best Practices
	+ Technique: Architecture Change Cases
	+ Template: Elements of a Change Case
	+ Example: Change Case
	+ Eliciting Change Cases
	+ Group Discussion: Change Case
* Architecture Requirement Techniques
	+ Requirements Management Activities
	+ Best Practices
	+ Baselining Requirements
	+ Desirable RM Repository Characteristics
	+ Example: Behavior-Driven Development (BDD)
	+ Why Traceability?
	+ Identifying Candidate Tactics, Patterns, and Styles
	+ Requirements-Tactics-Patterns-Styles
	+ Making Architectural Decisions
	+ Architectural Measurement
	+ Implementing Architectural Measurement
	+ Example Metrics
* Quality of Service (QoS) Requirements
	+ Qualities of Service and Design
	+ Performance: Requirements
	+ Performance: Response Time Pattern
	+ Performance: Transaction Time Patterns
	+ Performance: Throughput Pattern
	+ Scalability: Capacity Patterns
	+ Reliability & Availability
	+ Mean Time Between Failures (MTBF)
	+ Availability: Pattern
	+ Extensibility
	+ Maintainability
	+ Manageability
	+ Security
	+ Cultural Adaptability
	+ Portability
	+ Testability
	+ Usability
	+ Upgradeability
	+ Recoverability
	+ Recovery Time Objective (RTO)
	+ Recovery Point Objective (RPO)
	+ Prioritizing Quality of Service Requirements
	+ Inspecting QoS Requirements for Trade-off Opportunities
	+ Quality of Service Testing
* Business Architecture
	+ Business Architecture Models & Diagrams
	+ Business Process Concepts
	+ Example: Medicaid Business Process Model
	+ Example: Medicaid Business Process Definition
	+ Business Function Concepts
	+ Example: HL7 EHR Functional Model
	+ Example: Process Flow Diagram
	+ Resource: Business Analysis Book of Knowledge (BABOK)
	+ Resource: Business Architecture Body of Knowledge (BIZBOK™)
* Data Architecture
	+ Data Modeling
	+ Conceptual Data Model
	+ Example: Conceptual Data Model
	+ Example: Property & Casualty Conceptual Data Model
	+ Example: Data Entities
	+ Logical Data Model
	+ Normalization
	+ Abstraction
	+ Example: Logical Data Model
	+ Physical Data Model
	+ Example: Physical Data Model
	+ Data Modeling Notation
	+ Entity Relationship Diagram (ERD)
	+ Cardinality
	+ Annotated Relationships
	+ Subtype Relationship
	+ Resource: DAMA DMBOK
* Data Domain Systems
	+ First, Some Practical Observations
	+ Data vs Information
	+ The Need to Bridge the Gap
	+ The Three vs. of Big Data
	+ Limitations of Relational Databases
	+ Limitations of Relational Databases (Cont'd)
	+ What are NoSQL (Not Only SQL) Databases?
	+ What are NoSQL (Not Only SQL) Databases?
	+ The Past and Present of the NoSQL World
	+ NoSQL Database Properties
	+ NoSQL Benefits
	+ NoSQL Database Storage Types
	+ The NoSQL Systems CAP Triangle
	+ Limitations of NoSQL Databases
	+ Big Data Sharding
	+ Sharding Example
	+ Mix-and-match Approach
	+ Amazon S3
	+ Amazon Storage SLAs
	+ Amazon Glacier
	+ Data Lifecycle Management with Amazon S3
	+ Microsoft Azure Data Management Capabilities
	+ Hadoop
	+ Hadoop Distributed File System
	+ HBase
	+ Apache Spark
	+ The Spark Platform
	+ Running Spark on a Cluster
	+ MongoDB
	+ MongoDB Use Cases
	+ Apache Cassandra
	+ Apache Cassandra Design
	+ Cassandra's Main Features and Qualities of Service
* Supporting QoS Requirements
	+ Tactics
	+ Availability Tactics
	+ Supporting System's High Availability
	+ The CAP Theorem
	+ Mechanisms to Guarantee a Single CAP Property
	+ Modifiability Tactics
	+ Horizontal and Vertical Scalability
	+ Leveraging Cloud Scaling Services
	+ Performance Tactics
	+ Achieving the Performance You Need
	+ Security Tactics
	+ Single Sign-On (SSO) with Federated Identity Management
	+ OpenID
	+ OpenID Communication Diagram
	+ OAuth 2.0
	+ OAuth 2.0 Communication Diagram
	+ OpenID Connect
	+ OpenID Connect Communication Diagram
	+ Operational Security in the Cloud
	+ DevOps Security Concerns
	+ Testability Tactics
	+ Achieving Testability with Test-Driven Development and Continuous Integration
	+ Typical Setup for OSS-based Continuous Integration
	+ Responsive Web Design (RWD) Support for Usability
* Solution Architecture Styles
	+ Catalog of Architectural Styles
	+ Asynchronous Messaging
	+ Message Oriented Middleware (MOM)
	+ Example MOM implementation Platforms
	+ MOM Messaging
	+ MOM Qualities
	+ Publish/Subscribe Messaging
	+ Point-to-Point Messaging (P2P)
	+ MOM Related Standards
	+ MOM Example
	+ Service Oriented Architecture (SOA)
	+ Service-Oriented Interaction Model
	+ SOA Characteristics
	+ Microservices
	+ Microservices Architecture
	+ Microservices vs. Enterprise Service Bus (ESB)
	+ Many Flavors of Web Services
	+ Understanding REST
	+ Principles of RESTful Services
	+ SOAP and RESTful Web Services
* Patterns
	+ What are Patterns?
	+ Elements of a Pattern
	+ Pattern Levels
	+ Pattern Types
	+ How to Start Using Patterns?
	+ Common Architectural Patterns
	+ Layers Pattern
	+ Example: Retail Layered Architecture
	+ Object-Oriented Design Patterns
	+ OO Design Patterns
	+ Structural Design Pattern: Facade Pattern Example
	+ Enterprise Integration Patterns
	+ Messaging Systems: Overview
	+ Example Pattern: Pipes and Filters
	+ Example: Monitoring Credit Bureau
	+ EAA Patterns
	+ Model-View-Controller (MVC) Pattern
	+ SOA Patterns
	+ Example: Saga Pattern
	+ Business Process Patterns
	+ Example: Synchronizing Merge Pattern
	+ Configuration Management Patterns
	+ New Patterns Continue to Emerge
	+ Group Discussion: Patterns
* Technical Architecture
	+ What is Technical Architecture?
	+ Two Components of Technical Architecture
	+ Software Architecture
	+ What a Technical Architecture is Not
	+ Architectural Views
	+ Rational Unified Process (RUP) 4 + 1 Views
	+ The Implementation View
	+ The Deployment View
	+ Technology Modeling
	+ The Essential Project: Technology Modeling overview
	+ Layers of the Enterprise Architecture
	+ Relationship with Other Architectures
	+ Relationship between Business Architecture and TA
	+ Relationship between EA, SA, and TA
	+ SA vs. TA
	+ Technical Architecture's Scope
	+ The Technical Architect's Areas of Expertise
	+ The Technical Architect's Tasks
	+ Target System Elements Identification
	+ Technical Architecture Governance
	+ System Capacity Planning
* Defining the Cloud
	+ A Bit of History
	+ Cloud Computing at a Glance
	+ Electrical Power Grid Service Analogy
	+ The NIST Perspective
	+ Five Characteristics
	+ On-demand Self-Service (NIST Characteristic)
	+ Broad Network Access (NIST Characteristic)
	+ Resource Pooling (NIST Characteristic)
	+ Rapid Elasticity (NIST Characteristic)
	+ Measured Service (NIST Characteristic)
	+ The Three Cloud Service Models: IaaS, PaaS and SaaS
	+ The Four Cloud Deployment Models (NIST)
	+ The NIST Cloud Definition Framework
	+ A Hybrid Cloud Diagram
	+ Cloud Services
	+ Managed vs. Unmanaged Services
	+ Shared Responsibility Model
	+ The AWS (Simplified) Service Stack
* Architecture Deliverables
	+ Documentation Best Practices
	+ Architecture Requirements Document
	+ Template: Requirements Specification
	+ IEEE Architectural Description Document
	+ Template: Architectural Description Document
	+ TOGAF Architecture Definition Document
	+ Templates: Architectural Definition Document
	+ Group Discussion: Architecture Definition Documents
	+ Interface Specifications
	+ Interface Specification Best Practices
	+ Interface Design Document
	+ Template: Interface Design Document
	+ Database Design Document
	+ Template: Database Design Document
	+ Platform Design Document
	+ Template: Platform Design Document
	+ Architecture Decision Document
	+ Template: Architecture Decision Document
	+ Verbal Supports: CREST
	+ Group Discussion: Presentations
* Reference Architecture
	+ Reference Architecture Components
	+ Reference Architecture Context
	+ Architecture Principles
	+ Qualities of a Good Set of Principles
	+ Templates: Principle & Principle Catalog
	+ Applying Architecture Principles
	+ Policies
	+ Template: Policy
	+ Example: Governance Policies
	+ Example: SOA Policy
	+ Example: Policies
	+ Reference Models
	+ Example: Reference Model
	+ Example: Architecture Use Cases
	+ Example: SOA Reference Architecture
	+ Practices: Standards & Guidelines
	+ Example: Interoperability Standards
	+ Resource: Financial Industry Organizations
	+ Resource: Health Industry Organizations
	+ Resource: Retail Industry Organizations
	+ Resource: Technical Organizations
	+ Industry Organizations
	+ Insurance Industry Standards ROI
	+ Requirements
	+ Example: Mobile Security Reference Architecture
	+ Example: MSRA Requirements
	+ Architecture Building Blocks: Reusable Requirement Sets
	+ Resource: NIST Security Requirements
	+ Example: COTS Standard Requirements Set
* Packaged Software and SaaS
	+ Alternatives to Custom Development and Hosting
	+ Open Source Software
	+ Frameworks
	+ Cloud Computing
	+ Integration of Mixed Solutions
	+ Implications for Architecture
	+ Packaged Software Advantages & Disadvantages
	+ SaaS Advantages and Disadvantages
	+ Open Source Advantages and Disadvantages
	+ Integration Strategies
	+ The API Economy
	+ COTS
	+ Typical COTS Architecture
* Building Modern Applications
	+ Next-Generation Methodologies, Approaches, Tools, and Applications
	+ Web 2.0
	+ Rich Internet Client Applications
	+ Single Page Applications (SPA) with AngularJS
	+ Two-way Data Binding (the AngularJS Way)
	+ Other Client Side MV(C) Frameworks
	+ "Rich Client" - "Thin Server" Architecture
	+ Mobile Platforms
	+ Types of Mobile Applications
	+ Native Mobile Applications
	+ Mobile Web Applications
	+ Hybrid Mobile Applications
	+ Hybrid App Tools and Frameworks
	+ RIA as a Driving Force to Turn the "Thin Server" into Microservice(s)
	+ So, How Can Microservices Help Me?
	+ The Data Exchange Interoperability Consideration
	+ Microservices in Their Purest Form: AWS Lambdas
	+ The Microservices Architecture Design Principles
	+ Decentralized Processing
	+ Crossing Process Boundary is Expensive!
	+ Managing Microservices
	+ Traditional Enterprise Application Architecture (Simplified)
	+ Microservices Architecture Example (Simplified)
	+ Design for Failure
	+ Fault Injection During System Testing
	+ Architecting in the Cloud
	+ The Building Blocks of a Fault-tolerant Application on AWS
	+ Dev and Ops Views
	+ What is DevOps?
	+ More DevOps Definitions
	+ DevOps and Software Delivery Life Cycle
	+ Main DevOps Objectives
	+ The Term "DevOps" is Evolving!
	+ Infrastructure as Code
	+ Prerequisites for DevOps Success
	+ Alignment with Business Needs
	+ Collaborative Development
	+ Continuous Testing and Integration
	+ Continuous Release and Deployment
	+ Continuous Application Monitoring
	+ Standing Up DevOps
	+ Select DevOps Techniques and Practices
	+ Containerization and Virtualization
	+ Machine Images On Demand
	+ Virtualization
	+ Hypervisors
	+ Docker Containers
	+ Docker as Platform-as-a-Service
	+ Docker Integration
	+ Docker Application Container Public Repository
	+ Kubernetes
	+ Other Containerization Systems
	+ Where to Use Virtualization and Containerization
* Conclusion