
How You Will Benefit
This training course covers the fundamentals of solution architecture concepts and techniques. The course covers stakeholder identification, including identifying the stakeholder viewpoints. It covers the concept of architecturally significant that drives the architecture artifacts, including system process models, system use cases, change cases, and architecture requirements. The course material and techniques are based on industry best practices such as 4+1 architecture, business-driven architecture, and requirements-driven architecture. The course includes hands-on labs related to core architecture skills such as developing non-functional requirements, developing architecture views, using architecture patterns, and performing architecture reviews.

Objectives
In this training, attendees will learn how to:

• 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

Course Outline
Chapter 1. Introduction
• 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

Chapter 2. 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
• Solution Architecture
• Example: Solution Architecture Stakeholders
• Solution Architecture Deliverables
• EA Involvement in SA
• Architecturally Significant

Pre-requisite
• N/A
Chapter 3. Core Solution Architecture Methods

- Group Discussion: Architecture
- Resource: Software Engineering Institute (SEI)
- Resource: SWEBOK
- Resource: OpenUp
- Resource: Microsoft Library
- Group Discussion: Methodologies
- Summary

Chapter 4. Stakeholder Management

- Stakeholders
- 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
- Summary

Chapter 5. 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
- Summary
Chapter 6. Solution Architecture Styles
- 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
- Summary

Chapter 7. Defining the Cloud
- 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
- Summary

Chapter 8. Quality of Service (QoS) Requirements
- Qualities of Service and Design
- Performance: Requirements
- Performance: Response Time Pattern
- Performance: Transaction Time Patterns
- Performance: Throughput Pattern
- Scalability
- 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

Chapter 9. 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
A longtime—focused training solutions that equip employees with the knowledge, skills, and expertise needed to succeed in their day-to-day work, to advance in their careers, and to build organizational capacity. For more information, contact us at (613) 727-7729 or visit www.algonquincollege.com/corporate

Chapter 10. Patterns
- 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 Concern
- 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
- Summary

Chapter 11. Architecture Requirement Techniques
- Requirements Management
- Requirements Management Activities
- Best Practices
- Baselining Requirements
- Desirable RM Repository Characteristics
- Example: Behavior Driven Development
- Why Traceability?
- Identifying Candidate Tactics, Patterns and Styles
- Requirements-Tactics-Patterns-Styles
- Making Architectural Decisions
- Architectural Measurement
- Implementing Architectural Measurement
- Example Metrics
- Summary

Chapter 12. Views and Viewpoints
- Views and Viewpoints
- Views
- 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
- Group Discussion: Viewpoints
- Summary
Chapter 13. Architecture Concepts
- Fundamental Architecture Concepts
- Abstraction
- Coupling
- Cohesion
- Decomposition & Modularization
- Encapsulation & Information Hiding
- Separation of Interface & Implementation
- Summary

Chapter 14. 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
- Summary

Chapter 15. 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™)
- Summary

Chapter 16. 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
- Summary
Chapter 17. 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?
- 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
- Summary

Chapter 18. 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
- Summary

Chapter 19. 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 Implication: Accept Design Influence  
COTS Implication: Plan for Stability  
COTS Implication: Sustain Competency  
COTS Implication: Vendor Lock-In  
COTS Implication: Balance Business Needs & Architecture  
COTS Inherent Risks  
COTS Risk Management Strategy #1  
COTS Risk Management Strategy #2  
COTS Risk Management Strategy #3  
Group Discussion: COTS  
Typical COTS Architecture  
Summary

Chapter 20. 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
ALGONQUIN COLLEGE CORPORATE TRAINING specializes in industry-focused training solutions that equip employees with the knowledge, skills and expertise needed to succeed in their day-to-day work, to advance in their careers and to build organizational capacity.

For more information, contact us at (613) 727-7729 or visit www.algonquincollege.com/corporate
Handout 6. Mobile Application Architecture
- The Model View Controller (MVC) Pattern
- Advantages of MVC
- The Model Layer
- Controller Responsibilities
- Controller in Android
- View Layer in Android
- Model Layer Issues in Mobile Applications
- Controller Layer Issues
- View Layer Issues
- Cross Cutting Issues
- Mobile Architecture and Pattern Resources
- Summary

Appendix A. Reference Library
- Overview
- Industry Organizations
- References & Further Reading

Appendix B. Architecture Glossary
- ABC
- DEF
- GHI
- MNO
- PQR
- STU
- VWXYZ

Appendix C. Pattern List

Appendix D. Solution Architecture Styles
- Web 2.0
- The WOA Difference
- Web 2.0 Technologies
- REST: Constraints
- REST: Uniform Interface
- REST Example

- Plain-Old-XML (POX) Example
- JavaScript Object Notation (JSON)
- Really Simple Syndication (RSS) & Atom
- Syndication Example
- Business Process Management
- Further Qualifying BPM
- BPM Architecture
- BPM Architecture Model
- BPM Standards
- Master Data Management (MDM)
- Understanding Master Data
- Qualities of Master Data
- MDM Implementation Styles
- MDM Example
- Summary

Appendix E. Business Process Modeling
- BPMN Event Symbols and Definitions
- Activities, Gateways, and Objects
- Flow Connectors

Appendix F. UML Reference
- UML Class
- UML Interface
- UML Associations
- UML Association Class
- UML Aggregation & Composition
- UML Generalization

Appendix G. Class Diagrams
- Package Diagram
- Package Diagram Illustrated
- Class Diagram
- Example: Annotated Class Diagram
- Example: Application Framework Class Diagram
- Example: Architectural Solution Diagram
- Class Diagram Notation
- Multiplicity
- Aggregation
- Composition
- Summary
Appendix H. Interaction Diagrams
- Subsystem Interaction Diagrams
- Pattern Interaction Diagrams
- Interaction Diagrams
- Sequence Diagram Illustrated
- Sequence Diagram
- Sequence Diagram Notation
- Annotated Sequence Diagram
- Message Notation
- Annotated Message Notation
- Example: Sequence Diagram
- Summary

Appendix I. Deployment Diagrams
- Deployment Diagram
- Notation: Nodes
- Stereotypes
- Associations
- Example: Deployment Diagram
- Summary

Appendix J. Develop Class Diagram

Appendix K. The Solution Architecture
Toolbox Summary
- What Goes in A Toolbox
- Practitioner Toolbox Items

Appendix L. Lessons Learned
- Process Recommendations
  Summary
- Structural Recommendations
  Summary
- Anti-Patterns
- Anti-Pattern: Bleeding Edge
- Anti-Pattern: Gold-plating
- Anti-Pattern: Strive for Perfection
- Anti-Pattern: Stuck in the Weeds
- Anti-Pattern: Technology Above All
- Group Discussion
- Summary

Appendix M. Application Architecture
- The 4+1 Views Approach
- Take a Multiple View Approach
- UML: Unified Modeling Language
- UML Notation Examples
- Use Case View
- Logical View
- Process Views
- Implementation View
- Deployment View
- Summary

Appendix N. Use Cases
- Use Cases
- Architecture Use Cases
- Use Case Diagram Notation
- Example: Use Case Diagram
- Template: Use Case
- Use Case Patterns: Team Composition
- Use an Iterative Approach
- When Is A Use Case Done?
- Use Case Patterns: Use Case Set
- The Single-minded Use Case
- Use Case Patterns: Scenarios & Steps
- Use Case Patterns: Relationships
- Summary

Lab Exercises
Lab 1. Stakeholder Identification
Lab 2. Identifying Requirements Anti-Patterns
Lab 3. Quality of Service (QoS) Requirements
Lab 4. Define Tactics for a Quality Attribute
Lab 5. Identifying Tactics for a Pattern
Lab 6. Requirements Management
Lab 7. Viewpoints and Views
Lab 8. Domain Architecture
Lab 9. Architecture Techniques
Lab 10. Extra: Identify Architecture Use Cases & Change Cases
Lab 11. Extra: Solution Domain Rubric
How to Register

Call (613) 727-7729
Email training@algonquincollege.com
Also available for Group Delivery.
Call for course dates or view the current schedule on-line at algonquincollege.com/corporate

Location: 100 Gloucester Street, Suite 300
Ottawa, ON K2P 0A4

Time: 9:00 a.m. to 5:00 p.m.

WEB AGE SOLUTIONS INC

This course is offered in collaboration with Web Age Solutions. Web Age Solutions provides cutting-edge training solutions for individuals in need of improving skills, including hands on experience working with the latest technologies in live project situations.

Course cancellations must be received in writing within ten (10) business days prior to the course start date for a full refund to be applied. Participant substitutions are acceptable.