

Introduction to Software Architecture

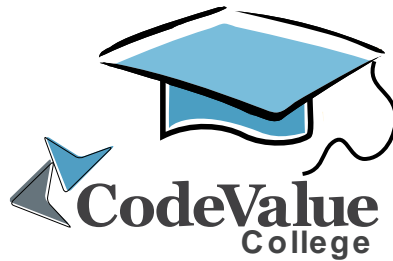
For the Windows & .NET Developer

Course Summary Table

Duration:	24 hours
Target Audience:	Experienced windows programmers and solution architects, interested in architecting and designing high quality enterprise level solutions targeting on premise and cloud on Microsoft Platforms
Objectives:	Understand the underlying architecture and design principles for developing high quality large scale line of business applications
Pre Requisites:	Deep knowledge of Windows OS development platform Practical experience developing windows application

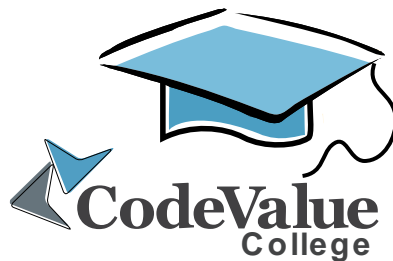
Abstract

Software architecture is about identifying and defining the building blocks - the software components, and the relationship between those components. Designing large scale, high quality software system is a mix of art and science. In this course the participant will learn the nature of the software product; How to capture and understand the functional and non-functional requirements, how to handle the constraints; how to analyze the problem domain and how to find the right solution that captures the current and future needs. This course provides the knowledge and tools that the modern architect must have in order to manage a nowadays project complexity. The course shows the software product life cycle and explain the role of the architect in each of these phases.

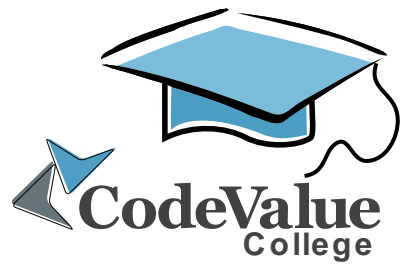


Syllabus

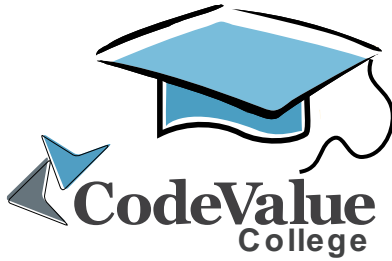
- ▼ Module 1: The Software Product and the Software Architecture
 - ▼ The System Structure
 - ▼ The Nature of the Software Product
 - ▼ Defining a “Good Software Product”
 - ▼ Software Architecture Goals
 - ▼ The Software Architecture Process
 - ▼ Key Architecture Principles
 - ▼ Testing & Validating the Architecture
 - ▼ The software architect's role
 - ▼ The Architect and the Staff
 - ▼ The Architecture and the Project Phases
- ▼ Module 2: Software Requirements
 - ▼ Functional, Non-Functional & Constraints
 - ▼ The SRS
 - ▼ Fishing for Requirements
 - ▼ UML based Use Cases
 - ▼ Requirements – The Agile Approach
 - ▼ User Stories
 - ▼ Design Sketches
 - ▼ Software Prototype
- ▼ Module 3: Software Architecture Analysis
 - ▼ Analysis, Architecture & Design
 - ▼ Analysis Principles
 - ▼ Defects in Requirements
 - ▼ Architecture Quality Attributes
 - ▼ Context and Significant Use Cases
 - ▼ Microsoft Patterns & Practices Analysis Process
 - ▼ Object Oriented Analysis
 - ▼ Agile Analysis
 - ▼ IDesign Method
 - ▼ Make your own Method
 - ▼ From Analysis to Design
 - ▼ Analysis Process Refinement
 - ▼ Risk and Issues Mitigations
 - ▼ Conducting POCs
 - ▼ Reviewing the Architecture
 - ▼ Applying The Architecture
- ▼ Module 4: Software Design
 - ▼ Key Design Rules
 - ▼ Design Concepts
 - ▼ Decomposition



- ▼ Cohesion
 - ▼ Type of Cohesion
- ▼ Coupling
 - ▼ Type of Coupling
 - ▼ The Price of Decoupling
 - ▼ Fan-In, Fan-Out and stable module
 - ▼ Dependency Injection
 - ▼ Inversion of Control
- ▼ Encapsulation
- ▼ The Two View of the System
 - ▼ The Static View
 - ▼ The Dynamic View
- ▼ Key Design Principles
 - ▼ Separation of Concerns (SoC)
 - ▼ Principle of Least Knowledge
 - ▼ Single Responsibility Principle (SRP)
 - ▼ Don't Repeat Yourself (DRY)
 - ▼ The Open/Close Principle (OCP)
 - ▼ Liskov Substitution Principle (LSP)
 - ▼ Design by Contract
 - ▼ Minimize Upfront Design
- ▼ Identifying Cross Cutting Concerns
- ▼ State Management
- ▼ Module 5: Architectural Patterns
 - ▼ Client/Server
 - ▼ Component-Based
 - ▼ Object Oriented
 - ▼ Domain Driven Design
 - ▼ Layered Architecture
 - ▼ N-Tiers
 - ▼ Message Bus
 - ▼ SOA
- ▼ Module 6: Handling Data
 - ▼ Data Representation
 - ▼ The Schema/Type Concept
 - ▼ Binary
 - ▼ Text, Mime 64
 - ▼ Key-Value
 - ▼ SQL
 - ▼ XML
 - ▼ JSon
 - ▼ Data Coherency & Consistency
 - ▼ Storing Data
 - ▼ File System



- ✔ The Development Team
- ✔ The Analysis Phase Outcome
- ✔ UI/UX Analysis
- ✔ The Offered Architecture
- ✔ The High Level Design
- ✔ Cross Cutting Concerns
- ✔ Packaging & Deployment
- ✔ Course Summary



Course Compatibility Questionnaire

Please answer the following questions as accurately as possible:

Name: _____ Email: _____
 Company: _____ Phone: _____

Language / Platform / Role	Years of Experience						Level of Familiarity				
C/C++	0-1	1-2	2-3	3-4	4-5	5+	1	2	3	4	5
C#	0-1	1-2	2-3	3-4	4-5	5+	1	2	3	4	5
Java	0-1	1-2	2-3	3-4	4-5	5+	1	2	3	4	5
Windows Server	0-1	1-2	2-3	3-4	4-5	5+	1	2	3	4	5
Windows Azure	0-1	1-2	2-3	3-4	4-5	5+	1	2	3	4	5
Amazon Web Services	0-1	1-2	2-3	3-4	4-5	5+	1	2	3	4	5
QA Engineer	0-1	1-2	2-3	3-4	4-5	5+					
Software Developer	0-1	1-2	2-3	3-4	4-5	5+					
Team Leader	0-1	1-2	2-3	3-4	4-5	5+					
Project Manager	0-1	1-2	2-3	3-4	4-5	5+					
Software Architect	0-1	1-2	2-3	3-4	4-5	5+					
Chief Architect	0-1	1-2	2-3	3-4	4-5	5+					

What is your current role?

What are your expectations from the course?

Thanks!

<http://college.codevalue.net/>