COURSE SYLLABUS

Microsoft .NET Solution Architectures
(MCP Exam 70-300)

50 Cragwood Rd, Suite 350
South Plainfield, NJ 07080

Victoria Commons, 613 Hope Rd Building #5,
Eatontown, NJ 07724

130 Clinton Rd,
Fairfield, NJ 07004

Avtech Institute of Technology Course

Instructor:
Course Duration: 40
Date/Time:
Training Location:

Course: Analyzing Requirements and Defining Microsoft .NET Solution Architectures (MCP Exam 70-300)

Text / Lab Books

Recommended:

The student kit includes a comprehensive workbook and other necessary materials for this class.
Microsoft Online Resources:
TechNet: Designed for IT professionals, this site includes How-tos, best practices, downloads, technical chats, and much more.
MSDN: The Microsoft Developer Network (MSDN) is a reference for developers, featuring code samples, technical articles, newsgroups, chats, and more.
Training & Certification Newsgroups: A newsgroup exists for every Microsoft certification. By participating in the ongoing dialogue, you take advantage of a unique opportunity to exchange ideas with and ask questions of others, including more than 750 Microsoft Most Valuable Professionals (MVPs) worldwide.

Course Description

This course is designed for experienced developers moving into a role that requires the skills to bridge business and technology environments. Experienced developers are those with the Microsoft Certified Application Developer (MCAD) credential, pursuing the Microsoft Certified Solution Developer (MCSD) credential.

The primary audience for this course is IT project professionals who are responsible for planning, building, and deploying Microsoft solutions. These IT project professionals include Microsoft customers and partners, and Microsoft Consulting Services.

The following models and activities will be explained in the classroom: , Introduction to the Microsoft Solutions Framework,, Introduction to Designing Business Solutions, Gathering and Analyzing Information, Envisioning the Solution, Creating the Conceptual Design, Analyzing Requirements, Creating the Logical Design, Identifying Objects for the Logical Design, Creating the Physical Design, Working on the Physical Design, Designing the Presentation Layer, Creating the User Interface, Designing the Data Layer, Creating a Data Schema, Designing Security Specifications, Threat Modeling and Mitigation, Completing the Planning Phase, Reviewing a Test Plan and Technical Specification, Stabilizing and Deploying the Solution,

Prioritizing Bugs, Building an MSF Team, Managing Project Risks, Drafting a Vision/Scope Document, Associating Work Items with Plans, Developing Your Solution, Stabilizing Your Solution, and Deploying Your Solution.

Learning Objectives

1. Envisioning the Solution

1.1. Introduction to Designing Business Solutions
- 1.1.1. Overview of Microsoft Solutions Framework
- 1.1.2. Phases in the MSF Process Model
- 1.1.3. Introducing the Case Study—Adventure Works Cycles Application

1.2. The Envisioning Phase
- 1.2.1. Creating a Vision/Scope Document
- 1.2.2. Creating the Project Structure Document
- 1.2.3. Analyzing Risks

1.3. Analyze the feasibility of the solution
- 1.3.1. Analyze the business feasibility of the solution
- 1.3.2. Analyze the technical feasibility of the solution
- 1.3.3. Analyze available organizational skills and resources

1.4. Analyze and refine the scope of the solution project
- 1.4.1. What's in a Name?: Improved Tablespace Management
- 1.4.2. gets a boost thanks to a sparser SYSTEM
- 1.4.3. support for defining a default tablespace for users
- 1.4.4. new SYSAUX, and even renaming
- 1.4.5. Default Tablespace for Nonessential Schemas

1.5. Identify key project risks

2. Gathering and Analyzing Business Requirements

2.1. Gather and analyze business requirements
- 2.1.1. Preparing for an Interview
  - 2.1.1.1.Deriving Use Case Statements for the Sales Automation
  - 2.1.1.2.Developing Draft Requirements from Initial Information
  - 2.1.1.3.Developing a Usage Scenario
- 2.1.2. Analyze business processes
- 2.1.3. Analyze the organizational structure, both current and projected
- 2.1.4. Analyze vertical market position and industry position
- 2.1.5. Analyze personnel and training needs
- 2.1.6. Analyze the organizational political climate
- 2.1.7. analyze business reach or scope
- 2.1.8. Analyze current and future regulatory requirement

2.2. Analyze business requirements for the solution
- 2.2.1. Identify business requirements
- 2.2.2. Identify dependencies, both inside and outside
- 2.2.3. Identify features of the solution
- 2.2.4. Define design goals, such as extensibility requirements

- 2.2.5. Define data requirements, types, and flows
- 2.2.6. Create data flow diagrams

2.3. Gather and analyze user requirements
- 2.3.1. Identify use cases
- 2.3.2. Identify usage scenarios for each use case
- 2.3.3. Identify globalization requirements
- 2.3.4. Identify localization requirements
- 2.3.5. Identify accessibility requirements

2.4. Gather and analyze operational requirements
- 2.4.1. Identify maintainability requirements
- 2.4.2. Identify scalability requirements
- 2.4.3. Identify availability requirements
- 2.4.4. Identify reliability requirements
- 2.4.5. Identify deployment requirements
- 2.4.6. Identify security requirements

2.5. Gather and analyze requirements for hardware, software, and network infrastructure
- 2.5.1. Identify integration requirements
- 2.5.2. Analyze the IT environment, including current and projected applications, and current and projected hardware, software, and network infrastructure
- 2.5.3. Analyze the impact of the solution on the IT environment

3. **The Design Models**

3.1. Creating the Conceptual Design
- 3.1.1. An Introduction to the Planning Phase
- 3.1.2. An Overview of the Functional Specification
- 3.1.3. An Overview of the Conceptual Design Process
- 3.1.4. Building the Conceptual Design
- 3.1.5. Optimizing the Conceptual Design

3.2. Creating the Logical Design
- 3.2.1. An Overview of Logical Design
- 3.2.2. Creating a Logical Design
- 3.2.3. Documenting Logical Design Output
- 3.2.4. Optimizing Logical Design

3.3. Creating the Physical Design
- 3.3.1. An Overview of Physical Design
- 3.3.2. Physical Design Analysis
- 3.3.3. Physical Design Rationalization
- 3.3.4. Physical Design Implementation

3.4. Designing the Data Layer
- 3.4.1. Design the Data Store
- 3.4.2. Optimizing Data Access
- 3.4.3. Implementing Data Validation

3.5. Designing Security Specification
- 3.5.1. Overview of Security Application Development
- 3.5.2. Planning for Application Security
- 3.5.3. Using the .NET Framework Security Features

3.5.4. Designing Authorization, Authentication, and Auditing Strategies
3.6. Completing the Planning Phase
3.6.1. Incorporating Design Considerations
3.6.2. Planning for Administrative Features
3.6.3. Planning for Future Phases
3.6.4. Creating the Technical Specifications
3.7. Stabilizing and Deploying the Solution
3.7.1. The MSF Stabilizing Phase
3.7.2. Testing and Piloting for Stabilization
3.7.3. The MSF Deploying Phase
3.7.4. Deploying to a Production Environment

4. Developing Specifications

- Transform requirements into functional specifications. Considerations include performance, maintainability, extensibility, scalability, availability, deploy-ability, security, and accessibility
- Transform functional specifications into technical specifications. Considerations include performance, maintainability, extensibility, scalability, availability, deploy ability, security, and accessibility

4.1. Select a development strategy
4.1.1. Select strategies for auditing and logging
4.1.2. Select strategies for error handling
4.1.3. Select strategies for integration
4.1.4. Select strategies for globalization
4.1.5. Select strategies for localization
4.1.6. Select strategies for data storage
4.1.7. Select strategies for data storage
4.1.8. Select strategies for state management
4.1.9. Include constraints in the development plan to support business rules. Constraints includes data validation
4.2. Select a deployment strategy
4.2.1. Select strategies for deployment
4.2.2. Select strategies for licensing
4.2.3. Select strategies for data migration
4.3. Select a security strategy
4.3.1. Select strategies to help ensure data privacy, such as encryption, signing, and sealing
4.3.2. Select strategies to help ensure secure access
4.4. Select an operations strategy
4.4.1. Select strategies for data archiving and data purging
4.4.2. Select strategies for upgrades
4.4.3. Create a support plan
4.5. Create a test plan
4.6. Create a user education plan

5. Creating the Conceptual Design

5.1. Create a conceptual model of business requirements or data requirements. Methods include Object Role Modeling (ORM)

5.1.1. Transform external information into elements
5.1.2. Apply a population check to fact types
5.1.3. Identify primitive entity types in the conceptual model
5.1.4. Apply uniqueness constraints to the conceptual model
5.1.5. Apply mandatory role constraints to the conceptual model
5.1.6. Add value constraints, set-comparison constraints, and subtype constraints to the conceptual model
5.1.7. Add ring constraints to the conceptual model

5.2. Validate the conceptual design

6. Creating the Logical Design

6.1. Create the logical design for the solution

6.1.1. Create the logical design for auditing and logging
6.1.2. Create the logical design for error handling
6.1.2.1. Create the logical design for exception handling
6.1.3. Create the logical design for integration
6.1.4. Create the logical design for globalization
6.1.5. Create the logical design for localization
6.1.6. Create the logical design for security
6.1.6.1. Create the logical design for data privacy, Option include encryption, signing, and sealing
6.1.7. Include constraints in the logical design to support business rules
6.1.8. Create the logical design for the presentation layer, including the user interface (UI)
6.1.9. Create the logical design for services and components
6.1.10. Create the logical design for state management
6.1.11. Create the logical design for synchronous or asynchronous architecture

6.2. Create the logical data model

6.2.1. Define tables and columns
6.2.2. Normalize tables
6.2.3. Define relationships
6.2.4. Define primary and foreign keys
6.2.5. Define the XML schema

6.3. Validate the proposed logical design

6.3.1. Review the effectiveness of the proposed logical design in meeting business requirements. Business requirements, include performance, maintainability extensibility, scalability, availability, deploy ability, security, and accessibility
6.3.2. Validate use cases, scenario walk-throughs, and sequence diagrams
6.3.3. Create a proof of concept for proposed physical design

7. Creating Standards and Processes

7.1. Establish standards, Standards can apply to development documentation, coding, code review, UI, and testing

7.2. Establish processes. Process include reviewing development documentation, reviewing code, creating builds, tracking issues, managing source code, managing change, managing release, and establishing maintenance tasks
7.3. Methods include Microsoft Visual Studio .Net Enterprise Templates
7.4. Establish quality and performance metrics to evaluate project control, organizational performance, and return on investment

Prerequisite

Candidates for this exam should have a minimum of two years of experience in the following areas:

- Analyzing customer needs and creating documents that specify requirements for software solutions in multiple business domains.
- Modeling processes, modeling data, designing components, and designing user interfaces.
- Designing, developing, and implementing software solutions.
- Integrating new applications into legacy environments.
- Developing Microsoft Windows-based applications and Web applications by using Microsoft tools and technologies

Contact Hours

_____ Contact Hours (Lecture ___ Hours / Lab ____ Hours)

Semester Credit Hours

_____ semester credit hours

Teaching Strategies

A variety of teaching strategies may be utilized in this course, including but not limited to, lecture, discussion, written classroom exercises, written lab exercises, performance based lab exercises, demonstrations, quizzes and examinations. Some quizzes may be entirely or contain lab based components. A mid-course and end course examination will be given.

Method of Evaluating Students

Grade Distribution

Class Attendance	10
Mid Term	30
Finals	50
Special Projects Makeup projects	10
Total	**100%**

Grading Policy

At the end of each course, each student is assigned a final grade as follows:

Point Range	Interpretation	Grade	Quality Points
90 – 100	Excellent	A	4.0
80 – 89	Very Good	B	3.0 – 3.9
70 – 79	Average	C	2.0 – 2.9
60 – 69	Poor	D	1.0 – 1.9
Below 60	Failure	F	0
N/A	Withdrawal	W	0
N/A	Pass	P	0
N/A	Incomplete	I	0

A student earning a grade of D or above is considered to have passed the course and is eligible to pursue further studies. A student receiving a grade of F has failed the course. A failed course must be repeated and passed to meet Avtech Institute's graduation requirements, in addition to an overall program GPA of 2.0.

Requirements for Successful Completion of the Course

At a minimum, students must achieve the following:

- A passing grade of **D** or above
- Completion of all required examinations
- Submission of all required lab exercises and projects and;
- Adherence to the school attendance policy.

Equipment Needed

Industry standard desktop computer for lab exercises.

Equipment Breakdown Lab room

Videos and Projector

Library Assignments

To be determined by the instructor.

Portfolio Assignment

Student program outcome portfolios are required to demonstrate student competencies. In conjunction with your course structure, please select a project/paper that best demonstrates what you have learned in this course and add it to your program portfolio.

Course Policies

Disruptive Behavior

Disruptive behavior is an activity that interferes with learning and teaching. Inappropriate talking during class, surfing inappropriate website, tardiness, cheating, alcohol or drug use, use of cell phone, playing lout music during class, etc. all disrupt the learning process.

Copyright Infringement

Specific exemptions to copyright infringement are made for student use in the context of learning activities. Graphic design students often download images from the Internet, or scan images from publications. As long as this work is for educational purpose, and subject to faculty permission, this is not a problem.

Plagiarism

Faculty cannot tolerate the ***misrepresentation of work as the student's own***. This often involves the use by one student or another student's design, whether voluntarily or involuntarily. In the event that plagiarism is evident and documented, all students involved in the conscious decision to misrepresent work must receive an F as the grade for the project. A second occurrence may result in suspension for the rest of the quarter, and return to the school only after a review by the Academic Standards Committee.

Attendance

Attendance and Lateness

In education and the workplace, regular attendance is necessary if individuals are to excel. There is a direct correlation between attendance and academic success. Attendance is mandatory. All students must arrive on time and prepared to learn at each class session. At the faculty member's discretion, students may be marked absent if they arrive more than 15 minutes late to any class. More that five absences in a class that meets twice per week or more that two absences in a class that meets once per week may result in a failure.

Make-Up Work

Late Projects and Homework

All projects and homework must be handed in on time. Homework should be emailed to your instructor if you are going to miss a class. Work that is submitted one week late will result in the loss of one full grade; and work that is submitted two weeks late will result in the loss of two full grades; more than two weeks late you will receive a failing grade on the project.