

CS490

Software Development Senior Project

[Onsite]

Course Description:

This course provides a comprehensive case study for the student to research, design, develop, test and deploy a software application to solve a specific business problem. The student is required to dynamically apply knowledge and skills acquired through the program of study and to complete the project to meet the technical requirements specified by the course. Teamwork, project management and presentation skills are also integrated as part of the course requirement.

Prerequisite(s) and/or Corequisite(s):

Prerequisites: Completion of a minimum of 164 credits earned in the program of study including CS400 Web Services and Applications or equivalent and CS410 Enterprise Applications with .NET or equivalent

Credit hours: 4

Contact hours: 50 (30 Theory Hours, 20 Lab Hours)

Syllabus: Software Development Capstone Project

Instructor: _____

Office hours: _____

Class hours: _____

Major Instructional Areas

1. Project management
2. Software Development Lifecycle
3. Logical and physical system design
4. Business object design, implementation, and deployment
5. Database design, implementation, and deployment
6. Client application design, implementation, and deployment
7. System testing and deployment
8. Team dynamics
9. Documentation
10. Presentation skills

Course Objectives

1. Apply project management concepts and techniques to a software development project.
2. Use Microsoft Office Project to help plan and manage a software development project.
3. Determine the requirements for an n-tier software development project.
4. Apply the knowledge acquired in the program to design an effective software solution for a given problem.

5. Apply the knowledge acquired in the program to implement an effective software solution for a given problem.
6. Apply the knowledge acquired in the program to test the system.
7. Work in teams on a medium-scope project.
8. Create the necessary documentation for the software solution.
9. Deliver a professional presentation on the design and functionality of the software solution.
10. Complete a comprehensive skills assessment for the program of study.

SCANS Objectives

SCANS is an acronym for Secretary's Commission on Achieving Necessary Skills. The committee, created by the National Secretary of Labor in the early 1990s, created a list of skills and competencies that the committee feels are necessary for employees to function in a high-tech job market.

1. Maintain information to maximize retention and expression of knowledge.
2. Work effectively as part of a team and contribute ideas, suggestions, and effort to the group.
3. Identify how technological systems work and operate effectively.
4. Demonstrate competence in understanding systems.
5. Use interpersonal skills and communication techniques.
6. Demonstrate competence in selecting a technology, including determining the desired outcomes and applicable constraints.
7. Apply decision-making skills to specify goals and constraints.
8. Identify the need for data, obtain it from existing sources or create it, and evaluate its relevance and accuracy.
9. Demonstrate competence in monitoring and correcting performance.
10. Demonstrate competence in configuring, installing, and integrating various hardware and software systems.

Course Outline

Unit	Activities
1– Introduction to the Senior Project	<ul style="list-style-type: none"> • Review project management terminology and apply the terminology to the senior project. • Discuss the senior project or Request for Proposal (RFP) to identify the following: <ul style="list-style-type: none"> ○ Objectives and scope ○ Requirements and specifications • Set up teams and define the roles and responsibilities of each team member. • Identify functional and non-functional requirements. • Submit next unit: Project Documentation Part I: Project scope, Requirements document. • Submit next unit: Project Plan: Introduction, project organization
2– System Design and Platform Selection	<ul style="list-style-type: none"> • Identify the subsystems in the system. • Write an abstract description of each subsystem. • Define the interfaces between subsystems. • Select the development platform for each subsystem. • Create a list of acceptance testing criteria. • Perform a project risk analysis. • Identify hardware and software requirements. • Decide team member responsibilities for each subsystem. • Submit next unit: Project Documentation Part II: System architecture design. • Submit next unit: Project Plan: Risk analysis, hardware and software requirements.
3– Design	<ul style="list-style-type: none"> • Perform object-oriented decomposition for each subsystem. • Define the interface for each class.

Unit	Activities
	<ul style="list-style-type: none"> • Create the data model for the database. • Research the PayPal API. • Tour Project 2007. • Create a Work Breakdown Structure (WBS). • Create a project schedule. • Create a monitoring and remediation strategy. • Submit next unit: Project Documentation Part III: Object design, Data design • Submit next unit: Project Plan: WBS, project schedule, monitoring and remediation strategy
4– Prototypes and Stubs	<ul style="list-style-type: none"> • Check the project milestones. Check the individual work and teamwork status. • Assess the risks to project completion and make adjustments as necessary. • Implement a prototype for each user interface subsystem. • Implement a stub for each public business, data, and utility class. • Create documentation for each public class that explains how it should be used. • Hand off the stubs to the other team members. • Submit in the next unit: Project Documentation Part IV: Class documentation. • Submit next unit: Project Plan: Finalized project plan, project status report. • Submit next unit: Implementation: A stub for each class, a prototype for each user interface subsystem.
5– Subsystem	<ul style="list-style-type: none"> • Check the project milestones. Check the individual work and teamwork status.

Unit	Activities
Development, Part 1	<ul style="list-style-type: none"> • Assess the risks to project completion and make adjustments as necessary. • Design unit tests for each component. • Present the prototypes to the other team members. • Design data input validation rules for the presentation subsystems. • Design unit tests for the presentation subsystems. • Create the database schema. • Populate the database with test data. • Begin implementation of components. • Begin refinement of the user interface subsystems. • Individual skills assessment. (This element may start earlier or later, depending on class needs. The activity will last until every individual student's interview is completed.) • Submit next unit: Project Documentation Part V: Unit tests, input validation rules. • Submit next unit: Project Plan: Project status report. • Implementation: Database populated with some test data.
6- Subsystem Development, Part 2	<ul style="list-style-type: none"> • Check the project milestones. Check the individual work and teamwork status. • Assess the risks to project completion and make adjustments as necessary. • Design integration tests for each subsystem. • Create a bug-tracking mechanism for integration tests. • Continue implementation of each subsystem. • Submit next unit: Project Documentation Part VI: Integration test plan.

Unit	Activities
	<ul style="list-style-type: none"> • Submit next unit: Project Plan: Project status report.
<p>7– Subsystem Integration</p>	<ul style="list-style-type: none"> • Check the project milestones. Check the individual work and teamwork status. • Assess the risks to project completion and make adjustments as necessary. • Complete implementation of each subsystem component. • Complete testing of each subsystem component. • Integrate the components of each subsystem. • Perform integration testing of each subsystem. • Identify and categorize bugs. • Refine documentation for interfacing with the subsystem. • Create user documentation. • Submit next unit: Project Documentation Part VII: Subsystem interface documentation, bug reports, user documentation. • Submit next unit: Project Plan: Project status report.
<p>8– System Integration and Deployment</p>	<ul style="list-style-type: none"> • Check the project milestones. Check the individual work and teamwork status. • Assess the risks to project completion and make adjustments as necessary. • Create a deployment package for each subsystem. • Perform regression testing on any subsystem bugs. • Create a test plan for the system. • Integrate the system and test. • Identify, categorize, and assign responsibility for correcting bugs. • Submit next unit: Project Documentation Part VIII: System test

Unit	Activities
	plan, bug reports <ul style="list-style-type: none"> • Submit next unit: Project Plan: Project status report
9– Complete System Integration and Documentation	<ul style="list-style-type: none"> • Check the project milestones. Check the individual work and teamwork status. • Assess the risks to project completion and make adjustments as necessary. • Perform regression testing on any remaining bugs. • Correct any remaining problems. • Create the presentation. • Submit next unit: Project Documentation Part IX: Regression test results, revised risk assessment, presentation outline. • Submit next unit: Project Plan: Project status report.
10– Project Closure	<ul style="list-style-type: none"> • Perform acceptance testing of the system. • Perform project closure. • Perform a practice presentation of the solution. • Submit next unit: Final project documentation • Submit next unit: Deployment packages and instructions for all subsystems
11– Final Professional Presentation	<ul style="list-style-type: none"> • Perform a peer review and evaluation of each team. • Make the final presentation.

Instructional Methods

The activities in this course are designed to simulate a real-world software development project. In the first class, you will be assigned to a project team and be presented with a case study for a real project. Throughout the course, you will perform various steps in the software development lifecycle, including designing, implementing, testing, documenting, and deploying the system. You will also perform project management tasks. You will prepare and deliver a presentation

that would be used when presenting a solution to a customer for acceptance. Your project will be evaluated on documentation, teamwork, presentation, and functionality of the software system.

Your instructor—or instructors—for this course will perform the roles of an advisor, a consultant, and a supervisor. Students in each group will perform the role of a professional consulting firm serving customers in a real-life situation. Meetings will be conducted regularly so that the instructor can provide student teams with consultation and guidance on the project and discuss various implementation and problem-solving strategies.

Complete **project documentation** in accordance with the project management guidelines is required. Grades will be awarded based on the quality and coverage of all the areas and items listed in the Documentation Requirements section.

For **team presentation**, grades will be awarded based on the quality in the following areas:

- Conformation to formalities—stated rules
 - PowerPoint presentation demonstrating final project solution
 - Hand-outs for audience
 - Professional business attire
- Clarity of explanation of solution functionality and design methodology
- End-user product support training plan

- For the **teamwork** portion, you will receive a Teamwork Evaluation Form at the beginning of the course. Your completed form will be collected at the end of the course. Areas for evaluation will be:
 - Participation
 - Team organizational contributions
 - Interpersonal communication performance
 - Subject area expertise contributions

For the **software system**, grades will be awarded based on the quality in the following areas:

- **Functionality:** whether or not the system meets the functional requirements
- **Usability:** whether the system includes an intuitive user interface and clear user documentation
- **Maintainability:** whether the design and code are sufficiently documented to allow for modification due to changes in the business or environment
- **Deployment:** whether a customer would be able to follow the directions provided to deploy, or make ready to use, the solution

To ensure that you acquire curriculum-related knowledge, your instructor will give you **individual skills assessment** questions at the beginning of the course. During the course, the instructor will schedule an interview in which you will be asked to answer these questions orally. The purpose of this interview is for you to review knowledge and skills learned in the entire program and help you to prepare for job interviews and career success.

Instructional Materials and References

Student Textbook Package

None

All textbooks for the core Software Development courses will be used directly or indirectly as references for this course.

References

ITT Tech Virtual Library

Log on to the ITT Tech Virtual Library at <http://library.itt-tech.edu/> to access online books, journals, and other reference resources selected to support ITT Tech curricula.

Books

You may click “Books” or use the Library Catalog on the home page to find the following books.

Books> Books24x7

- Andersson, Eve, Philip Greenspun, and Andrew Grumet. *Software Engineering for Internet Applications*. Cambridge, MA: The MIT Press, 2006.
- Biafore, Bonnie. *Visio 2007 Bible*. Indianapolis: John Wiley & Sons, 2007.
- Chatfield, Carl S., and Timothy D. Johnson. *Microsoft Office Project 2007 Step by Step*. Redmond, WA: Microsoft Press, 2007.
- Harris, Paul Eastwood. *Planning and Scheduling Using Microsoft Office 2007: Including Microsoft Project 2000 to 2003*. Victoria, Australia: Eastwood Harris Pty Ltd., 2007.
- Jalote, Pankaj. *An Integrated Approach to Software Engineering*. 3rd ed. New York: Springer Publishing, 2005.
- Jones, T. Capers. *Estimating Software Costs: Realism to Estimating*. 2nd ed. New York: McGraw-Hill, 2007.
- Marmel, Elaine. *In the Trenches with Microsoft Office Project 2007*. Redmond, WA: Microsoft Press, 2009.
- McManus, John. *Risk Management in Software Development Projects*. St. Louis: Butterworth-Heinemann, 2004.
- Munson, John C. *Software Engineering Measurement*. Boca Raton, FL: Auerbach Publications, 2003.
- Page, Alan, Ken Johnston, and B.J. Rollison. *How We Test Software at Microsoft*. Redmond, WA: Microsoft Press, 2009.
- Peckham, Joan, and Scott J. Lloyd. *Practicing Software Engineering in the 21st Century*. Hershey, PA: IGI Publishing, 2003.
- Peters, Lawrence J. *Getting Results from Software Development Teams*. Redmond, WA: Microsoft Press, 2008.
- Stover, Teresa S., and Bonnie Biafore. *Microsoft Office Project 2007 Inside Out*. Redmond, WA: Microsoft Press, 2007.
- Stover, Teresa S. *Microsoft Project 2003 Inside Out*. Redmond, WA: Microsoft Press, 2004.
- Thayer, Richard H., and Mark J. Christensen. *Software Engineering, Volume 1: The Development Process*. 3rd ed. Indianapolis: John Wiley & Sons, 2005.

- Tomayko, James E., and Orit Hazzan. *Human Aspects of Software Engineering*. Charles River Media, 2004.
- van Vliet, Hans. *Software Engineering: Principles and Practice*. 3rd ed. John Wiley & Sons, 2008.

Books> NetLibrary

- Lewis, James P. *Fundamentals of Project Management*. WorkSmart Series. 3rd ed. New York: AMACOM Books, 2007.
- Westland, Jason. *The Project Management Life Cycle: A Complete Step-by-Step Methodology for Initiating, Planning, Executing & Closing a Project Successfully*. Philadelphia: Kogan Page, 2006.

Reference Resources

You may click “Reference” or use the E-Journal Lookup on the home page to find the following reference resources.

Reference> Additional reference resources: Project Management

- allIPM
Articles, forums, white papers, jobs, and other project management-related content
- gantthead.com
Articles, news, forums, and information about the field of project management
- Microsoft Project
Tour, list of features, and FAQs about Microsoft Project
- Project Management Institute
The Project Management Institute (PMI) is a not-for-profit project management professional association.

Other References

The following resources can be found **outside** of the ITT Tech Virtual Library.

Web sites

- CIO.com - Business Technology Leadership
<http://www.cio.com> (accessed Jan. 12, 2010)
An online magazine for IT leadership
- ITtoolbox Project Management Knowledge Base
<http://projectmanagement.ittoolbox.com/> (accessed Jan. 12, 2010)
An online community for peers to share knowledge about information technology
- Microsoft Developer Network

<http://msdn.microsoft.com> (accessed Jan. 12, 2010)

A library of programmer documentation for Visual Studio, SQL Server, and other Microsoft platforms
- Microsoft Office Project

<http://www.microsoft.com/project/en/us/default.aspx>
(accessed Jan. 12, 2010)

Product documentation and training presentations for Microsoft Project
- Microsoft Office Visio

<http://office.microsoft.com/en-us/visio/FX100649211033.aspx?CTT=96&Origin=CL100636311033> (accessed Jan. 12, 2010)
Product information, help, training, and templates for various versions of Visio
- PayPal Developer Network

<https://www.x.com/index.jspa> (accessed Jan. 12, 2010)

API documentation and libraries for interfacing with PayPal
- Sun Java Developer Network

<http://java.sun.com> (accessed Jan. 12, 2010)

A library of programmer documentation, downloads, and training for Java
- The Value of Project Lifecycle Methodology
<http://www.lifecyclestep.com/open/401.0HomeValue.htm> (accessed Jan. 12, 2010)

Article on a site specializing in developing, consulting, and training in business methodologies

- Wideman Comparative Glossary of Project Management Terms
<http://www.maxwideman.com/pmglossary/> (accessed Jan. 12, 2010)
Definitions for commonly used terms in project management

All links to Web references outside of the ITT Tech Virtual Library are always subject to change without prior notice.

Course Evaluation and Grading

Evaluation Criteria

The final grades will be based on the following categories:

CATEGORY	WEIGHT
Team Project	
• Project Management Documentation	20%
• Teamwork	10%
• Presentation	10%
• System Design	20%
• System Solution	25%
Individual Skills Assessment	15%
Total	100%

Grade Conversion Table

The final grades will be calculated from the percentages earned in the course, as follows:

A	90-100%	4.0
B+	85-89%	3.5
B	80-84%	3.0
C+	75-79%	2.5
C	70-74%	2.0
D+	65-69%	1.5
D	60-64%	1.0
F	<60%	0.0