

ITT Technical Institute

IT308P

Software Development Capstone Project

Onsite Course

SYLLABUS

Credit hours: 4

Contact/Instructional hours: 66 (46 Theory Hours, 20 Lab Hours)

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

Prerequisites: Completion of a minimum of 80 credits earned in the program of study including IT306P Software Application Programming or equivalent

Course Description:

Development of a complex software application in an area jointly agreed upon by the student as well as the faculty member. The faculty member acts more as a facilitator and project manager for this final assignment.

Syllabus: Software Development Capstone Project

Instructor: _____

Office hours: _____

Class hours: _____

Major Instructional Areas

1. Major components of project management and teamwork
2. Analysis of and response to a Request for Proposal (RFP) or a business case
3. Analyses of technologies available and selection of the right technology for solving a problem
4. Project design, development, and implementation
5. Technical or business presentation of a solution

Course Objectives

1. Relate and apply important concepts of project management to the actual capstone project proposed for this course.
2. Use Microsoft Office Project or similar project management tools to help manage the actual capstone project.
3. Analyze the requirements for the capstone project.
4. Incorporate major software engineering concepts and methods throughout the design and development process of the proposed project to produce a software application using a chosen programming language.
5. Apply standard practice in the software development lifecycle to ensure functionality and usability based on original requirements.
6. Develop the complete documentation for a software system.
7. Perform user testing and functional testing for the software system by using different testing methods.
8. Present a developed software system professionally.
9. 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. Demonstrate competence in understanding how technological systems work and operate effectively within them.
4. Utilize interpersonal skills and communication techniques.
5. Demonstrate competence in selecting a technology, including determining the desired outcomes and applicable constraints.
6. Apply decision-making skills to specify goals and constraints.
7. Identify the need for data and evaluate its relevance and accuracy.
8. Demonstrate competence in monitoring and correcting system performance.

Course Outline

Unit	Activities
1— Capstone Project: Overview	<ul style="list-style-type: none"> • Read from <i>Essentials: Microsoft Office Project 2003</i>: Project 1, "Taking a Tour of Project 2003," pp. 2-34 • Project management concepts are introduced through the introduction of the capstone project. • Analyze the proposed capstone project to identify the following components: <ul style="list-style-type: none"> Objectives and scope <ul style="list-style-type: none"> ○ Needs ○ Requirements ○ Specifications ○ Resources required to deliver the product, including human resources, time, funds, space, and tools ○ Deliverables ○ Milestones ○ Tasks • Set up teams and assign roles to each team member. Use Microsoft Project 2003 or a similar project management tool to configure the preceding information into a project plan. • Identify resources, including time, human resources, and cost, and consider how to integrate each team member's responsibility with the overall needs of the project. • Understand and start preparing for the individual skills assessment. <p><u>Deliverable:</u></p> <ul style="list-style-type: none"> • Submit the team meeting agenda.
2— Project Planning	<ul style="list-style-type: none"> • Read from <i>Essentials: Microsoft Office Project 2003</i>: Project 2, "Specifying Overall Project Settings," pp. 37-59 Project 3, "Entering Tasks and Creating a Project Schedule," pp. 61-97 • Learn and gather technical skills for the development of the project. • Begin working on the capstone project software application. • Identify the major milestones and individual tasks. • Determine the programming language to use for the software

Unit	Activities
	<p>application.</p> <ul style="list-style-type: none"> • Deliver the preliminary project plan, such as phases, milestones, tasks, timelines, and major deliverables, for review. <p><u>Deliverables:</u></p> <ul style="list-style-type: none"> • Submit the team project sheet identifying the following: <ul style="list-style-type: none"> ○ Team member names ○ Team member responsibilities ○ Team project identification ○ Programming language ○ Preliminary project plan
3— Application Development	<ul style="list-style-type: none"> • Read from <i>Essentials: Microsoft Office Project 2003</i>: Project 4, “Scheduling Resources and Assigning Costs,” pp. 99-120 Project 5, “Modifying Task Information,” pp. 135-158 Project 6, “Modifying Tasks Using the Gantt Chart,” pp. 171-191 • Use Microsoft Office Project or a similar project management tool to produce a Gantt chart identifying how project resources are allocated. • Produce the requirements definition document identifying the system specifications, such as input, output, storage, and control functions. • Deliver a preliminary data flow diagram of the software application. Create the diagram using Microsoft Visio or other graphical application. • Continue working on the software application coding. <p><u>Deliverables:</u></p> <ul style="list-style-type: none"> • Submit the following: <ul style="list-style-type: none"> ○ A Gantt chart generated by MS Project or similar tool ○ Requirements definition document ○ Data flow diagram
4— Application Development (continued)	<ul style="list-style-type: none"> • Continue working on the software application coding. • Begin the development of the appendices to be included in the project documentation. Present the following graphically in the project appendix: <ul style="list-style-type: none"> ○ Database relationships ○ Context diagrams ○ Data flow diagrams (DFDs) <p><u>Deliverables:</u></p> <ul style="list-style-type: none"> • Submit the following: <ul style="list-style-type: none"> ○ Database relationship diagram ○ Context diagram ○ Data flow diagrams or other system design tool
5— Application Development (continued)	<ul style="list-style-type: none"> • Continue working on the software application coding.
6— Application Development (continued)	<ul style="list-style-type: none"> • Perform a draft application overview, identifying the following aspects: <ul style="list-style-type: none"> ○ Functionality: What data should be entered into the system? ○ Consistency: Will the screens or pages be similar in interface look and feel? ○ Performance: Is the system performance maximized with regard to data structure?

Unit	Activities
	<ul style="list-style-type: none"> ○ User-friendliness: Is the application design intuitive and easy to follow and understand? ● Develop the software application test plan. <p><u>Deliverable:</u></p> <ul style="list-style-type: none"> ● Submit the test plan.
7— Application Development (continued)	<ul style="list-style-type: none"> ● Continue coding on the capstone project software application. ● Discuss system comments with peers—the comments should include version numbers or dates to track changes and aid in development effort. ● Answer the individual skills assessment questions.
8— Application Development (continued)	<ul style="list-style-type: none"> ● Continue working on the software application coding. ● Begin developing the table of contents of the documentation. ● Consult the team on design phase uniformity to ensure uniformity in the following components: <ul style="list-style-type: none"> ○ Coding guidelines ○ Common or global module ○ System tracking—comments
9— Application Development (continued)	<ul style="list-style-type: none"> ● Continue working on the software application coding. ● Perform the system testing plan. ● Check the statuses of the project, individual work, and team work. <p><u>Deliverable:</u></p> <ul style="list-style-type: none"> ● Submit test plan results.
10— Mock Presentation	<ul style="list-style-type: none"> ● Finalize the software application coding. ● Complete system testing documentation. ● Review and check on all deliverables and presentation materials. ● Perform a mock presentation.
11— Software System Presentation	<ul style="list-style-type: none"> ● Present the capstone project. ● Demonstrate the software application. ● Discuss the software application development based on each team's experience. ● Submit Team Member Evaluation Sheet. <p><u>Deliverables:</u></p> <ul style="list-style-type: none"> ● Project presentation ● Software application on a CD, DVD, or USB drive ● Project documentation

Instructional Methods

Weekly classes will discuss and apply theories and practices of software application programming. Weekly laboratory time will be devoted to team activities designed to foster leadership and teamwork while you develop a fully functional software application system and present it to a panel of observers.

Part of the first four units will cover project selection and project management techniques using Microsoft Office Project or other open source project management tools to help manage your project for this course. To ensure that you have acquired the necessary knowledge to graduate from the program, an **individual skills assessment** questionnaire will be distributed at the beginning of the course. Your instructor will use this assessment to conduct a panel interview with you between Unit 5 and Unit 7, depending on your class size. This process provides a structured opportunity for you to review selected

topics covered in the entire program and to practice discussing your understanding of such topics in a spoken form, which is a necessary skill in your career development.

A major portion of the course will be devoted to the project—developing a software application based on a case study. Your project will be evaluated on the functionality of the software application, documentation, teamwork, and presentation.

In the **Project Documentation**, you will:

- Identify the teams and team members.
- Identify the responsibilities of team members.
- Identify the project.
- Identify the programming language that you will use.
- Identify a team/company logo.
- Identify a project plan—timeline with a Gantt chart generated by MS Project or a similar tool.
- Include a testing plan.
- Include the requirements definition document.
- Include the design phase documentation.
- Include project appendices.
 - Database relationships
 - Context diagrams
 - Data flow diagrams
- Include an application overview.
 - Functionality: This refers to the purpose of an application—does it do what it was designed to do?
 - Consistency: This refers to the uniformity of an application across pages—does it flow?
 - Performance: This refers to how efficiently a system responds to users' actions—does the structure optimize access?
 - User-friendliness: This refers to the interface design, colors, graphics, and borders in an application—is it visually appealing or hard to follow?

For the **Teamwork** portion, you will receive a standard 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

Team Presentation

Grades will be based on qualities in the following areas:

- Conformation to formalities—stated rules
- Clarity of explanation on application functionality and design methodology
- Team cohesiveness

Instructional Materials and References

Student Textbook Package

Wood, Dawn Parrish, Mary E. Pascarella and David R. Foley. *Essentials: Microsoft Office Project 2007, Custom Edition*. Pearson Custom Publishing, 2012

Note: In addition to this textbook, all textbooks for the previous SAP 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 “Search” function on the home page to find the following books.

Books> Ebrary

- Fujita, H., and M. Mejri, eds. *New Trends in Software Methodologies, Tools and Techniques*. Amsterdam, Netherlands: IOS Press, 2005.
- Lecky-Thompson, Guy. *Corporate Software Project Management*. Hingham, MA: Charles River Media, 2005.

Books> NetLibrary

- Gorton, Ian. *Essential Software Architecture*. Berlin, NY: Springer Science & Business Media, 2006.

Periodicals

You may click “Periodicals” or use the “Search” function on the home page to find the following periodicals.

Periodicals> EbscoHost> EBSCOhost Databases

- “Application integration aspirations.” By: Bednarz, Ann. *Network World*, 9/23/2002, Vol. 19 Issue 38, p68, 1p, 1 cartoon; (AN 7455963)
- “Software’s Common DNA.” By: Udell, Jon. *InfoWorld*, 8/8/2005, Vol. 27 Issue 32, p34-34, 1p; (AN 17869643)

Reference Resources

You may click “Reference Resources” or use the “Search” function on the home page to find the following reference resources.

- Computers> DevX
- Encyclopedias> AccessScience> Computing & Information Technology

Program Links

You may click “Program Links” or use the “Search” function on the home page to find the following program links.

- Information Technology - Software Applications Programming (ITSAP)> Professional Organizations> Association of Information Technology Professionals

Learning Guides

You may click “Learning Guides” or use the “Search” function on the home page to find the following learning guides.

- Online Tutorials> Computer Technical Tutorials
- Online Tutorials> Java Programming Tutorials
- Online Tutorials> TechTutorials.com

Other References

The following resources can be found **outside** of the ITT Tech Virtual Library, whether online or in hard copy.

Books

- Copeland, Lee. *A Practitioner's Guide to Software Test Design*. Boston: Artech House, Inc., 2003.
- Galin, Daniel. *Software Quality Assurance: From Theory to Implementation*. Boston: Addison-Wesley, 2003.
- Herron, David, and David Garmus. *Function Point Analysis: Measurement Practices for Successful Software Projects*. Boston: Addison-Wesley, 2000.
- Nygard, Michael T. *Release It: Design and Deploy Production-Ready Software*. Raleigh, NC: The Pragmatic Programmers, LLC, 2007.
- Tidwell, Jenifer. *Designing Interfaces*. Sebastopol, CA: O'Reilly Media, Inc., 2005.

Periodicals

- ACM TechNews
Available online at <http://technews.acm.org> (accessed Nov. 24, 2009)
- IEEE Software
Available online at <http://www.computer.org/portal/site/software/> (accessed Nov. 24, 2009)
- IET Software (The Institution of Engineering and Technology)
Available online at <http://www.ietdl.org/IET-SEN> (accessed Nov. 24, 2009)

Web sites

- CodeGuru
<http://www.codeguru.com> (accessed Nov. 24, 2009)
Articles, videos, columns, and code shared by a community of developers
- MSDN Home Page
<http://msdn2.microsoft.com/en-us/default.aspx> (accessed Nov. 24, 2009)
Microsoft's portal page for developer support
- Software Development Tools- Java, .NET, PHP, database, Testing UML, Project Management, SCM
<http://www.softdevtools.com> (accessed Nov. 24, 2009)
A portal to software development tools, such as programming editors and frameworks, software testing, software configuration management, databases, and project management, and industry news
- Webmonkey: The Web Developer's Resource
<http://www.webmonkey.com> (accessed Nov. 24, 2009)
A wiki for Web developers containing tutorials, tips, and advice for designing and building Web sites and programming Web applications for programmers of all skill levels
- ZDNet: Tech News, Blogs and White Papers for IT Professionals
<http://www.zdnet.com> (accessed Nov. 24, 2009)
Portal page to news, reviews, and articles on all facets of the IT industry

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	
• Documentation	20%
• Teamwork	10%
• Presentation	10%
• Software Application	45%
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

(End of Syllabus)