Advanced System Analysis and Design
IS 663, Session 103 – Fall 2017

Administrative Information

Class Time: Mondays, 6:00pm-9:00pm
Instructor: Mark Barnabei
Email: **mark.d.barnabei@njit.edu**
Phone: 609-947-6979

Office Hours: Virtual Office Hours: by appointment
Class Web Site: **All** course materials will be available via Moodle
Prerequisites: IS 531

Required Text

Required

"Software Engineering," Ian Sommerville, Addison-Wesley, England, 10th Edition

Recommended
"Software Engineering: A Practitioner's Approach," R. Pressman, McGraw Hill, NY 2003.
"Software Requirements: Objects, Functions and States," A. Davis, Prentice Hall, Englewood Cliffs, NJ, 1993.
UML Documentation & White Papers: http://www.rational.com/uml

Selected Online Readings

To be announced via Moodle

Course Summary

This course covers the theory, principles, and applications of the methodologies and tools of software analysis and design. Students will read selected material from the literature, actively participate in discussions, labs and exercises and prepare operational projects for real-world problems. We will spend a considerable amount of time interacting and learning through discussion of assigned readings and other material.

Course Objectives

When you complete this course you will have the ability to:

- Be familiar with different types of System Development Life Cycle (SDLC) models; Know how to choose appropriate SDLC models based on the nature of system development projects
- Effectively use UML diagrams and process models for system analysis phase
- Be able to use Rational Modeler to develop UML models

- Master the designing skill for interface, database and program design frequently used in business information systems
- Participate as an analyst/team member on a systems development team working with a **real world organization**
- Effectively utilize appropriate data gathering tools and techniques associated with the collection of system user requirements, constraints and expectations
- Describe, structure and plan an information systems development project's activities using basic Project Management techniques and tools
- Perform basic feasibility study activities associated with systems proposals
- Understand the basics of object-oriented system analysis and design methods
- Choose appropriate development methodology
- Understand and critique formal software requirements.
- Evaluate various architecture and design choices for specific systems and choose optimal solution
- Evaluate and exercise prototyping methodologies.

Grades

ACTIVITY	POINTS
Exam	30
Project	40
Assignments, Labs, Discussions and Participation	30
TOTAL POINTS	**100**

Letter grades will be assigned approximately as follows (the grades may be curved):

Marks	Letter Grade
90% and above	A
80% to 89%	B
70% to 79%	C
60% to 69%	D
Below 60%	F

Team Projects

One of the biggest difficulties for many students in this class is their lack of real world system development experiences -- Imagine trying to learn how to drive through a series of lectures without ever touching a car! That is why the team project is **extremely** important for your learning in this class. The team project should be based on a **real life business** situation where the team is engaged in a significant portion of a systems project. Working as a team, you are to demonstrate your mastery of the concepts, methods, tools, and techniques covered in class. You will be required to view the project from many angles - customer, analyst, developer,

tester, manager and end user. You will develop and analyze requirements, project plans, designs and will eventually prototype your design.

You will work together as a project team. Your team will be responsible for preparing key project deliverables consisting of (but not limited to):

Part I: Feasibility Analysis/Market Research/Competitors
Part II: Project Planning and Management
Part III: Software Requirements Specification (SRS) Document
Part III: Software Requirements Specification (SRS) Document

It is **extremely important** that you do not miss any of the milestones above --- failure to deliver on time will result in a 0 grade for your milestone.

Do not worry if you have no clue about the above milestones – this class is designed so that the lectures are "*synchronized*" with the real world project milestones. For example, one week after we discuss how to conduct interviews, your project team will schedule the actual interviews with your clients. I will highlight the knowledge points that will be applied in your projects in my lectures, and explain how you could use them effectively in the real world setting – so it is extremely important that you come to class!

At the end of the semester, we will have a presentation session where all teams present their work..

Assignments

You will be given five assignments throughout the semester. Some of them **are individual assignments** – you should independently work on the problems and find the solution. Others are **team-based** – you are expected to work closely with your teammates to collectively design a solution. Details of the assignments will be posted online.

Participation

You may have noticed that in-class performance is worth 10% of your total grade. Here is how I measure it:

During each class, I will give out several questions for you to think about. I will pose the questions to the class in general for anyone to answer. If I don't get a response I will randomly pick two or three of you to answer these questions (and to challenge me with follow-up questions, if you have any). In addition to attendance, your performance in answering these questions throughout the semester will be the basis on which I grade your in-class performances. Some questions will be group-based, in which case the whole group will be evaluated based on your answers.

Academic Standards

All students are expected to pursue the highest standards of academic honesty. Plagiarism or cheating on an assignment or examination can lead to an E on the assignment or examination, an E in the course, and other disciplinary action.

Plagiarism or academic dishonesty will not be tolerated. 'Plagiarism' means the intentional unacknowledged use or incorporation of any other person's work in, or as a basis for, one's own work offered for academic consideration or credit for public presentation. Plagiarism includes, but is not limited to, representing as one's

own, without attribution, any individual's words, phrasing, ideas, sequence of ideas, information or any other mode or content of expression. All work submitted for this class should be original; that is, it should be your own. Also, do not turn in work that you have turned in for other classes.

The NJIT Honor Code will be upheld, and any violations will be brought to the immediate attention of the Dean of Students.

TENTATIVE CLASS SCHEDULE

Date	Topic	Readings	Deliverables
9/11	Introduction	CH 1 , 11	
9/18	SDLC Models: Basics, Comparative Analysis	CH 2 "Software Chronic Crisis," W. Wayt Gibbs, Scientific American, September 1994, pp. 86-95. "No Silver Bullet" by Fred Brooks, 1987, Brooks87.pdf	Project Teams Formed
9/25	Process models (cont.) Requirements Engineering Process: Activities, Standards, Documentation	CH3 Wikipedia explanation of RUP: Wikipedia & RUP *A Rational Development Process*	Project Proposal
10/2	Requirements Modeling & Specification Overview: Structured and OO Approach, Use Cases, Informal and Formal Specifications, Data Flow Analysis; Methods: Data Flow Diagrams (DFD), Data Dictionary (DD), BNF, Process Description Examples	CH 4	
10/9	Requirements Modeling & Specification (Cont.) Methods: Use Cases and Scenarios Examples	CH 5	
10/16	Project Management, Risk Management Cost Estimation	CH 22, 23	
10/23	Requirements Management Software Tools for Requirements Management, Modeling & Specification Tools	CH 25	Project Plan Due
10/30	Design: Basic Concepts System & Software Architecture	CH 6	Req Due

	Design Methods & Tools	"A field study of the software design process for large systems" by Bill Curtis, Herb Krasner, and Neil Iscoe (1988) BC-SDP.pdf"	
11/6	Design Methods and Notations Transitioning from Requirements to design - heuristics and guidelines Introducing Design Quality: Basic Elements, Cohesion & Coupling, Reviews, Verification	CH 24 Abstractions for Software Architecture and Tools to Support Them", M. Shaw, R. DeLine, D. V. Llein, T.L. Ross, D. M.Young and G. Zelesnik, IEEE Transactions on Software Engineering, April 1995, V.21. No 4. UNICON-MS.pdf	
11/13	Object Oriented Analysis & Design (OOAD): Basics Concepts, Static & Dynamic Views, Object Dictionary, OOD Overview	CH 7 Dewayne E. Perry and Alexander L. Wolf. ``Foundations for the Study of Software Architecture". ACM SIGSOFT Software Engineering Notes, 17:4 (October 1992). – DP-AW-FSA.pdf	Architecture Doc Due
11/20	OOAD - continued UML: Introduction, Notation, Diagrams HCI	CH 16	
11/27	Design Patterns Change control	CH 7 (7.3), CH. 17	
12/4	Quality Frameworks: SEI Maturity Model & ISO 9000	CH 26	Design Doc Due
12/11	Course Summary, Q&A - Exam Preparation		Project Presentation Prototype Due
12/18	Final Exam		

XI. ANNOUNCEMENTS AND INSTRUCTIONS

Students are responsible for all postings on moodle. Students should check moodle at least two or three times a week for any updates. Any announcements or due dates on moodle take precedence and are final.

NOTE: THE SCHEDULES AND PROCEDURES IN THIS COURSE ARE SUBJECT TO CHANGE IN THE EVENT OF EXTENUATING CIRCUMSTANCES. YOU WILL BE NOTIFIED OF DEVIATIONS.