

Computer Science 420

Fall 2007

1:00 – 1:50 MWF
216 Thompson Hall

Instructor : Dr. M.A. Francel
E-mail address : `francelm@citadel.edu`
Office : 226 Thompson
Phone : 953-6987 (w)
: 958-9491 (h)

Office Hours:

11:00 – 12:00 and 2:00-3:30 MWF, 1:00-2:00 TR or
by appointment; or by phone

Feel free to call me at home for help with your work any day of the week,
including weekends, between 8am and 11pm.

Text : *Essentials of Software Engineering*
Author : Frank Tsui and Orlando Karam

Course Description:

An introduction to current techniques used in medium and large-scale software development. Topics include requirements analysis; architectural, functional, object-oriented, and user interface design; verification and validation; process and configuration management; and professional ethics.

Prerequisites:

CSCI-223 Data Structures

Course Objectives:

The objectives of the course are:

- to introduce the student to the terminology, methods, and artifacts used in the development of medium and large computer projects,
- to make students aware of the ethical and professional issues involved in building software,
- to practice and improve on group problem solving and communication skills.

Course Outcomes:

By the end of the course the student should be able to:

- understand the significance of the various products produced during application design
- understand the various processes used to manage a software design project
- have some experience with the development of a software project
- be able to discuss the ethics of a variety of professional situations
- know how to read and report (oral and written) on technical material

Class Attendance:

Class Attendance is expected. However if for some reason you do miss class it is up to you to (1) get any assignments due that class period to me before class, (2) learn the material presented in class that day before the next class, and (3) prepare any assignments given that day in class for the due date.

Class Discussion:

It is expected that everyone will take an active role in class. This means answering questions as well as taking part in class activities. Students who sleep in class will be treated as absent from class.

Homework:

Homework will be assigned, collected, and graded. It is an important part of the course. You are expected to devote appropriate time outside of class to the completion of this work. Do not get any outside help on any graded work except from me. Homework will not be accepted late for any reason. If you are unable to come to class on the day a homework assignment is due, it is up to you to see that the homework is passed in before class begins.

Quizzes:

Quizzes over the class readings and lectures will be given regularly. NO make ups will be given. Over the term all types of test questions will be used including: essay, short answer, true/false, multiple choice, and matching.

Technical Reports:

Software Engineering is a rapidly changing discipline. (Books, journals, conference proceedings, technical reports, published memos, trade magazines and the internet are constantly full of new information. While web content may be the easiest to change and update, it is also the least refereed by qualified individuals, making the information there potentially least reliable. Books are sources of good information, but quite often are “out of date” by the time they are published. Thus, journals, conferences, and trade magazines often contain the most reliable, current information in terms of on-going research.) Because of the constant changes in the discipline, it is necessary to regularly read the “trade journals.” To help you to develop this habit and to teach you how to read a technical article for its main ideas, the course includes technical reading/writing assignments. Over the term, you will be asked to make two types of technical reports.

Type 1: Here your assignment includes finding a journal article on some specific aspect of the general topic, reading the article, and writing a report on the article.

Type 2: Here your assignment is to give an overview of the current research in some specific aspect of a general topic. You will be expected to identify and report on several current articles that present results in the specific area you have identified within the general topic.

Exam Information and Dates:

All tests will be in-class, closed book exams. Exams will cover all material since the last test.

If you are a student who has been approved by Dr. Zaremba for testing accommodations, I will need to see your letter of accommodation at least one week before the first test.

Test 1	SEPT 19
Test 2	OCT 19
Test 3	NOV 28

Final Exam WED DEC 5 (1 pm)

Assessment:

20% technical reports
40% tests
10% homework + quizzes

30% final exam

Grading Scale:

A	100-90%
B	89-80%
C	79-70%
D	69-60%
F	below 60%

CSCI-420

Software Engineering

Study Units and Related Reading

Fall 2007

All reading refers to the text: *Essentials of Software Engineering*
Frank Tsui and Orlando Karam

ETHICS: Ethics as it relates to software engineering is extremely important and will be covered in conjunction with each and every unit.

Unit 1: Why Software Engineering?

Reading: Chapters 1, 2, 3

We begin by defining the key terms. Once this is done we review the history of software engineering, discussing what has been done correctly and what still needs improvement. We then give an overview of the software process including the activities, methods, and tools. We conclude the unit with a discussion of the key challenges facing software engineering and the professional and ethical responsibilities of software engineers.

Unit 2: Modeling the Process

Reading: Chapter 4, 5

In this unit we present an overview of several different types of process and life cycle models, including the waterfall model, the spiral model, the incremental models and various new and emerging process methodologies. We also discuss modeling techniques, diagramming methods and process support tools. The models introduced are illustrated using real world examples.

Unit 3: Capturing the Requirements

Reading: Chapters 6

The purpose of this unit is to look at requirements analysis and specification. We explain the difference between functional and nonfunctional requirements, present several ways to describe different kinds of requirements, and discuss how to prototype requirements. Other topics discussed include requirements gathering, requirements documentation,

requirements reviews, requirements quality and how to measure it, requirements testability, and how to select a specification method. The unit ends with application of the methods to an example situation.

Unit 4: Designing the System

Reading: Chapters 7, 8

A variety of design issues will be covered. The first major area is architectural design. This will be followed by the basic techniques for detailed design, including functional decomposition, relational database design, and object-oriented design. Next user interface design will be explored. Design principles, user interaction, information presentation and user support will all be addressed. The students will evaluate some local interface based on what they have learned. Lastly design metrics will be examined.

Unit 5: Testing and Quality Assurance

Reading: Chapters 10

In this unit both validation and verification will be examined. Students will be given the opportunity to actually review a document. Also several aspects of program testing will be explored and tried.

Unit 6: System Evolution

Reading: Chapters 12

The unit begins with a discussion of the need for training and documentation and presents several examples of training and documents that could accompany the information system that has been used as an example previously. This is followed with a discussion of the results of system change. The ways change can occur during the system's life cycle are introduced along with how system design, code, test process, and documentation must accommodate them. Typical maintenance problems are discussed as well as the need for careful configuration management.

Unit 7: Planning and Managing the Project (as time permits)

Reading: Chapter 11, 13

The aim of this unit is to understand project planning, scheduling and management. We introduce notions such as activities and milestones, work breakdown structure, activity graphs, risk management, quality control, configuration management, and costs and cost estimation. Again case studies will be used to introduce and demonstrate various ideas. Included here will be test planning and the role of testing in the software process.