

CS 179K Project in Computer Science: Software Engineering

Instructor	Gianfranco Ciardo	email: ciardo@cs.ucr.edu	url: http://www.cs.ucr.edu/~ciardo/
TA	Dragomir Yankov	email: dyankov@cs.ucr.edu	url: http://www.cs.ucr.edu/~dyankov/
Textbook	Instructor's handouts		
Prerequisites	CS 141 with grade of "C-" or better; CS 180; 8 additional upper-division units in Computer Science		
Units	4		
Office hours	Mon & Wed 2:00pm — 3:00pm	Room: Surge 346	
Class meetings	Tue, 11:10am — 12:00pm	Room: Sproul 2356	
Lab meetings	Thu, 6:00pm — 9:00pm	Room: Surge 170	

Catalog description

Small teams of students undertake a software project. The instructor acts as "customer", providing requirements. Teams must develop an independent strategy, including the planning, design, implementation, verification, and testing of some nontrivial software product. A written report and an oral presentation of the design, implementation, and performance aspects are required.

Goals

The goal of this course is to learn how to work on a large project in a team environment, develop a product in a professional manner, use the techniques and tools of our trade, and communicate with teammates and customers.

Contents

In the first week of the course, the instructor will introduce the `latex` and `bibtex` programs, which are required background for the project in this course, and the students must carry on a small individual assignment to demonstrate that they have familiarized themselves with these two programs. Then, teams of approximately five members will work on all the phases of software development for a small but significant project during the balance of the course. As in a real life situation, the project has been chosen by the customer (played by the instructor).

Each team will develop the traditional software work products: a *specifications document*, a *design document*, a *test plan*, and working *code*. In parallel, the class lectures will discuss the expected contents of each of work product, and cover important aspects of software development techniques. The work products, appropriately revised after having received feedback from the instructor, and after considering the experience gained during the following phases, must be integrated in a coherent final written report. This report must also include experimental performance results from running the software, such as timing and memory requirements collected on a few selected real-life inputs.

Each work product is a living document: if a team discovers an error or a better approach at some point during the development, earlier work products can and should be altered accordingly. Changes might also be prompted by a customer's request: that's real life as well!

Detailed schedule (subject to change)

<i>When</i>	Topic discussed in class	<i>Due @ 8:00am</i>
<i>Jan 4, 2005</i>	Introduction to \LaTeX and description of the project	
<i>Jan 11, 2005</i>	Review of the required project documents	<i>HW 1</i>
<i>Jan 18, 2005</i>	Configuration (<code>svn</code>) and build (<code>make</code>) management	
<i>Jan 25, 2005</i>	How to run design and code reviews	<i>Specifications</i>
<i>Feb 1, 2005</i>	How to present specifications and design	
<i>Feb 8, 2005</i>	Group presentations of specifications and design	<i>Design</i>
<i>Feb 15, 2005</i>	How to prepare a test plan	
<i>Feb 22, 2005</i>	Formal verification techniques	<i>Test plan</i>
<i>Mar 1, 2005</i>	Group presentations of the running code	<i>Code</i>
<i>Mar 8, 2005</i>	Feedback for final report	
<i>Mar 15, 2005</i>	— no class, final exam period —	<i>Final report</i>

Coursework

Homework:	5% of the grade (individual).
Specifications document:	15% of the grade (team).
Design document:	15% of the grade (team).
Presentation:	10% of the grade (team).
Test plan:	10% of the grade (team).
Code:	10% of the grade (team).
Final report:	15% of the grade (team).
Individual weekly reports:	10% of the grade (individual).
Team weekly reports:	5% of the grade (team).
Assessment by team members	5% of the grade (individual).

As in all team-oriented coursework, be especially aware that you have a responsibility to perform the required work in a professional and timely manner, not only toward yourself and the instructor, but also toward the other members of your team. Past experience suggests that a good team dynamic, which includes frequent team meetings and regular meetings with the instructor or TA, is just as essential as individual efforts to the successful completion of the project.

Team work has many advantages (mutual support, best use of individual skills, encouragement, camaraderie), but this is true only if everyone participates actively to the project. To ensure that every team member contributes fairly, each team must submit a weekly team report (one to two pages), and each individual must submit a weekly individual report (up to one page), both due at 10:00pm each Tuesday, from January 18, 2005 to March 15, 2005 included. Individual reports must clearly state the team member's contribution to the team's activities.

At the end of the course, each team member will be given a form to fill, in which all the team members must be evaluated. This evaluation constitutes part of the grade.

Except for code, which must be submitted in plain text, all other work products must be submitted as `latex` files, with all the required `bibtex` and `eps` files included. For generic drawings, you can use your favorite program; I like `tgif` (which is free and runs on many platforms) and `OmniGraffle` (which is a commercial product that runs only under Mac OS X), but `xfig` is probably the most commonly used in academia.

Logistics and academic honesty

Team members will have to communicate frequently by email among themselves and with the instructor and the TA. It is their responsibility to check their email frequently and reply promptly.

Each assignment is clearly labeled as a *individual* or *team* assignment. In the former case, a student must follow the UCR rules of conduct regarding academic honesty. In the latter case, the same rules apply to the team as a whole. In particular, note that the same project is used for all teams, and it is absolutely forbidden for team members of different teams to discuss the technical aspects of the project in any way.

Grading policy

Unless you successfully petition for a Satisfactory/No Credit (S/NC) grade, you will receive a letter grade, which will be determined as follows. Assuming your overall numerical grade is $x\%$, your letter grade is:

$-\infty < x < 57$: F	$57 \leq x < 60$: D-	$60 \leq x < 63$: D	$63 \leq x < 67$: D+
		$67 \leq x < 70$: C-	$70 \leq x < 73$: C	$73 \leq x < 77$: C+
		$77 \leq x < 80$: B-	$80 \leq x < 83$: B	$83 \leq x < 87$: B+
		$87 \leq x < 90$: A-	$90 \leq x < 93$: A	$93 \leq x < +\infty$: A+