

CS16xx: Software-System Design and Management

Provost's Innovation in Education Award 2009

Provost's Speaking across the Disciplines Fellowship 2009

Pitt CS Teaching Award 2010 (OMET scores in top 4% across A&S)

This is a proposal to turn the Spring 2010 edition of CS1699 Special Topics in Computer Science: "Software-System Design and Management" into a regular course offering.

December 8th 2010

Part I

Cover Sheet

A. Course

Department: Department of Computer Science

Number: CS16xx

Title: Software-System Design and Management

Credits: 3

B. Instructor

G. Elisabeta Marai (marai@cs.pitt.edu), Assistant Professor

C. Departmental Approval

Chair's Signature

Date

Note to Chairs: Your signature indicates that you and the appropriate faculty or committee in your department believe that there is a demonstrated need for this course in your departments curriculum, and you have adequate resources, including TA support when appropriate, to offer this course on a regular basis.

D. Rationale for New Course

Summary

This software engineering and project management course introduces novel material into the CS curriculum, including project management skills, usability testing, customer interviewing, and specification of formal requirements, “soft skills” which make our graduates more competitive in the global marketplace and less susceptible to off-shoring. The course also introduces a much needed, significant oral communication component into the CS curriculum. Oral communication has been identified as one of the four basic general education skills in the University of Pittsburgh’s undergraduate education program.

Rationale

Software Engineering is a core field of computer science, dedicated to creating software that is of higher quality, cheaper, maintainable, and quicker to build. While software engineering textbooks are deemed to be “*uniformly amongst the most boring writings in all of human literary output*” [1], the field holds treasures of excitement: for example, software engineers have designed Gmail, Facebook, Twitter, Amazon, and the iPod software. How do we account for, and address, this discrepancy?

A basic observation is that software engineering requires not only technical skills — which can be learned in a theoretical framework, but also “people skills” — which textbooks and typical CS courses do not teach. In the real world, software engineers have to interact with and interview real users — most of who are non-technical (drivers, dentists, supervisors etc.) — and then deploy their product to the same audience. Software engineers also tend to work on large-scale projects, in large teams, which often include both technical and non-technical personnel. It is this interaction, and the motivation that comes from having a real user and handling a real problem that makes software engineering exciting. [2] Therefore, we propose an immersive approach to teaching software engineering and project management, through which students will learn and practice simultaneously specific software engineering methods on a large, real-world project.

Because strong communication skills, and in particular oral communication skills, are paramount to the field, the students will also learn to communicate better by writing regularly, through feedback on every written assignment, by presenting several times in class, through required consultations with Writing and Rhetoric Tutors — an important component of this course, and through critiques of the presentations by the course staff and the tutors.

We note that CS students at the undergraduate level tend to have far less experience and training in oral communication and speechmaking than they do in writing. As CS faculty, we frequently deal with novice speakers; yet we rarely encounter students with virtually no writing background. Because of their relative inexperience as speakers, many students are uncomfortable and anxious at the mere thought of giving an oral presentation in front of a group of people. This anxiety makes confidence-building through training and peer-tutoring particularly important.

Pitt CS has two courses that fall under the “software engineering” category. One is a junior-level course, CS1530, which introduces students to medium-scale program design, small-group programming, and a collection of basic software design concepts. The other course, CS1631, is a higher-level course that emphasizes software design methodology. Both courses are primarily introductory methodology courses necessary in the CS curriculum, yet neither sufficiently prepares the students for the challenges of software development in the real world. The proposed course complements the current curriculum.

The main innovations behind the proposed course are two. First, the course uses real-world, community-based projects to introduce novel material into the CS curriculum. Such material includes project management skills, team-work, usability testing, customer interviewing, and specification of formal requirements, “soft skills” extremely valuable to our graduates. Second, the course introduces a much needed, significant oral communication component into the CS Department, and helps foster the cross-disciplinary collaborative effort

among CS, the English Department, and the Pitt Writing Center jump-started by the 2010 edition of this course.

The course has been test-driven as a special topics course in 2010, with excellent student feedback. Appendix A shows some of the relevant student feedback, which helps demonstrate the need for this course in the CS curriculum.

References:

[1] S. Krishnamurthi, "Software System Design", 2009.

<http://www.cs.brown.edu/courses/csci1900/2009/>

[2] S. Gaudin, "Dual Roles: The Changing Face of IT", 2005.

<http://itmanagement.earthweb.com/career/article.php/3523066>

Part II

Course Description

CS1699 is a software engineering course with particular emphasis on project management, usability testing, customer interviewing, specification of formal requirements, and oral communication, "soft skills" of particular importance to CS graduates.

This course is one of the last in the undergraduate curriculum and it can tie together earlier courses in a practical way. In addition to the focus on a team programming project, we discuss, debate, and think about the team software development process. In a significant departure from tradition, the students train with a public speech coach to learn how to effectively communicate orally -- from pitching ideas to a project manager, to interviewing a customer, resolving conflict with another team-member, giving feedback and presenting results.

A. Course Aims

We have three aims for the students of this course. First, learn enough about software engineering and teamwork to successfully implement software systems after they graduate. Second, become a person to whom others turn for advice on software engineering and development. Third, learn to think clearly about software project issues, spot problems early, articulate conclusions, communicate with others, and adapt to changes.

B. Course Objectives

When students finish the course, they should be able to:

- Determine requirements and specifications for a software project
- Apply strategies for effective communication
- Implement and evaluate different software engineering strategies
- Collaborate with a team to implement a moderately large software system by
 - designing the overall project,

- designing the pieces,
- implementing and integrating the pieces,
- demonstrating the result
- Clearly communicate their ideas to various audiences.

C. Course Methods

The methods for this course include lectures, assignments, class presentations, a term project, and exams.

In this course students identify, design, and implement significant software applications and learn and practice techniques of project management, design, coding, testing, maintenance and communication. Lectures and short assignments build up to the main project. This are relatively large projects, each requiring 10-15 students; 12 programmers is the average size of a typical software project [3].

As useful preparation for the main project, students conduct mock interviews with the course staff, who pose as employees of small companies (for example, employees of a small bookstore in Arizona, or the campaign staff of a young politician). Some of the course staff play technological ignoramuses and even Luddites — the type of partners that computer scientists usually struggle communicating with. There are several short written assignments in which the students present the assignment problem and propose solutions. The staff organizes elevator-speech competitions that the students use to promote their ideas. The class and staff vote on these ideas to identify the most promising approach. To prepare for these assignments, the students are required to meet regularly with the Public Speech Tutors (English majors trained each year as part of the Writing Center’s tutoring program.) The tutors and the course staff will also provide feedback after class assignments and presentations.

For the main project, the students have to interview actual clients, design and implement a software system, and then deploy their product to the same audience. Quick prototyping and iterative user feedback — both essential aspects the modern spiral model for software engineering — are inherently necessary. The students have to deal with problems in integrating large components, especially components that

are themselves evolving. A good way to tackle this is to have smaller groups work on different parts of the same system, thereby (inadvertently) creating trouble for one another and negotiating solutions. This forces students to collaborate and be team-players if they want to succeed, skills that are crucial and often neglected in the CS curriculum.

The projects typically come from the Pitt community. The 2010 proposed project brought together the departments of Astronomy and Computer Science; the waiting list includes software engineering projects with the Humanities, the School of Public Health, etc. The “customers” thoroughly understand and accept that the results will be mere early prototypes of full systems. In turn, the students have an opportunity to keep working on the next prototype of their term project as part of the departmental capstone experience.

A written and oral presentation at the end of the class-project is required, in addition to assignments, exams, and a term project.

D. Course Prerequisites

CS1501 and either: CS1520, CS1530, CS1550, CS1555, CS1566, or instructor's permission; programming experience expected.

E. Expected Size

20 to 40 students.

F. Terms of regular offer

Every other year.

Reference:

[3] Dreaming in Code, Scott Rosenberg, 2008.

Part III

Course Syllabus

A. Class Information

For the course aims, goals, and mechanics please see Part II.

All class-related information (including handouts, electronic readings, and assignments) should be accessed through the CS1699 web portal located at:

<http://vis.cs.pitt.edu/teaching/cs1699>

1:: Prerequisites and Attendance

Prerequisites: CS1501 and either: CS1520, CS1530, CS1550, CS1555, CS1566, or instructor's permission; programming experience expected.

The class meets **** in SENSQ ****. Attendance is mandatory. Class will start right on time; being consistently late (e.g., 2 or more times) is subject to the ``fair-play`` penalty described in Section 9. :-)

2:: Textbooks

The two following loved and cherished texts are required in this class:

- Dreaming in Code, by Scott Rosenberg, 2008.
- The Mythical Man-Month: Essays on Software Engineering, by Fredrick P. Brooks, Jr., 1995.

We will read additional articles as necessary (electronic copies will be provided for these).

3:: Staff

TBA

4:: Course Grade, Assignments, and Final Project

The course grade will be decided based on the following factors:

- Course projects (60%)
- Class participation (10%)
- Assignments and exams (30%)

Classes will be divided among discussions and debates, guest lectures, design critiques, student presentations, and group meetings.

Many of the assignments will be done in groups. Please note that in this course, as in real life after graduation, whom you get to work with is determined by us/your project manager, not by you. All members of a group will earn the same grade, barring exceptional circumstances. In a significant departure from tradition, the entire class will work on (different parts of) one project, and we will assign the project. The project has been carefully chosen to illustrate various important aspects of software engineering.

In addition, we will not impose any programming language choices on you. You may use any language you wish, but you will be asked to justify your choice of language.

Your primary grade will come from the course project. While the assignments will not define your grade, your performance on them can alter your grade (in particular, not paying enough attention to them can cause you to lose a letter grade or more). While you will be graded on the intermediate products of the project, the majority of your grade comes from (1) customer satisfaction and (2) demonstrable performance improvement for the end-user. One of your project activities will be to define reasonable metrics that demonstrate improvement. Failure to define such metrics will count against your grade implicitly, because you will be unable to satisfy one of the two main sources of evaluation.

Last, but not least, this course emphasizes oral communication skills -- reflected in both assignments and the class participation component of the grade. To help define successful oral communication, we provide two grading rubrics, one for student presentations and the other one for class discussion (see Section 9 below).

5:: Late Policy

We will not accept late assignments. The assignments will sometimes be timed to coincide with a lecture on a topic, so completing the assignment is crucial class preparation. After the class, the assignment will have much less value. The more deadlines you miss, and the more serious the ones you miss, the greater this will affect your grade.

6:: Collaboration

You may not discuss the assignment with anyone outside of the permitted group (by default, nobody else) with the sole exception of the course staff.

If you have questions, please send them to the course staff. If we think the question or answer are of general interest, we will respond on the mailing list. Failure to follow this rule may be viewed as violation of the collaboration policy.

The project is, naturally, a group activity. Groups will communicate internally in ways defined by the group leadership. Groups may not communicate with one another except in clearly permitted ways (as defined by the course project requirements). In particular, groups can (and should!) communicate to establish interfaces and protocols. However, group members cannot discuss any ideas, hints or any other information across group boundaries that may lead to a solution of the problems facing a group.

7:: Academic Honesty

The University has an [Academic Code](#) that governs all our transactions. This Code establishes the school's policy on cheating. We expect that you, as students and scholars, will abide by this faithfully and fully. Cheating in this course will result in an F for the course.

8:: On Writing

We care about your ideas, but we care equally deeply about the quality of your writing. We care about spelling, capitalization, punctuation, sentence construction, paragraphs, and so on. Avoid passive speech except where appropriate. Extremely good ideas expressed very poorly will earn a very poor grade.

9:: On Oral Communication and Discussion of Readings

We care equally deeply about the quality of your speech -- be it during team communication, results presentations, or interviewing potential users. We want you to be successful, and to this end this semester we will work with a Public Speech coach to improve your oral communication skills. Please come to meet the coach properly prepared: for example, if you are scheduled for a speech practice, have

your slides ready and have practiced by yourself a couple of times. We also provide grading rubrics for class presentation and class participation.

It is important that you read the required class readings for a given class *before* the class meets. These readings are carefully selected to stimulate class discussion; reading them after class is far less useful. Therefore, we ask you to submit to the course wiki short comments for each reading *by 8pm the day before class*. Each reading will feature on the wiki a "Lessons learned" section, and a "Topics for Discussion" section. Under "Lessons learned", please enter up to three sentences summarizing what you've learned from the paper or chapter. Under "Topics for Discussion" please enter three items that you'd like to propose for discussion. This will allow the presenters and discussion leads to come prepared to address class questions -- including amazing illustrations and examples and so forth :-). To encourage everyone to submit their comments on time, we implement the "fair-play" penalty for being late on the wiki. Entering comments late (or not at all) will result in having to prepare a mandatory 5-minute technical, entertaining skit for the following class meeting. (I.e., late for Monday == skit for Wed etc.)

10:: Students With Disabilities

If you have a disability for which you are or may be requesting an accommodation, you are encouraged to contact both your instructor and Disability Resources and Services, 216 William Pitt Union, (412) 648-7890/(412) 383-7355 (TTY), as early as possible in the term. DRS will verify your disability and determine reasonable accommodations for this course.

B. Schedule

Date	Topic	Readings	Assignments	Additional Materials
****	INTRODUCTION	****		

A&S New Course Proposal: CS16xx

01/06	Class Mechanics; Sw Eng. & Project Management		Personal background out	
01/08	Project Management & Self-Marketing	CS1699 Class Information; Dreaming in Code Ch.0	Elevator Speech exercise (in class) Personal background due Create wiki account and enter comments and background before 12pm	SF Bath salt pitch BlogUpps pitch
01/11	OpenSource Sw; Brooks's Law; Sw failures	Dreaming in Code Ch.1 Dreaming in Code Ch.2	Wiki comments due by 8pm previous day	
****	REQUIREMENTS	****		
01/13	Interviewing and Requirements Elicitation	Techniques for Requirements Elicitation Questioning techniques (Team 4)	Team Wiki contributions (Requirements Elicitation Assignment) due by 8pm previous day	
01/15	Requirements and Use Cases	Wiki Requirements Elicitation		
01/18	no class (MLK day)			
01/20	Mock Interview Questions Exercise (in class)	Sw Requirements/How to Write Requirements (01/15 handout)	Mock Requirements assignment (in class) out	

A&S New Course Proposal: CS16xx

01/22	Mock Requirements Critique		Mock Requirements due by 10pm previous day on the wiki	
01/25	Final Project description (warmup)	Pair Programming (pdf)	Email assignment out Wiki comments due by 8pm previous day	
01/27	Email-draft Critique	Tips for soliciting an interview via email: Email in the Workplace Writing Effective Emails	Email draft due via wiki by 8pm previous day	
****	TEAM DYNAMICS	****		
01/29	Build, Buy, or Borrow? Team Structure and Teamwork	DiC Ch.3 & Ch.4	Wiki comments due by 8pm prev. day	
02/01	Managing Dogs and Geeks	DiC Ch.5	Wiki comments due by 8pm prev. day Project Interviews due this week	
02/03	Software Design and Integration	DiC Ch.6	Wiki comments due by 8pm previous day	
02/05	Giving and Receiving Feedback; Resolving Conflict		Mock Feedback (in class); Conflict Exercise (in class)	

A&S New Course Proposal: CS16xx

			Final Project Requirements due via wiki by 12pm Mon (same style as the mock requirements)	
****	SPECIFICATIONS	****		
02/12	Specs	DiC Ch.7	Wiki comments due by 8pm previous day Composite Final Project Requirements due via wiki by 12pm (same style as the mock requirements)	
02/15	Final Project Requirements Critique		Specifications assignment out	
02/17	Scheduling	DiC Ch.8	Wiki comments due 8pm prev. day	
02/19	Guest Lecture (William Pitt Union 1pm) Richard Stallman Use the class time to work on your specs	How to Write Specs (Joel on Software)	Wiki comments due 8pm prev. day (Final Project Specs due by Mon 8pm)	
****	DESIGN	****		
02/22	Development Strategies	DiC Ch.9	Wiki comments due 8pm day before Final Project Specs due by 8pm	Scrum on wikipedia

A&S New Course Proposal: CS16xx

02/24	Engineers and Artists	DiC Ch.10	Wiki comments due 8pm day before Design assignment out	Law of Leaky Abstractions (Joel on Software) Example toy design documents (imperfect, reference with caution)
****	SCHEDULING	****		
02/26	Monitoring Progress	DiC Ch.11	Wiki comments due by 8pm prev. day	Finish DiC
03/01	Design Critique		Design assignment due by 2pm; 10 minute group presentation (in class)	
03/03	Team Mechanics	MMM Ch.3 (The Surgical Team)	Colors Exercise (in class) Wiki comments due 10pm prev. day	
03/05	No Silver Bullet	MMM Ch16	Wiki comments due by 10pm prev day	No Silver Bullet online
****	PRODUCTIVITY TOOLS	****		
03/08, 03/10, 03/12	No class -- Spring break			
03/15	Bug Tracking	Painless Bug Tracking (Joel on Software) Writing Solid Code Chapter 8: The Rest is	Wiki comments due by 10pm prev day	

A&S New Course Proposal: CS16xx

		Attitude	Team Roles due by 10pm today	
03/17	Curriculum Discussion	What to do when you're screwed	Wiki comments due by 10pm prev. day	
03/19	Group meetings		Wiki Final Specs and Design due 10pm today	
03/22	Narayana Tummala (Tech Lead Manager, Google)	Bring in your burning questions	Each group should commit their component Interface to the repo (if using C, this would be your header files)	
03/24	Interface Design	Slanty Design Guidelines for User-Centered Web Design NIH Criteria for Evaluating Health Websites WebContent.gov Content Criteria		
03/26	Group meetings		Engineering report due 10pm today	
03/29	All-Hands Engineering Meeting: Project checkpoint		Prepare 10 minute group presentation describing your component's final integration to the rest of the system and progress to date (see Engineering report)	

A&S New Course Proposal: CS16xx

03/31	Mary Beth Sapienza (Project Manager, BNY Mellon); Integration Demo;		Integration Demo in class	
04/02	Group meetings		Engineering report due 10pm today	
04/05	Profiling; Performance Analysis	Profiling Tools under Linux A Survey of Systems for Detecting Serial Run-Time Errors (section III) Open Source Profilers for Java		Debugging Tools for Java
****	QUALITY ASSURANCE	****		
04/07	Testing; Technical Reviews & Inspections	Wikipedia Sw Testing Wikipedia Code Reviews Wikipedia Code Walkthroughs A Guide to Code Inspections		
04/09	All-Hands Engineering meeting		Engineering report due 10pm today; 10 minute group presentation due in class	
****	CLOSING OUT A PROJECT	****		
04/12	Customer Feedback, Post-mortems, Close- out Reports	(presenters: please contact instructor)		
04/14	All-Hands Engineering Meeting; Outsourcing	McConnell, Rapid Development, Part III Ch.28	Alpha Release due (in class)	

A&S New Course Proposal: CS16xx

04/16	Group meetings		Engineering report due 10pm today	
04/19	Group meetings		Informal report to manager due	
04/21	Beta Release		Beta Release demo (in class)	
04/23	Launch (Demo Day)		Launch due (in class)	
04/28	Closing Out		Project Workbook due today	

Part IV

Sample Course Materials

A. Sample grading rubric (class presentation)

The following rubric (based on a J. Skupien – Communication Dept handout and a 11/16/09 cs3610 in-class exercise) should help ensure that you are prepared for future class presentations.

Presentation Component	Unacceptable 0 Points	Acceptable 1 Point	Good 2 Points	Excellent 3 Points
Purpose (aka “stating what we expect to have learned by the end of the presentation”)	No stated goal/purpose	Poorly stated, vaguely formed outcomes	Clearly stated	Well formed: specific, measurable and realistic
Introduction, Overview & Background	No introduction, overview, or background	Awkward, sketchy or unclear introduction, overview and/or background	Confident, clear and fluent introduction, overview and background; could be more polished	Confident, clear and fluent introduction, overview and background
Style & Delivery: use of effective verbal and nonverbal communication skills (e.g. voice volume, inflection, eye contact etc.)	Poor style (long pauses, “Umm...” and other mannerisms, poor eye contact)	Either fluent delivery but read, or awkward delivery but spontaneous	Generally good delivery and spontaneity but could improve	Excellent verbal and nonverbal style; good voice projection with inflection, spontaneous delivery, good eye contact
Structure, Pace & Amount: pace, flow, and delivery speed	Too slow/fast; notably short or long; information omitted; bad mapping of time for content (e.g. extensive rambling)	Better pace but still uncomfortable; presenter stretched to fill time or include everything; omits one or two technical points or details	Consistent pace throughout; information covered to meet all objectives; fluid transitions	Pace adjusted based on participants; all objectives met with supplemental information provided; reading material restructured to increase understanding; fluid transitions
Use of Audio/Visuals	All text	Visuals included, but few, ineffective and occasionally unnecessary	Supports presentation with effective slides, pictures, video clips, news articles but could be improved	Supports presentation with effective slides, pictures, video clips, news articles and/or brings in relevant guests etc.
Use of Technology: appropriate and skilled use of equipment	No use of technology (may be appropriate in some classes)	Awkward or incorrect use of equipment; untested equipment; no backup	Correct use but could be more practiced	Smooth, seamless, well practiced use of technology
Audience Involvement	Puts audience to sleep; lectures to audience with little interaction	Asks occasional questions	Uses structured activity or exercise	Spontaneous and lively interaction with task focus; relates the topic to the students’ lives and uses concrete examples, exercises, stories, quotes

A&S New Course Proposal: CS16xx

				and questions to involve the audience
Discussion Leading: facilitates discussion and is receptive to feedback	Little or no discussion; pouring straight through slides	Aimless discussion without clear organization or purpose; fielding posed questions in curt manner	Prepared discussion questions; targeted discussion	Prepared discussion questions; targeted w/ spontaneous extrapolation (fielding on-the-fly questions); gets shy students involved by bringing up their wiki comments

B. Sample Assignment

Requirements Exercise

Your task is to write a requirements document (see the How to Write Requirements handout) for the grading application described below. You must turn in use cases, but keep scenarios handy in case we ask to see them. If you have questions about how to classify something, ask!

You must submit your requirements in PDF format via the course wiki by 10pm on Thursday, Jan 21st. This is not a group assignment, so each of you should submit their own take on the requirements.

The goals of this assignment are:

- practice interviewing stakeholders
- practice writing requirements as use cases
- practice presenting your findings

You will be graded for the completeness of your use cases, so try to get as many detailed scenarios as you can, as well as additional requirements that do not fit naturally in the workflow of the scenarios. Please keep in mind that you will lose points if you miss essential constraints.

Although use cases are a natural fit with the Task Analysis requirement-elicitation technique, you probably want to review the Questioning Techniques presented by Team 4 on the wiki as well. After all, to get the scenarios from stakeholders, you do need to interview them.

Finally, in class we will discuss each solution, so please come prepared to give a 2 minute overview of your use cases. You may use the wiki PDF as a prop. For the class presentation, please keep in mind the feedback you got from the Public Speech Coach for your previous presentation.

Because you will submit your solutions concurrently, we can not accept late submissions for this assignment. Edits after 10pm will disqualify your submission.

Your client

Your Smarty Inc. company has been hired by the Punxsutawney CS department, a semi-public university Computer Science department trying to make it to the electronic century (or bust). The department faculty and graduate students have an insane amount of grading and reporting to do; the department would like to commission an application that would streamline most of this process.

A&S New Course Proposal: CS16xx

Your interviews

The department has given you access to four staffers. They are:

Overworked Graduate TA
Laid-back Graduate TA
Obsessive-Compulsive Instructor
Geeky yet hip Tech Staff

Interviews will take place during class on Wednesday, Jan 20th. They will be in a "speed dating" format. The staffers will be seated at four different tables. You get ten minutes with each staffer, after which you have to move to the next table, no exceptions. You might get a chance to ask additional questions via email, if the staffers like you, but do not count on it.

Appendix A

Sample Student Feedback

from **Jake Gress** <JDG30@pitt.edu>
to Liz Marai <marai@cs.pitt.edu>

date Wed, Nov 10, 2010 at 4:17 PM
subject Re: [cs1699] Fwd:

Professor Marai,

Hello. How are you? [...] I got word on Monday that I was being offered the ITLDP position with BNY Mellon. I gave my acceptance today. It is truly a dream job for me and I have been on cloud 9 since I have received the news! This position does not start until June though, so I have some suspenseful waiting to do!

I also wanted to let you know how important your 1699 course was in helping me get this job. I really could not say this about any other class I took within the CS department. Because the ITLDP position centers itself around Project Management and Leadership, there was a lot of information both from the 1699 and Dreaming in Code that I harped back on. This was especially useful with some of the interview questions. Just wanted to give you an update and say thanks again. I hope you will be able to offer 1699 again this spring.

Best Regards,
Jake

from **Yin, Andrew** <afy4@pitt.edu>
to "Marai, Georgeta Elisabeta" <marai@pitt.edu>

date Wed, Jun 16, 2010 at 9:43 AM
subject Special Topics Class CS1699

Hi Dr. Marai,

I just want to thank you for the great experience I had in our special topics class last semester. I learned alot from it and it's all coming to use right now as I'm interning with UPMC. I've only been here for a few days, but I can't help but notice how that class prepared me the most for the real work environment. Will you be offering it again next year? I have a few friends who will probably be interested.

A&S New Course Proposal: CS16xx

Thanks
Andrew