

# Syllabus for CS 499-001 Senior Design Project Fall 2013

## Instructor:

Dr. Jane Hayes ([www.cs.uky.edu/~hayes](http://www.cs.uky.edu/~hayes)).  
Room 228, Hardyman Building  
Office hours MW 1100-1150 in FPAT 773C or by appointment

## Course information:

Course homepage <http://selab.netlab.uky.edu/homepage/CS499fall13.htm>

Course: CS 499 Senior Design Project  
Section: 001  
Meets: MWF 1200-1250  
Location: Ralph G. Anderson Building -Rm.203 RGAN

## Description:

This is a project course. Students will work in small groups to design and implement systems of current interest to computer scientists. The course will also provide a high-level overview of the software engineering discipline: software requirements, software design, software construction, software management, and software quality.

## Course Outcomes:

### CS Outcomes:

Specifically, students will be able to:

- CS1. use accepted project development processes in the project implementation
- CS2. implement a large project
- CS3. work as part of a team
- CS4. present results of their work orally
- CS5. document their work in a written report

also

- The student shall be aware of ethical considerations in software engineering. (C1)

### Teacher Course Evaluation Supplemental Questions:

The course has helped me to improve my ability, my understanding, or my knowledge in the following categories:

- 37. An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs
- 38. An understanding of professional, ethical, legal, security, and social issues and responsibilities
- 39. Use the standard project development steps (specification, design, etc.) in implementing a project
- 40. Implement a large project
- 41. Develop and present a talk on the status of a project

42. Develop a written report on a large project

43. Improve my ability to function effectively on teams to accomplish a common goal

**Course Materials:**

**Recommended, but not required, texts:**

Shari Lawrence Pfleeger and Joanne M. Atlee  
*Software Engineering: Theory and Practice, \* Fourth Edition\**  
Prentice Hall  
ISBN: 0136061699

Frederick P. Brooks, *Mythical Man Month, 2<sup>nd</sup> Edition*, Addison  
Wesley  
ISBN: 0-201-83595-9

Martin Fowler  
*UML distilled: a brief guide to the standard object modeling language*  
(NOTE: 3<sup>rd</sup> edition available in library)

Gamma, Helm, Johnson & Vlissides  
*Design Patterns: elements of reusable object-oriented software*  
Addison-Wesley. ISBN 0-201-63361-2.

**You do not have to obtain these**, though you may choose to do so.  
Also, copies have been placed on reserve in the Engineering Library  
(3<sup>rd</sup> floor Anderson Hall)

**Other readings, as assigned:**

These are available via hyperlink in this syllabus or are on our course web page. See list below.

**Course web page:**

Course materials will be available on the course web page. The course web page and e-mail will be important methods of distributing information for the course.

**Grading:**

Your grade in CS 499 will be determined according to these weights:

- Project grade: **75%** (this is calculated as:  
Team grade – teamwork 15pts (as evidenced by team mtgs, 360 review forms,  
blog/developer notes), project processes 10pts (evidenced by project artifacts  
and presentations and team mtgs), project outcomes 30 pts (evidenced by project  
demo and artifacts)  
Individual grade – participation 20pts (as evidenced by attendance, course  
participation, blog/developer notes))

- Senior survey/resume/peer review: **4%**
- Presentations: **21%**

Where:

- A= 92 - 100%
- B= 83 - 91%
- C= 74 - 82%
- D= 65 - 73%

F = 64 and below

### **Whining May Lower Grades [1]:**

You are always welcome and encouraged to discuss exams and homework with your professor; it is an excellent way to learn from your mistakes. If the grading does not make sense to you, please ask. You may not yet have understood your mistake -- or there may be an error in the grading. However, whining, demanding a re-grade instead of requesting one, or saying that you deserve more points is a good way to convince a professor to re-grade your entire assignment or exam, perhaps with more careful attention to your mistakes.

### **Attendance:**

Students are expected to attend and participate in all scheduled classes. Arrival after attendance has been taken at the start of class will be considered an absence. The following are acceptable reasons for excused absences: 1) serious illness; 2) illness or death of family member; 3) University-related trips (S.R. 5.2.4.2.C); 4) major religious holidays; 5) other circumstances that the instructor finds to be "reasonable cause for nonattendance." It is the student's responsibility to provide sufficient documentation regarding the nature of the absence, and the instructor retains the right to ask for such proof.

### **Late Policy:**

Assignments must be submitted in person at or before **class time** on the day the assignment is due, unless otherwise indicated by the instructor. Assignments turned in after class starts are **late**. Credit will be deducted for late assignments. Assignments will not be accepted after solutions have been distributed.

### **Academic Honor Code:**

Individual work (homework, exams) must be your own. No sharing of computer code or other work will be allowed. Group projects allow the sharing of ideas and computer code within the group. No sharing of work **between** groups will be acceptable. The University of Kentucky's guidelines regarding academic dishonesty will be strictly enforced. "All incidents of cheating and plagiarism are taken very seriously at this University. The minimum penalty for a first infraction is a zero on the assignment. [3]" **See attached policy on plagiarism, also [here](#).**

### **Accepting Responsibility for Failure [2]:**

Failure is an unpleasant word, with bleak connotations. Yet it is a word that applies to every one of us at different stages of our lives. No one is exempt. Our icons, gurus, religious leaders, politicians, rock stars, and educators all fail. It is simply a reality of being human. It is also a label that we fight desperately to avoid. And it is this fight to avoid failure that drives us forward towards our life accomplishments. So--why can't we take responsibility for our own failure when it does occur?

We need to accept responsibility for a very important reason--namely, maturity. We cannot reach a full level of maturity until we accept ownership of our own mistakes. As an educator, I am confronted with this problem on a daily basis. When a student is late for class, it is because a parent failed to wake them up. A failed test becomes the responsibility of the teacher, the system, society, an after school job, but never the fault of the test taker. An incomplete assignment is inevitably due to the needy demands of a friend, or an electrical failure. I feel particularly blessed because the power circuits leading to my home must be exceptionally fine, as I have yet to experience the myriad of blackouts that have plagued my students.

Nevertheless, the daily onslaught of excuses has left me questioning the value of our education system. What, after all, is the point of "higher learning" if we fail to master the basic task of owning up to our own mistakes?

As we proceed through our education system and indeed life, our excuses for failure become more grandiose and perhaps more grotesque because the crude reality is that we have failed to mature in any significant sense of the word. To continually shift responsibility away from ourselves is worse than being a coward. Even a coward will admit that their failure is a result of their own lack of courage.

Accepting failure takes strength of character, honesty, and humility. It provides a building block for future achievements. When we deny culpability, we rob ourselves of the chance to learn from our mistakes. We condemn ourselves to a lifetime pattern of avoidance and deception. Like Marley's ghost, dragging his chains of missed humanitarian opportunities behind him, we crawl forward pulling our chains of pathetic excuses behind us--never fully maturing, never fully reaching our true potential. This stale baggage is far more character eroding than any of our individual failures could ever be.

### Computer Facilities:

You will be assigned an account for this course in the Multilab, a PC laboratory administered by the Computer Science department and located in Room 203 of the Engineering Annex. For information regarding these laboratories, see links under "facilities" from the Computer Science homepage (<http://www.cs.uky.edu/>). You may use alternative computer systems for developing and testing your work, provided that your submitted work will compile and run under the proper software environment as directed in class.

### Group Projects:

The group project for the course will require you to work together with other students in the class. You will be evaluated on your individual contribution to the group project and presentations of the project results. The instructor will make group assignments. Group members are not guaranteed to receive the same grade; evaluation of the group will be individualized to determine individual understanding, commitment, and mastery of the project goals (see below). More information on this can be found below the schedule. As part of the project, written reports will be required.

**Proper language usage is required.**

### Schedule:

Week	Date	Readings	Topics	Project
1	Wed 8/28/13		Software engineering visitor, syllabus	
1	Fri 8/30/13	Developer notes/Engineering notebook reading, Business plan readings	Syllabus; Career Center Visitor; Entrepreneurship, business plan visitor	
2	Mon 9/02/13	<b>Labor day – no class</b>	<b>No class</b>	
2	Wed 9/04/13	Requirements, story points readings	<b>Customer presentations</b>	
2	Fri 9/06/13	Requirements, story points readings	<b>Requirements/user stories/story points; Team building exercise [in class</b>	Teams formed by start of class

			<b>activity]</b>	
3	Mon 9/09/13	V&V, agile, project management readings	Project mgmt./verification and validation/agile process	Teams bid on projects, Selections emailed to teams, <u>hand out project plan assignment</u>
3	Wed 9/11/13	Architecture, TDD, design readings	Architecture, design, test driven development , use case development from requirements <b>[in class activity]</b>	<b>Resumes due,</b> teams hold internal team meetings, meet with sponsors to establish requirements, start to develop user stories
3	Fri 9/13/13	Poker planning readings	Poker planning, stand up meeting [in class activity]	Internal team meetings, meet with sponsors to establish requirements, work on project plan
4	Mon 9/16/13	Design, Reqts to design to code, walkthrough, CM, tracing readings	Design, design patterns, code walkthroughs, configuration control, version control (github, svn), tracing, System sequence diagram <b>[in class activity]</b>	Team progress reviews with instructor, internal team meetings, meet with sponsors to establish requirements, develop project plan, start to set up development environment
4	Wed 9/18/13	Code walkthrough readings	Design proposal <b>[in class activity]</b> , Code walkthroughs <b>[in class activity]</b>	<b>Project plan assignment due, Turn in developer notes, documents, peer review form;</b> internal team meetings, meet with sponsors to establish acceptance test plans, set up development environment; <u>hand out architecture assignment</u>
4	Fri 9/20/13	Javadocs, Unit testing readings	Javadocs [in class activity], Unit testing [in class activity]	Internal team meetings, sponsor meetings, work on architecture assignment, continue setting up development environment
5	Mon 9/23/13		Project meetings	Internal team meetings, sponsor meetings, work on

				architecture assignment, continue setting up development environment
5	Wed 9/25/13		Sponsor status meeting	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development, Prepare presentation, conduct team dry run of presentation (you may invite instructor to attend), correct presentation and documents
5	Fri 9/27/13		Project meetings	Internal team meetings, sponsor meetings, work on architecture assignment, continue setting up development environment
6	Mon 9/30/13		<b>Present</b> to class, customer, and instructor	<b>Architecture assignment due; Turn in presentation, developer notes, documents, peer review form; <u>hand out coding assignment</u></b>
6	Wed 10/02/13		<b>Present</b> to class, customer, and instructor	
6	Fri 10/04/13		No class	No class
7	Mon 10/07/13		<b>Present</b> to class, customer, and instructor (if needed)	
7	Wed 10/09/13		Project meetings	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development
7	Fri 10/11/13		Project meetings	Progress review meetings with

				instructor, internal team meetings, meet with sponsors as required, ongoing software development
8	Mon 10/14/13		Project meetings	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development
8	Wed 10/16/13	Code of conduct, assigned ethics readings	Ethics and code of conduct, project meetings	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development
8	Fri 10/18/13	No class	No class	
9	Mon 10/21/13	Code of conduct, assigned ethics readings, system testing – midterm of semester	Ethics and code of conduct, system testing [ <b>in class activity</b> ]	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development
9	Wed 10/23/13		Project meetings	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development
9	Fri 10/25/13		Project meetings	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development
10	Mon 10/28/13	Maintenance readings	Change requests [in class activity]	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software

				development
10	Wed 10/30/13		Project meetings	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development
10	Fri 11/01/13		Project meetings	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development
11	Mon 11/04/13		Customer status meetings	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development
11	Wed 11/06/13		Customer status meetings	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development, Prepare presentation, conduct team dry run of presentation (you may invite instructor to attend), correct presentation and documents
11	Fri 11/08/13		<b>Present</b> to class, customer, and instructor	<b>Code assignment due, Deliver</b> product to sponsor and <b>demo</b> to instructor, <b>Turn in presentation, developer notes, documents, peer review form; <u>hand out maintenance assignment</u></b>
12	Mon		<b>Present</b> to class, customer, and	



	11/11/13		instructor	
12	Wed 11/13/13		<b>Present</b> to class, customer, and instructor	
12	Fri 11/15/13		Catch up, if needed	Elicit change request from customer, start working on change
13	Mon 11/18/13		Catch up, if needed	Elicit change request from customer, start working on change
13	Wed 11/20/13		Project meetings	Progress rvw meetings w/instructor, internal team meetings, meet w/ sponsors as required, ongoing s/w development
13	Fri 11/22/13		Project meetings	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing s/w development
14	Mon 11/25/13		Project meetings	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing s/w development
14	Wed 11/27, Fri 11/29/13	<i>No class</i>	<i>Fall break - Thanksgiving</i>	
15	Mon 12/02/13		Project meetings	Internal team meetings, meet w/ sponsors as required, ongoing s/w development
15	Wed 12/04/13		Project meetings	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing

				s/w development
15	Fri 12/06/13		Project meetings	<b>Senior surveys due</b> , Progress rvw meetings w/instructor, internal team mtgs, meet with sponsors as required, ongoing s/w development; Prepare presentation including delivered requirements and lessons learned, conduct team dry run of presentation (you may invite instructor to attend), correct presentation and documents
16	Mon 12/09/13		<b>Present</b> to class, customer, and instructor and wider audience <b>at <u>Marksbury building</u></b> (including requirements delivered and lessons learned)	<b>Maintenance assignment due, Deliver</b> product to sponsor and <b>demo</b> to instructor; <b>Turn in presentation, documents, developer notes, peer review form</b>
16	Wed 12/11/13		<b>Present</b> to class, customer, and instructor and wider audience <b>at <u>Marksbury building</u></b> (including requirements delivered and lessons learned)	
16	Fri 12/13/13		<b>Present</b> to class, customer, and instructor and wider audience <b>at <u>Marksbury building</u></b> (including requirements delivered and lessons learned)	

## Project Information

### Grading policies

Students are expected to contribute to their project and meet their obligations in a timely manner. The project will be evaluated based on whether it met requirements and expectations as well as the quality of the process used to carry out the project. The team and individual processes followed will be evaluated using the technical and management documents submitted as well as the individual blog/developer notes maintained by each team member. Academic dishonesty will not be tolerated.

Each aspect of the project grade is described in more detail below.<sup>1</sup>

### Teamwork

<sup>1</sup> \*\*liberal lifting from Portland State University capstone course syllabus – permission of K. Toth sought\*\*\*\*\*

Each member of the team has a responsibility to make sure that the team works. The teamwork grade is assigned based on the ability of the team to meet obligations including meeting with the customer, the instructor, and with each other. All meetings (internal, instructor, and customer) should start with a brief, meaningful **agenda** and proceed through the agenda to resolve issues and assign follow-up actions. Team presentations should have responsibilities clearly allocated and coordinated. Differences of opinion among team members should be rationally and professionally discussed and resolved with minimal conflict. **Failure to attend scheduled meetings** will result in significant grade deductions. If meeting times need to be changed, appropriate reasons and significant lead time to reschedule are required. Customers will evaluate team interactions and attendance at monthly customer meetings. All members of a team will receive the same team grade.

#### Project Processes

Team process quality will be derived from the project artifacts, the agendas for each meeting, the meetings, and the follow-on actions items from each meeting. You will be expected to **produce and post agendas and minutes** (in the form of follow-on action items) for each customer, instructor, and internal team meeting.

#### Project Outcomes

Project outcomes include software delivered to the customer, product demos, presentations, presentation slides (or visual aids), and post-mortem. The customer will be asked to provide feedback on the quality of the delivered software system and demonstrations. The evaluation of presentations will include organization, completeness, and clarity with respect to approach followed, results achieved, challenges/problems encountered, resolution strategies adopted, and lessons you learned from the project.

#### Participation

The participation grade will mainly consider attendance and participation at weekly progress meetings and presentations (individual reporting of issues, progress, and problem resolutions). All team members should get involved and communicate about the project during meetings and presentations. Demonstrated unfamiliarity with lecture materials during a status meeting will be reflected in the student's participation grade. Likewise, consistent failure to attend status meetings or the inability to coherently answer questions about the status of their deliverables will impact this grade. Note that Participation and Teamwork account for a significant portion of a student's grade, so students that "blow off" the project can expect a failing grade.

#### Blog/developer's notes

Software engineers continuously annotate their daily activities, decisions, key facts, references, actions, etc. These developer notes reduce project and company risks, preserve design decisions, support follow-up and team communications, protect against lost knowledge due to employee departures and reassignments, and in some cases are essential to maintaining a company's and/or their own intellectual property rights especially as related to patents.

You will keep developer notes as part of the project. The notes must meet a minimum standard of credibility. The instructor will randomly review developer notes at or before the weekly status meetings. Additionally, at the end of the project each student's notes will be collected or accessed and evaluated by the instructor - and returned after grades are assigned (**online notes are fine**).

#### Course organization

The course will start with lectures and class interaction. You will then form teams and will become increasingly self-reliant, responsible for accomplishing goals of understanding the customer's requirements and producing a feasible project plan. By mid-October, you should have jelled into a highly cooperative group, you should be dealing with the customer like a true client, and you should be seeking the buy-in of the instructor like you would of a senior program manager overseeing the various ongoing projects of a company.

The remaining parts of the course will familiarize you with a corporate team environment, where responsibility for getting things done belongs to the team and its members. You will shift from relying on

the customer and instructor for inputs and guidance to becoming totally responsible for providing progress, visibility, and constructive outcomes to the customer and your senior management. You should be functioning as a self-managed team of professionals distributing responsibilities according to your individual competencies, solving tough problems jointly and constructively, and sharing the work equitably.

#### Role of the Instructor

The instructor can help with customer interactions, labs, other resources, etc. Teams need to interact with the instructor efficiently: being prepared for meetings is one way to do that. It is important to have all team members attend regularly scheduled meetings to avoid miscommunications and ensure all issues are well vetted and understood. If an ad hoc or special meeting is required with little lead time, identify a team representative to interact with the instructor. It is that representative's responsibility to accurately report the conversation to the rest of the team.

#### Role of the customer

The customer is expected to provide the requirements and general scope for the project. The project team is expected to elicit the requirements from the customer, ensure understanding, and review the feasibility and estimate the resources/time schedule of the requirements. Interactions with the customer, whether in person or via email or skype, should be professional. The entire team should attend customer meetings especially during the elicitation phase. Beyond that it may be most efficient to have one team member work with the customer to handle issues as they arise. If there are problems such as infeasible project or scope too large to complete in the allotted time, the team should meet with the customer (in a timely fashion) to negotiate a down scoping and prioritization of the requirements.

#### Progress Review Meetings and Reporting

Students must always be available to meet during this reserved time slot. Class time will be used for both class sessions and individual team meetings. After teams are formed and projects assigned, meeting times will be arranged for each team to meet with the instructor – with preference for this to occur during regular class slots. Based on number of teams, etc., these meetings may need to occur outside of class time. Students must be prepared to meet outside of regular class time with the instructor and customer as required. Students are responsible for planning their activities and scheduling mutually acceptable meeting times to coordinate their joint team efforts, progress reviews with instructor, and requirements elicitation meetings with the customer.

The weekly meeting with the instructor will be roughly a 10 - 15 minute meeting highlighting: progress achieved, problems/issues to be resolved, planned next steps toward completion. From these meetings the instructor will derive team performance; dynamics, effort invested by individual team members, and will ensure steady, consistent progress.

Customer meetings should be driven by the requirements list, issues encountered with requirements, and requirements yet to be completed.

#### Metrics

Teams will provide certain measures related to team member activities (e.g., hours of effort) and artifacts on which the team is working (e.g., lines of code, number of test cases, etc.). Metric collection will be discussed during project management and estimation lectures.

#### Resources

Teams may use computers in Multilab – you will be assigned linux and Windows accounts there. You may also use your own computers. An svn server has also been set up by the department to assist with configuration control of artifacts. Information will be distributed on its use. Other free resources such as utilities, development environment, tools, etc. can be found on the Web. Customers may provide tools or software components. In this case, it is the teams' responsibility to ensure that appropriate licenses and other permissions are in place. On Multilab, installation of pirated, unlicensed or otherwise illegal software is forbidden and will be treated as an act of academic dishonesty.

**Deliverables/artifacts:** 1) Project plan assignment which has business plan (optional), user stories, acceptance test plans, story points estimate/schedule/risk; 2) Architecture assignment which has architecture, design, test plans; 3) Code assignment which has code, user's/admin manual, tests, test results or reports; 4) Maintenance assignment which has updated information from all three prior assignments based on the change request – changes to the deliverables must be noted with change bars, highlight, or such.

### **Accommodations due to disability:**

If you have a documented disability that requires academic accommodations, please see me as soon as possible during scheduled office hours. You must provide me with a Letter of Accommodation from the Disability Resource center (Rm 2, Alumni Gym, 257-2754, email [jkarnes@email.uky.edu](mailto:jkarnes@email.uky.edu)).

**The syllabus is subject to change, and you are responsible for keeping informed of any alterations.**

### **Possible outside readings:**

Case of the Killer robot, <http://www.onlineethics.org/Resources/Cases/killerrobot.aspx>

Software Engineering Code of Ethics and Professional Practice (Version 5.2), ACM/IEEE-CS Joint Task Force on Software, <http://www.acm.org/about/se-code/>

#### Developer notes

What do you use to keep notes as a developer?, <http://stackoverflow.hewgill.com/questions/787/56.html>

Engineering notebook, [http://selab.netlab.uky.edu/homepage/The\\_Engineering\\_Notebook-cs499.pdf](http://selab.netlab.uky.edu/homepage/The_Engineering_Notebook-cs499.pdf)  
Engineering Note Book examples; [www.whrhs.org/.../lib/.../SampleEngineersNotebookEntries.doc](http://www.whrhs.org/.../lib/.../SampleEngineersNotebookEntries.doc)

#### Code walkthrough readings

<http://gfsuite.noaa.gov/developer/CodeWalkthroughGuidelines.html>

<http://www.mit.edu/~mbarker/ideas/checkcode.html>

<http://it.toolbox.com/blogs/enterprise-solutions/systems-development-code-walkthrough-checklist-49283>

<http://www.cs.uky.edu/~raphael/checklist.html>">Helpful information on common programming mistakes

#### Agile readings

Scrum, <http://www.mountangoatsoftware.com/topics/scrum>  
User stories, <http://www.mountangoatsoftware.com/topics/user-stories>

TDD: TDD in a nutshell, <http://ayagebeely.blogspot.com/2010/05/tdd-in-nutshell.html>  
Introduction to TDD, <http://www.agiledata.org/essays/tdd.html>

Paper on agile cost and effort estimation, <http://lens.cos.ufrj.br:8080/eselaw/proceedings/2004/interestedareas/eselaw23>

### Poker planning readings

Planning poker, <http://radio.javaranch.com/lasse/2008/04/22/1208837097457.html>

Planning poker, <http://www.mountangoatsoftware.com/topics/planning-poker>

### Business planning readings

**How to write a winning business plan, <http://www.startups.co.uk/how-to-write-a-winning-business-plan.html>**

Agile Software Startups: The Myth Of The Perfect Business Plan,

<http://onstartups.com/tabid/3339/bid/190/Agile-Software-Startups-The-Myth-Of-The-Perfect-Business-Plan.aspx>

[1] Dr. Judy Goldsmith

[2] <http://www.scs.ryerson.ca/~dwoit/failure.html>.

[3] [www.uky.edu/Ombud/acadoffenses/letterOfWarningExample.doc](http://www.uky.edu/Ombud/acadoffenses/letterOfWarningExample.doc)