Syllabus for CS 499-001 Senior Design Project
Fall 2015

Instructor:

Dr. Jane Hayes (www.cs.uky.edu/~hayes).
Room 228**,** Hardymon Building
Office hours MW 1250-1320 Sloane Research Bldg room 104b (has copier in it) or
by appointment

Course information:

Course homepage http://selab.netlab.uky.edu/homepage/CS499fall15.htm

Course:	CS 499 Senior Design Project
Section	001
Meets:	MWF 1200-1250
Location:	303-SRB - Sloane Research Bldg

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. This course provides full GCCR credit for the Computer Science major and for Computer Engineering students who opt to take this capstone course.

Course Outcomes:

Students will gain experience in the design and implementation process using material from throughout their undergraduate career. They will gain experience working in groups. Specifically, students will improve their abilities, knowledge, understanding and skills to:

1. Design, implement and evaluate a computer-based system, process, component, or program to meet desired needs
2. Professional, ethical, legal, security, and social issues and responsibilities
3. Use the standard project development steps (specification, design, etc.) in implementing a project
4. Implement a large project
5. Communicate effectively with a range of audiences
6. Develop and present a talk on the status of a project
7. Develop a written report on a large project
8. Function effectively on teams to accomplish a common goal

TCE Questions

37 – This course improved my ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs
38 – This course has improved my ability to understand professional, ethical, legal, security, and social issues and responsibilities
39 – This course has improved my ability to use the standard project development steps (specification, design, etc.) in implementing a project
40 – This course has improved my ability to implement a large project
41 – This course has improved my ability to communicate effectively with a range of audiences
42 – This course has improved my ability to develop and present a talk on the status of a project

43 – This course has improved my ability to function effectively on teams to accomplish a common goal

Course Materials:

Recommended Texts:

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

Craig Larman
Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process, 2/E
ISBN-13: 9780130925695
http://www.utdallas.edu/~chung/SP/applying-uml-and-patterns.pdf

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

Martin Fowler
UML distilled: a brief guide to the standard object modeling language (3rd edition), Addison-Wesley, ISBN-10:0321193687

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.

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	50%
Project documentation/deliverables	20%
Assignments	10%
Class participation	10%
Attendance	10%

Where:

A=	92 - 100%
B =	83 - 91%
C=	74 - 82%
D=	65 - 73%
F =	64 and below

There is no final exam.

Note on GCCR: To satisfy the CCCR requirements the student has to attain at least an equivalent of a C grade for the communication and composition components of the required work. Specific requirements related to the GCCR are described in section "Overview of CS 499 as a GCCR course" of this document.

Mid-term grades will be posted in myUK by the deadline established in the Academic Calendar (**http://www.uky.edu/Registrar/AcademicCalendar.htm**)

Project Grade Criteria (50% of grade)

100 points

Achieve all 7 of these items

1. Good teamwork and approach to the project, kept to a schedule
2. Communicated well with customer
3. Kept up a professional looking web page
4. Used software engineering procedures that would discover problems as soon as possible, used all resources available to solve problems
5. Instructor has no substantial criticism of the project
6. Appropriate, understandable documentation
7. Went beyond the project specifications in some way

96 Same as 100, except for item 7

92 Same as 96, but with one or two minor negative comments from instructor

86 More than two negative minor comments, or a major problem

82 Partial project failure due to problems that could have been corrected by the team by following the items above

75 Project failed in some way due to problems that could have been corrected by the team by following the items above

70 Project failed, poor teamwork, poor procedures, many of the above items not followed

65 Project failed, most items not followed, lack of effort

Project grades are based on the judgment of the instructor on how well the projects met the above criteria. Note that the use of a version control system (CVS and SVN are examples) for the project will be required.

Project Documentation/Deliverables (20% of grade)

Good documentation is essential for a successful project. These are the items that will be graded:

Final documentation	6%
Team web page	6%
Code documentation	6%
CD of deliverable	2%

Standards for the final project documentation will be provided on the class web site. An example of a team web page will be provided. The instructor will warn teams that have inadequate web pages, or do not keep them current. If the deficiencies continue, points will be deducted. Your source code must follow product documentation standards. Your CD of deliverables must be well organized, and include a table of contents in a "readme" file. Proper language usage is required for all written material.

Note that all students on a project may not receive the same project grade (including documentation grade). At each project phase, each team member submits a report on what each team member did on the project, how much they cooperated on the project tasks, attended meetings, etc. The instructor will also make a judgment on each student's team participation based on:

- Class attendance
- Team meeting attendance
- Participation in team class presentations
- Participation in team meetings with the instructor
- Project knowledge as shown in presentations and meetings

If (in the judgment of the instructor based on the input from the team members and the observation of the instructor) it is clear that a team member has made insufficient contribution to the project, that student's grade (for the project and project documentation/deliverables) will be adjusted.

Assignments (10% of grade)

There will be a few written assignments during the semester to be done individually.

Class Participation/Web page (10% of grade)

Students are expected to participate in their team's class presentations, and keep a log of the student's project activities updated on the project web page. Each team presents its project to the class several times throughout the semester. During the semester, the teams present the project status to the instructor. All team members are expected to participate in these presentations. "Participation" means not only being present, but also understanding the project, and being able to answer questions about it. The instructor will judge, and can lower the participation grade of students who, in the judgment of the instructor, do not understand the project.

The grading for class participation:

Participated in team project midterm presentation	3%
Participated in team project final presentation	3%
Participated in meetings with the instructor	2%
Kept project web page log of student's activities up to date on a weekly basis	2%

Attendance (10% of grade)

Students are expected to attend and participate in all scheduled classes. An attendance sheet will be used. **Attendance for presentations from guest speakers from outside the university and team project presentations to the class will count as two attendances.** The dates of the invited guest speakers' presentations will be announced in class and put on the class web page.

Students can be excused for University accepted 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 contact the instructor regarding the nature of the absence, and the instructor retains the right to ask for proof.

Students anticipating an absence for a major religious holiday are responsible for notifying the instructor in writing of anticipated absences due to their observance of such holidays no later than the last day in the semester to add a class. Information regarding dates of major religious holidays may be obtained through the religious liaison, Mr. Jake Karnes (859-257-2754).

Students are expected to withdraw from the class if more than 20% of the classes scheduled for the semester are missed (excused or unexcused) per university policy.

Incomplete grade: Because of the team project nature of the class, the grade of incomplete (I) will be given only in cases of extreme hardship in accordance with the University policy.

See class schedule for deadlines with respect to specific assignments/deliverables.

Overview of CS 499 as a GCCR course:

CS 499 Senior Design Project provides full GCCR credit for the Computer Science major, and for Computer Engineering students who opt to take this capstone course instead of its EE counterpart. Check with your advisor and course instructor for more information.

There are three major components of the GCCR part of the CS 499 course. They constitute about 40% of the final grade of the course:

(a) Multiple written assignments (reports, documentation, user manuals, etc.) that total to at least 4,500 words of text (see details below) for the total of 150 points for the communication and composition.
Deliverables: written notes, reports, and technical documents.

(b) Oral assignments in English, in which teams of students give a formal presentation with at least 10 minutes long presentation by each student. There will be three presentations. The total score for the communication aspects of the class is 75 points.
Deliverables: Two PowerPoint oral presentations.

(c) *Software requirements specification* document for the software project, which requires the student to demonstrate information literacy in the discipline. The total score for this project is 75 points.
Deliverables: a formal document following the domain-specific technical writing and format.

Grading for the GCCR requirement: A: 265 – 300 point; B: 225 – 264 points, C: 185 – 224 point; D: 140 – 184 points; E otherwise.

Important: To satisfy the CCCR requirements the student has to attain at least an equivalent of a C grade for the communication and composition components of the required work.

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 specified in the project documentation and agreed to by the customer.

Group Projects:

The group project for the course will require you to work together with other students in the class; typically there are three or four students in each group. Each project will have a web page to be maintained by the project team. The contents of the web page will be discussed in class and the class web site. You will be evaluated on your individual contribution to the group project and presentations of the project results. The instructor reserves the right to 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:

You are expected to attend class **every day**, with the exception of project meeting days that do not involve your team. This will be determined later in the semester, probably in mid-September.

Week	**Date**	**Readings**	**Topics**	**Project**
1	Wed 8/26/15		Software engineering, syllabus	Scrum process memo assignment handed out, due 12/7
1	Fri 8/28/15	Developer notes/Engineering notebook reading	**Customer presentations,** explain project assignment	Project plan assignment handed out, due 9/21
2	Mon 8/31/15	Requirements, story points readings	Requirements/user stories/story points; **Team building exercise**	Teams formed by start of class, Teams bid on projects, Selections emailed to teams, Reqts elicitation/use case modeling assignment handed out, due 9/11
2	Wed 9/02/15	Business plan reading	Career Center Visitor; Entrepreneurship, pitch contest, business plan presentation – Brian Raney	Teams hold internal team meetings, meet with sponsors to establish requirements, start to develop user stories
2	Fri 9/04/15	V&V, agile, project management readings	Technical writing visitor; Project mgmt./verification and validation/agile process	Estimating projects memo assignment handed out, due 9/15
3	Mon 9/07/15	**Labor day**	**No class**	
3	Wed 9/09/15	Architecture, TDD, design readings	Architecture, design, test driven development , go over requirements [in class activity]	**Resumes due**
3	Fri 9/11/15	Poker planning readings	Poker planning, stand up meeting [in class activity]	Internal team meetings, meet with sponsors to establish requirements, work on project plan, **Reqts elicitation/use case modeling assignment due**, Config. control/svn

				memo assignment handed out, due 9/25
4	Mon 9/14/15	Design, CM, tracing readings	Design, configuration control, tracing	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 9/16/15	Design readings	Design [in class activity]	Team progress reviews with instructor, internal team meetings, meet with sponsors to establish requirements, develop project plan, start to set up development environment, **Estimating projects memo assignment due**
4	Fri 9/18/15	Unit testing readings	Unit testing [in class activity]	Team progress reviews with instructor, internal team meetings, meet with sponsors to establish requirements, develop project plan, start to set up development environment
5	Mon 9/21/15	Code walkthrough readings	Code walkthroughs [in class activity], explain project assignment	**Project plan assignment due**, **Turn in developer notes, documents, peer review form;** internal team meetings, meet with sponsors to establish acceptance test plans, set up development environment; hand out architecture assignment
5	Wed 9/23/15		Project meetings	Internal team meetings, sponsor

				meetings, work on architecture assignment, continue setting up development environment, TDD/refactoring assignment handed out, due 9/30
5	Fri 9/25/15		Project meetings, IPA visitor	Internal team meetings, sponsor meetings, work on architecture assignment, continue setting up development environment, **Config. control/svn memo assignment due**
6	Mon 9/28/15		Project meetings	Internal team meetings, sponsor meetings, work on architecture assignment, continue setting up development environment
6	Wed 9/30/15		Project meetings	Internal team meetings, sponsor meetings, work on architecture assignment, continue setting up development environment, **TDD/refactoring assignment due**
6	Fri 10/02/15		Sponsor 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
7	Mon		**Present** to class, customer, and	**Architecture**

	10/05/15		instructor Explain project assignment	**assignment due**; **Turn in presentation, developer notes, documents, peer review form**; hand out coding assignment, start of SCRUM CYCLE 1
7	Wed 10/07/15		**Present** to class, customer, and instructor	
7	Fri 10/09/15		**Present** to class, customer, and instructor	
8	Mon 10/12/15		Project meetings, IPA visitor	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development, Ethics short essay assignment handed out, draft due 10/16
8	Wed 10/14/15	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 10/16/15	Code of conduct, assigned ethics readings	Ethics and code of conduct [in class activity] – draft/feedback/revision	**Ethics short essay assignment due**
9	Mon 10/19/15	Midterm point of the semester	Project meetings	END OF SCRUM CYCLE 1 Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development
9	Wed 10/21/15		Project meetings	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development
9	Fri 10/23/15		Project meetings	Progress review meetings with

				instructor, internal team meetings, meet with sponsors as required, ongoing software development
10	Mon 10/26/15	~~Maintenance readings~~	~~Maintenance [in class activity]~~ Project meetings	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development
10	Wed 10/28/15		Project meetings	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development
10	Fri 10/30/15		Project meetings, ~~IPA visitor~~	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development
11	Mon 11/02/15	Maintenance readings	Maintenance [in class activity] ~~Catch up, if needed~~	END OF SCRUM CYCLE 2 **Scrum process assignment due** Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development
11	Wed 11/04/15		Catch up, if needed	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development
11	Fri 11/06/15		Project meetings	Progress review meetings with instructor, internal team meetings, meet with sponsors as

				required, ongoing software development
12	Mon 11/09/15		Project meetings	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development
12	Wed 11/11/15		Project meetings	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development
12	Fri 11/13/15		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
13	Mon 11/16/15		**Present** to class, customer, and instructor Explain project assignment	**END OF SCRUM CYCLE 3** **Code assignment due, Deliver** product to sponsor and **demo** to instructor, **Turn in presentation, developer notes, documents, peer review form;** hand out maintenance assignment, start SCRUM CYCLE 4
13	Wed 11/18/15		**Present** to class, customer, and instructor	Elicit change request from customer, start working on change
13	Fri 11/20/15		**Present** to class, customer, and instructor	Progress review meetings with

				instructor, internal team meetings, meet with sponsors as required, ongoing software development
14	Mon 11/23/15		Project meetings	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development
14	Wed 11/25/15, Fri 11/27/15	***No class***	***Fall break - Thanksgiving***	
15	Mon 11/30/15		Project meetings	Internal team meetings, meet with sponsors as required, ongoing software development
15	Wed 12/02/15		Project meetings	Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software development
15	Fri 12/04/15		Project meetings	**Senior surveys due,** Progress review meetings with instructor, internal team meetings, meet with sponsors as required, ongoing software 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 12/07/15		**Present** to class, customer, and instructor and wider audience **at Marksbury building** (including	END OF SCRUM CYCLE 4 **Maintenance**

			requirements delivered and lessons learned)	**assignment due, Deliver** product to sponsor and **demo** to instructor; **Turn in presentation, documents, developer notes, peer review form**
16	Wed 12/09/15		**Present** to class, customer, and instructor and wider audience **at Marksbury building** (including requirements delivered and lessons learned)	
16	Fri 12/11/15		**Present** to class, customer, and instructor and wider audience **at Marksbury building** (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.[1]

Teamwork

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

[1] **liberal lifting from Portland State University capstone course syllabus – permission of K. Toth sought*****

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. Every 499 student is expected to carefully review and follow the developer note guidelines for this course. The instructor will randomly review developer notes at the weekly status meetings. Additionally, at the end of the project each student's notes will be collected and evaluated by the instructor and returned after grades are assigned **(online notebooks are fine (teams can consider Google Docs, Trac, Trello, or some other means of internal communication/collaboration if online)).**

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 5-10 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, 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, test 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).

When unforeseen circumstances necessitate changes, they will be negotiated with the class and reflect the nature of the circumstances necessitating the change. These changes will be e-mailed, announced in class, and noted in the web page version of the syllabus with change bars.

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/

Engineering notebook, http://selab.netlab.uky.edu/homepage/The Engineering Notebook-cs499.pdf
ENB examples; www.whrhs.org/.../lib/.../**SampleEngineersNotebook**Entries.doc

Code walkthrough readings

http://gfesuite.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.mountaingoatsoftware.com/topics/scrum
User stories, http://www.mountaingoatsoftware.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.mountaingoatsoftware.com/topics/planning-poker

Barry W. Boehm, Software Engineering, IEEE Trans. On Computers, 25(12):1226-1241, 19. – see course web page

Boehm, B. A Spiral Model for Software Development and Enhancement, Computer, Vol. 21, no. 5, May '88, pp. 61-72. - see course web page

Parnas, D.L., On criteria to be used in decomposing systems into modules, CACM, vol. 15, no. 12, April '72, pp.1053-1058. http://www.cs.umd.edu/class/spring2003/cmsc838p/Design/criteria.pdf

Wirth, N. Program development by stepwise refinement, CACM, vol. 14, no. 4, 1971, pp. 221-227. http://sunnyday.mit.edu/16.355/wirth-refinement.html

Musa, J.D., and Ackerman, A.F., Quantifying software validation: when to stop testing? IEEE SW, May 1989, pp. 19-27. - see course web page

Chidamber, S.R. and C.F. Kemerer, A metrics suite for object-oriented design, IEEE TSE, vol. SE-20, no. 6, June '94, pp.476-493. http://portal.acm.org/citation.cfm?id=631131

Kiczales, G., Lamping, J. , Mendhekar, A., Maeda, C., Lopes, C.V., Loingtier, J.-M., and Irwin, J. Aspect--Oriented Programming. In European Conference on Object--Oriented Programming, ECOOP'97, LNCS 1241, pages 220--242, Finland, June 1997. Springer--Verlag. http://www2.parc.com/csl/groups/sda/publications/papers/Kiczales-ECOOP97/for-web.pdf

[1] Dr. Judy Goldsmith
[2] http://www.scs.ryerson.ca/~dwoit/failure.html.
[3] www.uky.edu/Ombud/acadoffenses/letterOfWarningExample.doc