UNIVERSITY OF KENTUCKY

New Course Report

5/18/2015 7:37:01 AM

1. General Information

1a. Submitted by the College of: ENGINEERING

Date Submitted: 5/13/2015

1b. Department/Division: Computer Science

1c. Contact Person

Name: Jane Hayes

Email: hayes@cs.uky.edu

Phone: 2573171

Responsible Faculty ID (if different from Contact)

Name:

Email:

Phone:

1d. Requested Effective Date: Semester following approval

1e. Should this course be a UK Core Course? No

2. Designation and Description of Proposed Course

2a. Will this course also be offered through Distance Learning?: No

2b. Prefix and Number: CS 498

2c. Full Title: Software Engineering for Senior Project

2d. Transcript Title: Software Engineering for Senior Project

2e. Cross-listing:

2f. Meeting Patterns

LECTURE: 3

2g. Grading System: Letter (A, B, C, etc.)

2h. Number of credit hours: 3

2i. Is this course repeatable for additional credit? No

If Yes: Maximum number of credit hours:

If Yes: Will this course allow multiple registrations during the same semester?

2j. Course Description for Bulletin: Current approaches -- practice and technologies -- for developing reliable software: specifications, testing, and verification. Individual and team assignments focused on applying these approaches to software systems. A significant communication and composition component related to specifying, designing, presenting, and documenting software systems.

2k. Prerequisites, if any: CS 115, 215, 216

2l. Supplementary Teaching Component:

3. Will this course taught off campus? No

If YES, enter the off campus address:

4. Frequency of Course Offering: Spring,

Will the course be offered every year?: Yes

If No, explain:

5. Are facilities and personnel necessary for the proposed new course available?: Yes

If No, explain:

6. What enrollment (per section per semester) may reasonably be expected?: 25

7. Anticipated Student Demand

Will this course serve students primarily within the degree program?: Yes

Will it be of interest to a significant number of students outside the degree pgm?: Yes

If Yes, explain: Computer engineering majors may find it of interest or any engineering major who would like to learn more about software engineering.

8. Check the category most applicable to this course: Traditional – Offered in Corresponding Departments at Universities Elsewhere,

If No, explain:

9. Course Relationship to Program(s).

a. Is this course part of a proposed new program?: No

If YES, name the proposed new program:

b. Will this course be a new requirement for ANY program?: Yes

If YES, list affected programs: Computer Science, it will be a pre-requisite for CS 499, Senior Design Project

10. Information to be Placed on Syllabus.

a. Is the course 400G or 500?: No

b. The syllabus, including course description, student learning outcomes, and grading policies (and 400G-/500-level grading differentiation if applicable, from **10.a** above) are attached: Yes

Distance Learning Form

Instructor Name:

Instructor Email:

Internet/Web-based: No

Interactive Video: No

Hybrid: No

1.How does this course provide for timely and appropriate interaction between students and faculty and among students? Does the course syllabus conform to University Senate Syllabus Guidelines, specifically the Distance Learning Considerations?

2.How do you ensure that the experience for a DL student is comparable to that of a classroom-based student's experience? Aspects to explore: textbooks, course goals, assessment of student learning outcomes, etc.

3.How is the integrity of student work ensured? Please speak to aspects such as password-protected course portals, proctors for exams at interactive video sites; academic offense policy; etc.

4.Will offering this course via DL result in at least 25% or at least 50% (based on total credit hours required for completion) of a degree program being offered via any form of DL, as defined above?

If yes, which percentage, and which program(s)?

5.How are students taking the course via DL assured of equivalent access to student services, similar to that of a student taking the class in a traditional classroom setting?

6.How do course requirements ensure that students make appropriate use of learning resources?

7.Please explain specifically how access is provided to laboratories, facilities, and equipment appropriate to the course or program.

8.How are students informed of procedures for resolving technical complaints? Does the syllabus list the entities available to offer technical help with the delivery and/or receipt of the course, such as the Information Technology Customer Service Center (http://www.uky.edu/UKIT/)?

9.Will the course be delivered via services available through the Distance Learning Program (DLP) and the Academic Technology Group (ATL)? NO

If no, explain how student enrolled in DL courses are able to use the technology employed, as well as how students will be provided with assistance in using said technology.

10.Does the syllabus contain all the required components? NO

11.I, the instructor of record, have read and understood all of the university-level statements regarding DL.

Instructor Name:

SIGNATURE|SEALES|William B Seales|CS 498 NEW Dept Review|20140331

SIGNATURE|BJSTOK0|Barbara J Brandenburg|CS 498 NEW College Review|20140421

SIGNATURE|SEALES|William B Seales|CS 498 ZCOURSE_NEW Approval Returned to Dept|20140421

SIGNATURE|BJSTOK0|Barbara J Brandenburg|CS 498 NEW College Review|20141021

SIGNATURE|JMETT2|Joanie Ett-Mims|CS 498 NEW Undergrad Council Review|20150512

SIGNATURE|SEALES|William B Seales|CS 498 ZCOURSE_NEW Approval Returned to Dept|20150513

SIGNATURE|JMETT2|Joanie Ett-Mims|CS 498 NEW Undergrad Council Review|20150514

New Course Form

https://myuk.uky.edu/sap/bc/soap/rfc?services=

Open in full window to print or save

Generate R

Attachments:

Browse... Upload File

	ID	Attachment
Delete	2456	CS498-Faculty-Approval-2013.pdf
Delete	3100	example homework with rubric.html
Delete	5041	Syllabus-CS498-001-SW-Eng-Senior-Design.pdf

First 1 Last

(*denotes required fields)

1. **General Information**

a. * Submitted by the College of: ENGINEERING Submission Date: 5/13/2015

b. * Department/Division: Computer Science

c.
* Contact Person Name: Jane Hayes Email: hayes@cs.uky.edu Phone: 2573171
* Responsible Faculty ID (if different from Contact) Email: Phone:

d. * Requested Effective Date: ● Semester following approval OR ○ Specific Term/Year [1]

e.
Should this course be a UK Core Course? ○ Yes ● No
If YES, check the areas that apply:

□ Inquiry - Arts & Creativity
□ Inquiry - Humanities
□ Inquiry - Nat/Math/Phys Sci
□ Inquiry - Social Sciences
□ Composition & Communications - I

□ Composition & Communications - II
□ Quantitative Foundations
□ Statistical Inferential Reasoning
□ U.S. Citizenship, Community, Diversity
□ Global Dynamics

2. **Designation and Description of Proposed Course.**

a. * Will this course also be offered through Distance Learning? ○ Yes [4] ● No

b. * Prefix and Number: CS 498

c. * Full Title: Software Engineering for Senior Project

d. Transcript Title (if full title is more than 40 characters): Software Engineering for Senior Project

e. To be Cross-Listed [2] with (Prefix and Number):

f. * Courses must be described by at least one of the meeting patterns below. Include number of actual contact hours[3] for each meeting pattern type.

3	Lecture		Laboratory[1]		Recitation		Discussion
	Indep. Study		Clinical		Colloquium		Practicum
	Research		Residency		Seminar		Studio
	Other	If Other, Please explain:					

g. * Identify a grading system:
● Letter (A, B, C, etc.)
○ Pass/Fail
○ Medicine Numeric Grade (Non-medical students will receive a letter grade)
○ Graduate School Grade Scale

h. * Number of credits: 3

i. * Is this course repeatable for additional credit? ○ Yes ● No
If YES: Maximum number of credit hours:
If YES: Will this course allow multiple registrations during the same semester? ○ Yes ○ No

j. * Course Description for Bulletin:

Current approaches -- practice and technologies -- for developing reliable software: specifications, testing, and verification. Individual and team assignments focused on applying these approaches to software systems. A significant communication and composition component related to specifying, designing, presenting, and documenting software systems.

k. Prerequisites, if any:

CS 115, 215, 216

l. Supplementary teaching component, if any: ○ Community-Based Experience ○ Service Learning ○ Both

3. *** Will this course be taught off campus?** ○ Yes ● No

If YES, enter the off campus address:

4. **Frequency of Course Offering.**

a. * Course will be offered (check all that apply): ☑ Fall ☑ Spring □ Summer □ Winter

b. * Will the course be offered every year? ● Yes ○ No

If No, explain:

5. *** Are facilities and personnel necessary for the proposed new course available?** ● Yes ○ No

If No, explain:

6. *** What enrollment (per section per semester) may reasonably be expected?** 25

7. **Anticipated Student Demand.**

a. * Will this course serve students primarily within the degree program? ● Yes ○ No

b. * Will it be of interest to a significant number of students outside the degree pgm? ● Yes ○ No

If YES, explain:

Computer engineering majors may find it of interest or any engineering major who would like to learn more about software engineering.

8. *** Check the category most applicable to this course:**

☑ Traditional – Offered in Corresponding Departments at Universities Elsewhere
□ Relatively New – Now Being Widely Established
□ Not Yet Found in Many (or Any) Other Universities

9. **Course Relationship to Program(s).**

a. * Is this course part of a proposed new program? ○ Yes ● No

If YES, name the proposed new program:

b. * Will this course be a new requirement [5]for ANY program? ● Yes ○ No

If YES [5], list affected programs::

Computer Science, it will be a pre-requisite for CS 499, Senior Design Project

10. **Information to be Placed on Syllabus.**

a. * Is the course 400G or 500? ○ Yes ● No

If YES, the *differentiation for undergraduate and graduate students must be included* in the information required in **10.b**. You must include: (i) identi additional assignments by the graduate students; and/or (ii) establishment of different grading criteria in the course for graduate students. (See SR

b. ☑ * The syllabus, including course description, student learning outcomes, and grading policies (and 400G-/500-level grading differentiation if appl **10.a** above) are attached.

[1] Courses are typically made effective for the semester following approval. No course will be made effective until all approvals are received.

[2] The chair of the cross-listing department must sign off on the Signature Routing Log.

[3] In general, undergraduate courses are developed on the principle that one semester hour of credit represents one hour of classroom meeting per week for a semester, exclusive of any laboratory meeting. Laboratory meeting, generally, re two hours per week for a semester for one credit hour. (from SR 5.2.1)

[4] You must also submit the Distance Learning Form in order for the proposed course to be considered for DL delivery.

[5] In order to change a program, a program change form must also be submitted.

Rev 8/09

UK
UNIVERSITY OF
KENTUCKY

College of Engineering
Department of Computer Science
329 Rose Street
Davis Marksbury Building
Lexington, KY 40506-0633

859 257-3961

www.cs.uky.edu

October 15, 2013

MEMORANDUM

TO: Jane Hayes
FROM: Brent Seales WBS
SUBJECT: New Course Proposal CS 498

The new course proposal for CS 498 Software Engineering Senior Project was vetted by the Computer Science faculty on Monday, 14 October 2013. After discussion by the faculty, the proposal was approved by unanimous vote for submission to the next level in the approval process.

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

10/30/2013	Mobile - client-side	
11/1/2013	Mobile - server-side	
11/4/2013	Mobile - integration	
11/6/2013	**Wrap up**	**Homework 4 due**
11/8/2013	Code walkthrough	**Homework 5 assigned**
11/11/2013	Code review	
11/13/2013	Unit testing	
11/15/2013	Exploratory testing	
11/18/2013	Bug reporting	
11/20/2013	Code coverage	
11/22/2013	Integration testing	
11/25/2013	System testing	
11/27/2013	**Holiday**	
11/29/2013	**Holiday**	
12/2/2013	**Wrap up**	**Homework 5 due**
12/4/2013	Maintenance	**Homework 6 assigned**
12/6/2013	Regression testing	
12/9/2013	Debugging	
12/11/2013	Debugging	
12/16/2013 @ 10:30	**Final exam**	Location TBD

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:

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

8/30/2013	**Wrap up**	**Homework 0 due**
9/2/2013	**Holiday**	
9/4/2013	Team process models	
9/6/2013	Requirements	
9/9/2013	Requirements elicitation	
9/11/2013	Requirements elicitation	**Homework 1 assigned**
9/13/2013	Requirements representations	
9/16/2013	Requirements validation	
9/18/2013	**Wrap up**	**Homework 1 due**
9/20/2013	Project planning	**Homework 2 assigned**
9/23/2013	Effort estimation	
9/25/2013	Fast prototyping	
9/27/2013	OO design patterns	
9/30/2013	OO design patterns	
10/2/2013	Web design patterns	
10/4/2013	Design validation	
10/7/2013	**Midterm exam**	**Homework 2 due**
10/9/2013	Version control, basic	**Homework 3 assigned**
10/11/2013	Version control, advanced	
10/14/2013	Coding styles	
10/16/2013	Documentation	
10/18/2013	OO	
10/21/2013	MVC	
10/23/2013	**Wrap up**	**Homework 3 due**
10/25/2013	Web services	**Homework 4 assigned**
10/28/2013	Mobile - intro	

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:

It is possible that you will work in groups on some assignments/projects for this course. If so, you will be evaluated on your contribution to the group project and presentations of the project results. The instructor may 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. As part of the project, written reports will be required. **Proper language usage is required.**

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).

Schedule:

Date	Topic	Notes
8/28/2013	Orientation, s/w engineering	**Homework 0 assigned**

Presentations:	15%
Assignments:	40%
Mid-term exam:	12%
Quizzes:	10%
Final exam:	13%

Where:

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

There will be several quizzes given. The dates are listed in the schedule below, though "pop" quizzes (not previously announced) may also be given.

Whining Lowers 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. 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

39 - This course improved my ability to function effectively on teams to accomplish a common goal.

40 - This course improved my understanding of professional, ethical, legal, security, and social issues and responsibilities.

41 - This course improved my ability to communicate effectively with a range of audiences.

42 - This course improved my ability to analyze the local and global impact of computing on individuals, organizations and society.

43 - This course improved my recognition of the need for, and an ability to engage in, continuing professional development.

44 - This course improved my ability to use current techniques, skills, and tools necessary for computing practices.

45 - This course improved my ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.

46- This course improved my ability to apply design and development principles in the construction of software systems of varying complexity.

Course Materials:

Required Text:
Shari Lawrence Pfleeger and Joanne M. Atlee
Software Engineering: Theory and Practice, **** Fourth Edition****
Prentice Hall
ISBN: 0136061699
You must obtain a copy of this text.

Recommended Texts:
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. Also, copies have been placed on reserve in the Engineering Library (3rd 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 498 will be determined according to these weights:

Attendance and participation:	10%

Syllabus for CS 498 001 Software Engineering for Senior Project

Instructor:

Dr. Jane Hayes (www.cs.uky.edu/~hayes)
Room 233, Hardymon Building
Office hours: In Robotics (CRMS) Bldg, Room 514D on T from 1020 – 1050am and R from 1345 - 1415 or by appointment

Dr. Ashlee Holbrook	ashleeh@gmail.com
Mr. Mark Hays	mahays0@engr.uky.edu

Course information:

Course homepage http://selab.netlab.uky.edu/homepage/cs485-004spring2012.htm

Course:	CS 498 Software Engineering for Senior Project
Section	001
Meets:	TR 1100 - 1215
Location:	Dickey Hall Rm B57 ****(Jan. 12 we will meet in Marksbury bldg. Theater)****

Description:

Current approaches - - practice and technologies - - for developing reliable software: specifications, testing, and verification. Individual and team assignments focused on applying these approaches to software systems. A significant communication and composition component related to specifying, designing, presenting, and documenting software systems.

Pre-requisites: CS 216 (which requires CS 115 and CS 215)

Course Outcomes:

(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;
(c) An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;
(d) An ability to function effectively on teams to accomplish a common goal;
(e) An understanding of professional, ethical, legal, security, and social issues and responsibilities;
(f) An ability to communicate effectively with a range of audiences;
(g) An ability to analyze the local and global impact of computing on individuals, organizations and society;
(h) Recognition of the need for, and an ability to engage in, continuing professional development;
(i) An ability to use current techniques, skills, and tools necessary for computing practices;
(j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;
(k) An ability to apply design and development principles in the construction of software systems of varying complexity.

TCE Questions
37 - This course improved my ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

38 - This course improved my ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs.

Conversing With The Client: Preparing, Summarizing, Reflecting

Purpose. The role of Product Owner requires you to keep in touch with the client, informing him/her of the project status and communicating changes in requirements back to your team. This kind of communication can be challenging, for a couple of reasons. First, the client lies outside of your world as a Software Engineering student; there are things that are familiar to you that your client knows nothing about, and vice versa. You need to establish a shared body of knowledge with your client. Second, your client is most likely a busy person, and you should make the most of your limited "face time". So you need to think not only about what topics to cover, but how to cover them in a quick and effective way.

This assignment centers around your first meeting with the client: in particular, the meeting where you first get into the backlog of user stories for the project. Obviously, one outcome is the creation or modification of a prioritized set of user stories. But you will also do some self-inquiry and reflection exercises to help make your client communication as good as it can be.

Every conversation is **situated** in a particular location, at a particular time, with particular participants with particular attitudes, styles, etc. These situated aspects don't really have much to do with the actual content of the conversation, but they can have a great effect. Here are some examples:

- Where is the conversation located? What are the seating arrangements? Can each of you see what the other is doing clearly? Or perhaps the meeting is not face-to-face - what are the limitations of communicating via phone or computer?
- Is the conversation centered around shared tools, like a whiteboard, a paper, a computer? Who is in control of the tools: for instance, if you're using a computer, who is doing the "steering" (manipulating the mouse and keyboard)?
- What are the identities of the participants? Do you perceive some as having greater authority, or greater power, than others?
- What is the style of the conversation? Is is following a set agenda, or is it more freeform? Is technical jargon used?
- What is the mood? Does the client seem serious, humorous, distracted, impatient? How about *you*?

We will observe these "details" and consider their consequences.

Outcomes. In this assignment you will get experience in the following skills:

1. assessing the client's needs and knowledge in advance, and identifying the *content* of the upcoming conversation: what to inform and ask the client about;
2. identifying the situated aspects of a conversation (outside of the actual verbal content) that can affect the value of the conversation;
3. reading written reports from previous Product Owners, and using the information in them to improve your own Product Owner performance;
4. creating or modifying a prioritized backlog of user stories;
5. writing a detailed narrative of a conversational situation (in this case, the conversation between client and Product Owner);
6. critiquing the actions described in the narrative, proposing suggestions for how to improve them.

There are five parts to this assignment: two to be completed **before meeting** the client for the first time, and the other three to be completed **immediately after the meeting** (the same day).

A challenging part of this assignment: How do you conduct a productive meeting *and* pay attention to the communication details of the conversation? **Don't try to do both without some help.** Possible solutions: Record the conversation (with permission from the client); bring a teammate along to take notes on the communication aspects, then compare notes after the conversation.

Please **read the entire assignment before starting it.**

Part 1 (before meeting). Time: 30 minutes.

Write answers to the following:

1. Give three things you know about your client. (If you don't know three things, go out and *learn* three.)
2. Give three things you would like to *learn* about your client during your meeting.
3. Give three things related to the project that your client (probably) knows and you would *like* to know.
4. Give three things related to the project that you know and your client would (probably) like to know.
5. Give three specific ways in which the above information will affect the way you conduct the meeting.

Part 2 (before meeting). Time: 30 minutes.

Read the two most recent instances of this assignment, submitted by previous Product Owners. Write answers to the following:

1. Given their experiences, what will you do differently in your experience with the client? Why?
2. What will you keep the same? Why?

Part 3 (after meeting). Time: 30 minutes.

Based on your conversation with the client, create a product backlog for the project (or, if one exists already, update it).

Some background on "backlog": In each sprint, the team chooses a set of requirements and then sets to work fulfilling them. The requirements are chosen from a preexisting product *backlog*, maintained by the Product owner. To kickstart the whole process, the product backlog must be populated with requirements, expressed as *stories*. To do this, the Product owner first consults with the client, to get a vision of the ultimate purpose of the product and a set of prioritized features and time constraints. The Product owner takes this information to the team, who then break it down into a backlog of feasible items, along with any necessary technical requirements that are outside the purview of the client. For backlog items to be feasible, they must be readily *estimable* and *testable*: each item must have an estimated amount of effort necessary to complete it, and a clear means to test whether the item has been completed successfully. Once the process is underway, the Product owner keeps the backlog up to date through communication with the client.

Part 4 (after meeting). Time: 30 minutes.

If you look at a written description of a play or film, you will usually find stage directions (describing where, when and how the action takes place) interleaved with the dialogue. Here's a small example.

> *(Oscar Wilde, The Importance of Being Earnest, Act II, Part 1)*
>
> Garden at the Manor House. A flight of grey stone steps leads up to the house. The garden, an old-fashioned one, full of roses. Time of year, July. Basket chairs, and a table covered with books, are set under a large yew-tree.
>
> [**Miss Prism** discovered seated at the table. **Cecily** is at the back watering flowers.]
>
> **Miss Prism.** [Calling.] Cecily, Cecily! Surely such a utilitarian occupation as the watering of flowers is rather Moulton's duty than yours? Especially at a moment when intellectual pleasures await you. Your German grammar is on the table. Pray open it at page fifteen. We will repeat yesterday's lesson.
>
> **Cecily.** [Coming over very slowly.] But I don't like German. It isn't at all a becoming language. I know perfectly well that I look quite plain after my German lesson.

Describe your meeting with the client in a similar fashion. Don't try to include all of the text word-for-word; you can just roughly summarize it. Try to capture as much of the stage directions as you can: the setup of the location, where you sat (or stood), what kinds of objects were used, the speaking style, etc.

Note: The client will **not** see any of the written results.

Part 5 (after meeting). Time: 30 minutes.

Write answers to the following:

1. List all the things you learned at the meeting.
2. List at least three things that you should have asked or said (but didn't).
3. Look back at your stage directions. Identify at least one aspect that you found effective, and explain why.
4. Identify at least one aspect that you would change, and explain why.

Grading criteria

1. Did you demonstrate knowledge about the client?
2. Did you identify a robust set of specific, meaningful issues to discuss at the meeting?
3. Did you demonstrate a careful reading of earlier Product Owner documents, through references to specific items in their descriptions?
4. Does your narrative display an attention to all aspects of your conversation with the client — both verbal and nonverbal?
5. In your reflection on the client meeting, do you identify specific, concrete positive and negative aspects?
6. Are the technical aspects of your writing sound: organization, grammar; spelling; complete sentences; tailored to the intended audience, with unfamiliar terms defined?

Grading rubric

Criterion	Outstanding	Score	Unsatisfactory
Knowledge about client	Provides specific, varied, relevant details about client.	5...4...3...2...1	Provides client information that is vague or frivolous, or focused on one or two aspects to the exclusion of all else.
Issues to discuss	Provides a range of focused questions, targeted to particular needs within the project.	5...4...3...2...1	Provides only questions that are too vague to answer effectively, or irrelevant to the current project needs.
Careful reading of documents	Refers to multiple specific items in earlier Product Owner documents.	5...4...3...2...1	Fails to make specific references to earlier documents.
Appropriate backlog format	Backlog arranged as set of stories with clear priorities.	5...4...3...2...1	Backlog lacks indication of priorities, or lacks discernible stories.
Appropriate story content	Stories expressed as clear, statements of desired software functionality, that can conceivably be achieved within the bounds of a sprint.	5...4...3...2...1	Stories fail to refer clearly and specifically to functionality that can be feasibly implemented.
Attention to all aspects of conversation	Covers a range of nonverbal aspects of the conversation, in addition to paraphrasing the verbal content.	5...4...3...2...1	Focuses on only one or two aspects of the conversation.
Identification of positive/negative aspects	Identifies multiple specific aspects of the conversation, both positive and negative; refers back to the written description of the conversation.	5...4...3...2...1	Identifies only a narrow range of conversational issues; fails to make reference to the written description.
Technical aspects of writing	Presents organized, grammatical prose; shows attention to the audience through reader-friendly prose, with complete sentences and definitions where appropriate.	5...4...3...2...1	Presents prose with significant technical faults; fails to present a reader-friendly document, due to fragmented prose or undefined terminology.

Resources for Instructors

Ethnography - Describing the "Workplace"

Paul Kutsche. *Field Ethnography: A Manual for Doing Cultural Anthropology.* Prentice Hall, 1997. ISBN 0138894523.

Michael Agar. *The Professional Stranger: An Informal Introduction to Ethnography.* Academic Press, 1996. ISBN 0120444704.

John Van Maanen. *Tales of the Field: On Writing Ethnography.* University of Chicago, 1988. ISBN 0226849627.

Scrum and Product Owner

Scrum Principles. http://www.scrumalliance.org/pages/scrum_101

Creating Product Owner Success. http://www.infoq.com/articles/agile-product-owner

User Stories. http://www.agilemodeling.com/artifacts/userStory.htm

Samples of Conversation Transcripts

Memorandum of Conversation: Dr. Henry Kissinger and Chatichai Choonhavan, November 1975. http://www.mekong.net/cambodia/kissinger_chatichai.htm

Example Screenplays. http://www.makingthefilm.com/screening4.html