Wright State University

CORE Scholar

Computer Science & Engineering Syllabi

College of Engineering & Computer Science

Winter 2012

CEG 468/668: Managing the Software Development Process

John A. Reisner
Wright State University - Main Campus, john.reisner@wright.edu

Follow this and additional works at: https://corescholar.libraries.wright.edu/cecs_syllabi

Part of the Computer Engineering Commons, and the Computer Sciences Commons

CEG 468/668: Managing the Software Development Process
Winter Quarter, 2012

Course Description

This course covers the challenges and issues associated with software project management. Emphasis will occur on two fronts: (1) the software project manager's view (i.e., "What considerations and obstacles confront project managers during software development?"), and (2) the organizational view (i.e., "How can organizations can foster a climate where software project management is performed effectively throughout an organization?"). Topics covered will include:

- Software project management via POMA: Planning, Organizing, Monitoring, and Adjusting
- The CMM (Capability Maturity Model) and CMMI
- The rise of agile methodologies in response to heavyweight CMM methodologies and processes
- Balancing the advantages and strengths of both "agile" and "disciplined" approaches to software project management

Course Textbook & Other References

B. Boehm and R. Turner, ***Balancing Agility and Discipline****: A Guide for the Perplexed*, Addison-Wesley, 2004. This is a required textbook for this course.

This course will draw from materials in other texts as well. The materials related to POMA will draw heavily from the book *Managing Software Projects* by Frank Tsui (Jones and Bartlett Publishers, 2004). The course will be structured in such as way that students will not need to obtain this textbook.

The course will also teach much on the CMM. Students will need to have access to a CMM or CMMI reference during this course. However, such materials are available on-line, and need not be bought. One such reference can be found at http://www.sei.cmu.edu/pub/documents/93.reports/pdf/tr24.93.pdf.

Note: the website will tell you that SEI is no longer maintaining the CMM, because it has been superseded by the CMMI. While that true, a reference to the CMM will suffice for the purposes of this course. (In fact, even though the CMM has been superseded by the CMMI, this course will focus more on the CMM, because the CMM is more focused on software (as opposed to full systems), and this is a software engineering course).

The aforementioned document is listed because it is a mere 81 pages long. More complete references exist online, but there is no need to print a 600-page reference.

Instructor Contact Info

John Reisner
Office Hours after class or by appointment
Work Phone: 255-3636 x7422 (Wright-Patterson AFB)
email: **john.reisner@wright.edu** (if you want a timely response, please CC: **john.reisner@afit.edu**)

The instructor is an adjunct faculty member. Most contact will be done via email, Pilot, or in after-class discussions. Other meetings can be arranged.

If, at any time, you are having trouble accessing course materials via Pilot, please send me an email immediately. (The sooner I am aware of a problem, the sooner I can fix it. Because I view Pilot as an instructor, I sometimes mistakenly believe materials have been posted, when in fact students cannot access them. Your support in this matter is greatly appreciated.)

Course Objectives

By the conclusion of this course, each student should be able to:
- Better understand some of the strategies used to manage the development of large-scale software systems.
- Understand the goals of various software process models.
- Explain the difference between a software process model and a software lifecycle model.
- Comprehend how the Software Engineering Institute's (SEI) Capability Maturity Model (CMM) can be used to measure and improve an organization's software development process.
- Understand some of the differences between the CMM and CMMI, and understand the motivations for the model's evolution.
- Describe the benefits, limitations, and misuses of CMM and CMMI evaluations within a software development community.
- Explain the reasons behind the advent of agile methods.
- Understand some of the advantages, disadvantages, and tradeoffs between agile methods and more formal alternatives.
- Describe some of the selection criteria for agile and structured methodologies.
- Describe the Turner and Boehm's "sweet spot."

Course Format

This course will be taught in a collaborative manner – meaning that, during class time, much of the material will be *discussed* among the class, rather than presented in a *strict lecture format*. Students will be expected to have done any readings or research assigned prior to the lecture, and able to contribute to the discussion in an informed, intelligent, and constructive manner. See notes under "Class Participation" in the Course Grading section, on the following page, for more information.

Due to the collaborative nature of many lectures, open laptops are <u>not</u> permitted in class.

The SUE Grading System

Many assignments in this class are designed to require thought, reinforcing understanding, and increase retention. Often, these assignments don't have answers that are "right or wrong;" rather, they are either well-supported and articulated, or they are not. Hence, my grading scheme follows three-tier scale: work is graded as Satisfactory, Unsatisfactory, or Exemplary.

If your submission is *Satisfactory*, then your grade will be S, which translates to a 90. If you receive a 90 on an assignment, you haven't "lost 10 points;" you have received ample credit for satisfactory work.

Unsatisfactory indicates the work did not meet expectations, due to a lack of effort, a lack of understanding, or apathetic carelessness. When the problem seems to be caused by a misunderstanding of basic ideas, then I will usually provide some personal feedback, with the aim of helping you understand the material better.

Exemplary work is refined, and shows effort, forethought, insight, thoroughness, and originality, and will be scored above a 90. Grades of 100 are meted out when an assignment is so enjoyable to read, that I find myself thinking, "This is as good as or better than anything I could put in an answer key."

After reading 15 or 20 essays on the same topic, I get a pretty good idea of which assignments are more well-presented than others. The ones that are "more than satisfactory" receive grades such as 92, 95, or 97, and a truly superior work may receive an 100. Again, do not ask me what was "wrong" if your grade is a 90. A 90 means you understood the assignment and did a good job of presenting your response.

Points are deducted for late assignments. Factors such as how late the work was turned in, and how much advanced notice was provided, affect the severity of the deduction; extenuating circumstances may be considered. In general, though, the later the work is turned in, the more points will be deducted. If you complete an assignment well before you will be able to turn it in, then email it to me ASAP; this can minimize points lost for lateness. (For example, if you finish your late work on a Thursday night, then email it to me that day -- that could result in fewer points lost than if you waited until the following Tuesday to hand it in). That said, **always** turn in a hard copy at the start of the next class.

Course Grading

25% Class Project
- Each student will contribute to a large-group project.
- This work will be done throughout the course, and turned in on the day of the last class.
- Not everyone in the group will necessarily receive the same grade. Grades are assigned based on individual contributions to the overall success or failure of the project.
- More information will become available as the class progresses. This information will be posted on Pilot for reference.

15% Homework Assignments
- Homework assignments are designed to facilitate deeper comprehension about a lecture topic (in other words, these are "think and respond" assignments).
- In contrast to the large-group project, these assignments are to be completed **individually**.
- There may be up to two assignments per week, but some weeks may have 1 or 0 assignments.
- Details about these assignments will be found on Pilot.
- Normally, these assignments will be due on Tuesday of the week following the assignment. In other words, you will have one week to complete an assignment corresponding with a Tuesday lesson, and five days to complete an assignment given on Thursday. Any exceptions to this policy will be mentioned when the homework is assigned.
- All work is to be handed in via hard copy. Assignments are due at the **start** of class.
- If you are unable to attend class, email the assignment to indicate it was completed on-time. Then, turn in your hard copy the next time you are in class. Such emails should be timestamped **before** class time (missing class does not give you a homework extension).
- Assignments will be graded using the SUE grading system (explained on the preceding page).

15% Individual Project
- Each student will complete either a term paper, or else a multimedia presentation, on a software engineering methodology discussed in the book.
- Each student's work will be reviewed by at least two peers.
- Your goal is to provide a source that will help your peers learn more about the topic.
- More information will be provided on Pilot.

5% Class Participation
- The instructor will note individual contributions to class discussions as the course progresses.
- Assessments are made based on *long-term* contributions. Students should not feel compelled to blurt out something during every lecture, but simply be attentive and contribute intelligently when appropriate.
- The instructor will attempt to accommodate these grades with respect to individual personality traits and potential language barriers.
- Remember, part of being a good participant is allowing others to participate as well. It is okay to get passionate about something, but dominating the class discussions repeatedly may mean that it's time to yield the floor.
- Other factors, such as sharing your ideas and resources on the Pilot discussion boards, or your peer reviews of individual projects, may be factored into this grade as well. *The end goal is to inspire you to become a contributing member of the class*.
- As a general rule, if you come ready to participate, you will do fine. If you come with an indifferent and apathetic attitude, this part of your grade will suffer accordingly.
- Use of laptops in class is forbidden.

20% Mid-term Exam
- Mixed-format exam, administered in class.

20% Final Exam
- Comprehensive, mixed-format exam, administered during scheduled exam time.

Grades over 92 will receive an A, over 84 will receive a B, over 76 will receive a C, and over 68 will receive a D. Those grades are *guaranteed*. **This scale can be (and typically is) curved**.

Course Schedule (subject to change)

PART I. Software Project Management Across the Project Lifecycle			
Wk (of)	**Session**	**Lesson Topics**	**Assigned Textbook Reading**
1 (Jan 3)	**1**	**Course Introduction**	
	2	**Process and Project Models**	
2 (Jan 10)	**3**	**PO – Planning, Organizing**	
	4	**Overview of the CMM**	**Appendix C**
3 (Jan 17)	**5**	**The CMM – a Panacea? Advent of Agile Methods**	**Appendix B,** *plus* **Chapter 1**
	6	**Project Management During Requirements**	
4 (Jan 24)	**7**	**CMM Level 2**	
	8	**Project Management During Design**	
5 (Jan 31)	**9**	**CMM Level 3**	
	10	**MIDTERM EXAM**	
6 (Feb 7)	**11**	**MA – Monitoring and Adjusting**	
	12	**Project Management During Coding**	
7 (Feb 14)	**13**	**Agile Methods Compare & Contrast**	**Appendix A**
	14	**Selection Factors**	**Chapter 2**
8 (Feb 21)	**15**	**"A Day in the Life"**	**Chapter 3**
	16	**CMM Level 4**	
9 (Feb 28)	**17**	**CMM Level 5**	
	18	**Attaining a Healthy Balance, Part 1**	**Chapter 5**
10 (Mar 7)	**19**	**Attaining a Healthy Balance, Part 2**	**Chapter 6**
	20	**Project Demos and "Debriefings"**	