Spring 2012

CEG 4120/6120-01: Managing the Software Process

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

Wright State University

CEG 4120/6120: Managing the Software Process

Winter Quarter, 2013

Course Description

This course will cover some of the challenges and issues associated with software project management. Emphasis will occur on two fronts: (1) the software project manager's view (that is, what considerations and obstacles confront project managers during software development), and (2) the organizational view (that is, how organizations can foster a climate where software project management is performed effectively throughout an organization). Specific 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 (Capability Maturity Model), however, no additional book is required, because plenty of adequate resources on the CMM are available online. Note: some websites will tell you that SEI is no longer maintaining the CMM, because it has been superseded by the CMMI. While this assertion is 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 In fact, even though the CMM has been superseded by the CMMI, this course will focus more on the CMM, because the CMM is more focused on software (as opposed to full systems), and this is a software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.

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***. All students will be expected to regularly contribute to these discussions 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 generally **NOT** permitted in class.

Course Project

Students will work on a large-group project for the duration of the course, typically in teams ranging in size from 5 to 7 students. This project will serve as a "test bed" of sorts to help students better understand some of the issues involved with managing larger-scale software development efforts.

The team's goal is not to "hack out" a working program, but to engineer the software, to include requirements analysis, design, planned testing, and the creation of user documentation. To help make this experience more realistic, students will be assigned a real-world project.

WI (Writing Intensive) Designation

This course has been designated as a writing-intensive course. This has two repercussions for the student:

1) There will be written assignments, to include a term paper.

2) All written assignments will be assessed in part on the quality of the writing. Students are expected to turn in polished work, not works of "first-draft" quality.

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 purposes of, and differences between, processes and polices within the context of software development
- 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 Turner and Boehm's "sweet spot."

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 **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 have the instructor's view of **Pilot**, I sometimes mistakenly believe materials have been posted when in fact students cannot access them. Your support in this matter is greatly appreciated.

The "SUE" Grading System

Overall, my goal is to assign homework that requires thought, thereby reinforcing understanding and increasing retention. Many of these assignments don't have answers that are "right or wrong," but, rather, they are either well-supported and articulated, or they are not. Hence, it is impossible to grade these assignments without some measure of subjectivity. I do everything I can to grade each assignment fairly and equitably. Ordinarily my grading follows three-tier scale: work is graded as Satisfactory, Unsatisfactory, or Exemplary.

If your submission was Satisfactory, then your grade will be S, which translates to a 90. Don't think of a 90 as "losing 10 points;" think of it as getting ample credit for satisfactory work.

Occasionally, I receive an assignment with great originality and insight, reflecting much forethought and effort. Not only do I find these assignments enjoyable to read, I sometimes find myself thinking, "This is as good as or better than anything I could put in an answer key." Such exemplary work is graded E, which translates to 100.

If submitted work indicates either a lack of understanding of basic concepts, or an apparent apathetic carelessness, then it will be graded as Unsatisfactory, and a numeric grade will be assigned accordingly. If I think the problem lies with misunderstanding the basic ideas, then I will usually provide some personal feedback, with the aim of helping you understand the material better.

Of course, I also reserve the right to stray from this guideline some. For example, I might grade with values such as 85 or 95 (corresponding to S- or S+), or even an S++, which would be a 98. After reading 15 or 20 essays on the same topic, I get a pretty good idea of which papers are more well-thought-out than others. The ones that are "more than satisfactory" receive grades such as 92, 95, or 97, while the truly superior works will receive an E (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.

I also reserve the right to deduct points for late assignments, depending upon how late the work was turned in, how much advanced notice I was given about when I could expect the work, and any extenuating circumstances that may have applied.

Course Grading

Individual Work: 35%

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 one or zero assignments.
- Answers to these homework assignments generally run about half page to one page in length, and should not take too long to complete.
- 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.
- Assignments are due at the **start** of the class/lab session; please have them printed out and ready to turn in at the start of class. If you are unable to attend class, email will be accepted. Emailed assignments should be timestamped **before** class time (skipping class does not give you a homework extension).

20% **Term Paper**

- Each student will write a term paper (6-8 pages) on a particular software development methodology.
- Papers will be read by the instructor, and peer reviewed by a subset of the class.

Group Work: 30%

05% **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.
- As a general rule, if you come ready to participate, you will do fine. If you come with an indifferent and apathetic attitude, your grade will suffer accordingly.
- Recreational browsing on laptops or tablets in class is forbidden.

25% **Class Project**

- Each student will make a contribution, individually or in small groups, 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.

Exams: 35%

15% **Mid-term Exam**

- Mixed-format exam, administered in class.

20% **Final Exam**

- Comprehensive, mixed-format exam, administered during scheduled exam time.

Course Grades

Final course grades will be assigned at the instructor's discretion, after all grades have been calculated. Grades over 94 will be A, over 86 will be B, over 78 will be C, over 70 will be D, although these numbers represent *guaranteed* grades; **this scale can be (and frequently is) curved.**

Course Schedule

(This is a rough schedule and is subject to change)

Wk (of)	Session	Lesson Topics	Corresponding Textbook Reading
1 (Jan 7)	1	Course Introduction	
	2	Process and Project Models	
2 (Jan 14)	3	Introduction of Class Projects	
	4	Overview of the CMM	Appendix C
3 (Jan 21)	5	**NO CLASS** – Dr. Martin Luther King Day	
	6	Problems with the CMM	
4 (Jan 28)	7	Advent of Agile Methods	Appendix B, plus Chapter 1
	8	Overview of Agile Methods	*Term Paper Assignments after this class*
5 (Feb 4)	9	Project Management During Requirements	
	10	CMM Level 2	
6 (Feb 11)	11	Project Management During Design	
	12	CMM Level 3	
7 (Feb 18)	13	POMA P: Planning	
	14	Project Management During Coding	
8 (Feb 25)	15	POMA O: Organizing	Appendix A
	16	Project Management During Testing	
9 (Mar 4)	17	CMM Level 4	
	18	CMM Level 5	
10 (Mar 11)	19	Software Metrics, Part 1	
	20	Software Metrics, Part 2	
11 (Mar 18)	21	POMA MA: Monitoring & Adjusting	
	22	Selection Factors	Chapter 2
12 (Mar 25)	23	"A Day in the Life"	Chapter 3
	24	Stump the Chump Day	
13 (Apr 1)	25	Methodology Survey, Part 1	*In-Class Presentations*
	26	Methodology Survey, Part 2	*In-Class Presentations*
14 (Apr 8)	27	Attaining a Healthy Balance, Part 1	Chapter 5
	28	Attaining a Healthy Balance, Part 2	Chapter 6
15 (Apr 15)	29	Project Demos and "Debriefings"	
	30	Project Demos and "Debriefings"	