

Wright State University  
**CEG 4120/6120: Managing the Software Process**  
Syllabus for Winter Quarter, 2024

## Where we are going

This course examines challenges and issues associated with **software project management**. We will emphasize two themes: (1) the software project manager's view (that is, the obstacles and tradeoffs that project managers face 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:

1. Process models in general, with particular emphasis on the CMM (Capability Maturity Model) and CMM
2. The rise of agile methodologies in response to heavyweight CMM methodologies and processes
3. Balancing the strengths of both "agile" and "disciplined" approaches to software project management

## How we will get there

Class meets from 6:10 pm to 7:30 pm on Mondays and Wednesdays in Russ 144, where topics will be taught in a *collaborative* manner. This means that, during class time, much of the material will be *discussed* among the class, rather than presented in a strict lecture format. Students should expect to regularly contribute to these discussions in an intelligent and constructive manner. Sometimes the class will break into pairs or groups of three for short exercises where students can wrestle with new concepts in hopes of better understanding them in a more lasting way. The best way to prepare for the exams in this class is by staying engaged in the classroom discussions.

Due to the collaborative nature of the lecture sessions, open laptops and cell phones are generally discouraged during class, particularly when these are being used for social networking rather than taking notes. I will do my best to structure this class so that the time spent in the classroom will foster learning in an engaging way; please be similarly courteous and come ready to participate and learn. (Exceptions can be made for those who are truly taking notes, and not pretending to take notes.)

The textbook for this class Paul E. McMahon's ***Integrating CMMI and Agile Development*** (Addison-Wesley, 2011). This is a required textbook for this course, although the text will not get used until after spring break.

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.

## Learning by doing

The course will feature homework assignments and group projects design to enhance the understanding of course material. Most notably, students will work on a large-group project for the duration of the course, typically in teams ranging in size from 4 to 6 students. This project will help students gain a better understanding some of the more complex issues involved with managing larger-scale software development efforts. (Think about it: most programmers aren't free-lance programmers who work by themselves.)

Although this is a software course, it is not a programming course. I'm not so interested in whether or not you as an individual can write code; I want to know: Can you work in a team environment, and, in such an environment, determine which processes will best facilitate the development of useful software for a client? The goal of a programming team is not confined to "hacking out" a working program, but to engineering the software, which includes eliciting, understanding, and analyzing system requirements; working out an architectural design; integrating components; planning for adequate testing; and creating useful documentation for the end user. The best way to learn these concepts is to do them, so each student will be put into a larger group where they will develop software for a real-world client, often in a technical domain.

## Learning by writing

This course has been designated as a writing-intensive (WI) course. This means that:

- 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.
- 3) Feedback will be given designed to help students become more proficient writers and to better prepare them for professional writing in the workplace.

More about the WI credit can be found in Appendix A.

## Specific topics you will master

By the conclusion of this course, each student should be able to:

- Describe some of the strategies and tradeoffs used to manage the development of large-scale software systems.
- Explain the goals of various software process models.
- Explain the purposes of, and differences between, processes and policies within the context of software development.
- Explain the difference between a software process model and a software lifecycle model.
- Describe how the Software Engineering Institute's (SEI) Capability Maturity Model (CMM) can be used to measure and improve an organization's software development process.
- Note 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.
- Describe some of the advantages, disadvantages, and tradeoffs between agile methods and more formal alternatives.
- List some of the selection criteria for agile and structured methodologies.
- Describe how the CMMI can help agile organizations and vice-versa.

## If you need to contact me

My name is John Reisner; I am an adjunct faculty member at Wright State. This means I am not usually on campus during the school day; however, I am always available for an "office hour" immediately after class. Other meetings can be arranged as necessary.

There is also a Discussion thread on **Pilot** where you can post questions about assignment or class material. Use of this thread is highly encouraged because it allows me to answer a question for everyone in the class rather than for one individual student. You can even post to this thread anonymously. However, if you need to discuss a more private matter, I can also be reached via email at [john.reisner@wright.edu](mailto:john.reisner@wright.edu). If you want a more timely response, please also CC: my daytime email at [john.reisner@afit.edu](mailto:john.reisner@afit.edu). In dire circumstances, you can also reach me at my work phone on Wright-Patterson AFB: (937) 255-5963.

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.

## How I will grade your work

I try to assign homework that reinforces understanding and increases retention. Many of these assignments don't have answers that are "right or wrong;" instead, they are either well-supported and articulated, or they are not. It is impossible to grade such assignments without some measure of subjectivity. I strive to grade each assignment fairly and equitably.

My grading scheme follows an overarching three-tier scale: work can be 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.

However, if your assignment demonstrates forethought, insight, and effort, and is presented with great originality and thoroughness, and is enjoyable to read, to the point where I find myself thinking, "This is as good as or better than anything I could put in an answer key," then the work is graded E, which translates to 100.

On the other hand, if the 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 a misunderstanding of basic ideas, then I will usually provide some personal feedback, with the aim of helping you understand the material better.

Most grades are given somewhere within this three-tier guideline. For example, I might grade with values such as 85 or 95 (corresponding to S- or S+), or even an S++, which might be a 97 or 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 above 90, 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 is a good grade, and it means you understood the assignment and did a good job of presenting your response.

I will accept late work, although points are generally deducted for late assignments. The severity of the deduction depends on how late the work was turned in, how much advanced notice I was given about when I could expect it, and any extenuating circumstances that may have applied. Unless clearly specified otherwise, all graded work is due in the Pilot Dropbox at midnight on the day BEFORE class, although midnight is a "soft" deadline. (I don't consider work turned in a couple hours late to be "late work.")

## Turning your graded assignments into a final grade

Assignments and projects will be weighted as follows:

### Homework: 20%

#### 20% Homework Assignments

- Homework assignments are designed to facilitate deeper comprehension about lecture topics
- In contrast to the large-group project, these assignments are to be completed **individually**.
- All assignments get turned in via the Pilot Dropbox **in .pdf format**.
- Generally speaking, homework is not assigned more than once per week.
- Details about each assignment will be posted on Pilot.
- Homework deadlines will be set when the homework is assigned. Most of the time, at least one weekend will elapse between when homework is assigned and when it is due.
- These assignments are generally due the night before a class.

### Term Paper: 20%

#### 10% Written Term Paper

- Each student will write a term paper (5-7 pages) on a particular software development methodology.
- Papers will be read by the instructor, and also peer reviewed by other students.
- Each student will read and evaluate at least one other student paper (6120 students will review two).

#### 10% Oral Presentation

- Each student will present their findings from the term paper as an oral presentation.
- Presentations can be given during class time or recorded on video and uploaded to YouTube.
- How we will accomplish this will be determined during an in-class discussion.

### Group Work: 25%

#### 25% Class Project

- Each student will contribute to a large-group (4- to 6-person team) software development project.
- The project will be a real-world development project, built for a real-life client.
- This work will be done throughout the course, with a final written report turned in on the last class day.
- **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. **Do NOT try to “skate”.**
- 6120 students should expect to play a greater role in project management functions.
- More information will become available as the class progresses. This information will be posted on Pilot for reference.

### Exams: 35%

#### 20% Two Mid-term Exams

- Mixed-format exams, administered in class; with multiple choice questions, plus short answer/essays.

#### 15% Final Exam

- Comprehensive, mixed-format exam (resembling mid-term), administered during scheduled exam time.

After all weighted grades have been calculated, final course letter grades will be assigned. Grades over 93 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 typically is) curved**. Overall, I look for natural “breaks” in final weighted grades scoring to prevent two students with nearly-identical performances from getting different letter grades. For example, I wouldn’t want a student with an average of 89.9 to get a B while a student with a 90.1 received an A. More about my grading, to include a historical record of letter grades given in prior sections, will be discussed at length on the first day of class.

## Keeping it honest; keeping it real

I take academic integrity seriously; I will not tolerate cheating. Work you turn in should be your own work. If you need help, contact me – don’t go online and pay someone to do your work for you as that won’t help you learn. If you are caught cheating in this class, you will get a zero for the assignment and lose one letter grade for the course. Subsequent violations will result in automatic failure (F) for the course. If you use ChatGPT or any other AI tool to help you with your assignments, that’s okay, so long as you also provide the exact prompts you asked the bot and the exact responses it gave you as a citation.

## The nitty-gritty (our course schedule)

(This is a rough schedule and is subject to change. Keep an eye on the Announcements in Pilot for the most up-to-date information.)

Wk (of)	Class	Lesson	Lesson Topic	Corresponding Reading
1 (Jan 8 & 10)	1	1	Course Introduction	
	2		Class Project Client Meeting	
2 (Jan 15 & 15)	<b>No Class - Holiday Honoring Dr. Martin Luther King, Jr.</b>			
	3	2	Project Risk	
3 (Jan 22 & 24)	4	3	The Software Crisis	
	5	4	Introducing the CMM	
4 (Jan 29 & 31)	6	5	Lifecycle Models vs. Process Models	
	7	6	The Advent of Agile Methods	
5 (Feb 5 & 7)	8	7	More on Agility vs Process	
	9	8	CMM Level 3 - The Defined Level	
6 (Feb 12 & 14)	10	9	Metrics in Software Management	
	11	10	CMM Levels 4 & 5 - The Defined Level	
7 (Feb 19 & 21)	12	11	Introducing the CMMI	
	13		<b>Midterm Exam 1</b>	
8 (Feb 26 & 28)	<b>No Classes – Spring Break</b>			
9 (Mar 4 & 6)	14	12	Choosing a Methodology: Selection Factors	Chapter 1
	15	13	Adding a Dash of Agility	Chapters 2 & 3
10 (Mar 11 & 13)	17	14	Adding a Pinch of Maturity	Chapters 4 & 5
	18	15	Adding an Ounce of Prevention	Chapter 6
Sun Mar 17	<b>First page of Term Papers Due on Pilot, 6 p.m.</b>			
11 (Mar 18 & 20)	19	16	Preliminary Project Demo Prep	<b>2024:</b> <i>No class (work on projects)</i>
	22	17	Preliminary Project Demos	<i>In-class demos</i>
12 (Mar 25 & 27)	23	18	Hybrid Approaches, Revisited	Chapter 8
	24		<b>Midterm Exam 2</b>	
Sun Mar 31	<b>Final draft of Term Papers Due on Pilot, 6 p.m.</b>			
13 (Apr 1 & 3)	25	19	Balancing Agility & Discipline	
	26	20	“I’d Like to Meet His Tailor” (Tailoring)	Chapter 10
14 (Apr 8 & 10)	27		Oral Presentations of Term Papers, Part 1	
	28		Oral Presentations of Term Papers, Part 2	
15 (Apr 15 & 17)	29	21	Course Review	
	30		Final Demos and “Debriefings”	<i>In-Class Presentations</i>
Wed, Apr 24	<b>Scheduled Date for Final Exam (5:45 p.m. - 7:45 p.m.)</b>			

## APPENDIX A

### Integrated Writing Assignments for this Course

(In order for you to get WI-credit for this course, Wright State requires that I include this information)

The following table details the writing assignments for this course:

Nature and goals of assignment	Approx Pg Count
As homework, students will prepare sample policies and procedures as prescribed by the Software Capability Maturity Model (SW-CMM/CMMI). This requires executive-style writing for policies, and the detailed description of technical methodologies for documented procedures.	3
Students will write a detailed term paper describing a software development methodology. This paper must include at least three outside sources (not including the course textbook or Wikipedia). This paper will be graded partly on content, and partly on presentation; it is expected that the paper be well-organized, polished, properly punctuated, with scant or no misspellings, sentence fragments, or grammatical errors. The paper should be clear and interesting to read; each student's paper will be peer reviewed by other students.	5-6
Students will write a comparative analysis of four or five different user interfaces for a single program. It is expected that the essay will deftly compare and contrast the implementation differences in a thoughtful manner, using proper English style.	2
Students will research a major software development failure, and write about why this project failed. The end product should synthesize principles taught in the course with news articles describing a failed software development effort, and is expected to be written in an organized and coherent manner.	2
<b>Total pages of individual writing assignments for the course</b>	12-13

These individual writing assignments are worth roughly 30% of the students grade (the written term paper is worth 10%; the other assignments are homework assignments, and homework is worth 20%).

Moreover, students in this course are assigned to a large group (5-6 students), and each group works on a large-group software development project. At the end of the term, each group turns in a software development folder (SDF), which is a compilation of the several artifacts, such as: a project plan and schedule, a summary of the development strategy used, a requirements specification, architectural design documents and diagrams, use case scenarios, source code, test results, and a rudimentary user's guide. Moreover, a written summary of the lessons learned during project development is included in the final turn-in.

The exact content of this folder will vary from group to group, depending on the approach the group elects to take, and the amount of documentation generated as they complete their project. The end result often runs between 40-60 pages, but most of this material is work that is completed as a natural part of project development, not written after-the-fact. Still, meticulous note-taking and thorough design documentation can pay handsome dividends in this course.

#### *Standard WSU IW boilerplate language*

#### **Wright State students will be able to produce writing that:**

- demonstrates their understanding of course content,
- is appropriate for the audience and purpose of a particular writing task,
- demonstrates the degree of mastery of disciplinary writing conventions appropriate to the course (including documentation conventions), and
- shows competency in standard edited American English.