EAST TENNESSEE STATE UNIVERSITY

CSCI-4350– Software Engineering II – 83686

Credits: 3 | Term: Fall 2026

Welcome to this class!

Welcome from William Kinser, Your Instructor

Welcome to 4350 (SE II) for Fall of 2026. This course will explore several topics related to how software projects are organized and executed. We will look at several of the most common models, their phases, their artifacts, and architecture approaches.

Regards,

William Kinser

An Essential Question We Will Ponder

How can you plan and execute software solutions following the Agile philosophy?

Where and when does our class meet?

Location & Meeting Time

Classroom Location: Brinkley Center (BC) 215

Class Meeting Schedule: Tuesday and Thursday from 12:40 pm to 2:40 pm

Accessibility Notes: Elevators are located near the main entrance and near Digital Media.

Who is my instructor and how can I reach them?

Contact Information

Instructor: William Kinser (Professor of Practice, Computing)

Email: kinserw@etsu.edu

When can I talk to my instructor about my class progress or any questions I have?

I welcome out-of-class discussions with my students! You can find me at the following times, or if these times do not work, you can schedule an appointment to meet.

Student Drop-In Hours:

10:30 am to 1:00 pm Mondays
8:30 am to 10:00 am Tuesdays
8:30 am to 2:30 pm Wednesdays
8:30 am to 10:00 am Thursdays

Drop-In Location: In the hall outside Brinkley Center 215 (second floor near the ramp/bridge)

What will I need to be successful in this class?
Course Description & Materials

Prerequisites: CSCI 4250, completed with a grade of 'C-' or better.

.

Required Materials

There is no required physical textbook for this course. All needed reference material will be provided on-line. You will need a device that allows you to participate in team-based software development using tools such as Visual Studio, GitHub, Trello, Jira, Selenium, etc. Specific tools used will be selected by teams.

Optional, but Helpful, Materials

See the below list of optional reference material available on Safari Books Online:

- **[UML]** Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *Unified Modeling Language user guide. (2nd Ed.)*. Upper Saddle River, NJ: Pearson Education, Inc. ISBN 0-321-26797-4.
- **[ES]** Rubin, K. S. (2013). *Essential Scrum: A practical guide to the most popular agile process*. Upper Saddle River, NJ: Pearson Education, Inc. ISBN 978-0-13-704329-3.
- **[CleanC]** Martin, R. C. (2009). *Clean Code: A Handbook of Agile Software Craftsmanship*. Upper Saddle River, NJ: Pearson Education, Inc. ISBN 978-0-13-235088-4
- **[PP]** Thomas, D., & Hunt, A. (2019). *The Pragmatic Programmer: Your Journey to Mastery*. Boston, MA: Addison-Wesley Professional. ISBN 978-0-13-595705-9
- **[DevOps]** Kim, G., Debois, P., Willis, J., Humble, J., and Allspaw, J. (2016). *The DevOps handbook*. Portland, OR: IT Revolution Press, LLC. ISBN: 978-1-942-78800-3 (Available for FREE on Sherrod Library Website and Safari Books Online)
- **[Planning]** Cohn, M. (2006). *Agile Estimating and Planning*. Upper Saddle River, NJ: Prentice Hall. ISBN: 0-13-147941-5 (Available for FREE on Sherrod Library Website and Safari Books Online)

Resource Readings – Recommended

- *Learn UML: Unified modeling language*. http://www.tutorialspoint.com/uml/index.htm.

What will I learn in this class?
Course Overview

Course Purpose and Objectives

Software development as an engineering discipline with emphasis on process, continuous improvement, SE phases (vision, requirements, design, implementation, testing, release, maintenance, upgrade/update), project management, verification and validation of processes and products for correctness and completeness, configuration management, automation, and software quality assurance. Communications (written and oral), participation in team projects, and use of automation tools are integral. Our goal is to plan, design, build, and validate robust, fault tolerant, scalable, and extensible solutions that meet or exceed client needs.

Major Course Topics

- We will revisit most of the topics from CSCI 4250 but do them more robustly: Professionalism, Legal issues, ethics in software engineering, Project Management, Source control, Agile, Scrum, and the Software Development Lifecycle (SDLC) phases, models, and artifacts from SE I. In addition, we will merge these with DevOps and Emergent Architecture.
- Time and scheduling permitting, we will run multiple Scrum sprints during the semester for real world product development.

I provide a schedule on D2L outlining the timing of content and deliverables: **schedule is subject to change, depending on content progress and factors beyond my control, e.g., university closures.**

Expected Learning Outcomes

By the end of this course, you should be able to:

1. Apply an appropriate software lifecycle model to a given scenario
2. Demonstrate an understanding of processes involved in the creation, evaluation, and use of project deliverables in each phase of the software development lifecycle
3. Demonstrate the ability to participate on project teams in various roles
4. Demonstrate an understanding of the measurement of software products and processes
5. Demonstrate an understanding of project scheduling and estimation
6. Demonstrate an understanding of the controlling disciplines used to assess and improve software quality, including configuration management, quality assurance, verification, and validation,
7. Demonstrate an understanding of DevOps,
8. Demonstrate an understanding of emerging architectures, such as microservices and Internet of Things,
9. Write and orally communicate about software engineering.

What am I expected to do in this class?
Course Policies

What is the attendance policy for this class?

Your presence is essential in this class for your best learning!

Attendance will be taken and tardiness will count as an absence because this class is designed to prepare you for a professional position where attendance and punctuality are expected every day. Being late to class is a distraction to the rest of the class.

Students should inform the instructor as soon as possible of **excused absences** (as defined by university policy https://www.etsu.edu/policies/documents/attendance-policy.pdf) to arrange delayed due dates and make-up work (as needed). While I will work with you to keep you current with the class's progress so that you are not at a disadvantage upon your return, it is up to you to do the necessary work that makes that a reality. If you do not contact me in a timely manner for make-up work, **no credit will be possible for the missed work**.

How is late work addressed?

Deadlines are clearly posted in D2L at the outset of the semester. Work must be submitted on time in order to receive credit unless you make arrangements with me in a timely manner (timely meaning "as soon as you know you will miss an assignment"). Acceptable reasons for missing a submission deadline should align with university attendance policy referenced above. Unacceptable reasons, such as simply forgetting to turn it in or missing the deadline due to technical issues at the last minute, will not be granted the opportunity to make-up work.

What are the expectations for my behavior in class?

In order for everyone to have a positive learning experience, it's important that we work together to have a safe, non-distracting, and engaged learning environment. Here are expectations to help us meet this goal:

I encourage students to bring their laptops, tablets, cell phones, and other development and/or touch-enabled devices to class—We will have SEVERAL in-class exercises where you will need some internet access and/or development tools. However, I request that students try to minimize distractions generated by those devices, e.g., use class time to work on coursework for this class and not for other classes. Students should not stream video/audio, play games, or browse social media during class time. Students should not work on assignments unrelated to this class during class time (e.g. their job or other classes). Students that distract other students will be asked to leave the classroom for the day.

All students are expected to actively participate in class activities with a good attitude and conduct themselves in a professional manner.

Will this syllabus change?

While I have tried to make this syllabus as accurate and complete as possible, I reserve the right to make changes within it at any time. I'll make every effort to notify you of such changes, and information about any changes will be posted from me on D2L.

How will my grade be determined in this class?

Course Grade

Coursework consists of:

- Lecture (where you will be expected to take notes),
- In-class exercises (where you will be expected to participate),
- Weekly worksheets
    - Course study guides prior to start of project
    - Project work
    - Presentation Reviews near end of the semester
- Learning Outcome tests (for 7 of the 9 LOs; team participation and communication are addressed separately),
- A team-based, software development project (see the Project subsection) with several iterations,
- Individual presentations on a topic related to Software Engineering.

The final course grade is calculated as:

Weekly worksheets	20%
Learning Outcome Tests (7) (10% each)	70%
Presentation	10%

Grading Scale: Final Grade

Final grades will be given according to the following scale:

Numeric Grade	Letter Grade
93.000 - 100.000	A
90.000-92.999	A-
87.000-89.999	B+
83.000-86.999	B

80.000-82.999	B-
77.000-79.999	C+
73.000-76.999	C
70.000-72.999	C-
67.000-69.999	D+
60.000-66.999	D
0.000-59.999	F

The Grade of Incomplete

In general, the grade of incomplete is only given in the cases of serious, uncontrollable situations that will prevent the student from completing the assigned coursework. A request for a grade of incomplete must be submitted in writing with the reason for the request and any appropriate documentation. I reserve the right to discuss a student's request with the department chair and/or other instructors. Students have up to one year to complete the requirements of the course and resolve the incomplete. After that one-year period, the grade automatically converts to an 'F'. **Remember that an incomplete could have financial aid implications; students should check into those issues before making a request**.

How will I show what I'm learning in this class?

Assignment & Assessment Overview

Learning Activities

Students are expected to attend class, practice active listening during lectures, and participate in class exercises and team based assignments in a professional manner. The ultimate goal of the course is for each student to learn the material. If a student has a situation or learning style not conducive to the planned course execution, they can meet with the instructor to discuss and potentially define an alternate learning plan.

Learning Outcome Tests

There will be seven (7) topic specific tests. Each is available on D2L after the topic has been fully covered in class. Each is due prior to Early Semester Progress Reporting (ESPR/Mid-Term Grades). Tests are time limited to an hour by default. At the instructor's discretion, these tests may be administered during class time or be taken outside of class. In either scenario, students are expected to take them like a formal exam (no notes, no internet, no assistance from other students or other sources). The ETSU honor policy is expected to be followed. Cheating will result in a zero for the exam in question.

If a student is not happy with their performance on any of the tests, they can retake them up to two (2) times. The highest grade will be used if a test is attempted more than once. The retakes are only

available at specific times during the semester (after the initial test due date and before “dead week”). It is only possible to attempt a retake if you have attempted the prior test. That is, you can’t skip a test and count on taking a retest for your grade.

Final Exam (if applicable)

There is no final exam for this course.

Group Discussions/Projects

The project grade is determined by the completion of the Sprint Worksheets. There is a different worksheet for each sprint.

When group projects are assigned, it is each individual's responsibility to contribute to the work being done. If a team member is making a good faith effort to be involved in a group project, but their contribution is being rejected, the student should contact the instructor immediately.

If a team member is not contributing, the team should first try to engage this team member. If unsuccessful, the instructor should be notified. The instructor will contact the non-contributing student to communicate their options related to the group project. **Options, at the instructor’s discretion, may include the student remaining on the team, moving to a different team, performing the project individually, performing an alternate assignment individually, or the student receiving a reduced grade (typically a 0) for the project.**

Individual Presentation

Each student must research a topic related to Software Engineering (choosing from a list provided or getting approval for their own topic of choice), construct a presentation that is informative and represents the culmination of their college experience in presenting. The delivery should be well practiced with a smooth cadence, inflection, and clear enunciation. The presentation can be delivered via video (e.g. YouTube, Panapto, or the video services at the university library) or in-person (please work with the instructor to schedule a time during class to do in-person presentations).

Each presentation should have an introduction, body, and conclusion. The use of visuals is strongly encouraged but is not limited to only slides (e.g. you can do a demonstration, a how-to video, a podcast style, etc.) It must be between 10 and 15 minutes in length. Citations for sources are expected but may be informal. They will be evaluated against the following rubric:

Criteria	Does Not Meet	Meets	Exceeds	Criterion Score
Organization	0 points Student does not present a logical sequence of information.	3 points Student presents a logical sequence of information.	5 points Student presents a strong, clearly logical sequence of information.	/ 5
Subject Knowledge	0 points Student cannot answer questions about subject.	8 points Student is able to answer central/basic questions about subject adequately.	10 points Student is able to provide in depth answers to probing questions about subject.	/ 10
Graphics	0 points Student uses superfluous graphics or no graphics.	3 points Student uses graphics that support text and presentation.	5 points Student uses creative graphics that effectively support text and enhance presentation.	/ 5
Mechanics	0 points Student has numerous spelling errors and/or grammatical errors throughout presentation that impede understanding extensively.	3 points Student has few misspellings and/or grammatical errors throughout presentation that do not impede understanding.	5 points Student has no misspellings and/or grammatical errors throughout presentation.	/ 5
Eye Contact	0 points Student reads report with little or no eye contact (with audience/camera).	3 points Student occasionally reads but maintains adequate eye contact.	5 points Student refers to notes as appropriate but maintains excellent eye contact.	/ 5
Elocution	0 points Student consistently mumbles, mispronounces terms, and/or speaks too quietly.	3 points Student speaks clearly and pronounces most words correctly and audibly.	5 points Student varies tone of voice and maintains audience attention.	/ 5
Timing	0 points No presentation	60 points Length was outside the assigned window	65 points Length was within the assigned window	/ 65

Total 100

The presentation is worth 10% of the final grade

Software Engineering Competency Model (SWECOM)

SWECOM is a competency model based on individuals developing their knowledge, behavioral attributes, and cognitive skills to support learning and growing proficiency in a set of technical skills (in this case—engineering software). Students should view SWECOM as a helpful guide for developing their skills in preparation for a career in developing software. Examples from each category follow—these examples include, **but are not limited to**:

- Knowledge:
    - Key software engineering definitions
    - Software development lifecycle (SDLC) topics
    - Process implementation and management
    - Process assessment and improvement
    - Students may demonstrate high levels of knowledge in both their writing and their quiz scores
- Behavioral attributes:
    - Showing initiative
    - Demonstrating enthusiasm
    - Exhibiting willingness to work
    - Demonstrating honesty, integrity, and dependability
    - Communicating, both in writing and verbally
- Cognitive skills:
    - Reasoning (including using abstractions and forming associations)
    - Measuring and analyzing data
    - Performing root cause analysis
    - Problem solving (including top-down, bottom-up, and divide-and-conquer strategies)
        - Also includes recognizing patterns and applying both iterative and incremental problem solving
    - Brainstorming, prototyping, and modeling
- Technical skills:
    - Producing artifacts related to areas of the SDLC
        - Requirements artifacts, e.g., use cases and user stories

        - Design artifacts, e.g., architectural component-and-connector diagrams
        - Software construction artifacts, e.g., functional code that addresses requirements
        - Testing artifacts, e.g., unit tests
    - Using CASE and other software production tools effectively, e.g., Trello and BitBucket

What might impede my learning in this class?
ETSU Code of Ethics and Course Gen AI Policies

ETSU Code of Ethics

Learning is effortful and sometimes uncomfortable; however, your learning happens best when you use your own brain instead of relying on others (or technology) to do the thinking for you.

Therefore, please avoid plagiarizing or other forms of cheating, as outlined in our university's Academic Integrity policy. And, if you know someone else is engaging in these behaviors, be sure to report this because it is part of our ETSU Honor Code.

Generative AI Use in This Class

Generative AI (Gen AI) can sometimes help and sometimes hinder learning. In this class, you can use AI to help you (or your team) brainstorm ideas. For the semester project, you can leverage AI to generate code, test cases, and other technical aspects of the project. Whenever you use AI, you should cite your source and clearly indicate what portions were generated by AI. Always review content created by AI for accuracy and completeness. While it is quick to produce content, it tends to make assumptions and "hallucinate" which results in errors.

In addition, my expectation is that everyone, including me, follows ETSU's policy and guidelines for Gen AI use.

What else is important for me to know for my success?
Additional Information

Syllabus Attachment Information

Syllabus Attachment Information: The University's approved Syllabus Attachment Information page provides information about relevant University and Academic Policies that important for you to know. https://www.etsu.edu/curriculum-innovation/syllabusattachment.php