CS59300-ASE AI-Assisted Software Engineering

Instructor: Tianyi Zhang, Assistant Professor of Computer Science

Email: tianyi@purdue.edu

Lecture: 1:30PM-2:45PM Tue and Thurs @ LWSN B134

Office Hours: 12:30PM-1:25PM @ LWSN 3154H

Teaching Assistant: Wei-Hao Chen, chen4129@purdue.edu

Instructional Modality: Face-to-Face

Course Credits: 3.0

Prerequisites: Python programming skills and basic understanding of machine learning are required. Knowing how to use PyTorch is recommended.

Course Description

Have you ever wondered about these:

- Will programming jobs no longer exist because of large language models like ChatGPT?[1]
- Will software engineering become prompt engineering in the next decade?
- How far are we from the “black art” of natural language programming as Dijkstra called it 40 years ago?[2]
- What essential software engineering skills are needed in the age of AI?

This course will help you answer those questions by introducing the foundations, systems, and research challenges of modern AI-assisted software engineering. In this course, you will learn (1) how LLMs are trained and adapted for software tasks; (2) how LLM agents combine reasoning, tools, and context to tackle complex tasks; and (3) how LLMs and agents support the full software lifecycle, from requirements and design through implementation, testing, deployment, maintenance, and security. While there has been a long history of applying AI and Machine Learning to software engineering tasks[3], we will focus on recent advances in large language models and agentic AI and on how these technologies are reshaping software development practice and research in this course.

---

[1] Matt Welsh, The end of programming. *Communications of the ACM*, January 2023.

[2] E. W. Dijkstra. On the foolishness of “natural language programming". In *Program Construction*, pages 51–53. Springer, 1979.

[3] Richard J Waldinger and Richard CT Lee. PROW: A Step toward Automatic Program Writing. IJCAI, 1969.

This course begins with an introduction to AI-assisted software engineering (Week 1), followed by LLM fundamentals (Weeks 2-3), including language modeling, transformers, pre- and post-training, prompt engineering, RAG, and test-time scaling. The next part covers agent fundamentals (Weeks 4-7), including agent loops, MCP, agent skills, agent architectures, and harness engineering. From Weeks 8-15, we will talk about how LLMs and agents are applied to major software engineering activities across the lifecycle. This course includes three homework assignments, five pop quizzes, and a midterm exam. Besides, you will work in groups on a semester-long project to design and conduct a research project in AI-assisted software engineering.

Learning Resources, Technology, Texts

There is no required textbook for this course because AI-assisted software engineering is an emerging and rapidly evolving area. But the instructor will post recommended readings before each lecture. Students are encouraged to read them for extra material not covered in the lecture.

The following books and online resources may be helpful for students who would like to learn more about deep learning, transformers, LLMs, and agents:

- Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning
- Stanford CS336: Language Modeling from Scratch
- Austin Huang, Suraj Subramanian, Jonathan Sum, Khalid Almubarak, and Stella Biderman. The Annotated Transformer
- Hugging Face LLM Course
- Hugging Face AI Agent Course
- DAIR.AI. Prompt Engineering Guide

I recommend you reading some surveys to get a basic understanding about the literature.

- Liu, Junwei and Wang, Kaixin and Chen, Yixuan and Peng, Xin and Chen, Zhenpeng and Zhang, Lingming and Lou, Yiling. "Large Language Model-Based Agents for Software Engineering: A Survey." TOSEM 2025.
- Kaixin Wang, Tianlin Li, Xiaoyu Zhang, Chong Wang, Weisong Sun, Yang Liu, Aishan Liu, Xianglong Liu, Chao Shen, Bin Shi. "Software Development Life Cycle Perspective: A Survey of Benchmarks for Code Large Language Models and Agents." Arxiv 2026.

We will use Ed Discussion to make announcements, ask & answer questions, look for teammates, etc. The course instructor has already added everyone to this course on Ed Discussion based on the class roster from Purdue Brightspace. If you have not been added to this course on Ed Discussion, please email the instructor.

All lecture slides, assignment instructions, and other materials will be posted on Purdue BrightSpace.

Learning Outcomes

At the end of this course, students should be able to:

- explain the core concepts of LLMs and AI agents and how they support software engineering
- effectively use modern coding agents for different kinds of software engineering tasks
- design and implement LLM-based methods or agentic systems for software engineering tasks
- evaluate AI-assisted software engineering tools through quantitative experiments and user studies
- design, conduct, and communicate research projects in AI-assisted software engineering

Assignments

Your achievement of the course learning outcomes will be assessed through three homework assignments, pop quizzes, a midterm exam, and a semester-long course project. Your final grade will be based on the cumulative scores earned across these components.

Homework assignments [30%, 10% each assignment]
Midterm exam [25%]
Course project [35%]
- Project proposal [5%]
- Midterm project report [5%]
- Final project report [20%]
- Final presentation [5%]

Pop Quizzes [10%, 2% each quiz]

Further details about the homework assignments and course project are provided in later sections of this syllabus. Assignment instructions, grades, and feedback will be posted on Brightspace.

Late submissions are accepted with a 20% penalty per day, except for the final project report and presentation slides, which have a firm deadline of December 13 and will not be accepted late.

Grading Scale

Percentage Grade (%)	Letter Grade
[97, 100]	A+
[93, 97)	A
[90, 93)	A-
[87, 90)	B+
[83, 87)	B
[80, 83)	B-
[77, 80)	C+
[73, 77)	C
[70, 73)	C-
[67, 70)	D+
[63, 67)	D
[60, 63)	D-
[0, 60)	F

Attendance Policy

This course follows Purdue's academic regulations regarding attendance, which states that students are expected to be present for every meeting of the classes in which they are enrolled. While we will not check attendance in each class, we will use other ways such as pop quizzes and in-class activities to check your attendance and participation in the class. Please come to the class continuously and participate in class discussions.

For absences due to common illness, students should email the instructor regarding the class they will miss. A doctor's note is required if an illness causes a student to miss a pop quiz, exam, or homework deadline and the student needs to request a deadline extension or reschedule a quiz or exam.

For excused absences covered by University policy, such as grief/bereavement, military service, jury duty, or parenting leave, students should follow the instructions on the Office of the Dean of Students website and submit the appropriate absence request form. Late submissions and rescheduled quizzes or exams will be accommodated based on the dates of leave approved by the University.

Course Schedule

Part I. Introduction	
Tue 8/25	Course Overview
Thu 8/27	Introduction to AI-assisted Software Engineering
Part II: LLM Fundamentals	
Tue 9/1	Language Modeling and Generation
Thu 9/3	Transformers: The Architecture Behind Modern LLMs
Tue 9/8	LLM Pre-training, Post-training, and Knowledge Distillation
Thu 9/10	Prompt Engineering, RAG, and Test-time Scaling
Part III. Agent Fundamentals	
Tue 9/15	Agent Loop: Planning, Reasoning, Acting, and Feedback
Thu 9/17	Agent Tool Use and Model Context Protocol
Tue 9/22	Agent Skills and Self-Improvement
Thu 9/24	Tutorial 1: Claude Code and Codex
Tue 9/29	Context Engineering and Memory
Thu 10/1	Tutorial 2: Build Your Own Agent
Project proposal due on 10/4 midnight	
Tue 10/6	Agent Architectures and Multi-Agent Systems
Thu 10/8	Harness Engineering and Long-Horizon Tasks

HW1 due on 10/10 midnight	
Part IV. Applications in Software Engineering	
Tue 10/13	**No Class (Fall Break)**
Thu 10/15	LLMs and Agents for Requirement Engineering
Tue 10/20	LLMs and Agents for Code Generation
Thu 10/22	**Midterm Exam**
Tue 10/27	LLMs and Agents for Software Testing
Thu 10/29	LLMs and Agents for Software Debugging and Fault Localization
Mid-point project summary due on 11/1 midnight	
Tue 11/3	LLMs and Agents for Program Repair
Thu 11/5	LLMs and Agents for Code Review
HW2 due on 11/8 midnight	
Tue 11/10	LLMs and Agents for Formal Verification
Thu 11/12	LLMs and Agents for Software Maintenance
Tue 11/17	LLMs and Agents for Cybersecurity
Thu 11/19	LLMs and Agents for DevOps
11/24 and 11/26: No Class (Thanksgiving)	
HW3 due on 11/29 midnight	
Tue 12/1	Open Challenges and Future Directions
Thu 12/3	Final Project Presentation

Tue 12/8	Final Project Presentation
Thu 12/10	Final Project Presentation
Final project report and presentation slides due on 12/13 midnight	

Homework Assignments

There will be three homework assignments during the semester, each designed to provide hands-on experience using coding agents for realistic software engineering tasks. Students may choose any coding agent they prefer. Detailed instructions, submission requirements, and grading rubrics will be posted on Brightspace.

Homework 1: Build a Web Application with a Coding Agent

Use a coding agent to develop a small web application with a graphical user interface, backend logic, and persistent storage. The purpose of this assignment is to gain experience using coding agents for end-to-end software development, including UI implementation, database integration, API integration, debugging, and deployment.

This semester, you will build an AI Research Assistant that helps users search, organize, and analyze research papers. This application should allow users to search for papers, save papers to a local library, upload PDF papers, and use an LLM to summarize papers and answer questions about their content. You are expected to submit the link to the GitHub repository of this application, a demo video, and a report that reflects on your experience of using the coding agent to develop this application, e.g., where the coding agent works well, where it struggles, etc. HW1 is due on Oct 10 midnight.

Homework 2: Generate Test Cases with a Coding Agent

This assignment asks you to use a coding agent to generate and improve test cases for the web application developed in Homework 1. The goal is to explore how effectively coding agents can understand an existing application and generate tests for its functionality.

You will evaluate the generated tests using measures such as code coverage, test correctness, and fault-detection capability, and will iteratively improve the tests by providing additional context or execution feedback to the coding agent. You are expected to submit a report that points the instructor to the final test suite in the GitHub repository,

describes the test coverage and fault-detection results, and reflects on their experience of using the coding agent to generate test cases. HW2 is due on Nov 8 midnight.

Homework 3: Fix Bugs in a Real-World Software Repository

Unlike Homework 1 and Homework 2, where you work with software you developed yourself, this assignment requires you to work with an unfamiliar codebase written by others. Working with existing software written by others is a common part of professional software engineering. When you join a company after graduation, you are often expected to quickly become familiar with large, unfamiliar repositories and diagnose and fix bugs in others' code. This assignment is designed to provide practical experience with how coding agents can support that workflow.

You will be given a GitHub issue from a well-known GitHub repository. You will use a coding agent to explore the repository, understand relevant components and dependencies, reproduce and localize the bug, implement a fix, and validate the resulting patch. You are expected to submit the test case(s) that reproduce the GitHub issue, the patch that fixes it, and a report that reflects on your experience of using the coding agent to fix the issue. HW3 is due on Nov 29 midnight.

Course Project

You are expected to work on a course project either alone or in groups (2 to 4 students in a group). You can pick any topics related to AI-based software engineering, e.g., a new LLM-based code generation pipeline, a new code review agent, a new software testing agent, etc. Between Week 1 and Week 6, please stop by during office hours to discuss your project ideas with the instructor to get early feedback on the relevance, novelty, feasibility, and significance of your ideas.

A short project proposal is due on Oct 4 midnight (Week 6). This proposal should describe the project idea and the motivation of this idea (e.g., why it is important, why we should care). The proposal could be any length but no longer than 4 pages. It will be evaluated based on the quality and clarity of the idea, not the length of your writing.

A mid-point project summary is due on Nov 1 midnight (Week 10). This summary should describe the envisioned approach/methodology/design as well as which parts have been done so far. The summary could be any length but no longer than 4 pages.

In Week 15 and Week 16, each team will deliver a presentation of their project. The presentation will be about 20 minutes. You will get another 5 minutes for Q&A. We will

draw lots to decide the presentation order. Your presentation slides need to be submitted on Brightspace by Dec 13 midnight.

The final project report is due on Dec 13 midnight. Your project report should be in the double-column ACM conference format. The instructor will provide both a word and latex template. Your final project report should not exceed 10 pages, not including references and appendices. Feel free to reuse sections from your mid-point project report in your final report.

Your final report should be structured like a conference paper. It should contain:

- Abstract
- A well-motivated introduction
- Related work with proper citations
- Description of your methodology
- Evaluation results
- Discussion of your approach and threats to validity
- Conclusions and future work

Please also include a link to your GitHub repository in your final report. Your GitHub repository should contain the source code and data used in your project, as well as a README file to describe how to reproduce the results.

Important dates:

- Project proposal due on 10/4 midnight
- Mid-point project summary due on 11/1 midnight
- Final project presentation on Week 15 and Week 16
- Final project report and presentation slides due on 12/13 midnight

Quizzes

We will have five pop quizzes during this course. Each quiz should take about 5 to 10 minutes. Each quiz will assess your understanding about the research topics covered in the previous two or three weeks. The quizzes will include True or False questions, multiple-choice questions, and/or open-ended questions. To prepare for the quizzes, make sure you have reviewed the lecture slides.

The instructor cannot accommodate quizzes on a different date, except for verified reasons. For common illnesses, a doctor’s note needs to be emailed to the instructor. For other exceptional cases covered by University policy, such as grief/bereavement, military service, jury duty, or parenting leave, the student or the student’s representative should

contact or go to the Office of the Dean of Students website to complete appropriate forms for instructor notification.

Academic Integrity

Incidents of academic misconduct in this course will be addressed by the course instructor and referred to the Office of Student Rights and Responsibilities (OSRR) for review at the university level. Any violation of course policies as it relates to academic integrity will result minimally in a failing or zero grade for that assignment, and at the instructor's discretion may result in a failing grade for the course. In addition, all incidents of academic misconduct will be forwarded to OSRR where university penalties, including removal from the university, may be considered.

Academic integrity is one of the highest values that Purdue University holds. Individuals are encouraged to alert university officials to potential breaches of this value by either emailing integrity@purdue.edu or by calling 765-494-8778. Concerns may be reported anonymously.

AI Policy

The use of generative AI tools, including large language models and coding agents, is permitted and encouraged in this course except for quizzes and exams, since a central objective of this course is to teach students how to use AI tools effectively and responsibly for software engineering tasks. When required by an assignment or course project, students must disclose the AI tools they used and describe how those tools contributed to their work.

Students must use AI tools in accordance with Purdue University policies and course requirements. Students remain responsible for the correctness, quality, originality, and integrity of all submitted work and are expected to critically evaluate, test, and revise AI-generated outputs. Confidential, proprietary, personally identifiable, or otherwise sensitive information should not be entered into external AI tools without appropriate authorization. Course materials, assessments, and other copyrighted or proprietary content should not be uploaded to AI tools without permission.

Failure to comply with these requirements, including prohibited or undisclosed use of generative AI when disclosure is required, may result in academic consequences consistent with Purdue University academic-integrity policies and course requirements.

Nondiscrimination Statement

Purdue University is committed to maintaining a community that recognizes and values the inherent worth and dignity of every person; fosters tolerance, sensitivity, understanding, and mutual respect among its members; and encourages each individual to strive to reach his or her potential. A hyperlink to Purdue's full Nondiscrimination Policy Statement is included in the Academic Resources table on your Brightspace homepage.

Accessibility and Accommodations

Purdue University is committed to providing accessible learning experiences for all students. All course materials and activities will be made accessible in accordance with the Digital Instructional Materials – Accessibility Checklist.

Students who require disability-related accommodations should contact the Disability Resource Center (DRC). Approved accommodations will be communicated to the instructor through a Course Accommodation Letter. Students are responsible for requesting their accommodations, and instructors are responsible for implementing them. Students should contact the DRC early—ideally at the beginning of the semester—to allow sufficient time for review and approval of their request. Do not wait until a few days before an exam to reach out to the DRC.

Students whose Course Accommodation Letter (CAL) includes testing accommodations must first release their CAL to the instructor and then schedule their exams with Purdue Testing Services. While the university states that exams must be scheduled at least four days (96 hours) before the exam date listed in the syllabus, students are strongly advised to schedule their accommodated exams at least two weeks in advance, as there may be limited availability at Purdue Testing Services for the requested exam date and time. For final exams, students must schedule by the Friday before Quiet Period. The instructor will provide the exam to Purdue Testing Services, which will administer the exam and return it to the instructor for grading. If Purdue Testing Services is at capacity or otherwise unable to proctor an exam, students should notify the instructor immediately so that alternative arrangements can be made. Students who do not follow this process or meet the required scheduling deadlines may not be able to receive their testing accommodations for that exam.

For questions about accommodations or accessibility services, contact the Disability Resource Center at drc@purdue.edu or 765-494-1247.

Mental Health/Wellness Statement

Purdue University is committed to supporting students' mental health and well-being. College life can be challenging, and students may experience stress, anxiety, depression, or other concerns that affect their academic work and daily life. Seeking help is an important life skill, and students are encouraged to make use of the many well-being resources available at Purdue.

Students can find an overview of campus support through the Office of the Dean of Students Well-Being Resources and the Health and Well-Being Resources available on the Brightspace homepage. Purdue Counseling and Psychological Services (CAPS) also provides Therapy Assistance Online (TAO), a free and confidential web- and app-based resource with self-guided tools, short videos, exercises, and other resources for managing anxiety, depression, stress, and related concerns.

If you are concerned about a fellow student and are unsure how to help, anyone in the Purdue community may submit a Student of Concern Report to connect the student with appropriate support.

Emergency Preparedness

Purdue University provides guidance for responding to emergencies through the Emergency Procedures Guide. Students are encouraged to review this guide and sign up for Purdue ALERT notifications. During class, please keep your phone available to receive emergency alerts when necessary. In an emergency, follow Purdue ALERT instructions and directions from University personnel and the instructor. Building-specific evacuation and shelter information is available through Purdue's Building Emergency Plans webpage.
If an emergency disrupts normal course operations, Brightspace will serve as the primary point of contact for course announcements, updates, and changes to course requirements or deadlines. Unless otherwise instructed, assignments should continue to be submitted through Brightspace. Course requirements, deadlines, or instructional arrangements may be adjusted as necessary based on University guidance and the circumstances of the emergency.

Disclaimer

This syllabus is subject to change. You will be notified of any changes as far in advance as possible via an announcement on Brightspace. Monitor your Purdue email daily for updates.