2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #50534

WIP: Use of Reflective Writing to Guide the Use of Generative AI Tools in a Software Engineering Course

Dr. Bruce R Maxim, University of Michigan - Dearborn

Bruce R. Maxim has worked as a software engineer, project manager, professor, author, and consultant for more than forty years. His research interests include software engineering, human computer interaction, game design, social media, artificial intelligence. Dr. Maxim's professional experience include managing research information systems at a medical school, directing instructional computing activities for a medical campus, and working as a statistical programmer in a department of epidemiology. He served as the chief technology officer for a game development company.

Anwar Ghammam, University of Michigan - Dearborn

Anwar Ghammam, PhD is an assistant Professor in the Department of Computer and Information Science at the University of Michigan-Dearborn. Her research sits at the intersection of Artificial Intelligence and Software Engineering, with a focus on improving software quality, software maintenance and DevOps through AI-driven techniques. Her work has been published in leading venues such as ICSE, ICSME, MSR, TOESM, and she actively contributes to advancing AI-powered solutions for modern software development challenges.

Mayuri Roy, University of Michigan

Mayuri Roy is a Senior Software Engineer with over 10 years of experience at JP Morgan Chase and DBS Bank, specializing in AI-powered microservices and scalable ML systems. She is currently pursuing a Master's in Artificial Intelligence at the University of Michigan, where her thesis research focuses on the efficiency trade-offs between traditional machine learning and fine-tuned large language models for large-scale financial risk prediction. Specifically, she is investigating whether the predictive accuracy gains from LLMs justify their computational cost when processing billions of loan records, using techniques such as LoRA, QLoRA, and full fine-tuning on architectures like BERT, Llama-3, and Mistral-7B. Her work contributes a practical decision framework for deploying LLMs in production financial systems where scale, latency, and cost are critical constraints. Her expertise spans Python, PyTorch, TensorFlow, Java, and Kubernetes-based MLOps. She bridges enterprise engineering with cutting-edge AI to deliver intelligent, secure, and business-aligned solutions in the financial services domain.

Sindhuja Gandi, University of Michigan - Dearborn

Sindhuja Gandi is a Software Consultant and recent graduate of the Master of Science program in Computer and Information Science at the University of Michigan–Dearborn. She has industry experience in business analysis and enterprise systems. Her research interests include generative artificial intelligence, machine learning, and natural language processing. Her current work focuses on integrating Generative AI tools into software engineering education, with an emphasis on reflective writing and prompt engineering for software design, testing, and project management.

Mr. Atharva Kishor Gujarathi, University of Michigan - Dearborn

Atharva Kishor Gujarathi is a graduate student pursuing a Master of Science in Data Science at the University of Michigan-Dearborn. His research interests include generative AI, software engineering processes, data engineering, and machine learning applications in real world systems. He has industry experience as a software development engineer, where he worked on full stack web applications, data driven platforms, and scalable software systems. His current work explores the integration of generative AI tools in software engineering education and development workflows.

WIP: Use of Reflective Writing to Guide the Use of Generative AI Tools in a Software Engineering Course

Abstract

This paper describes the authors' experiences introducing the use of generative AI (GenAI) in a software engineering course. On our campus hybrid courses have both in-person and online students participating in a single course taught by the same instructor. The authors argue that use of GenAI in commercial software development is inevitable. We believe that students should be given opportunities to experiment with GenAI tool use on some software engineering activities. In this paper, the authors discuss how they integrated GenAI tools to help students complete several software engineering process tasks. The authors created a prompt engineering module to introduce students to the use of ChatGPT. They adapted eight active learning modules from a junior-level course to include the use of ChatGPT. The course was delivered using a flipped classroom. Students were encouraged to use GenAI in the completion of software design, testing, and project management tasks. Students were asked to use reflective writing to assess the quality of AI-generated artifacts and identify unsatisfactory work products. Students were allowed to use ChatGPT as part of a seven-week term project, where student teams develop web-based software engineering tools. Their use of GenAI programming tools was not assessed formally. Most students, both in-person and online, successfully completed the team project, The students demonstrated comparable levels of engagement across both in-person and online course delivery modalities. In-person students reported that ChatGPT helped them complete software design and process tasks more efficiently. Analysis of student survey responses and course analytics reinforce the efficacy of our structured approach for teaching responsible GenAI use in software project courses.

Background

Generative Artificial Intelligence (GenAI) represents a cutting edge field within artificial intelligence that focuses on developing algorithms capable of producing content that mirrors human-like outputs, such as text, images, audio and video [1], [2]. One of the most promising and widely popular applications using GenAI today is ChatGPT [3], a GenAI-based large language model (LLM) developed by OpenAI [3], which has gained global recognition for its ability to handle complex language processing tasks and generate responses in an interactive conversational way. The recent advent of ChatGPT, with its fast growing user base of approximately 700 million users [4] and free version offerings, has made it easily accessible [5], thus introducing the domain of GenAI to a large general audience of users. Additionally, it has spurred growing interest in both research and industry, driving a wide range of applications across multiple fields. [6].

The shift to the use of code libraries has had a profound impact on the efficiency of commercial software development. This higher level of code abstraction makes it difficult for novice programmers to determine the root causes of defects that may only be observed in the run-time behavior of the software. AI tools may be a useful way to augment developer capabilities in several phases of the software development life cycle (SDLC) besides coding (i.e. requirements analysis, design, testing, maintenance). Likely, significant interaction between software developers and AI tools will be necessary for some time to come [7].

GenAI tools can help to increase the efficiency and productivity of software developers. Software engineers want some freedom to configure their development environment to allow them to exercise their own creativity. Software engineers have a deeper and more holistic understanding of the SDLC that extends beyond the technical assistance provided by an AI tool. This expertise is the key to ensuring the quality of a software product. Engineers rely on their own creativity when refactoring code, they have not seen before or when reusing code in clever ways [8].

Recent advances in GenAI tools have led to their widespread adoption in professional software engineering practice, where they are increasingly embedded in developers' daily workflows [9-16]. Empirical evidence from industry surveys indicates that LLM-based tools are commonly used across multiple phases of the software development lifecycle such as implementation, maintenance, verification, and knowledge-intensive tasks [9,10].

In software engineering practice, GenAI tools are most frequently applied to code-based tasks, including code generation [13], code completion [14], explanation [15], and documentation [16]. Practitioners report productivity gains through reduced development time, and improved support when working with unfamiliar or legacy codebases [9,10]. Beyond implementation, recent studies demonstrate the applicability of LLMs to software maintenance and quality-related tasks such identifying code smells and supporting refactoring decisions particularly when prompts are enriched with contextual information and software metrics [11].

Other studies have explored the use of GenAI in requirements engineering, showing that prompt-based LLMs can support the classification and analysis of software requirements with little or no task-specific training [12]. They achieved performance comparable to supervised models while significantly reducing the need for large, annotated datasets, making them particularly attractive for industrial settings where labeled data is limited or costly to obtain.

Among the many areas influenced by GenAI, education has seen some of the most visible changes, particularly in software engineering. Tools such as ChatGPT are now commonly used by students as personal learning assistants, helping them quickly access information, ask questions, and generate or explain code based on specific needs [17]. With the ability to answer a wide range of questions, these tools encourage more independent learning, allowing students to delve deeper into subjects that interest them, reinforcing classroom learning and promoting curiosity [18,19].

However, the growing popularity of these tools has also raised concerns about their possible impact on students' learning and educational achievements. Some educational institutions have even started taking measures to prohibit the use of certain AI platforms to maintain conventional teaching methods and skills [20]. One of the major reasons behind restricting the usage of ChatGPT by educational institutions tends to be concerns related to direct copy-pasting of content, which may decrease student engagement and diminish critical thinking skills. Educators face challenges in preventing the use of ChatGPT for completing assignments, coding tasks, and homework, which could lead to issues like plagiarism, overreliance on technology, misinformation, and ethical problems [5]. Despite efforts to limit its usage in the educational domain, its widespread availability and easy access make it challenging to control its adoption by students [21].

Based on the authors' experience with teaching and ChatGPT, we believe that rather than banning ChatGPT, instructors should teach students how to utilize such tools effectively to enhance their learning experiences. We believe the advantages offered by GenAI tools in the educational domain outweigh the disadvantages, since their widespread usage is almost inevitable. Research has demonstrated that tools like ChatGPT can significantly improve student learning by acting as a personal tutor, providing immediate feedback, and generating new ideas [22], [23].

GenAI may have untapped potential due to students' limited knowledge of how to engage with these tools effectively. One study [24] found that despite students' intentions to use ChatGPT, there is a noticeable gap in their ability to engage with it effectively. Additionally, prior research indicates that enhancing students' interaction skills with large language models (LLMs) is crucial for maximizing the educational benefits, such as increased motivation and engagement [25], and enhancing their learning interest [26].

We believe that learning can be enhanced if students learn to make effective use of prompt engineering and to critically assess the resulting GenAI recommendations. Research on student learning shows many benefits to the use of reflective writing in problem solving situations [27]. Therefore, rather than restricting access to GenAI tools like ChatGPT, educators should focus on equipping students with the necessary skills to leverage these tools effectively for their academic improvement. This approach underpins the motivation for this work. This paper makes the following contributions:

1. We created a GenAI video module on prompt engineering that can be completed asynchronously outside of class time. This makes it useful to online students and to in-person students participating in a flipped classroom.

2. We created eight active learning exercises that utilize GenAI tools and asked students to formulate ChatGPT prompts requesting suggestions for completing several software engineering design and project management tasks. Students were asked to reflect on each recommendation and correct any deficiencies in the artifacts created by ChatGPT.

3. We asked students to reflect on their experience using ChatGPT as a software engineering design and project management tool. How effective was ChatGPT in suggesting solutions? Did using ChatGPT enhance their understanding of the software engineering processes? What were the challenges or benefits of adding ChatGPT to their workflow?

In this paper, the authors discuss how they integrate GenAI tool use in software process tasks. The authors created a prompt engineering module and adapted eight active learning modules from a junior-level software engineering course to include the use of ChatGPT. The course was delivered using active learning in a flipped classroom. Student assignments in CIS 375 (Software Engineering 1) routinely incorporate the use of reflective writing.

Active Learning

Engineering educators regard experiential learning as the best way to train the next generation of engineers [28]. It is reasonable to believe that the soft skills practiced in active learning classrooms can improve the capabilities of software engineering students and better prepare them for their capstone projects [29]. Active learning is "embodied in a learning environment where the teachers and students are actively engaged with the content through discussions, problem-solving, critical thinking, debate, and a host of other activities that promote interaction among learners, instructors and the material" [30]. Prince defines active learning as any classroom activity that requires students to do something other than listen and take notes [31]. Active learning opportunities can complement or replace lectures to make class participation more interesting to students. Active learning using a flipped classroom approach can also foster development of an attitude of lifelong learning among students [32].

Active learning helps students develop problem-solving, critical reasoning [33], and analytical skills, all of which are valuable tools that prepare students to make better decisions, become better students, and better employees [22]. Raju and Sankar undertook a study to develop teaching methodologies that could bring real-world issues into engineering classrooms [34]. The results of their research led to recommendations to engineering educators on the importance of developing interdisciplinary technical case studies that facilitate the communication of engineering innovations to students in the classroom.

Active learning helps students learn by increasing their engagement in the educational process [35], [36]. The group work that often accompanies active learning instruction helps students develop their soft skills [37]. Some instructors believe that the project activities inherent in real-world software development encourage students to improve their written and oral communication skills [38].

Day and Foley used class time exclusively for exercises by having their students prepare themselves through the study of materials provided online [39]. Research suggests that the success of flipped classroom approaches depends on the nature of the course being taught. The investment in time required for instructors to develop quality out-of-class materials and in-class active learning experiences can be substantial [40]. The instructor's time commitment becomes greater if both in-person and online students are in the same course.

The authors chose to make use of a flipped classroom approach for the software engineering course discussed in this paper. Students viewed short video lectures before beginning the class activities.

Project-based Learning

Project-based learning (PBL) is a dynamic classroom approach in which students actively explore real-world problems and challenges and acquire transferable knowledge. PBL has consistently demonstrated it can lead to positive learning outcomes such as self-directed learning habits, critical thinking skills, and deep disciplinary knowledge while engaging students in collaborative, authentic learning situations [41]. While PBL was first incorporated into medical school curricula in 1969, it is currently used in a wide variety of courses [42]. For instance, within the field of engineering, Warnock and Mohammadi-Aragh investigated the impact of PBL on student learning in a biomedical materials course and found that students made significant improvements in their problem-solving, communication, and teamwork skills [43].

PBL has been used in senior level engineering courses with the same positive results [44]. Although students in one PBL software engineering course reported that the projects were more time intensive than a typical course project, they were receptive to the approach since they thought it was related to the professional environment and provided them with opportunities to relate theory and practice. This contrasted with students taught using a traditional lecture and project approach to the course who viewed completing a traditional course project more negatively [45]. Each of our project courses contains a significant group project that requires several weeks to complete

Student Engagement

Student engagement refers to the degree of interest and attention shown in course activities. Student engagement can be a predictor for course completion and retention rates [46]. Active learning techniques such as think-pair-share exercises [47], pair programming [48], peer instruction [49], and flipped classrooms [50] have been demonstrated to increase student engagement [35]. Many of these interventions are used for introductory level instruction, primarily to address broadening participation in large classes [51].

In software engineering courses, the use of real-world, community-based projects may be an effective way to engage students with meaningful problem solving while teaching them software engineering concepts [52]. Students often become more invested in their projects when they see that their products are more than simply paper designs. We made sure that the daily course activities help students practice skills they need to complete the larger team projects. We surveyed online and in-person students to see if there were differences in their perceived levels of engagement with the course materials.

Reflective Writing

Research on student learning shows many benefits to the use of reflective writing in clinical or professional experiences. This suggests its use as an authentic assessment technique. Students asked to reflect on their learning experiences are better able to retain and transfer their learning to new contexts. The act of reflecting requires retrieval, elaboration, and generation of information can make learning more durable for students [53].

Promoting reflective thinking is important to helping learners develop strategies to apply added information to unpredictable situations in real life. Knowledge is created through the transformation of experience. Reflective writing could be one method for promoting reflective thinking that allows learners to consider their experiences and transform them into knowledge that can be applied in new contexts. Reflective writing is an effective method for promoting metacognitive thinking. Reflective writing can be a useful tool for communication between students and mentors in experiential learning activities [54].

Reflection provides opportunities for students to think about their performance, consider which strategies were effective, and contemplate how to improve their process. In work contexts, individuals who engage in reflection have lower error rates when learning new skills [55]. In asynchronous on-line courses, student reflective activities are important since students and instructors do not have opportunities for face to face communication. If gathered over time, student writings can guide instructors in refreshing course content. If reflections are collected over the course delivery, students can use them to monitor their own progress. In face to face classes, reflective writing can be used to initiate in-class discussions in small group activities [56].

In active learning, students working in small groups or by themselves, are required to summarize the lessons learned from each hands-on assignment. If students are assigned to read a textbook before coming to class, it may be helpful to have them summarize their reactions to the reading in writing. Writing critiques of student presentations in-class also encourages the development of critical thinking, which is a valuable lifelong learning skill. It can be time consuming for instructors to grade large numbers of reflection documents, so this effort can be reduced by making use of peer evaluation strategies or allowing the submission of group reflection documents.

Instructional Delivery

One of the authors teaches an undergraduate software engineering course CIS 375. This course is offered in-person on campus and merged with an online section that allows enrolled students to complete the course requirements asynchronously online. The authors determined that a PBL (project-based learning) approach was well suited for the delivery of junior level software project courses.

We use the class activities to motivate students to design software products and use software engineering techniques to solve real-world programming problems. The investigators included small group activities with the expectation that students would provide written or oral summaries (either live in-person or virtually using video) of the strategies used to complete their tasks and their lessons learned. Online students need to complete group work asynchronously. Both groups of students reflect on lessons learned from exercises as written or oral presentations.

Course Overview: CIS 375 Software Engineering 1

The junior level software engineering course, CIS 375 (Software Engineering 1), offered by the Computer and Information Science (CIS) department is organized as a fifteen week, four credit-hour course, meeting two days a week (Table 1).

Table 1: Activity Question Rubric

Topic	Activity
Course Introduction	Ice breaking activity. Group chapter reflection
Process Models	Waterfall process - paper airplane construction activity Process improvement game Scrum trigger video - group viewing with discussion
Requirements Engineering	Requirements video - group viewing with discussion. Requirements ambiguity exercise User story and use case exercises
Requirements Modeling	CRC modeling exercise UML modeling exercise
Reviews and Inspections	Prompt Engineering Module SRS document review exercise Inspection video - group viewing with discussion Formal code inspection review – roleplay (activity 8)
Cost and Time Estimation and Project Scheduling	Project cost estimation exercise Project scheduling exercise (activity 10)
Risk Management and Software Architecture	Project risk management exercise (activity 11) Architecture tradeoff analysis exercise
SRS Peer Review and Configuration Management	Software requirements document with peer review Version control system assessment exercise
User Experience Design 1	Web page usability assessment exercise UI Design Pattern Creation (activity 15)
User Experience Design 2	Roleplay web user interface design (activity 16) Project planning document peer reviews
Software Quality	Defect life cycle exercise Creating test cases from requirements (activity 18)
Testing – part 1	Test plan analysis and critique exercise (activity 19) Software design document peer review
Testing – part 2	Cost effective testing activity. Web page accessibility and redesign exercise (activity 22)
Security Engineering	Security inspection video - group view with discussion Test plan peer review
Term Project Delivery	Tiny Tool Prototype Demonstrations

The in-person meetings are recorded and made available for viewing after the class meetings end. Students in both sections had access to prerecorded video lectures. Students are allowed to enroll in the asynchronous section of CIS 375 and complete the course activities by themselves at home. For most online students, viewing the video lectures and reading the textbook were their primary sources of instruction

Prior to attending class meetings, the in-person students were expected to read the sections of the required course textbook [57] and view two 20-minute video lectures created as part of the weekly course module. The online students were expected to do the same.

During Fall 2024, students were expected to complete the course activities without any use of ChatGPT. This cohort had 38 students (30% female) attending the in-person section of the CIS 375 and 20 students (23% female) registered in the online section. These students were organized into 12 project teams for outside of class project work. During Fall 2025, CIS 375 students were given instruction on the use of prompt engineering and completed laboratory activities requiring the use of ChatGPT. This cohort had 45 students (24% female) attending the class in-person and 25 (18% female) students registered in the online section. These students were organized into 14 project teams for outside of class project work.

Laboratory Activities

The laboratory activities make use of just in time learning. The in-person (FF) activities in CIS 375 are taught using a flipped classroom. Students watch lecture videos before beginning the class activities. The activities often consisted of small group design or problem-solving activities. Online (AO) students are asked to complete similar activities at home by themselves.

All students are asked to write reflections on each weekly activity. Both in-person (FF) students and online (AO) students participate in peer reviews of work products produced by other students or teams. The sequence of GenAI activities appears in Figure 1.

Typical reflection questions for the GenAI exercises include:

- Did using GenAI make it easier to organize your thoughts or structure your answers? In what specific way?
- How did you verify that the AI-generated information or examples were accurate and relevant to your work?
- If you were to redo this task without GenAI, which parts would be most challenging for you and why?
- In future projects, which steps would you prefer to complete manually? Which would you prefer to use GenAI tools?
- How can GenAI tools be used responsibly to support (not replace) your own problem-solving process?

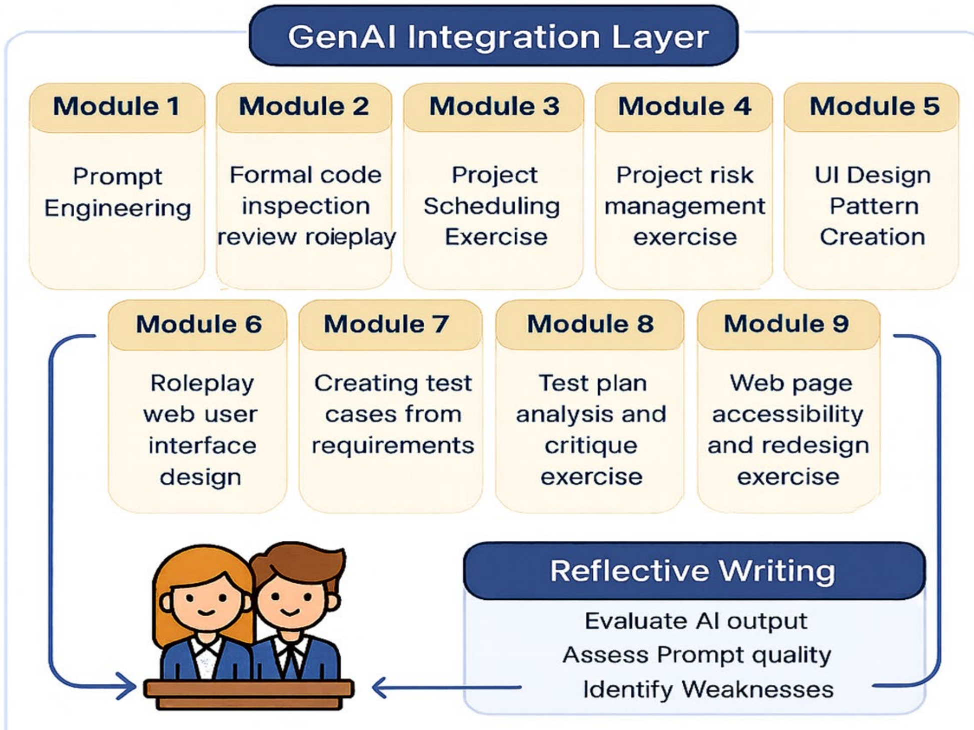

Figure 1: Sequence of GenAI Modules

In the formal code inspection activity, students viewed prompt engineering and code inspection videos before coming to class. Students organized into groups of 6 and asked ChatGPT to analyze the code for three C++ functions using a code quality checklist provided by the instructor. Once each student created an issues list, inspection roles were assigned: reader, recorded, facilitator, and inspector. Each group of students conducting the code inspection discussed the issues that really needed correction and the group made a consensus recommendation for the C++ functions.

In the scheduling activity, student groups of 3 or 4 used ChatGPT to assist in the creation of an agile project schedule. ChatGPT listed the project deliverables and milestones from a list of ATM system user stories. ChatGPT was used to create a draft project schedule based on four 4-week sprints assuming the usual Scrum ceremonies. ChatGPT was asked to create a work breakdown structure with durations for a typical sprint and represent it as an activity graph and Gantt chart.

For the risk management activity, students worked in groups of 3 or 4 to create a risk table and risk information sheets for the ATM system. ChatGPT listed suggested project, technical, and business risks that should be considered. ChatGPT suggested severity ratings for the consequences for each risk. ChatGPT was then asked to create risk table with columns labeled – risk, impact, probability and identify risks that need to be managed. Students identified their 3 most significant risks and asked ChatGPT to create risk information sheets to help document how to monitor and manage these risks.

Creating design patterns is hard for students. Students worked individually with ChatGPT to create paper prototypes and keystroke models for an alarm clock app from a set of simple user stories. Once done, students paired up and simulated their partner's paper prototype while completing the user stories. Two pairs of students joined up to form a group of four students and looked for features common to each prototype. Once identified, the group of four asked ChatGPT to create a design pattern that could be used by developers of other mobile apps.

The website design role play involved having students work in groups of four and use ChatGPT to create a list of features and user stories for two page web site from a set of clothing store business requirements. ChatGPT was used to create a list of functional and non-functional requirements which was passed to another team to create a prototype for the web site design. Each group took the requirements received and asked ChatGPT to create sketches of the web pages. The draft web pages were sent back to the requesting team for iterative feedback and revision requests.

GenAI was used for test case creation. Students worked in groups and asked ChatGPT to suggest strategies to test an online ordering system, conduct unit testing of an image processing system, and thread testing of the image processing library. Student groups asked ChatGPT to generate test cases for three requirements from a student registration system (password change, financial system access authorization, student account status verification for course registration). They were also asked to criticize and correct these suggestions.

Student groups were asked to use ChatGPT to perform a scenario analysis on a test plan (functionality tested and the adequacy of testing proposed). Groups used ChatGPT to evaluate test plan scope, resources, schedule, test case development, pass-fail criteria, and incident reports, summary reports, and metrics. Student groups were asked to present their findings to the class using 5 or 6 human created power point slides.

Student project groups asked ChatGPT to create a graphical storyboard for their web based tiny tool based on their design document. They asked ChatGPT to create a list of accessibility features the team should consider in their web tool design. Students were asked to consider which changes should be included. Teams asked ChatGPT to create a storyboard showing their tiny tool with their proposed accessibility changes added.

Online students were assigned versions of these activities that were designed to be completed by students working alone. Online students were allowed to work with partners. Sometimes online students were required to work with in-person students virtually. All project teams contained both in-person and online students.

Assessment

The authors created three research questions to compare the experiences of students taking recent offerings of CIS 375 under the in-person (FF) delivery of active learning materials to students taking the same courses under asynchronous online (AO) delivery.

RQ1: Does the delivery mode (FF or AO) affect student performance in taking either offering of CIS 375?

RQ2: Does the course delivery mode (FF or AO) affect student perceptions of engagement as reported on surveys taken in either offering of CIS 375?

RQ3: Does the course delivery mode (FF or AO) affect CIS 375 student perceptions of the effectiveness of GenAI to complete SE tasks as reported on the Fall 2025 survey?

RQ1 examines differences in student performance based on assessment measures in each course offering. RQ2 examines differences in student perceptions of engagement at the end of each offering of CIS 375. RQ3 examines student reactions to their experiences using ChatGPT for software engineering tasks other than programming.

Student Performance

Each of the course assignments (lab writeups, reading reflections, documents, and final project) was evaluated using Canvas rubrics designed by the instructor for each type of submission. Each rubric contains two to ten criteria; each scored from 0 to 5. The total course grade was based on the points earned on all assignments, including peer evaluations on the project performance score card (Figure 2).

Developer Name:			40 max
Team Meetings	Points	#	Earned
daily stand-up held (1 point per meeting attended)	1	0	0
backlog item moved to done (1 point for each item personally completed)	1	0	0
Software Development (must earn at least 5 points from this group)			
Programmer or UI/UX Programmer			
Programmer - 1 user story implemented as solo developer	5	0	0
Prorgammer - 1 user story completed using pair prorgamming (3 points each)	3	0	0
UI/UX - key screen implmented	3	0	0
UI/UX - key screen wire frame	2	0	0
Repository Manager, Document Manager			
Document manger - one hour of work	1	0	0
Repository manager - one hour of work	1	0	0
Tester, QA			
QA work - one hour of work	1	0	0
tester - test case written	1	0	0
tester - written test case executed and documented	1	0	0
		Total =	0

Figure 2: Final Prototype Score Card

Table 2 compares FF and AO student performance from Fall 2024. Table 3 compares FF and AO students from Fall 2025.

No statistical comparisons of student performance on the individual course assignments were made between students in the in-person (FF) section and the asynchronous online (AO) section in either offering, Students were allowed to skip 2 of 21 lab write-ups and 2 of 18 reading reflections. No statistical comparisons were made between students Fall 2024 and the Fall 2025 offerings of CIS 375.

In both offerings of CIS 375 the student sections (FF and AO) worked independently of each other with the students in-person completing the class assignments in-class or at least discussing them heavily before submitting together as a group. The online students worked independently and submitted their work without much or any interaction with their peers. The term projects in CIS 375 were completed by teams composed of both in-person and online students.

For Fall 2024 (Table 2), students in the FF section had slightly better overall grades (96.4% vs. 93.1% respectively) than students in the AO section of the course. The FF students had slightly lower number of missing labs (1.24 vs. 2.50) and number of missing reading reflections (0.84 vs. 1.15). The in-person students had slightly fewer late assignments (11.78 vs 11.95) and higher prototype contribution scores (39.24 vs. 37.25). One-tail Student t-tests comparing the FF and OA students revealed that the significant differences between these groups at the 95% confidence level only existed for course grades which favored the FF students.

Table 2: CIS 375 Student Performance Fall 2024

	FF (N=38)	**AO (N=20)**
Average Course Grade	96.4%	93.1%
Average Number of Late Assignments per Student	11.78	11.95
Average Number of Missing lab Assignments per Student	1.24	2.50
Average Number of Missing Reflections per Student	0.84	1.15
Average Prototype Score (40 max.) per Student	39.24	37.25

For Fall 2025 (Table 3) students in the FF section had slightly better overall grades (96.3% vs. 93.2% respectively) than students in the AO section of the course. The FF students had a lower number of missing labs (1.11 vs. 2.88) and late assignments (7.07 vs. 11.88). The in-person students had slightly more missing reflection assignments (0.71vs 0.63) and higher prototype contribution scores (39.02 vs. 36.33). One-tail Student t-tests comparing the FF and OA students revealed that only significant differences between these groups at the 95% confidence level were for the average course grades, number of late assignments, and number of missing lab assignments favoring the FF students.

Table 3: CIS 375 Student performance Fall 2025

	FF (N=45)	AO (N=25)
Average Course Grade	96.3%	93.2%
Average Number of Late Assignments per Student	7.07	11.88
Average Number of Missing lab Assignments per Student	1.11	2.88
Average Number of Missing Reflections per Student	0.71	0.63
Average Prototype Score (40 max.) per Student	39.02	36.33

Student Engagement Survey Comparisons

Students were surveyed during the final weeks of the Fall 2024 (Table 4) and Fall 2025 offerings (Table 5) to gather their perceptions of engagement with the active learning materials.

Table 4: CIS 375 Final Survey Student Perceptions of Engagement for Fall 2024
FF (N = 34) vs. AO (N = 18)

Survey Statement	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	Course (Median)
1. There were opportunities for me to actively engage in learning	0 2 (11%)	0 1 (6%)	1 (3%) 2 (11%)	8 (24%) 7 (39%)	25 (74%) 6 (33%)	FF (5.0) AO (4.0)
2. Course activities were useful way to learn	0 2 (11%)	0 0	0 3 (17%)	13 (38%) 7 (39%)	21 (62%) 6 (33%)	FF (5.0) AO (4.0)
3. Course activities let me apply what I learned	0 2 (11%)	0 0	4 (12%) 1 (6%)	8 (24%) 8 (44%)	22 (65%) 7 (39%)	FF (5.0) AO (4.0)
4. Course is an example of active learning	0 2 (11%)	0 1 (6%)	0 2 (11%)	12 (35%) 6 (33%)	22 (65%) 7 (39%)	FF (5.0) AO (4.0)
5. I felt more engaged during the activities than during video lectures	0 2 (11%)	0 0	1 (3%) 3 (17%)	5 (15%) 6 (33%)	28 (82%) 7 (39%)	FF (5.0) AO (4.0)

Students in all sections rated a series of instructor designed survey statements using a five-point Likert scale (1 = Strongly Disagree to 5 = Strongly Agree). The Mann-Whitney U test (one-tailed, $\alpha = 0.05$) was used to compare response distributions between the in-person (FF = Face-to-Face) and asynchronous online (AO) sections. The median test evaluated the hypothesis that FF student responses would be more favourable than AO responses for each survey item.

For Fall 2024 (Table 4), the analysis revealed no statistically significant differences between FF and AO students on any of the five core engagement questions. While median responses for FF students were consistently at 5.0 (Strongly Agree) compared to medians of 4.0 for AO students, these distributions were not significantly different at the 95% confidence level. This indicates a statistically equivalent perception of the active learning model's effectiveness across both delivery modes.

For Fall 2025 (Table 5), the results for the five core engagement questions again showing no significant differences. However, the medians for both FF and AO students are consistently 5.0. This consistency reinforces the finding that the course's active learning pedagogy translates successfully to both instructional formats. For 2025 survey the authors added survey questions evaluating student perceptions of Generative AI use in the lab activities (Table 6).

Table 5: CIS 375 Final Survey Student Perceptions of Engagement for Fall 2025 FF (N = 41) vs. AO (N = 25)

Survey Statement	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	Course (Median)
1. There were opportunities for me to actively engage in learning	4 (10%) 2 (8%)	2 (5%) 0	1 (2%) 1 (4%)	8 (20%) 8 (32%)	26 (63%) 14 (56%)	FF (5.0) AO (5.0)
2. Course activities were useful way to learn	4 (10%) 2 (8%)	1 (2%) 0	0 0	8 (20%) 10 (40%)	28 (68%) 13 (52%)	FF (5.0) AO (5.0)
3. Course activities let me apply what I learned	4 (10%) 2 (8%)	2 (5%) 0	0 1 (4%)	5 (12%) 11 (44%)	30 (73%) 11 (44%)	FF (5.0) AO (5.0)
4. Course is an example of active learning	4 (10%) 2 (8%)	2 (5%) 0	0 1 (4%)	5 (12%) 12 (48%)	30 (73%) 10 (40%)	FF (5.0) AO (5.0)
5. I felt more engaged during the activities than during video lectures	4 (10%) 2 (8%)	1 (2%) 0	0 3 (12%)	8 (20%) 9 (36%)	28 (68%) 11 (44%)	FF (5.0) AO (5.0)

Table 6: CIS 375 Final Survey Student Perceptions of GenAI for Fall 2025
FF (N = 41) vs. AO (N = 25)

Survey Statement	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	Course (Median)
1. Generative AI tools helped me understand the SE content	4 (10%) 2 (8%)	0 1 (4%)	7 (17%) 7 (28%)	13 (32%) 9 (36%)	17 (41%) 6 (24%)	FF (4.0) AO (4.0)
2. GenAI suggestions improved the quality of my deliverables	4 (10%) 2 (8%)	0 1 (4%)	4 (10%) 5 (20%)	18 (44%) 8 (32%)	15 (37%) 9 (36%)	FF (4.0) AO (4.0)
3. Using GenAI saved me time and made the assignments easier	4 (10%) 2 (8%)	0 0	3 (7%) 3 (12%)	8 (20%) 12 (48%)	26 (63%) 8 (32%)	FF (5.0) AO (4.0)
4. I was confident in evaluating AI content as correct and relevant.	4 (10%) 2 (8%)	0 0	5 (12%) 7 (28%)	10 (24%) 8 (32%)	22 (54%) 8 (32%)	FF (5.0) AO (4.0)
5. Using GenAI increased my confidence in completing SE tasks.	4 (10%) 2 (8%)	1 (2%) 1 (4%)	7 (17%) 7 (28%)	10 (24%) 9 (36%)	19 (46%) 6 (24%)	FF (4.0) AO (4.0)
6. Using GenAI encouraged me to critically evaluate suggestions	4 (10%) 2 (8%)	1 (2%) 1 (4%)	6 (15%) 6 (24%)	8 (20%) 9 (36%)	22 (54%) 7 (28%)	FF (5.0) AO (4.0)
7. I was aware of and followed the ethical guidelines for using GenAI in coursework	4 (10%) 2 (8%)	0 1 (4%)	1 (2%) 3 (12%)	10 (24%) 6 (24%)	26 (63%) 13 (52%)	FF (5.0) AO (5.0)
8. Using GenAI helped me focus more on design and reasoning, rather than repetitive or mechanical tasks	4 (10%) 2 (8%)	1 (2%) 0	2 (5%) 3 (12%)	6 (15%) 9 (36%)	28 (68%) 11 (44%)	FF (5.0) AO (4.0)

Only one statistically significant difference ($p = 0.033$) emerged: FF students reported significantly stronger agreement than AO students that "Using GenAI saved me time and made the assignments easier to complete." While most students claimed to respect and follow ethical guidelines. The authors did not assess the use of GenAI on the term project.

An analysis of response trends revealed that FF students also reported higher percentages of agreement on nearly all other GenAI perception items - including AI's help in understanding content ($p = 0.069$) and confidence in evaluating output ($p = 0.051$) - though these differences did not reach the 95% threshold for statistical significance. This pattern suggests that while all students viewed GenAI as a beneficial course tool, the in-person setting may have fostered a more uniformly positive and efficient experience with the technology.

Open-ended survey questions asked students to indicate what the most and least engaging parts of the course were. The FF students most frequently cited collaborative, hands-on activities such as in-class group projects, labs, and presentations. AO students also emphasized the value of group projects but specifically noted the importance of connecting with their teammates and the applied nature of the term project work. Both FF and AO students identified individual reading reflections as their least engaging activity, often attributing this to its solitary nature in contrast to the preferred collaborative work.

Related Works

Recent research has examined the integration of GenAI, particularly LLMs such as ChatGPT, as adaptive learning supports in educational contexts. Across disciplines, LLMs have been positioned as tools capable of providing personalized explanations, iterative tutoring, and immediate feedback tailored to student needs [58–60]. In computing education, empirical studies report gains in computational thinking, confidence, and motivation when students use ChatGPT to assist with programming tasks [61], [62]. In project-based learning settings, GenAI has been received as a collaborative assistant that supports idea refinement, debugging, and incremental solution development [63], [64].

Within software engineering (SE) education, this adaptive support has embedded into structured quality-focused activities. AlOmar et al. [65] integrate ChatGPT into static analysis and code review exercises, demonstrating that learning effectiveness depends on prompt specificity and students' critical evaluation of AI-generated explanations. Similarly, Menolli et al. [66] examine LLM-assisted refactoring in undergraduate courses, reporting that students addressed 86% of identified code smells, with ChatGPT contributing to 66% of the refactoring effort. Importantly, deeper conceptual understanding emerged when students reflectively assessed AI suggestions rather than applying them mechanically. Together, these studies frame LLMs as scaffolds that enhance reasoning about software quality rather than as autonomous coding agents.

A related strand extends beyond code-level assistance to examine LLM use across multiple phases of the SE lifecycle. Baresi et al. [67] report that students perceive LLMs as most helpful for implementation tasks (coding, debugging, testing) but less effective for higher-level activities such as requirements engineering and architectural reasoning. Dosaru et al. [68] propose a comparative instructional framework in which AI-generated solutions are artifacts for critique, encouraging validation, trade-off analysis, and metacognitive reflection. These studies highlight the importance of instructional design in enabling LLMs to support higher-order reasoning across lifecycle activities.

Despite these promising findings, the broader educational literature consistently raises concerns about academic integrity, over-reliance, and superficial engagement with AI-generated outputs [59], [60]. Faculty adoption remains limited due to insufficient training and uncertainty regarding responsible use [69–71]. Accordingly, recent work emphasizes that the educational value of LLMs depends on structured integration, ethical framing, and sustained human oversight [72], [73]. Systematic reviews further stress the need to redesign assessments and pedagogy, so students engage critically with AI while maintaining foundational reasoning skills [74], [75].

Our work extends this literature by systematically embedding LLM-supported activities across multiple SE lifecycle phases within an active-learning and reflective framework. Rather than focusing solely on coding assistance or isolated quality tasks, we investigate how GenAI can scaffold higher-order reasoning, structured decision-making, and critical evaluation while mitigating risks associated with over-reliance.

Threats to Validity

We recognize that one of the limitations of this study was that we did not have a control group. Having the same instructor teaching both course offerings may account for the small numbers of significant differences in the performance measures in CIS 375. The instructor practices mastery learning in project classes, which means students may resubmit their work for regrading after fixing any deficiencies. This tends to result in higher grades for student who exercise this option.

A few students in both sections (FF and AO) failed to turn in any lab write-ups. This may be the reason for the differences between the evaluation measures for the FF and AO students in Fall 2025. AO students seemed to struggle with the use of ChatGPT to complete the 8 new modules.

The pairing of an asynchronous distance learning section with a face-to-face section of the same course does not guarantee students receive the same educational experience, The live class sessions were captured, verbatim on video, for later viewing online by the FF and AO students. This allowed AO students the opportunity to witness the live lecture and class activities as a virtual classroom observer.

AO students were not allowed to attend FF sessions in either 2024 or 2025. It is not clear how often AO students viewed the lecture capture videos. On the final survey most AO students indicated they viewed 5 or more class recordings, but this could not be verified. It is possible the asynchronous students experienced more uncertainty when attempting to complete the activities alone.

One area of uncertainty when measuring the student responses is the unknown amount of interaction between students in the two sections in the same course. Students in the CIS department know each other from other classes that they have taken together. Even though a student registered in the asynchronous online section was not allowed to attend any in-person class meetings, it is quite possible that a friend from an in-person course section may have shared their course experiences with them giving them additional insight into group activities completed in the classroom.

Student engagement can only be measured indirectly in online courses using surveys and course analytics. In previous studies conducted by the authors, direct observation of student behavior was used to provide insight into the levels of engagement of in-person students. The authors could not use direct observation of students in the AO sections. The average number of late and missing assignments is the best data that could be obtained.

Conclusions and Future Direction

In this paper the authors demonstrated that it is possible to offer software engineering project courses to online students without significant loss of student satisfaction or perception of engagement. The introduction of GenAI tools in 2025 did not change this. This might be considered as evidence that it is possible to manage a hybrid software project course and keep most online students engaged. The active learning components of the classes and the levels of student interaction that accompany should be credited for making this possible. The active learning components of the course, the hybrid project teams, and peer evaluation work also should be credited with the success of the project work.

The use of reflective writing as an alternative to traditional testing was well received by the students. Both FF and AO students read the textbook and were forced to interact with it every week. This was not true in previous course offerings. Based on students' comments, reflective writing helped students to analyze the suggestions produced by ChatGPT.

The short, daily lab write ups and weekly reading reflections provided students with many opportunities for the instructor to provide feedback on their work. This also allowed the instructor many opportunities to identify student issues before the end of the semester. We encourage instructors to develop authentic assessment practices and modify them as needed to manage their hybrid course deliveries to achieve higher levels of student satisfaction and engagement. It was interesting to note that the in-person students felt ChatGPT was more of a time saver than the on-line students. Perhaps this was because the in-person students were able to discuss both prompt creation and assessment of the artifacts produced in small groups.

It may be important to continue developing ways in which asynchronous students are encouraged to participate in more experiences with face-to-face students outside formal class meetings. We are continuing to search for additional authentic assessment techniques that will help to catch slackers early during team project work. While we emphasize agile software development in our courses the assessment and management techniques we used could be adapted to fit other engineering process models.

We are continuing to develop tools to provide scaffolding assistance for student activities for both FF and AO students. One of the authors also teaches game design courses using active learning, authentic assessment, and GenAI tools. The current plan is to create additional scaffolded lab activities using GenAI tools for several software design courses.

Acknowledgments

This project was partially supported by grants from the University of Michigan-Dearborn Advancement of Teaching Fund and the College of Engineering and Computer Science.

References

[1] S. Feuerriegel, J. Hartmann, C. Janiesch, and P. Zschech, “Generative ai,” *Business & Information Systems Engineering*, vol. 66, no. 1, pp. 111–26, 2024.

[2] P. Denny, J. Leinonen, J. Prather, A. Luxton-Reilly, T. Amarouche, B. A. Becker, and B. N. Reeves, “Prompt problems: A new programming exercise for the generative ai era,” in *Proceedings of the 55th ACM Technical Symposium on Computer Science Education*, Portland, OR, 2024, pp. 296–302.

[3] K. I. Roumeliotis and N. D. Tselikas, “ChatGPT and open-ai models: A preliminary review,” *Future Internet*, vol. 15, no. 6, p. 192, 2023.

[4] A. Bick, A. Blandin, and D. J. Deming, “The Rapid Adoption of Generative AI,” Technical Report 32966, National Bureau of Economic Research, Cambridge, MA September 2024.

[5] T. Wu, S. He, J. Liu, S. Sun, K. Liu, Q.-L. Han, and Y. Tang, “A brief overview of ChatGPT: The history, status quo and potential future development,” *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 5, pp. 1122–1136, 2023.

[6] S. Sharma and R. Yadav, “Chat gpt–a technological remedy or challenge for education system,” *Global Journal of Enterprise Information System*, vol. 14, no. 4, pp. 46–51, 2022.

[7] V. Terragni, A. Vella, P. Roop, and K. Blincoe. “The Future of AI-Driven Software Engineering,”. *ACM Transactons on Software Engineering Methodology,* pp. 1-20, May 2025.

[8] S. Inman, A. Murillo, S. D'Angelo, A. Brown, and C. Green, "Seamful AI for Creative Software Engineering," *IEEE Software,* pp. 59-66, February 2025.

[9] V. M. de Brito and K. Farias. "Understanding the Role of Large Language Models in Software Engineering: Evidence from an Industry Survey." *arXiv preprint arXiv*:2512.21347, 2025.

[10] G. Giray, et al. "An Empirical Study of Generative AI Adoption in Software Engineering." *arXiv preprint arXiv*:2512.23327, 2025.

[11] A. Gomes, et al. "Attentionsmelling: Using Large Language Models to Identify Code Smells." *Simpósio Brasileiro de Engenharia de Software (SBES)*. Recife/PE, 2025, pp. 271-281.

[12] M. Binkhonain, and R. Alfayez. "Are prompts all you need? Evaluating prompt-based Large Language Models (LLM) s for software requirements classification." *Requirements Engineering*, pp. 1-21, 2025.

[13] A. Buscemi, "A comparative study of code generation using ChatGPT 3.5 across 10 programming languages." *arXiv preprint arXiv*:2308.04477, 2023.

[14] R. A. Husein, H. Aburajouh, and C. Catal. "Large language models for code completion: A systematic literature review." *Computer Standards & Interfaces* 92, 103917, 2025.

[15] D. Nam, et al. "Using an llm to help with code understanding." *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*. Lisbon, Portugal, 2024, pp. 1-13.

[16] S. S. Dvivedi, et al. "A comparative analysis of large language models for code documentation generation." *Proceedings of the 1st ACM international conference on AI-powered software*. Porto de Galinhas, Brazil, 2024, pp. 65-73.

[17] L. Zheng, J. Niu, L. Zhong, and J. F. Gyasi, “The effectiveness of artificial intelligence on learning achievement and learning perception: A meta-analysis,” *Interactive Learning Environments*, vol. 31, no. 9 p. 5650–5664, 2023.

[18] T. Ito, M. S. Tanaka, M. Shin, K. Miyazaki, et al., “The online pbl (project-based learning) education system using ai (artificial intelli- gence),” in DS 110: *Proceedings of the 23rd International Conference on Engineering and Product Design Education* (E&PDE 2021), VIA Design, VIA University in Herning, Denmark, 9th-10th September 2021, 6 pages.

[19] B. Branco, B. M. Castelo, et al. "Assessing the Role of GPT-4 in Software Engineering Learning: An Experience Report." *Simpósio Brasileiro de Engenharia de Software*, Recife/PE, September 2025, 12 pages.

[20] E. A. Alasadi and C. R. Baiz, “Generative ai in education and research: Opportunities, concerns, and solutions,” *Journal of Chemical Education*, vol. 100, no. 8, pp. 2965–2971, 2023.

[21] D. Baidoo-Anu and L. O. Ansah, “Education in the era of generative artificial intelligence (ai): Understanding the potential benefits of ChatGPT in promoting teaching and learning,” *Journal of AI*, vol. 7, no. 1, pp. 52– 62, 2023.

[22] M. A. Almulla, “The effectiveness of the project-based learning (pbl) approach as a way to engage students in learning,” *Sage Open*, vol. 10, no. 3, p. 2158244020938702, 2020.

[23] J. Su and W. Yang, “Unlocking the power of ChatGPT: A framework for applying generative ai in education,” *ECNU Review of Education*, vol. 6, no. 3, pp. 355–366, 2023.

[24] C. Zheng, K. Yuan, B. Guo, R. H. Mogavi, Z. Peng, S. Ma, and X. Ma, "Charting the future of ai in project-based learning: A co-design exploration with students," *arXiv preprint arXiv:*2401.14915, 2024.

[25] Q. Xia, T. K. Chiu, M. Lee, I. T. Sanusi, Y. Dai, and C. S. Chai, "A self-determination theory (sdt) design approach for inclusive and diverse artificial intelligence (ai) education," *Computers & Education*, vol. 189, p. 104582, 2022.

[26] M. P. C. Lin and D. Chang, "Enhancing post-secondary writers' writing skills with a chatbot," *Journal of Educational Technology Society*, vol. 23, no. 1, pp. 78–92, 2020.

[27] B. R. Maxim and B. A. Garcia, "Managing Students in Hybrid Software Project Classes", *Proceedings of 54th IEEE Annual Frontiers in Education Conference*, Washington, DC, October 2024, pp. 1-8.

[28] L. Samavedham, and K. Ragupathi, "Facilitating 21st century skills in engineering students," *The Journal of Engineering Education,* Vol. XXVI No. 1, pp.38-49, 2012.

[29] B.R. Maxim, S. Acharya, S. Brunvand, and M. Kessentini, "WIP: Introducing active learning in a software engineering course," *Proceedings of the 2017 Annual Meeting of the American Society for Engineering Education,* Columbus, OH, June 2017, pp.1-12.

[30] B. R. Maxim, and J. J. Yackley, "Using Active Learning and Gamification to Teach Software Engineering Courses in Game Design Courses." *Proceedings of the 2023 Annual Meeting of the American Society for Engineering Education*," Baltimore, MD, June 2023, pp. 1-21.

[31] M. Prince, "Does Active Learning Work? A Review of the Research," *Journal of Engineering Education*, Vol. 93, pp. 223-231, 2024.

[32] S. Luster-Teasley, S.C. Hargrove-Leak, and C. Waters, "NSF TUES: Transforming undergraduate environmental engineering laboratories for sustainable engineering using the case studies in the sciences instructional method" *Proceedings of the 2014 Annual Meeting of the American Society for Engineering Education,* Indianapolis, IN, June 2014, 10.18260/1-2–22873.

[33] V. Jungic, H. Kaur, J. Mulholland, and C. Xin, "On flipping the classroom in large first-year calculus courses," *International Journal of Mathematical Education in Science and Technology*, 46(4), 1–8, 2015.

[34] P.K. Raju, and C.C. Sanker, "Teaching real-world issues through case studies," *Journal of Engineering Education*. Vol. 88 No 4 pp501-508, 2018.

[35] K.M. Nickels, "Do's and don'ts of introducing active learning techniques," *Proceedings of the 2000 Annual Meeting of the American Society for Engineering Education,* St. Louis, Missouri, June 2000, 6 pages

[36] J.P. Lavelle, M.T. Stimpson, and E.D. Brill, "Flipped out engineering economy: Converting a traditional class to an inverted model," In A. Krishnamurthy & W. K. V. Chan (Eds.), *Proceedings of the 2013 Industrial Systems Engineering Research Conference,* San Juan, PR, 2013, pp. 397–407.

[37] K. Yelamarthi, S. Member, and E. Drake, "A flipped first-year digital circuits course for engineering and technology students," *IEEE Transactions on Education*, 58(3), 179–186, 2015.

[38] M. Ardis, S. Chenoweth, and F. Young, "The 'Soft' Topics in Software Engineering Education," *Proceedings of 38th Annual Frontiers in Education Conference (Vol. 1, Oct 2008),* IEEE Press, Saratoga Springs, NY, 2008, pp. F3H1-F3H6.

[39] J.A. Day and J.D. Foley, "Evaluating a Web Lecture Intervention in a Human-Computer Interaction Course," *IEEE Transactions on Education* 49(4):420-431, 2006.

[40] G.R. Morrison, S.M Ross, J.E. Kemp, and H. Kalman, *Designing effective instruction*. John Wiley & Sons, 2010.

[41] L. Zhang and Y. Ma, "A study of the impact of project-based learning on student learning effects: a meta-analysis study", *Frontiers in Psychology,* Jul 17;14:1202728, 2023.

[42] F. Elkhamisy, A. Zidan, and M. Fathelbab, "Using project-based learning to enhance curricular integration and relevance of basic medical sciences in pre-clerkship years", *Alexandria Journal of Medicine*, 58(1), 1–7, 2022.

[43] J.N. Warnock and M.J. Mohammadi-Aragh, "Case study: use of problem-based learning to develop students' technical and professional skills," *European Journal of Engineering Education*, Vol. 41, No, 2, 2016, pp.142-153,

[44] J. Dunlap, "Problem-based learning and self-efficacy: How a capstone course prepares students for a profession," *Education Technology Research and Development* Vol. 53, No.1, pp. 65–83, 2004.

[45] M. Souza, et. al. "Students perception on the use of project-based learning in software engineering education," SBES 2019: *Proceedings of the XXXIII Brazilian Symposium on Software Engineering*, Salvador, Brazil, 2019, pp. 537–546.

[46] P.C. Schlechty, *Engaging Students: The Next Level of Working on Work*, Wiley and Sons, 2011.

[47] R. Kothiyal, Majumdar, S. Murthy, and S. Iyer, "Effect of think-pair-share in a large CS1 class: 83% sustained engagement," *Proceedings of the ninth annual international ACM conference on international computing education research (ICER '13)*. San Diego, CA, ACM, New York, NY, USA, 2013, pp. 137-144.

[48] M. Nagappan; L. Williams, M. Ferzli, E. Wiebe, K. Yang, C. Miller, and S. Balik, "Improving the CS1 experience with pair programming," *Proceedings of the 34th SIGCSE technical symposium on Computer science education (SIGCSE '03*). Reno, NV, ACM, New York, NY, USA, 2003, pp. 359-362.

[49] L. Porter, D. Bouvier, Q. Cutts, S. Grissom, C. Lee, R. McCartney, D. Zingaro, and B. Simon, "A multi-institutional study of peer instruction in introductory computing," *Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16).* Online, Canada, ACM, New York, NY, USA, 2016, pp. 358-363.

[50] T. Greer, Q. Hao, M. Jing, and B. Barnes, "On the effects of active learning environments in computing education," *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19),* February 27-March 2, 2019, Minneapolis, MN, USA. ACM, New York, NY, USA, 6 pages.

[51] B. Hoffman, R. Morelli, and J. Rosato, "Student engagement is key to broadening participation in CS," *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19),* February 27-March 2, 2019*,* Minneapolis, MN, USA. ACM, New York, NY, USA, 7 pages.

[52] J.A. Stone and E. Madigan, "Experiences with community-based projects for computing majors," *Journal of Computer. Science in the Colleges*, Vol. 26, No.6, pp.64-70, June 2011.

[53] B. R. Maxim, B. Thomas, and B. A. Garcia, "WIP: Managing and Assessing Students in Hybrid Software Project Classes", *Proceedings of the 2024 Annual Meeting of the American Society for Engineering Education*, Portland, OR, June 2024, pp. 1-19.

[54] C. Kingkaew, T. Theeramunkong, T. Supnithi, P. Chatpreecha, K. Morita, K. Tanaka, ans M. Ikeda. "A Learning Environment to Promote Awareness of the Experiential Learning Processes with Reflective Writing Support." *Education Sciences*.13(1):64, 2023.

[55] V. Clinton, "Reflections versus Extended Quizzes: Which is Better for Student Learning and Self-Regulation?" *Journal of the Scholarship of Teaching and Learning,* 18(1), 2018.

[56] N. Ragonis and O. Hazzan, "Reflection Pre-learning in Computer Science Courses," *Communications of the ACM BLOG@ACM*, January 2022.

[57] R.S. Pressman and B. R, Maxim, *Software Engineering: A Practitioner's Approach*, New York, NY, McGraw-Hill, 2020.

[58] S. Steinert, K. E. Avila, S. Ruzika, J. Kuhn, and S. Ku¨chemann, "Harnessing large language models to enhance self-regulated learning via formative feedback," *arXiv preprint arXiv:2311.13984*, 2023.

[59] G. Cooper, "Examining science education in ChatGPT: An exploratory study of generative artificial intelligence," *Journal of Science Education and Technology*, vol. 32, no. 3, pp. 444–452, 2023.

[60] R. Michel-Villarreal, E. Vilalta-Perdomo, D. E. Salinas-Navarro, R. Thierry-Aguilera, and F. S. Gerardou, "Challenges and opportunities of generative ai for higher education as explained by ChatGPT," *Education Sciences*, vol. 13, no. 9, p. 856, 2023.

[61] R. Yilmaz and F. G. K. Yilmaz, "The effect of generative artificial in- telligence (ai)-based tool use on students' computational thinking skills, programming self-efficacy and motivation," *Computers and Education: Artificial Intelligence*, vol. 4, p. 100147, 2023.

[62] V. Roger-Monzo´, "Impact of generative artificial intelligence in higher education: Student perceptions," in *INTED2024 Proceedings*, Valencia, Spain, 2024, pp. 2631– 2635.

[63] A. Jaber, K. Lechasseur, K. Mottakin, and Z. Song, "A study report in the web technologies course: What makes feedback effective for project- based learning?," in *2023 ASEE Annual Conference & Exposition*, Baltimore, MD, 2023, 7 pages.

[64] J. Su and W. Yang, "Artificial intelligence in early childhood education: A scoping review," *Computers and Education: Artificial Intelligence*, vol. 3, p. 100049, 2022.

[65] E. A. AlOmar and M. W. Mkaouer. "Cultivating software quality improvement in the classroom: An experience with chatGPT," *2024 36th International Conference on Software Engineering Education and Training (CSEE&T)*. Würzburg, Germany, 2024, pp. 1-10.

[66] Menolli, André, Bruno Strik, and Luiz Rodrigues. "Teaching refactoring to improve code quality with chatgpt: An experience report in undergraduate lessons," *Proceedings of the XXIII Brazilian Symposium on Software Quality*. 2024, pp. 563-574.

[67] Baresi, Luciano, et al. "Students' Perception of ChatGPT in Software Engineering: Lessons Learned from Five Courses," *2025 IEEE/ACM 37th International Conference on Software Engineering Education and Training (CSEE&T)*," 2025, pp. 158-169.

[68] Dosaru, Daniel-Florin, et al. "A Code Analysis Tool to Help Students in the Age of Generative AI," *European Conference on Technology Enhanced Learning*. Cham: Springer Nature Switzerland, 2024, pp. 222-228.

[69] B. Obenza, A. Salvahan, A. N. Rios, A. Solo, R. A. Alburo, and R. J.,Gabila, "University students' perception and use of ChatGPT: Generative artificial intelligence (ai) in higher education," *International Journal of Human Computing Studies*, vol. 5, no. 12, pp. 5–18, 2024.

[70] A. Kelly, M. Sullivan, and K. Strampel, "Generative artificial intelligence: University student awareness, experience, and confidence in use across disciplines," *Journal of University Teaching & Learning Practtice*, pp. 1-16,vol. 20 No. 6, 2023.

[71] C. Bopp, A. Foerst, and B. Kellogg, "The case for llm workshops," in *Proceedings of the 55th ACM Technical Symposium on Computer Science Education*, Portland, OR, 2024, pp. 130–136.

[72] M. Liu and F. M'Hiri, "Beyond traditional teaching: Large language models as simulated teaching assistants in computer science," in *Proceedings of the 55th ACM Technical Symposium on Computer Science Education* , Portland, OR, 2024, pp. 743–749.

[73] Y. Dai, A. Liu, and C. P. Lim, "Reconceptualizing ChatGPT and generative ai as a student-driven innovation in higher education," *Procedia CIRP*, vol. 119,Elsevier B.V., 2023, pp. 84–90.

[74] Q. Qin, R. Santos, and R. Spinola. "On the Role and Impact of GenAI Tools in Software Engineering Education." *arXiv preprint arXiv:2512.04256*, 2025.

[75] N. Kiesler, et al. "The Role of Generative AI in Software Student Collaboration." *Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education*, Nijmegen, The Netherlands, 2025, pp. 72-78.