Structured Integration of Artificial Intelligence in Software Engineering Education: From Perception to Practice

Dr. Robert Harold Lightfoot Jr, Texas A&M University

Robert Lightfoot received his Ph.D. from Texas A&M University in Interdisciplinary Engineering, focusing on Computer Science and Engineering Education. His master's degree is in software engineering from Southern Methodist University, and his bachelor's degree in computer science from Texas A&M. Before joining Texas A&M, he worked at Ericsson (now Sony-Ericsson) in the network division, then DSC (Digital Switch) for the Motorola Cellular Infrastructure Group. Robert Lightfoot is now an Associate Professor of Practice at TAMU in the Computer Science department and a member of the Engineering Education Faculty.

Dr. Shawna Thomas, Texas A&M University

Dr. Thomas is an Instructional Assistant Professor in the Department of Computer Science and Engineering at Texas A&M University. She is a member of the Engineering Education Faculty in the Institute for Engineering Education & Innovation at Texas A&M. Sh

Brinley Boyett, Texas A&M University

Structured Integration of Artificial Intelligence in Software Engineering Education: From Perception to Practice

Robert H. Lightfoot Jr.
Department of Computer Science & Engineering
Texas A&M University
rob.light@tamu.edu

Shawna L. Thomas
Department of Computer Science & Engineering
Texas A&M University
sthomas@tamu.edu

Brinley Boyett
Department of Computer Science & Engineering
Texas A&M University

March 25, 2026

Abstract

The widespread adoption of large language models (LLMs) such as ChatGPT has accelerated educators' need to define responsible and effective ways to integrate artificial intelligence (AI) into authentic learning environments. While early work explored student perceptions of AI-supported learning, fewer studies have examined structured instructional frameworks that position AI as a transparent, documented component of student work. Building on our prior research in the Computer Science and Engineering department at a large R1 university, which evaluated students' perceptions of AI's influence on learning environment, technological literacy, and teaching effectiveness [1], the present work implements and evaluates a structured framework for AI use across multiple assignments in a junior-level software engineering course.

In this new phase, students were required to engage with LLMs through a transparent and ethical workflow that constrained the scope of model assistance. Each team-based deliverable specified: (1) a provided LLM prompt template emphasizing student authorship of ideas and data; (2) documentation of the complete AI chat history as an appendix; and (3) reflection on the benefits and limitations of AI's contributions to the final work. These procedures were integrated into complex, authentic tasks such as producing software design and database documentation. The approach aimed to treat AI as a collaborative drafting and reasoning partner rather than a shortcut to problem completion.

Data were gathered from students through assignment artifacts, structured reflections, and instructor observations to examine outcomes across three dimensions: (1) learning environment, as reflected in team collaboration and engagement; (2) technological literacy, including ethical reasoning, prompt engineering, and evaluation of AI outputs; and (3) teaching effectiveness, measured by students' ability to apply structured AI use while maintaining independent problem-solving. Preliminary analyses indicate that the structured framework fostered higher accountability and deeper understanding of AI's strengths and limitations. Students reported that documenting their AI interactions clarified the boundary between human and machine contributions and improved their ability to critically assess LLM outputs.

This work demonstrates how structured integration of AI tools can enhance both student learning and teaching effectiveness while safeguarding academic integrity. It offers a replicable model for instructors seeking to move beyond ad hoc AI usage toward systematic, ethically informed classroom practices. By shifting from perception to practice, this study contributes to the ongoing dialogue on how AI can responsibly augment—not replace—human cognition and creativity in engineering education.

1 Introduction

Artificial intelligence (AI)—and, in particular, large language models (LLMs)—has transitioned in a short time from optional novelty to routine infrastructure in many university learning environments. For computing educators, this shift raises two simultaneous imperatives: first, to prepare students for software and data-centric workplaces where AI

is ubiquitous; and second, to steward ethical, transparent, and educationally sound use of these tools in coursework. Early institutional responses understandably focused on policy—either restricting use or permitting it with broad guidelines. While such policies were necessary, they are not sufficient to cultivate the competencies students need: prompt design, critical evaluation of outputs, documentation of tool use, and the metacognitive habit of distinguishing human reasoning from machine assistance.

Our previous study [1] investigated student perceptions of AI-supported learning in two upper-division computer science courses at a large R1 university. We found that when instructors offered clear expectations, students reported a more positive learning environment and increased technological literacy, while teaching effectiveness remained primarily dependent on the educator's pedagogical approach. These findings suggested that AI can enhance learning when its use is intentional and well-scaffolded, but also underscored a gap between permissive "allowance" and structured *practice*.

The present paper addresses that gap by moving from perception to implementation. We propose and evaluate a practical framework for structured AI use in a junior-level software engineering course. The framework requires students to (a) initiate model interactions with a constrained, ethics-forward prompt template that prohibits idea generation and external sourcing; (b) append the full model interaction history to deliverables for end-to-end transparency; and (c) reflect on the scope, limits, and quality of AI contributions relative to their own work. We integrated this workflow into authentic team-based tasks, including a database design document and other design-oriented artifacts common to software engineering practice.

Our investigation is guided by three research questions:

1. How does a structured AI-use framework affect student engagement and the learning environment in a software engineering course?

2. To what extent does transparent documentation of AI interactions enhance technological literacy, including ethical reasoning and critical evaluation of LLM outputs?

3. How do structured AI-use practices influence teaching effectiveness and student independence in problem-solving?

This paper contributes three things. First, it operationalizes responsible AI use into a replicable set of instructional practices that go beyond simple permission or prohibition. Second, it presents evidence from student artifacts, reflections, and instructor observations that structured transparency supports accountability without diminishing students' ownership of ideas. Third, it positions this framework within ongoing work on AI in engineering education—including studies on faculty and student perceptions [14, 2, 4], structured guidance and policy [5], and evolving educator roles [12]—to argue that methodical integration is a necessary next step for aligning classroom practice with professional expectations.

The remainder of the paper reviews related work, details the course setting and structured framework, describes our data collection and analysis methods, and presents findings organized around learning environment, technological literacy, and teaching effectiveness. We close by discussing implications for educators seeking to embed AI responsibly in software engineering and related computing courses, along with limitations and avenues for future research.

2 Background and Related Work

The rapid normalization of large language models (LLMs) in higher education has catalyzed research along three complementary threads: (i) classroom integration models and policy guidance; (ii) stakeholder perceptions and ethical practice; and (iii) instructional roles and competencies in AI-supported learning. We summarize each strand and identify gaps addressed by our structured-use framework.

2.1 Classroom Integration Models and Policy Guidance

Simply permitting or forbidding AI tools is insufficient [7, 9]; students require concrete, repeatable processes to engage LLMs responsibly in authentic coursework. Prior work in our context proposed assignment-level scaffolds that bounded model use, clarified allowed assistance (e.g., drafting, formatting, commentary), and reinforced academic integrity through transparent documentation [5]. Parallel efforts in engineering education and computing pedagogy highlight the need for designs that make model use *visible* to both students and instructors, enabling formative feedback

on prompt quality, source attribution, and evaluation of AI outputs. Our present framework operationalizes these recommendations through a three-part structure: constrained prompts, full transcript inclusion, and reflective analysis tied to deliverables common in software engineering.

2.2 Stakeholder Perceptions and Ethical Practice

A substantial body of work has explored evolving attitudes of students and faculty toward LLMs. Multi-institutional and division-level studies report both enthusiasm for productivity gains and concern about academic integrity, bias, and overreliance [14, 2, 4]. Our prior study found that clear guidance was associated with improved technological literacy and a more positive learning environment [1]. Beyond attitudes, researchers urge explicit attention to responsible use, including respect for course policies, honest authorship, and discipline-aligned professional norms [6, 11]. The present work extends this line by embedding ethical practice into the mechanics of assignments, making integrity and reflection integral to the workflow rather than an after-the-fact declaration.

Beyond the classroom, mainstream scientific discourse reflects this tension between productivity benefits and integrity risks. Large-scale reports have documented LLMs' fluency alongside well-known failure modes such as hallucinated facts and fabricated citations, reinforcing the need for human verification and transparent disclosure of AI assistance [10]. Relatedly, publishers have begun articulating expectations for disclosure (and, in some venues, restrictions) while noting that currently available AI-detection tools are imperfect and should not be used as the sole basis for misconduct determinations [10]. These developments align with our approach of embedding disclosure, verification, and reflection into the mechanics of assignments—making ethical practice visible and assessable rather than aspirational.

2.3 Instructional Roles and Competencies

The introduction of AI tools shifts, but does not diminish, the educator's role. Studies note that instructors increasingly act as designers of learning workflows, curators of examples, and coaches for metacognitive and evaluative skills [12]. Our earlier findings echoed this: students perceived teaching effectiveness to be more tightly coupled to pedagogy than to tool presence [1]. Accordingly, our structured-use framework is intended to augment—not replace—core instructional activities such as critique, code review, and design reasoning. In doing so, it targets competencies central to software engineering education: communicating assumptions, tracing requirements to artifacts, and distinguishing human decisions from tool-generated suggestions.

In practice, engaged instructors in our implementation met *weekly* with graduate and undergraduate TAs to align guidance with course learning objectives, calibrate feedback on AI transcripts and reflections, and surface common failure modes early. This hands-on orchestration aligns with broader guidance that positions teachers as central coordinators of AI-enabled learning—responsible for disclosure expectations, verification routines, and human oversight [13, 8]. More generally, evidence from the active-learning literature underscores that instructor-designed structures and facilitation substantially impact learning outcomes in STEM contexts, reinforcing the need for intentional, instructor-led integration of new tools [3].

2.4 Gap and Contribution

Across these strands, two gaps persist: (1) a lack of *replicable* classroom procedures that operationalize responsible AI use in team-based, design-oriented computing courses; and (2) limited analyses that triangulate student artifacts, reflections, and instructor observations to characterize outcomes beyond self-report. This paper addresses both gaps by detailing, implementing, and evaluating a structured AI-use framework aligned with authentic software engineering deliverables, and by presenting mixed sources of evidence around learning environment, technological literacy, and teaching effectiveness.

3 Methodology

We adopt a multi-source design that extends the single-course, perception-centered approach in our earlier work by incorporating: (i) instructor perspectives on student learning outcomes after a full semester of structured AI use; (ii) a cross-course study of CS instructors' adoption patterns and practices with AI in their courses; and (iii) a comparative analysis of student outcomes under structured versus ad hoc (unstructured) AI use.

Component	Purpose	Student Action	Evidence / Artifact	Typical Pitfalls & Instructor Role
Constrained Prompt Template	Bound AI's role to drafting/formatting/ explanation; preserve authorship and sourcing	Start the LLM session with the provided template; supply only course-approved information; prohibit idea origination or external sources	Prompt text captured at top of transcript appendix	*Pitfalls:* prompt creep; hidden external sources. *Instructor:* spot-check first prompt; model exemplars of precise constraints.
Full Transcript Appendix	Ensure transparency and inspectability of model use	Export full chat history; redact PII; include as PDF or text appendix	Appendix file/section linked in submission	*Pitfalls:* missing turns; selective copy. *Instructor:* require complete session; random audits.
Reflection Note (200–300 words)	Separate human reasoning from AI contributions; develop evaluative skill	Describe what AI did vs. what you did; identify issues (hallucination, bias); explain verification steps	Reflection paragraph with labeled bullets	*Pitfalls:* vague reflection; rationalizing errors. *Instructor:* provide a 3-criteria rubric; give brief formative comments.
Role Limits in Assignment	Align AI use with learning outcomes & integrity	Use AI for drafting/organization/grammar; not for idea generation, external facts, or code design decisions (unless explicitly allowed)	Self-attestation line in deliverable; matches transcript evidence	*Pitfalls:* over-reliance; substitution. *Instructor:* align exams with the same reasoning moves; assess clarity of human decisions.
Rubric Rows for AI Use	Make appropriate use and evaluation assessable	Address rubric rows for: (a) clarity of human reasoning; (b) appropriateness of AI use; (c) quality of AI evaluation	Rubric subscores reported back to students	*Pitfalls:* hidden criteria. *Instructor:* publish rubric anchors and examples.
Feedback Loop	Shift effort from detection to coaching	Use transcript excerpts in office hours/code-review to target prompt design & verification habits	Inline feedback referencing excerpt IDs	*Pitfalls:* late feedback. *Instructor:* schedule early checkpoint using short transcript.

Table 1: Structured AI framework: components, actions, artifacts, pitfalls, and instructor roles.

3.1 Course and Educational Context

The focal course remains a junior-level software engineering course at a large R1 university. Teams complete design-oriented deliverables, including a database design document, with a structured AI workflow. Parallel data collection efforts include additional upper-division computing courses taught by faculty who permit AI use with varying degrees of structure.

3.2 Structured AI-Use Workflow

We operationalize responsible AI use via three requirements embedded in assignments: (1) a constrained, ethics-forward prompt template; (2) end-to-end transparency through inclusion of the full model interaction history (appendix); and (3) short reflections on the boundaries, limitations, and contributions of AI relative to student work. This workflow bounds AI's role to drafting, organization, and explanatory support rather than idea generation or sourcing external content.

3.3 Data Sources and Cohorts

Dataset A: Instructor Outcome Views. We conducted semi-structured interviews and collected guided reflections from **four** professors teaching a junior-level software engineering course across **seven** lecture sections (total enrollment **N = 507** students) during a single semester. Although all instructors agreed to adopt a *structured AI-use* policy for this course, expectations and emphases *varied modestly* across sections. To support consistency and surface emerging issues, instructors held *weekly* calibration meetings with graduate and undergraduate TAs to align guidance with course learning objectives, review common failure modes observed in AI transcripts, and coordinate formative feedback.

Interviews targeted perceived changes in (i) student behaviors (independence, prompt literacy, verification habits), (ii) quality of deliverables (clarity, traceability of human decisions, documentation), (iii) feedback cycles (time spent on detection vs. coaching), and (iv) exam performance and homework–exam alignment. Guided reflections (short memos completed at midterm and end-of-term) asked instructors to characterize the benefits and drawbacks of the structured workflow, name one policy they would tighten or relax, and describe a representative example where the AI transcript directly improved feedback.

Dataset B: CS Instructor Adoption Study. We conducted a descriptive analysis of AI policies and practices across the *core Computer Science sequence* (freshman through junior level). Publicly available syllabi and course policy documents were collected for required lecture/lab offerings in the core. Our aim was to determine whether a *consistent AI ideology* exists across courses and levels, and to characterize where guidance diverges.

Dataset C: Structured vs. Ad Hoc Cohorts. Comparative evidence of student outcomes in courses that employed (i) structured AI workflows versus (ii) ad hoc or unstructured AI use. Outcome variables include homework scores, exam performance, rubric subscores for design reasoning and explanatory clarity, and evidence of misalignment (e.g., high homework/low exam patterns).

3.4 Measures and Instruments

Instructor perspectives (A): Interview protocol with prompts targeting changes in (a) student independence and reasoning quality; (b) prompt literacy and evaluation of AI outputs; (c) time-on-task and feedback efficiency; and (d) integrity events (suspected/confirmed).

Adoption patterns (B): Faculty survey/inventory of course-level AI policies and practices; collection of representative assignment prompts/policies (when available).

Student outcomes (C): Course-grade artifacts (homework, exams), rubric-based subscores for design artifacts, submission metadata (e.g., revision counts, timing), and alignment indicators (e.g., homework–exam deltas).

3.5 Analysis Plan

Qualitative: Thematic coding of instructor interviews (Dataset A) for perceptions of learning behaviors, instructional affordances, and constraints. For adoption (Dataset B), coding of policy texts to classify structure level and pedagogical intentions.

Quantitative: Descriptive statistics and group comparisons for Dataset C (structured vs. ad hoc). Where feasible, we compute homework–exam deltas, effect sizes, and nonparametric tests (e.g., Mann–Whitney) for robustness. We also summarize rubric subscores by cohort to assess design reasoning clarity and explanatory quality.

3.6 Ethics / IRB and Data Management

All analyses use de-identified artifacts and aggregated statistics. Interview data are anonymized; instructor and course identifiers are removed. Student data are analyzed in compliance with institutional policies. Internal, unpublished datasets are treated as confidential research data; no identifiable or proprietary content is included in the manuscript. We focus on reporting aggregate indicators, representative (anonymized) excerpts, and derived metrics.

4 Results and Discussion

4.1 R1: Instructors' Views After a Semester of Structured AI Use

Key themes: (i) clearer separation of human reasoning vs. AI drafting; (ii) improved transparency enabling targeted feedback; (iii) decreased time spent on misconduct investigations, increased time on coaching and critique; (iv) exam preparedness stable or improved when structure was followed. We illustrate each theme with representative excerpts and frequency summaries.

To address the first research question—*How does a structured AI-use framework affect student engagement and the learning environment in a software engineering course?*—we draw on qualitative data collected through semi-structured interviews with all instructors teaching the course during the semester of study. These interviews capture instructors' shared reflections on student engagement, learning behaviors, and classroom dynamics following the introduction of a structured, transparent AI-use framework.

All instructors participating in this study had taught the course in prior semesters before the availability or integration of large language models (LLMs), allowing for informed comparison between pre-AI and AI-integrated offerings. Interviews focused on instructors' experiences designing assignments with explicit AI-use expectations, observing student engagement during project work, and evaluating the overall learning environment across teams.

Across interviews, instructors consistently reported that a structured AI-use framework positively influenced the learning environment by shifting student attention away from surface-level concerns and toward substantive engineering work. In particular, instructors observed that students were less anxious about writing quality, formatting, and document organization when AI was explicitly permitted for documentation tasks. Knowing that AI could assist with structure and clarity, students appeared more willing to engage deeply with the technical content of their projects, including design rationale, implementation decisions, and validation of results.

Instructors further noted that providing a common documentation template and a shared AI prompt contributed to a more focused and collaborative learning environment. Because teams followed similar documentation structures, class discussions and feedback could center on engineering trade-offs and problem-solving strategies rather than inconsistencies in writing or presentation. This consistency also appeared to support student engagement during team interactions, as students shared common expectations about how their work would be documented and evaluated.

From an instructional perspective, instructors reported that the structured AI-use framework fostered a more productive classroom dynamic. Rather than policing unauthorized AI use, instructors were able to engage students in conversations about appropriate, ethical, and effective use of AI tools. This transparency reduced ambiguity and encouraged students to view AI as a legitimate component of the learning process rather than a hidden shortcut. Instructors perceived this shift as contributing to a more open and trust-based learning environment.

Overall, instructor interviews suggest that a structured and transparent AI-use framework can enhance student engagement and improve the learning environment in a software engineering course. By clearly defining when and how AI may be used, instructors observed that students were better able to focus on core learning objectives, engage more confidently with complex engineering tasks, and participate in a learning environment characterized by clarity, consistency, and shared expectations.

4.2 R2: CS Instructors' Use of AI Across Courses

To address the second research question—*To what extent does transparent documentation of AI interactions enhance technological literacy, including ethical reasoning and critical evaluation of LLM outputs?*—we draw on structured practitioner observations rather than a standalone quantitative dataset. Dataset B consists of recurring, reflective discussions among instructional staff directly involved in courses that integrated transparent and documented AI use.

Specifically, four instructors (each with multiple semesters of pre-AI experience teaching the same or similar courses) and seven teaching assistants (many of whom had served as TAs prior to the introduction of AI-supported assignments) met weekly throughout the semester. These meetings focused on instructional design decisions, student behavior, grading experiences, and observed learning outcomes related to AI-supported work. While informal in structure, these discussions were longitudinal, comparative (pre-AI versus AI-integrated offerings), and grounded in shared instructional artifacts such as assignment prompts, AI usage guidelines, and grading rubrics.

Across courses, instructors required students to explicitly document their interactions with large language models, including prompts used and the resulting AI-generated output. This transparency appeared to significantly shape how students engaged with AI tools. One consistent observation was that students demonstrated reduced anxiety about

writing mechanics and formatting, knowing that the LLM would assist with documentation quality. As a result, students were able to focus more deliberately on the technical and factual components of their engineering projects, such as system design decisions, implementation details, and experimental results.

Importantly, this shift did not appear to reduce rigor. Instead, instructors observed that students treated AI as a support tool rather than a substitute for understanding. Because students were provided with a common documentation template and a shared prompt structure, they were required to critically evaluate whether the AI-generated documentation accurately reflected their work. This process implicitly required students to review, verify, and revise LLM outputs, supporting the development of critical evaluation skills central to technological literacy.

Teaching assistants and graders reported substantial improvements in clarity and readability across submissions. The consistency in document structure made it easier to locate key technical content, reducing ambiguity and misinterpretation during grading. Furthermore, because teams used a common AI prompt, graders could focus their evaluation on the correctness, completeness, and depth of the underlying engineering work rather than on surface-level writing differences. This consistency also contributed to more reliable and equitable assessment practices.

From an ethical standpoint, instructors emphasized that the transparent and mandated use of AI shifted student behavior toward more responsible and above-board practices. Rather than attempting to conceal AI use, students were expected to disclose it, reflect on it, and take ownership of the final product. Instructors noted that this approach aligned closely with professional expectations students are likely to encounter in industry, where engineers routinely use automated tools but remain accountable for correctness, clarity, and ethical decision-making.

Collectively, these observations suggest that transparent documentation of AI interactions supports technological literacy in multiple dimensions. Students gained experience using AI tools ethically, learned to critically evaluate and refine AI-generated content, and became familiar with professional-style technical documentation workflows. While Dataset B does not provide quantitative measures, the consistency of observations across instructors, courses, and instructional roles provides converging evidence that transparent AI integration can enhance ethical reasoning, critical evaluation of LLM outputs, and preparedness for professional practice.

4.3 R3: Structured vs. Ad Hoc Outcomes

To address the third research question—*How do structured AI-use practices influence teaching effectiveness and student independence in problem-solving?*—we compare cohorts operating under two different instructional framings of AI use. Both cohorts completed comparable homework assignments and assessments; however, one cohort operated under an ad hoc AI environment with no explicit instructional guidance on using AI for learning, while the other cohort was given a structured AI-use framework that explicitly positioned AI as a learning aid rather than a performance tool.

In the ad hoc cohort, students were subject to the standard course policy prohibiting AI use for submitted homework, but no additional guidance was provided on how AI might be used responsibly to support learning. In contrast, the structured cohort was explicitly instructed to use AI as a learning tool—for example, to explore approaches, clarify concepts, or check understanding—but was reminded that homework submissions need not be perfect to receive full credit. This reframing emphasized formative learning over polished performance and explicitly decoupled homework quality from summative evaluation.

Across cohorts, we compared homework scores, final exam scores, and homework–exam deltas. Preliminary patterns suggest that the ad hoc cohort exhibited slightly higher homework scores, including a larger number of perfect homework submissions. However, this difference did not translate into a meaningful improvement in exam performance. A Welch two-sample t-test indicated that the difference in final exam scores between Group A ($M = 83.02$, $SD = 8.84$, $n = 129$) and Group B ($M = 81.12$, $SD = 8.32$, $n = 192$) was not statistically significant, $t \approx 1.92$, $df \approx 260$, $p \approx 0.06$. The effect size was small (Cohen's $d \approx 0.22$), suggesting a minimal practical difference between the groups.

Instructors interpreted the higher homework performance in the ad hoc cohort with caution. Based on observed student behavior and the absence of structured guidance, instructors believe that some students likely used AI tools to generate or complete homework solutions despite formal restrictions. In this context, homework performance may reflect tool-assisted completion rather than independent problem-solving or conceptual mastery.

By contrast, the structured cohort demonstrated fewer perfect homework submissions, consistent with the explicit messaging that homework was intended as a learning activity rather than a polished artifact. Students in this group were encouraged to attempt problems, document their reasoning, and reflect on misunderstandings without penalty for incomplete or imperfect solutions. As a result, homework scores were modestly lower on average, but students still received credit for engagement and effort.

Importantly, this reframing did not negatively impact exam performance. Despite fewer perfect homework scores, students in the structured cohort performed comparably on final exams, suggesting that structured AI use supported learning without inflating homework grades. From an instructional perspective, this outcome aligns with improved teaching effectiveness: homework functioned as formative practice supporting student independence, while exams remained a more reliable measure of individual understanding.

Taken together, these findings suggest that structured AI-use practices may reduce grade inflation in homework while preserving—or modestly improving—summative learning outcomes. By positioning AI as a learning aid and redefining homework as a low-stakes space for exploration, instructors observed greater alignment between homework engagement and independent problem-solving assessed on exams. While observed differences in exam performance were not statistically significant, the pattern of results supports the interpretation that structured AI-use frameworks can enhance teaching effectiveness by promoting student independence rather than polished but potentially AI-assisted homework performance.

5 Conclusion

This paper extends prior perception-focused work on AI in engineering education by operationalizing a *structured AI-use* framework and evaluating it across multiple, complementary data sources. Building on earlier findings that transparent expectations improve learning environment and technological literacy [1], we implemented a concrete classroom workflow that (a) constrains prompts to preserve student authorship and sourcing, (b) documents end-to-end model interactions, and (c) requires short reflections distinguishing human reasoning from AI assistance. We then analyzed outcomes from three additional perspectives: instructor views after a semester of structured use (Dataset A), a cross-course study of AI adoption among CS instructors (Dataset B), and comparative evidence from cohorts operating with structured versus ad hoc practices (Dataset C).

Across these strands, three integrative conclusions emerge. First, **structure creates visibility and accountability**. Instructors reported that requiring transcripts and reflections made AI use inspectable and coachable, enabling feedback on prompt quality and evaluation of outputs rather than post hoc misconduct inquiries. Second, **structure supports, but does not supplant, agency**. Students reported confidence benefits when AI was framed as a bounded aid; however, without intentional opportunities to practice autonomous, strategic tool use, student agency developed more slowly than ethical responsibility. Third, **misalignment is measurable**. In ad hoc settings, elevated homework performance alongside weaker exam results suggests unexamined substitution rather than reinforcement; structured use mitigated this pattern when assignments and exams were aligned with the same reasoning expectations.

Taken together, these findings reframe classroom AI as a curricular literacy rather than a binary permission. When operationalized through transparent artifacts and bounded roles, AI can augment feedback cycles and reasoning practice while preserving authorship and integrity.

6 Implications

6.1 Pedagogy and Course Design

In computing courses with design-oriented deliverables, AI should be positioned as a *documented drafting aide* tied to explicit reasoning targets. We recommend embedding: (i) a course-level statement that defines allowed roles for AI; (ii) assignment prompts that include a constrained kick-off template; (iii) transcript appendices; and (iv) brief reflections that locate AI contributions relative to student decisions. Align homework and exams on the same cognitive moves (e.g., requirement tracing, justification, critique) to reduce homework–exam deltas observed in ad hoc contexts.

6.2 Program and Departmental Policy

Departmental AI guidance should articulate not only *permissions* but a shared *vision*: when and where AI belongs in the curriculum, to what ends, and what graduates must be able to do with (and without) AI. We recommend moving from prohibition-only language toward *scaffolded integration* that prepares students for AI-enabled workplaces while preserving integrity, transparency, and human authorship.

Vision and Intended Outcomes. Programs should publish a one-page AI statement that (i) defines AI's role in learning (augmentation, not substitution); (ii) states expected behaviors (disclosure, verification, correct attribution); and (iii) names measurable graduate outcomes (below). Policies can require disclosure, citation-like attribution for model use, and visible process artifacts, while also naming disallowed roles (e.g., idea origination, external sourcing) for specific assignments. Harmonizing policy language across core courses improves student clarity and reduces covert use.

Time and Place for Integration (Curriculum Map). For example, we could establish a sequenced map that specifies the *level-appropriate* use of AI:

- **First Year (Foundations):** Exposure and ethics. Students practice basic prompt literacy, disclosure, and simple verification on low-stakes tasks.
- **Second Year (Core Building):** Structured use in design/homework with transcript appendices and short reflections. AI supports drafting, not ideation or external sourcing.
- **Third Year (Systems/SE/HCI):** Team-based artifacts with stronger verification routines, traceability of human decisions, and assessment alignment (homework/exam parity).
- **Senior/Capstone:** AI as a professional tool under domain-specific constraints (e.g., licensing, privacy, security). Students justify or reject AI use on a per-task basis.

Graduate-Level Mastery Targets (By Graduation). Our departments should name explicit competencies and assess them:

- **Prompt & Process Competence:** Craft constraints; elicit justification; request checks; iterate productively.
- **Verification Competence:** Detect hallucinations/omissions; triangulate against specs, tests, or sources; document what was verified and how.
- **Design & Authorship Competence:** Produce clear, traceable design documents that separate human decisions from AI suggestions; defend choices without model assistance.
- **Ethics & Compliance Competence:** Disclose AI use; respect IP/privacy; follow course and organizational policy; cite or attribute model contributions appropriately.
- **Selective Use Competence:** Decide *when not* to use AI; explain risks (security, bias, brittleness) and choose alternatives when warranted.

Governance and Assessment Loop. Create a lightweight *AI-in-the-Curriculum* committee to:

- maintain exemplar policies, rubrics, and structured prompts;
- review *annual* evidence dashboards (e.g., homework–exam deltas, rubric rows on AI use, integrity incidents);
- run brief *calibration sessions* for instructors/TAs each term (shared failure modes, exemplar feedback);
- update guidance based on data (e.g., tighten/relax transcript requirements, adjust reflection prompts).

Program-Level Language (sample). *"AI tools may be used as **documented drafting aides** when permitted by the assignment. Students must disclose use, include a transcript excerpt or appendix when required, and verify key claims against specifications, tests, or cited sources. AI must not originate ideas, requirements, or external facts unless explicitly authorized. Courses align homework and exams on comparable reasoning moves to ensure independent mastery."*

This policy architecture treats AI as a curricular literacy with clear *time and place*, measurable outcomes, and a programmatic feedback loop—supporting consistent student development from novice exposure to capstone-level professional judgment.

6.3 Assessment and Evidence

Introduce rubric rows for (a) clarity of human reasoning, (b) appropriateness of AI use, and (c) quality of AI evaluation (error detection, bias checks, boundary testing). Track cohort-level homework–exam deltas and rubric subscores to monitor for substitution effects. Instructors can repurpose AI transcripts as formative checkpoints to target feedback on prompt engineering and verification practices.

Program–Level Assessment (Seminar/Capstone). Because ethics in our CS department is taught and verified in the capstone, departments can also *assess AI use* at this terminal stage (or in a junior seminar) to ensure graduates demonstrate professional judgment with AI:

- **Capstone AI Use Statement:** Each team submits a 1–2 page "AI Use and Verification" addendum summarizing when/why AI was used, what was *not* delegated, and how key claims were verified (tests/specs/citations). Faculty score with a short rubric.
- **Scenario Mini-Cases (Seminar):** Students respond to short vignettes (IP/privacy, data leakage, bias) and draft a constrained prompt + verification plan. Score on ethical reasoning and appropriateness of intended AI use.

Alignment to ABET Student Outcomes. AI assessment artifacts can be mapped to ABET outcomes (e.g., *ethics and professional responsibility*, *design*, *communication*):

- **Ethics & Professional Responsibility (ABET 4):** Disclosure, appropriate role limits, privacy/IP compliance, and an explicit verification record.
- **Design (ABET 2):** Clear separation of human decisions from model suggestions; traceability from requirements to design artifacts.
- **Communication (ABET 3):** Coherent justification of design choices in writing and oral defense; precise citation/attribution of AI contributions.

Signature Assignments & Evidence Adopt a small set of repeatable, program-level "signature" tasks with shared rubrics:

- **S1 (Junior Core):** Structured prompt + transcript appendix + 250-word reflection (drafting/verification focus).
- **S2 (Senior/Capstone):** AI Use and Verification addendum + oral defense (professional judgment focus).
- **S3 (Seminar Option):** Scenario mini-cases (ethics/policy application focus).

Rather than requiring continuous data collection across all courses each semester, we recommend that departments collect assessment data at strategically selected curricular milestones. In large computer science programs, this approach is more feasible and aligns evidence collection with points where students are expected to demonstrate integration, independence, and professional readiness.

Specifically, departments can collect and review an *evidence dashboard* at key moments such as a gateway software engineering course, a senior seminar, or the capstone design sequence. These courses naturally emphasize synthesis of knowledge, teamwork, documentation, and ethical decision-making, making them well-suited for evaluating the impact of structured AI-use practices. In this model, data collection occurs at moments of high instructional signal rather than at fixed calendar intervals.

At each milestone, the evidence dashboard may include rubric score distributions, homework–exam deltas, counts of academic integrity incidents (reported in aggregate only), and transcript-audit pass rates for required AI documentation or reflection components. Because these artifacts are already embedded in existing assessment workflows, they can often be gathered with minimal additional instructor burden. Importantly, the dashboard is intended to support trend analysis and reflective improvement rather than high-stakes evaluation of individual instructors or students.

Departments may also consider differentiated views of the dashboard for specific student populations. For example, an honors or high-achieving student cohort could be examined separately to explore how structured AI-use practices support advanced autonomy, leadership, and professional writing expectations. This lens allows departments

to identify whether students who are already performing at a high level are using AI to deepen understanding and refine professional practices, rather than merely accelerating task completion.

By shifting from time-based collection to milestone-based evidence gathering, departments can balance rigor with practicality. This approach enables scalable, program-level insight into how structured AI-use frameworks influence learning, integrity, and teaching effectiveness across the curriculum, while remaining adaptable to the constraints of large and diverse computer science programs.

Calibration and Moderation. Run brief, *pre-semester* TA/instructor calibration using 2–3 anonymized exemplars (strong/weak). Mid-semester, moderate a 10% sample across sections to tune inter-rater reliability and share common feedback stems.

Closing the Loop. Each year, the program committee reviews the dashboard and recommends small, data-informed adjustments (e.g., earlier transcript checkpoints; revised reflection prompts; stronger exam alignment). Document changes and track their effects in the subsequent cycle.

6.4 Faculty Development

Faculty development should focus on providing practical, reusable supports that help instructors integrate structured AI-use practices consistently across courses. Recommended resources include exemplars of structured prompts, transcript appendices, and short reflection forms that instructors can adapt to local contexts. These materials should be paired with brief "failure mode" galleries (e.g., hallucination carry-over, over-generalized patterns, or incorrect assumptions) and coaching scripts that model how instructors and TAs can respond when AI outputs are partly correct but pedagogically unhelpful.

To support consistent assessment, departments can run short *calibration sessions* each term using two to three anonymized student exemplars. These sessions should be aligned with shared rubric dimensions such as clarity of human reasoning, appropriateness of AI use, quality of verification, and ethics or policy compliance. Sharing *rubric anchors* and common feedback stems helps instructors and TAs apply consistent standards across signature tasks, such as structured prompt–transcript–reflection assignments in junior-level core courses, AI-use addenda and oral defenses in capstone, or mini-case analyses in senior seminars.

In addition to graded rubric rows, faculty development efforts should encourage the inclusion of *non-graded outcome indicators* that track whether students have met program-level learning outcomes related to responsible and effective AI use. These outcome indicators are not intended to influence course grades, but rather to document student progress toward competencies such as ethical AI use, transparency in AI interactions, critical evaluation of AI-generated outputs, and professional accountability. In practice, these indicators can be implemented as simple outcome checkmarks or mastery flags within a learning management system (e.g., Canvas outcomes), and are supported by most modern LMS platforms.

Using non-graded outcomes allows departments to gather longitudinal evidence of student development without increasing grading burden or incentivizing surface-level compliance. For example, a student may receive full credit for an assignment while still being flagged as "developing" in verification quality or ethical justification, providing instructors and advisors with actionable information without penalizing exploratory learning. Aggregated at the program level, these outcomes offer insight into how effectively students are meeting AI-related learning objectives across multiple courses.

To further lower adoption barriers, departments can provide quick-start faculty kits that include an oral-defense checklist, a one-page *transcript audit* guide, and sample LMS outcome mappings. Instructors are encouraged to feed aggregated results—both graded rubric data and non-graded outcome indicators—into the program's evidence dashboard at designated curricular milestones. This approach closes the loop between instruction, assessment, and continuous improvement, while supporting scalable faculty development in large computer science programs.

6.5 Research Agenda

Future work should extend the present findings in targeted and scalable ways rather than expanding the scope of inquiry substantially. In the near term, studies should triangulate existing sources of evidence—student artifacts, exams, and instructor interviews—across multiple offerings of the same course to strengthen confidence in observed patterns and

reduce reliance on any single measure. This includes examining homework–exam deltas over time to better understand how structured AI-use practices influence alignment between formative and summative assessment.

A second line of inquiry should examine how varying degrees of structure in AI-use frameworks influence student agency and independence in problem-solving. Rather than introducing entirely new interventions, future work can explore modest design variations, such as adjusting the timing or format of reflections, to better understand how structure supports learning without over-constraining exploration. These investigations would help clarify whether structured AI use primarily benefits novice learners by reducing unproductive exploration, while still allowing more advanced students to exercise autonomy.

Instructor buy-in represents an important, though often underexamined, factor in the sustainability of AI-supported instructional practices. Future studies should examine how instructors adopt, adapt, or selectively implement structured AI-use frameworks, and how variation in implementation relates to student outcomes and assessment alignment. Understanding the conditions under which instructors perceive structured AI use as supportive rather than restrictive will be critical for broader adoption.

Finally, to establish external validity, the framework should be examined across additional courses and institutional contexts. Replication in other core computer science courses, such as systems or databases, as well as at peer institutions with differing policies and cultures, would help determine which aspects of the framework generalize and which require local adaptation. Such work would position structured AI-use practices as a repeatable, evidence-informed approach rather than a course-specific solution.

Collectively, this research agenda emphasizes incremental refinement, instructor feasibility, and generalizability. By focusing on alignment, agency, and adoption rather than expansive new constructs, future studies can build a coherent body of evidence that supports responsible, scalable integration of AI into computer science education.

6.6 Limitations

Findings rely on internal, de-identified datasets and instructor self-reports; some student measures are perception-based and subject to response bias. Comparative analyses, while suggestive, are not randomized; observed homework–exam deltas may be confounded by cohort factors. We mitigate these limitations by reporting aggregated patterns, using convergent data sources, and focusing on replicable classroom procedures.

A Reusable Materials: Prompt Template, Transcript Guidance, and Reflection Rubric

A.1 A.1 Constrained Prompt Template (Copy/Paste)

Instructor Note: Students paste this as the first message to the LLM and then add only course-approved information.

> **Title:** Drafting Assistance for *¡Assignment Name¿* Using My Provided Content Only
>
> I would like help *drafting and formatting* an *¡artifact type¿* using the attached/template structure.
> **Do not** originate ideas or content for me. **Do not** draw on outside sources or prior knowledge.
> I will supply all domain information. You may: restructure, clarify wording, suggest headings, and ask me focused questions
> *only* when required by the template and missing from what I provided.
>
> **Scope limits:** You may help with grammar, organization, and consistency checks.
> **Disallowed:** new requirements, new facts, new design decisions (unless I explicitly authorize).
>
> Here is my information to use verbatim/paraphrase only: *¡paste team-authored content¿*
>
> If you need something I have not provided, ask a specific question so I can supply it.

A.2 A.2 Transcript Appendix Guidance

- Export the *entire* model interaction history (all turns) as PDF or text; redact personal identifiers.
- Place the prompt template as the first line; label subsequent turns with timestamps or #1, #2, . . .
- If images/files were uploaded, list filenames and a one-line description of each (no private data).

- Do not remove "failed" or exploratory turns; mark them with a one-line note (e.g., "discarded due to hallucination").
- Link excerpt numbers in your reflection (e.g., "see Transcript #7" for the claim I verified).

A.3 A.3 Short Reflection Prompt and Rubric

Student Prompt (200–300 words). In bullet form, state: (1) which parts were authored by you vs. the AI; (2) one issue you found in the AI output (e.g., hallucination, bias, incorrect template use) and how you detected it; (3) how you verified key claims; and (4) what you would change about your next prompt to get better results.

3-Row Rubric (5-point scale each).

1. **Clarity of Human Reasoning (0–4):** 0=unclear; 2=mixed; 4=clear decisions and justifications are attributable to the student/team.
2. **Appropriateness of AI Use (0–4):** 0=disallowed uses; 2=minor scope creep; 4=stays within drafting/formatting/explaining.
3. **Quality of AI Evaluation (0–4):** 0=no verification; 2=surface checks; 4=explicit checks with evidence (e.g., cites Transcript # and correction).

Reporting: Instructors may provide each row score and a single-sentence feedforward note.

References

[1] Brinley Boyett and Robert H. Lightfoot Jr. Examining educators' impact on learning environment, technological literacy, and teaching effectiveness through integrating ai in the classroom. In *Proceedings of the 2025 ASEE Annual Conference & Exposition*. American Society for Engineering Education, 2025. Paper ID 49519.

[2] H. Chang, B. Liu, Y. Zhao, Y. Li, and F. He. Research on the acceptance of chatgpt among different college student groups based on latent class analysis. *Interactive Learning Environments*, 2024. doi: 10.1080/10494820.2024.2331646.

[3] Scott Freeman, Sarah L. Eddy, Miles McDonough, Michelle K. Smith, Nnadozie Okoroafor, Hannah Jordt, and Mary Pat Wenderoth. Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, 111(23):8410–8415, 2014. doi: 10.1073/pnas.1319030111.

[4] Nicia Guillen-Yparrea and F. Hernandez-Rodriguez. Unveiling generative ai in higher education: Insights from engineering students and professors. In *2024 IEEE Global Engineering Education Conference (EDUCON)*, 2024. doi: 10.1109/educon60312.2024.10578876.

[5] Tonia Haikal and Robert Lightfoot Jr. Enhancing education through thoughtful integration of large language models in assigned work. In *Papers on Engineering Education Repository (ASEE)*, 2024. doi: 10.18260/1-2--45377.

[6] International Technology Education Association. *Standards for Technological Literacy: Content for the Study of Technology*. ITEA, Reston, VA, 2000.

[7] Enkelejda Kasneci, Kathrin Sessler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, and Gjergji Kasneci. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103, 2023. ISSN 1041-6080. doi: https://doi.org/10.1016/j.lindif.2023.102274. URL https://www.sciencedirect.com/science/article/pii/S1041608023000195.

[8] Rose Luckin, Wayne Holmes, Mark Griffiths, and Lauréate B. Forcier. Intelligence unleashed: An argument for AI in education. https://discovery.ucl.ac.uk/1475756/, 2016.

[9] Sarin Sok and Kimkong Heng. Chatgpt for education and research: A review of benefits and risks. *SSRN Electronic Journal*, 01 2023. doi: 10.2139/ssrn.4378735.

[10] Chris Stokel-Walker and Richard Van Noorden. The promise and peril of generative ai. *Nature*, 614:214–216, 2023. doi: 10.1038/d41586-023-00340-6.

[11] R. Sukumar, V. Gambhir, and J. Seth. Investigating the ethical implications of artificial intelligence and establishing guidelines for responsible ai development. In *2024 International Conference on Advances in Computing, Robotics and Secure Communication (ACROSET)*, 2024.

[12] M. S. I. Taufikin, None Supa'At, Nur Azifah Nikmah, Jayasmita Kuanr, and None Parminder. The impact of ai on teacher roles and pedagogy in the 21st century classroom. In *2024 International Conference on Knowledge Engineering and Communication Systems (ICKECS)*, pages 1–5, 2024. doi: 10.1109/ickecs61492.2024.10617236.

[13] UNESCO. Guidance for generative ai in education and research. https://www.unesco.org/en/articles/guidance-generative-ai-education-and-research, 2023. Accessed 2026-01-19.

[14] L. L. A. White, T. Balart, K. Shryock, and K. Watson. Understanding faculty and student perceptions of chatgpt. ASEE Paper, 2024.