

A Five-Dimensional Integrated Teaching Practice for Software Engineering to Improve Ethical Awareness under Generative AI

Yajing Bai^{1,a,*}, Qing Lin^{1,b}, Li Li^{1,c}

¹School of Intelligent Science and Engineering, Xi'an Peihua University, Xi'an, 710125, China

^a002923@peihua.edu.cn, ^blq2008@peihua.edu.cn, ^cll@peihua.edu.cn

*Corresponding author

Abstract: Generative AI has profoundly transformed modern software engineering workflows. While enhancing development productivity, it introduces notable pedagogical challenges, such as students' excessive reliance on AI generated outputs and insufficient awareness of AI hallucination risks, which weakens their capacity for independent engineering reasoning. Conventional Software Engineering curricula predominantly prioritize technical skill training yet lack systematic instruction to foster professional accountability and AI related ethical literacy. To address these gaps, this paper proposes a cognition reflection practice enlightenment action five dimensional collaborative teaching framework. Value oriented and AI ethical evaluation components are embedded within full lifecycle engineering deliverables, with each ethical analysis sub module assigned a weight of 20 % in the feasibility analysis report, requirement analysis report, and overall final project report. Supported by an industry university competition certificate integrated mechanism that incorporates professional certifications, discipline contests and enterprise sourced practical materials, this multi-year reform has been implemented across more than 20 undergraduate classes involving nearly 1500 students at a local application oriented university. The proposed framework establishes a closed loop teaching pipeline spanning knowledge acquisition, ethical deliberation, hands on practice, value internalization, and normative behavior formation. Empirical teaching observations suggest that the approach cultivates students' rational perceptions of generative AI and strengthens their recognition of software engineers' social accountability. This paper further discusses study limitations and prospective research directions and provides replicable references for values driven computing education in the large model era.

Keywords: Generative AI; Software Engineering; engineering education reform; AI ethics; professional responsibility; five dimensional teaching framework; industry university competition certificate integration; project embedded assessment

1. Introduction

Large model driven generative AI coding assistants have reshaped software engineering practices and drawn growing attention from computing educators^[1]. Despite substantial productivity gains, empirical studies document widespread over trust among student learners toward AI generated artifacts. Many students accept AI produced source code and written documents without rigorous validation, exposing themselves to logical defects, copyright infringements, and algorithmic bias originating from AI hallucinations^[2]. Modern day software engineers are expected to possess solid technical proficiency while accepting social responsibility for systems they develop^[3].

Software Engineering constitutes a core undergraduate level discipline course. Generative AI functions merely as a productivity augmenting auxiliary tool. High impact engineering decisions including requirement trade offs, architectural design, and quality risk governance still hinge on human judgment and accountability^[4]. Conventional teaching often segregates technical skill building from value oriented cultivation. Ethical related content tends to remain fragmented, and few curricula embed quantifiable ethical assessment within authentic engineering deliverables^[5]. Many institutions treat generative AI exclusively as a novel development utility and omit targeted instructional designs to nurture critical AI usage competence and computing ethics^[6].

Existing studies concerning AIGC enhanced Software Engineering education largely focus on case repository construction or OBE oriented syllabus redesign^[7]. Relatively few empirical investigations

develop complete closed loop teaching workflows with quantifiable evaluation embedded in project deliverables while integrating certification, competition, and industrial resources for values oriented talent development^[8].

Against this backdrop, this paper reports multi-year cyclic teaching reform carried out at a local application oriented university. A cognition reflection practice enlightenment action five dimensional collaborative teaching framework is put forward, accompanied by document embedded ethical assessment and industry university competition certificate integrated arrangements. The reform has reached over 20 classes with nearly 1500 undergraduate participants. This article elaborates framework connotations, implementation workflows, assessment design, practical constraints, and lessons drawn from practice.

2. Theoretical Background

2.1. Values Oriented Requirements for Software Engineering under Generative AI

Software systems underpin critical societal infrastructure covering government services, finance, and healthcare. Consequently, software artifacts yield far reaching societal impacts, rendering professional responsibility a required graduate competency for computing majors. The ACM IEEE CS curriculum guidelines advise integrating computing ethics and social impact analyses throughout core coursework rather than treating ethics as isolated elective modules^[9].

Generative AI brings new dimensions to ethics focused education. Prior research identifies three prominent instructional challenges: students underestimate hazards stemming from AI hallucinations and lack validation routines; intellectual property boundaries for AI generated outputs confuse learners; students tend to transfer engineering accountability to large model systems^[10]. Accordingly, modern Software Engineering instruction should instill a core consensus: human engineers bear primary accountability for software systems even when generative AI participates in development workflows.

2.2. Identified Pedagogical Gaps in Project Oriented Courses

Project based learning (PBL) has been widely adopted within Software Engineering education^[11]. Nevertheless, conventional PBL predominantly evaluates deliverables based on functional and technical performance. Value relevant learning outcomes such as ethical awareness and social value considerations are frequently regarded as informal, non graded side effects. Three notable gaps persist:

1) Decoupling between technical practice and value oriented reflection: Ethical deliberation is seldom made mandatory within requirement specifications, feasibility analyses, or final project reports.

2) Absence of quantifiable evaluation metrics: Professional responsibility represents internalized personal disposition. Final exam centered assessment cannot adequately capture such learning outcomes^[12].

3) Insufficient generative AI oriented instructional interventions: Numerous PBL assignments lack explicit specifications governing AI assisted development and fail to guide critical large model utilization.

Motivated by these pedagogical gaps, this reform embeds values-oriented and AI-ethical requirements into full-cycle project-document workflows to establish a structured multi-stage teaching solution.

3. Insufficient generative AI oriented instructional interventions

The proposed teaching solution forms a closed-loop model consisting of five progressive phases: cognition, reflection, practice, enlightenment, action, each associated with specific teaching objectives and tailored learning activities. Its holistic architectural overview is presented in Figure 1.

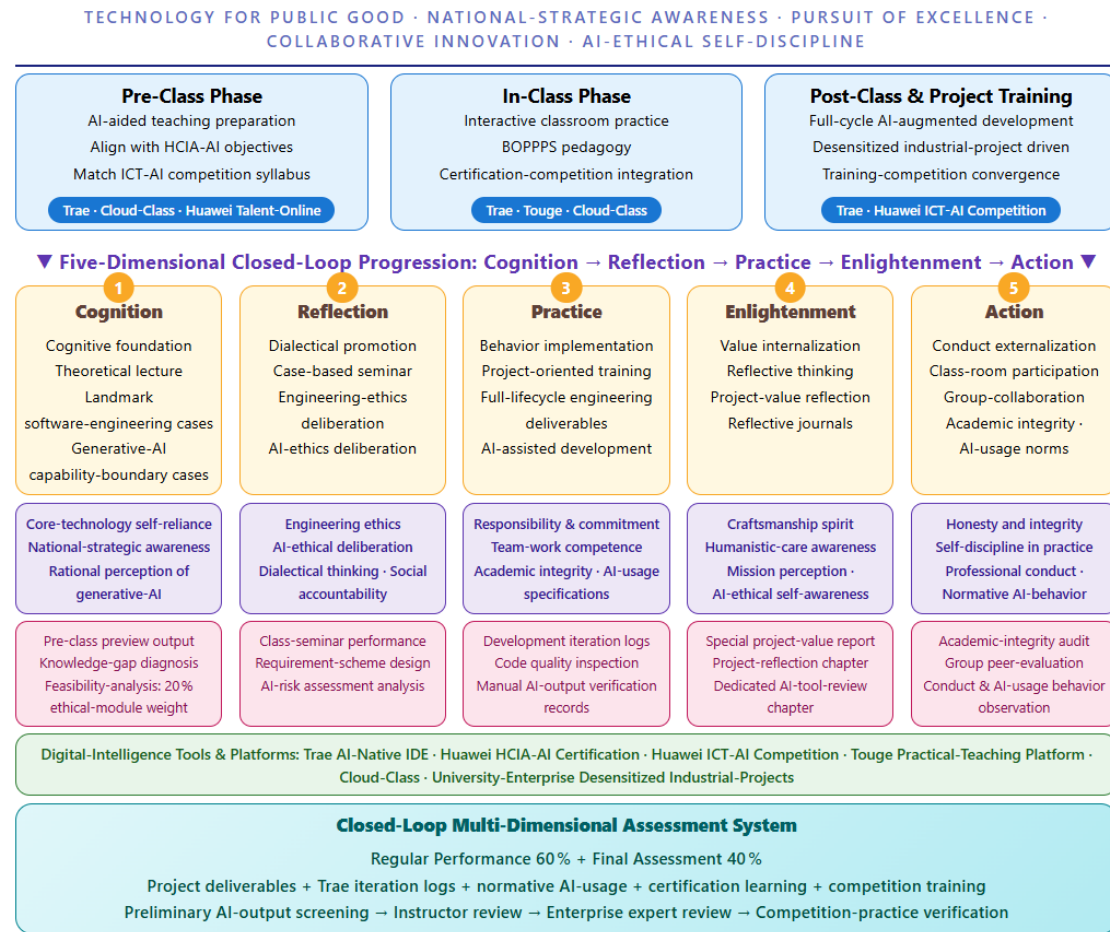


Figure 1. Architecture of the cognition-reflection-practice-enlightenment-action five-dimensional collaborative teaching framework.

3.1. Cognition: Build Foundational Awareness

Theory centered lectures combine landmark software engineering case studies with real world demonstrations of AI induced risks. Instruction covers software crisis theories, software quality criteria, and practical examples of security vulnerabilities originating from AI hallucinations. Students develop two core understandings: human responsibility remains irreplaceable within software projects, and generative AI assistants possess well defined capability boundaries.

3.2. Reflection: Engineering and AI Ethics Deliberation

Seminar driven instruction is deployed in chapters addressing feasibility analysis, project management, and human computer interaction. Historical software failure cases and inclusive financial system scenarios trigger discussions balancing technical feasibility against public interest demands. Seminar themes include suitable versus inappropriate AI assisted subtasks, identification of AI hallucination risks, and copyright management practices for AI produced materials. Drawing on discipline competition topics, students investigate algorithmic fairness concerns and responsibility attribution mechanisms within AI augmented software development workflows.

3.3. Practice: Project Based Training and Document Assessment Reform

People livelihood oriented simulated projects, such as aging friendly digital services and public convenience mini programs, are adopted for practical training. Mandatory deliverables are restructured with ethical requirements embedded across multiple reports. The ethical and AI ethical analysis sub modules account for 20 % of the score of each of the three key documents: feasibility analysis report, requirement analysis report, and overall final project report.

1) Preliminary special report: Students examine landmark domestic software engineering cases and discuss professional obligations for practitioners working within the generative AI paradigm.

2) Feasibility analysis report: Retain conventional technical economic feasibility sections and add sub sections for social value discussion and AI risk analysis.

3) Requirement analysis report: Append constraints addressing humanistic care, vulnerable group protection, and data security. For projects incorporating AI enabled functions, dedicated explanations of algorithm fairness and privacy protection designs shall be supplied.

4) Overall final project report: Include a value reflection chapter plus an independent AI tool review chapter documenting AI usage contexts, manual verification workflows, and resolution strategies for AI hallucination related incidents.

Formal training regulations are enforced: AI shall operate exclusively as an auxiliary tool. Requirement analysis, architectural design, and core logic implementation must be completed independently by students. All AI generated code and textual fragments require source attribution alongside manual inspection. Integrated development environment interaction logs are preserved so instructors can audit learners' critical usage of AI outputs. Project grading considers both system functional performance and analytical depth demonstrated in deliverables regarding social value reasoning and AI risk judgment.

3.4. Enlightenment: Promote Value Internalization

Reflective instruction draws on public domain cases documenting software system societal impacts and real world AI ethical failures. Students internalize engineers' social mission and accept the core consensus: human engineers hold primary system level accountability irrespective of large model capabilities. Associated AI security and ethics certification related knowledge supports learners' transition from passive knowledge reception toward active value recognition.

3.5. Action: Externalize Normative Behaviors

This dimension targets observable conduct, encompassing classroom participation norms, group collaboration etiquette, document writing conventions, and standardized generative AI utilization. Requirements incorporate academic integrity rules, prohibitions against wholesale reuse of complete AI generated deliverables, and mandatory marking plus verification for AI produced materials. Competition participating students sign academic integrity commitment letters; contest submissions must demonstrate original design work together with manual validation of AI derived outputs. Behavioral observation extends to competition preparation activities.

3.6. Internal Logic of the Framework

The five dimensional sequence forms a progressive closed loop causal chain: Cognition delivers knowledge input; Reflection facilitates ethical deliberation; Practice realizes hands on engineering training with AI assisted development; Enlightenment enables internal value recognition; Action translates internalized beliefs into observable real world conduct. This chain spans pre class preparation, in class instruction, after class expansion, and hands on practical training to simultaneously achieve multi-faceted learning objectives.

4. Implementation and Multi-Dimensional Assessment

4.1. Five in One Teaching Objectives

1) Knowledge objective: Master full lifecycle Software Engineering theories; comprehend generative AI application contexts within software development; recognize human engineers' irreplaceable roles.

2) Engineering competence objective: Conduct requirement investigation, scheme design, and test planning for small to medium sized software artifacts; produce standardized project deliverables; identify AI hallucination risks and validate AI generated outputs.

3) Value oriented objective: Comprehend software engineers' social accountability; perform social impact and ethical analyses within engineering deliverables.

4) Moral conduct objective: Uphold academic integrity; fulfill duties during group collaboration; internalize professional ethics norms.

5) AI critical thinking objective: Distinguish merits and drawbacks of generative AI; identify risks including hallucinations, algorithmic bias, and copyright infringement; uphold the principle of “AI assistance with human led accountability”; maintain independent engineering reasoning.

4.2. Integrated In Class / Out of Class Teaching Path

Blended teaching merges case oriented instruction, project driven learning, and BOPPPS pedagogy, supported by AI native IDE tools, online practical platforms, enterprise resources, and certification competition materials.

Pre class: Assign reading resources addressing software engineering social impact and AI ethics; release pre training materials aligned with certification competition syllabi; clarify generative AI usage specifications.

In class: Adopt the instructional pattern of “knowledge point + case + value expansion + AI ethical discussion”, supplemented by short thematic seminars. AI IDE tools are utilized for requirement validation and code debugging exercises, and error correction training is delivered based on real-world large language model failure cases.

Practical phase: Assign people livelihood oriented project topics; rigorously enforce deliverable requirements incorporating embedded ethical modules. Guest lectures on industrial AI development specifications and associated ethical risks are delivered by industry engineers.

Advanced after class training: Group based projects are built upon desensitized industrial project prototypes. Fundamental training consolidates theoretical knowledge through online platforms; advanced projects adopt AI collaborative development while mandating student independent core design work. Qualified students prepare for professional certification; competition teams are assembled from course outputs to achieve bidirectional conversion between coursework and contest achievements.

4.3. Multi-Dimensional Assessment System

Pure final exam driven evaluation is discarded. A composite assessment scheme is implemented with 60% regular performance and 40 % final assessment. Regular performance metrics include classroom seminar participation, project deliverable quality, and learning conduct observation. A 20 % score weight for ethical and AI ethical evaluation is assigned to each of the three core technical reports, differentiating mechanically reproduced AI generated template text from genuine independent student analyses. Final assessment options include written examination or comprehensive project defense, evaluating both Software Engineering domain expertise and AI ethical cognition. This evaluation framework translates abstract value orientation and professional ethics requirements into traceable documentary evidence and observable learning behaviors.

5. Support Conditions, Limitations and Counter Measures

5.1. Team and Platform Support

The reform team comprises Software Engineering curriculum group instructors together with value education faculty. Lead course instructors possess HCIA AI senior level certification and have completed specialized AI native IDE training. Team members combine university teaching experience with industrial engineering backgrounds. Scheduled collective lesson preparation activities refine teaching cases, scoring rubrics, and behavioral observation indicators.

Available resources include on campus computer laboratories. Enterprise cooperation partners supply real world project materials alongside industrial AI development specifications. AI native IDE and online practical training platforms support laboratory execution and learning status monitoring. University level teaching reform grants fund case library revisions and teaching research exchanges.

5.2. Study Limitations

This instructional practice is implemented within a local application oriented university; participant

samples predominantly consist of domestic undergraduate students enrolled at this institution. Direct replication for research intensive universities or higher vocational colleges necessitates adaptive adjustments according to learner baseline characteristics. In addition, this investigation mainly draws upon classroom observation, deliverable based analysis, and qualitative teaching feedback; controlled cross university comparative experiments have not yet been conducted. Future research will cooperate with multiple institutions to expand sample size and carry out cross institutional validation.

5.3. Risks and Counter measures

1) Superficial integration of technical content and ethics education: Conduct unit wise collective lesson preparation. Disciplinary and value education instructors jointly refine instructional designs. Deliverable writing guidelines are compiled, requiring ethical analysis to be grounded in project business scenarios. Academic integrity requirements and AI usage constraints are embedded in task documents. Regular class auditing procedures are organized to iteratively improve teaching quality.

2) Subjective evaluation risks and student submitted stereotyped AI generated ethical paragraphs: Refine scoring rubrics and behavioral observation worksheets; establish three tier evaluation criteria: template reproduction / superficial narration / in depth independent project rooted analysis. Multi-source evidence, including project documents, classroom presentations, and group peer reviews, is synthesized, while qualitative descriptions are combined with quantitative grading to ensure traceable assessment outcomes.

3) Accelerated obsolescence of AI ethics related teaching cases: Establish a multi-member content collection group consisting of instructors, student representatives, and enterprise specialists. Software industry trends and national AI governance policies are tracked, and in-house case libraries are iteratively updated accordingly.

4) Learner competence stratification and faculty workload originating from certification competition activities: Deploy stratified teaching approaches; certification and competition participation remain non mandatory for all students. Targeted intensive training is provided for high-achieving subgroups. Standardized practical training documentation is developed, and instructors' digital teaching and AI ethics instructional competencies are enhanced through enterprise-sponsored training and internal teaching research workshops.

6. Reform Outcomes and Conclusion

6.1. Reform Outcomes

Generated deliverables include revised value oriented teaching case collections, updated syllabi, comprehensive practical document writing specifications, ethical assessment scoring rubrics, and behavioral observation worksheets. The five dimensional collaborative teaching framework and indicator matrix are formally consolidated.

For student participants: Learners strengthen awareness of software engineers' social accountability, develop rational stances toward generative AI tools, and acquire preliminary capabilities for detecting AI hallucination risks. Students can independently complete social value and AI ethical analyses within engineering deliverables; improvements are observed regarding academic integrity awareness and teamwork competencies.

For instructors: Faculty gain enhanced competencies for ethics integrated Software Engineering instructional design; collaborative teaching experience accumulates between technical instructors and value education educators. The established methodology can serve as a reference resource for additional computing related courses offered within the university.

This reform does not demand high cost hardware investments. It primarily relies on instructional design refinement, case resource development, and practical document mechanism renovation. Accordingly, the proposed framework exhibits transferable reference value for comparable application oriented universities.

6.2. Conclusion

Generative AI reshapes software development paradigms. Nevertheless, human centered systematic

engineering thinking and professional accountability cannot be superseded by large model systems. Values oriented Software Engineering education is not supplementary moral oriented preaching appended onto technical curricula; rather, it constitutes a systematic educational project integrating knowledge delivery, competence cultivation, value shaping, and AI critical thinking development.

This paper establishes the cognition reflection practice enlightenment action five dimensional collaborative teaching framework. Through project document assessment reform, each core technical report is assigned a 20 % score weight for value orientation plus AI ethical modules, translating professional ethics criteria into observable learner conduct. The reported practice addresses multiple practical teaching pain points: disconnection between technical training and value cultivation, difficulties quantifying ethical competence outcomes, insufficient AI ethics instruction, and outdated teaching case resources.

Curriculum teaching reform represents a long term iterative undertaking. Follow up research will continuously track software industry and large model technology evolution, dynamically refresh teaching case resources, optimize assessment indicators, deepen university enterprise collaboration, and further explore responsible talent cultivation pathways for Software Engineering within the generative AI era.

Acknowledgements

This work is supported by University level Teaching Reform Project of Xi'an Peihua University under Grant PHKCSZ202513.

References

- [1] S. Kumar, S. S. Datta. *Large language models in software engineering education: A systematic mapping study*[J]. *IEEE Transactions on Education*, 2024, 67(2): 145-154.
- [2] R. A. Lutz, T. D. Bowman. *Student pitfalls with generative AI code assistants: Hallucination, over trust and attribution issues*[J]. *Computer Science Education*, 2025, 35(1): 42-63.
- [3] C. Fiesler, N. Garrett. *Computing ethics education: Where the curriculum meets real world impacts*[J]. *Communications of the ACM*, 2023, 66(7): 54-59.
- [4] M. Petrenko, L. Kovalchuk. *Software engineering fundamentals in the age of generative artificial intelligence*[J]. *Computer Applications in Engineering Education*, 2024, 32(3): 789-806.
- [5] M. J. Wolf, K. Miller. *Embedding social responsibility assessment into project based software engineering courses*[J]. *Journal of Computing Sciences in Colleges*, 2023, 38(8): 91-98.
- [6] H. H. Hoffmann. *Teaching software engineering with generative AI: Curriculum gaps and instructional strategies*[C]//2024 IEEE International Conference on Software Engineering Education and Training (CSEE&T). Lisbon: IEEE, 2024: 112-119.
- [7] L. Zhang, Y. Wang. *OBE based software engineering course redesign for generative AI era*[C]//2024 13th International Conference on Educational Informatics. Cham: Springer, 2024: 217-224.
- [8] T. S. Costa, R. Moreira. *Industry certification and competition enhanced software engineering education: A systematic review*[J]. *Education and Information Technologies*, 2025, 30(4): 4781 -4803.
- [9] ACM IEEE CS. *Computer Science Curricula 2023: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*[R]. New York: ACM IEEE CS, 2023.
- [10] J. Weidinger, et al. *Ethical risks and safeguards of generative large language models*[J]. *AI and Ethics*, 2024, 4(1): 29-44.
- [11] J. E. Burge, R. J. Mitchell. *Project based learning in software engineering education: A meta review*[J]. *IEEE Transactions on Education*, 2022, 65(4): 281-289.
- [12] S. Barua. *Assessing ethical learning outcomes in computing education: Challenges and rubric based approaches*[J]. *Journal of Information Systems Education*, 2023, 34(2): 103-112.