2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC

Host Sponsor UNC Charlotte

Paper ID #52991

# AI-Supported Systems Engineering: A Proposed Undergraduate Course Design with Client-Based Projects

**Dr. Maryam Heidaripour, University of the Pacific**

Maryam Heidaripour is an Assistant Professor of Design, Innovation, and Entrepreneurship at the University of the Pacific. Her research explores how emerging technologies, social change, and environmental challenges are reshaping engineering education and practice. Grounded in the view that change emerges through collective action, her work brings engineers and collaborators together to shape new approaches to the field.

# AI-Supported Systems Engineering: A Proposed Undergraduate Course Design with Client-Based Projects

## Introduction

Traditional systems engineering pedagogy emphasizes disciplined lifecycle processes, requirements traceability, and formal documentation practices grounded in standards and canonical texts such as the *INCOSE Systems Engineering Handbook* [1, 2]. These approaches establish rigor and accountability, but in undergraduate settings they are often implemented through document-heavy and prescriptive instructional models that provide limited realism and constrained opportunities for professional judgment [3, 4].

At the same time, artificial intelligence (AI) is increasingly embedded in the development and operation of engineered systems. In industry practice, AI-enabled workflows now support activities such as requirements analysis, traceability maintenance, model documentation, and stakeholder data synthesis, reshaping how systems are conceived, validated, and governed [5, 6]. As AI becomes a routine component of systems practice, engineers are expected not only to use AI tools effectively, but also to reason about their limitations and impacts.

Prior research emphasizes that AI-enabled systems fundamentally challenge traditional engineering practice. Amershi et al. [5] argue that machine-learning-based systems introduce new sources of complexity related to data dependence, model behavior, and lifecycle evolution, rendering conventional software engineering processes insufficient. From this perspective, AI is not simply another software feature, but a socio-technical system component that requires distinct workflows, governance mechanisms, and documentation practices. Complementing this view, the NIST 2023 Annual Report highlights the rapid expansion of AI and other critical technologies and underscores the need for robust standards, measurement science, and governance frameworks to ensure responsible and trustworthy deployment [6]. However, these shifts in systems practice are not yet reflected in how systems engineering is commonly taught at the undergraduate level.

This mismatch is particularly consequential for undergraduate engineers working with small and mid-sized enterprises (SMEs). SMEs often lack formal systems engineering infrastructure, specialized roles, and advanced tooling, placing greater emphasis on disciplined reasoning, documentation, and traceability as mechanisms for coordination and accountability. In such contexts, students cannot rely on organizational scale or specialization to compensate for weak systems thinking; instead, they must engage directly with tradeoffs and uncertainty in ill-structured, socio-technical problem settings.

Addressing this gap requires educational approaches that strengthen systems engineering rigor while developing responsible AI literacy, incorporating human-centered and participatory design, and remaining scalable for constrained environments. Rather than evaluating a single instructional technique, this work examines how undergraduate systems engineering education can be restructured to better align with emerging AI-assisted tools and practices.

This paper presents *Systems Engineering with Client-Based Projects*, an undergraduate, SME-partnered systems engineering course designed to meet these needs. It introduces a fully developed course design and evaluation framework as a curricular model; the paper offers a detailed, theory-informed proposal intended for replication and adaptation. The course treats AI as a scaffold for rigor and reflection, explicitly integrating responsible AI practices, traceability, and stakeholder engagement into authentic system design work. Through collaboration with regional SME partners, students prototype lightweight, auditable systems while learning to balance technical rigor with human, organizational, and ethical considerations.

This paper contributes a transferable course design and suggests an evaluation framework and addresses the following research questions:

1. How can AI be integrated into undergraduate systems engineering courses to strengthen rigor without displacing human judgment?
2. How can systems engineering education support responsible AI practice in constrained, real-world contexts such as SMEs?
3. What educational structures, assignments, and evaluation methods support successful implementation?

Although situated in systems engineering, this work is fundamentally a design education contribution: it treats requirements, traceability, and foresight artifacts as design representations that evolve through iterative cycles of problem framing, stakeholder negotiation, and prototyping. The course adopts a studio-informed, client-based model that emphasizes design under uncertainty and sociotechnical tradeoffs; core themes in the Design in Engineering Education Division (DEED). In doing so, it extends design pedagogy to AI-supported workflows by making students' design reasoning, judgments, and accountability explicit in the artifacts they produce.

## Related Work and Educational Context

### *Systems Engineering Education*

At the curriculum level, prior work suggests that undergraduate engineering education remains fragmented and prescriptive. Robinson-Bryant [4] describes undergraduate curricula as emphasizing well-defined problems and correct answers, often at the expense of holistic systems reasoning. In systems engineering contexts, this has frequently translated into document-centric and manual pedagogies that are poorly aligned with contemporary, AI-enabled system development. Similarly, the International Council of Systems Engineering (INCOSE) recognizes the need for stronger academic preparation in systems engineering and advocates for engineering degree programs with deeper integration of systems engineering principles and systems thinking [1].

From the perspective of disciplinary integration, these educational gaps are particularly consequential given the increasing prevalence of multidisciplinary and ill-structured engineering problems. Asher et al. [7] observe that modern engineering challenges frequently span multiple disciplines and argue that discipline-specific instruction alone is insufficient for preparing students to manage complexity, risk, and coordination at the system level. Their work positions systems engineering as a unifying framework for multidisciplinary problem solving and provides evidence that exposure to systems engineering principles can improve student learning outcomes in such contexts.

At the level of professional readiness, Wasson [2] argues that the absence of a required undergraduate course in systems engineering fundamentals constitutes a critical educational gap, leaving graduates underprepared to perform up-front analysis, requirements flow-down and traceability, analysis of alternatives, lifecycle thinking, and data-driven decision making. Squires et al. [3] further contend that systems engineering maturity typically develops over many years of professional experience and propose technology-enabled, experience-based learning as a means of accelerating competency development beyond process- or document-focused instruction.

*Design Education and Client-Based Pedagogies*

Engineering design education research emphasizes that students learn design through iterative cycles of framing, making, testing, and reflecting, particularly in ill-structured contexts where requirements and success criteria evolve [8, 9]. Studio-based and client-based pedagogies make this process visible through critiques, design reviews, and repeated engagement with stakeholder needs. This course builds on those traditions by treating systems engineering artifacts (requirements, traceability, rationales) as evolving design representations and by using AI to prompt iteration, surface assumptions, and support reflective judgment rather than replace design decision-making.

*AI and Data-Centric Engineering Education*

Recent literature on AI in engineering education underscores the need for more intentional curricular integration. Schleiss et al. [10] argue that effective AI education cannot be achieved through isolated tools and add-on courses but instead requires alignment with disciplinary practice and institutional context. Building on this view, another research [11] proposes a role-based competency framework that maps AI literacy to the professional roles engineers perform in practice, such as AI consumer, user, and developer. This approach emphasizes embedding responsible AI use within routine engineering workflows across the curriculum.

Complementary work in systems engineering focuses on the technical potential and limitations of AI in practice. Poulsen et al. [12], through a comprehensive review, identify substantial opportunities for AI to augment systems engineering and model-based systems engineering activities across the V-model, including requirements engineering, architecture, modeling, and verification. At the same time, they highlight persistent socio-technical barriers, such as limited data quality, lack of explainability, and trust concerns that constrain the realization of this potential.

More narrowly scoped empirical studies illustrate how these opportunities and constraints play out at the level of systems engineering tasks. Levy et al. [13] demonstrate that generative AI can support requirements engineering by classifying requirements, assessing their quality against

INCOSE criteria, and generating preliminary test artifacts. However, they emphasize that AI outputs require expert oversight due to risks such as hallucination and misinterpretation, framing AI as an auxiliary reasoning aid, and not an automated decision authority.

This body of literature positions AI as a means of augmenting systems engineering practice when its use is grounded in core SE principles, human judgment, and governance.

*Anticipatory Systems and Future-Oriented Design*

Anticipatory thinking has also been positioned as a core competency in systems engineering and design. Poli [14] frames anticipation as an integrative design capability that enables teams to link present design choices to plausible future outcomes through shared artifacts and collaborative processes. Nadin [15] further argues that anticipation is a fundamental property of living and complex systems, underpinning adaptation, learning, and decision-making. In this context, an anticipatory system is one that contains a predictive model of itself and/or its environment, enabling present action based on representations of possible futures [16].

Within engineering and design contexts, some argue from an industry perspective that organizations can deliberately build anticipatory capacity by leveraging model-based systems engineering to reason about evolving future contexts rather than relying on static, point-in-time approaches [17]. From an educational perspective, recent work has made the case for incorporating design futures into engineering education, emphasizing that anticipatory methods enable students to explore alternative futures and make more robust design decisions under uncertainty [18]. Related work positions anticipation as an ethical responsibility, particularly in the context of emerging technologies with long-term societal impacts [19, 20]. Together, these contributions frame anticipatory thinking as a core systems engineering competency that spans both professional practice and engineering education.

Despite this growing body of work, no existing undergraduate educational model integrates AI-supported systems rigor, participatory stakeholder engagement, anticipatory ethics, and authentic SME contexts within a single, coherent course. This gap motivates the course design presented in this paper. Compared to traditional undergraduate systems engineering courses, the proposed model differs in four key respects. While conventional courses often treat requirements and traceability as static documentation exercises, this course positions them as evolving design representations revised through iterative critique and stakeholder feedback. AI is not introduced as a standalone topic or tool demonstration but is embedded directly within core systems engineering artifacts as a structured second-pass reasoning aid. Projects are conducted with live SME partners rather than hypothetical case studies, introducing authentic organizational constraints and ambiguity. The course also adopts a studio-based, critique-driven structure that emphasizes iteration, tradeoff documentation, and reflective judgment rather than linear artifact completion.

## Course Design Framework

The course is organized around a hybrid design framework that addresses limitations of traditional systems engineering pedagogy while incorporating AI-enabled capabilities. The framework brings together four complementary strands that structure student engagement with real-world system design problems: systems engineering rigor, stakeholder-centered participation, iterative design cycles, and anticipatory systems thinking (Figure 1).

While established systems engineering texts emphasize lifecycle rigor, traceability, and formal documentation, complementary perspectives on participatory design, iterative knowledge generation, and future-oriented reasoning are distributed across adjacent research traditions (Table 1). This work synthesizes these perspectives into a coherent, course-level framework tailored to undergraduate systems engineering education. In DEED terms, the framework treats requirements models, traceability matrices, and foresight memos as design representations that mediate problem framing, exploration of alternatives, and justification of tradeoffs. These artifacts are intentionally revisited across iterations, allowing students to experience design as an evolving process from *requirements* to *implementation*.

Table 1: Conceptual origins of the four strands integrated in the course design framework

| **Strand** | **Origin** |
|---|---|
| SE rigor | Classical systems engineering |
| Stakeholder-centered framing & participation | Soft Systems Methodology, Human-Centered Design |
| Iterative design cycles | Design research |
| Anticipation | Futures studies, anticipatory systems |

Within this framework, AI functions as an enabling overlay (Figure 1). It is used to support activities such as requirements auditing, traceability maintenance, stakeholder data synthesis, and scenario exploration, while human judgment and ethical responsibility remain central to all decisions. By positioning AI as an infrastructural support rather than the focal skill, the framework enables students to engage with emerging AI-enabled systems while preserving the disciplinary foundations of systems engineering.

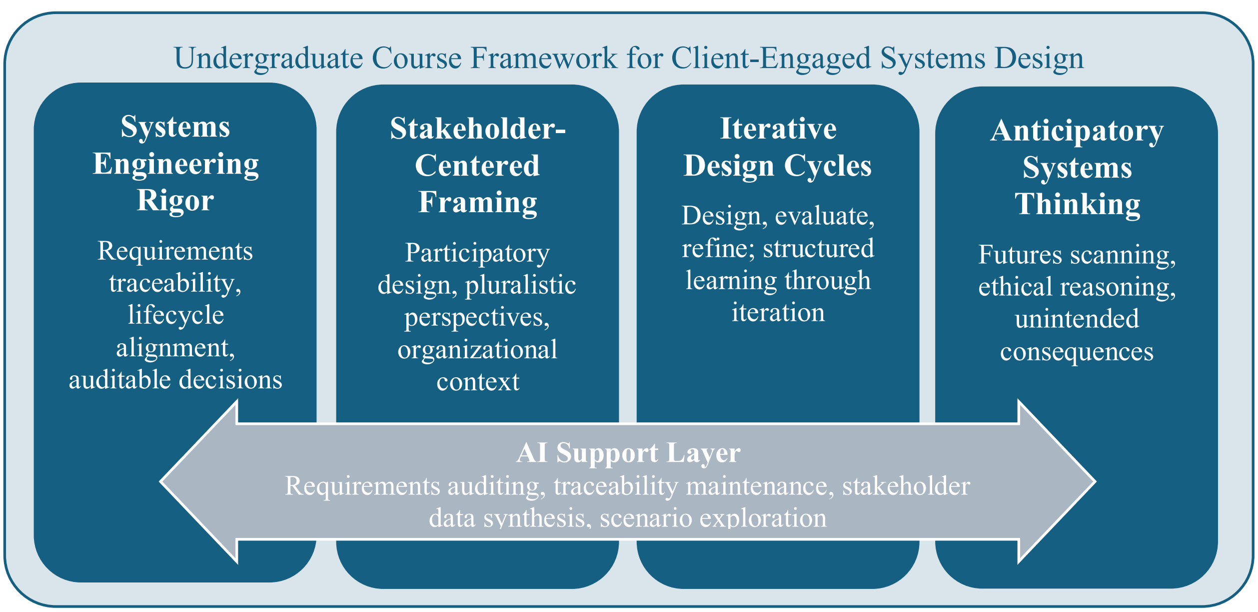


Figure 1: Hybrid course framework; four pedagogical strands with AI as a support layer.

Each strand plays a distinct pedagogical role in shaping how students approach system design; each is briefly described below.

*Systems Engineering Rigor* emphasizes core systems engineering rigor, including requirements traceability, auditable decision-making, and lifecycle alignment [1]. Students are required to produce and maintain requirements and traceability matrices that are link to stakeholder needs. This strand draws on established SE principles to ensure that design work remains transparent and reviewable, even as projects evolve through iteration. In addition, rather than oversimplifying systems engineering processes to fit classroom constraints, the course selectively adapts them to remain lightweight while retaining their essential functions.

*Stakeholder-Centered Framing* emphasizes participatory design practices to foreground stakeholder perspectives in system definition and decision-making. Building on ideas from Soft Systems Methodology [21], students engage diverse stakeholders to elicit multiple perspectives and articulate desired outcomes in ill-structured problem contexts. These activities inform engineering requirements, ensuring that technical artifacts are grounded in social and organizational realities. Hence, stakeholder engagement is not considered an informal precursor to the work, but as an integral component of system definition.

*Iterative Design Cycles* structures the course around iterative design cycles. While systems engineering already incorporates iteration through lifecycle feedback loops, reviews, and decision gates intended to refine requirements and manage risk as systems mature, the course foregrounds a complementary form of iteration drawn from design practice. Student teams engage in iterative cycles centered on lightweight prototyping, testing, and evidence-based learning. System concepts are implemented at a minimal viable level, evaluated against technical and stakeholder criteria, and revised based on observed outcomes. Rather than treating iteration as means for validation or correction, these cycles emphasize early exploration and failure as productive activities; structured reviews and decision gates anchor these design iterations within systems engineering.

*Anticipatory Systems Thinking* incorporates anticipatory systems thinking, encouraging students to consider future consequences, unintended effects, and ethical implications of AI-enabled systems. Students engage in structured foresight activities, such as scenario exploration and ethical canvases, to examine impacts related to safety, equity, labor, and long-term system behavior. Anticipation is operationalized through deliverables and artifacts rather than abstract discussion, ensuring that ethical reasoning is integrated into design decisions and documented alongside technical considerations.

## Course Structure and Implementation

The course is offered as an upper-division undergraduate special topics course open to engineering students who have completed prior coursework in programming, data analysis, or systems-oriented design. Enrollment is structured for approximately 15-20 students, organized into teams of three to four to support sustained project-based work. The course follows a studio format with weekly extended sessions that integrate brief lectures, hands-on design activities, and structured reviews. Each week centers on active artifact development within a single system context that spans the semester.

Design reviews serve as formal critique sessions in which teams defend their requirement interpretations, traceability structures, and prototype decisions using evidence from stakeholder input and testing. These reviews mirror professional practice and create repeated opportunities for students to refine their design rationale. In early assignments, AI-generated requirements may not be copied directly into final deliverables; any adopted content must be rewritten, contextualized, and justified by the team.

Industry engagement is established through partnerships with small and mid-sized enterprises recruited via regional networks and campus-industry liaisons. Each partner proposes a focused problem suitable for an undergraduate team and participates in scheduled check-ins to provide domain insight and review deliverables. Project scopes are deliberately limited to ensure feasibility within a single semester and to reflect the resource constraints common in SME environments. Partner onboarding includes expectation alignment, data governance agreements, and clarification of confidentiality and publication norms.

Within this structure, real-world systems engineering projects are defined as client-driven challenges grounded in ongoing organizational needs. These projects require students to navigate stakeholder ambiguity, system boundaries, and lifecycle considerations while operating within practical constraints. Examples include developing a decision-support workflow, organizing requirements for a data-informed internal tool, strengthening traceability for a compliance-sensitive process, or prototyping an AI-assisted analytic pipeline. The emphasis is on systems definition, requirement quality, traceability integrity, stakeholder alignment, and risk reasoning, while full-scale deployment remains outside the scope of the course.

Each team produces a structured documentation and prototype package that includes stakeholder analysis, a prioritized and quality-assessed requirements set, an updated traceability matrix, documented design decisions with justification, an anticipatory foresight analysis, and a lightweight prototype demonstrating feasibility. Production-level robustness, regulatory certification, and operational deployment fall outside the scope of the course.

*Learning Objectives*

The course learning objectives emphasize systems thinking, ethical reasoning, and professional practice. By the end of the course, students are expected to:

- Apply systems engineering principles to define, analyze, and document complex systems, including requirements traceability and auditable design decisions.
- Frame ill-structured design problems by negotiating stakeholder needs, defining system boundaries, and translating ambiguity into well-formed requirements.
- Evaluate and justify design tradeoffs across technical, organizational, human, and ethical constraints using documented rationale.
- Demonstrate responsible AI literacy by critically assessing when and how AI tools support system design while maintaining human judgment and accountability.

- Apply anticipatory reasoning to examine future impacts, risks, and unintended consequences of AI-enabled systems.

These objectives are reinforced through repeated interaction with SME clients, iterative artifact development, and explicit reflection on both technical and socio-technical dimensions of system design.

*Core Assignments and Learning Artifacts*

Learning is organized around four core assignments that mirror professional systems engineering activities and produce cumulative artifacts. Each assignment builds on prior work, reinforcing continuity across the semester. All major deliverables are submitted in two iterations (draft and revised), with revisions guided by stakeholder feedback and structured design reviews. Teams must document what changed, why it changed, and how evidence informed the revision.

Assignment 1: Requirements Audit
Students begin with a structured requirements audit of the client system. Using stakeholder interviews and existing documentation, teams draft an initial set of requirements independently. AI-assisted analysis is introduced only after this first draft to surface ambiguity, inconsistency, and gaps. Students compare AI outputs against their own requirements and reconcile discrepancies in a structured rationale memo. Acceptance, modification, or rejection of AI-suggested revisions must be justified using stakeholder evidence and established requirement quality criteria. This assignment develops judgment in defining well-formed requirements and evaluating AI-generated content.

Assignment 2: Traceability Automation
Teams next develop and maintain a traceability structure linking stakeholder needs, system requirements, design decisions, and evaluation criteria. An initial traceability matrix is established early and updated as requirements evolve. AI tools may support consistency checking and identification of missing or broken links; however, all AI-flagged issues require human review and documented justification. This assignment reinforces lifecycle thinking and positions traceability as a living representation of design reasoning.

Assignment 3: Anticipatory Foresight
Each team produces a foresight memo examining potential long-term impacts and unintended consequences of the proposed system. Students analyze issues such as equity, safety, organizational change, and future system evolution. Ethical and anticipatory reasoning are documented within formal artifacts, linking future-oriented considerations directly to design decisions.

Assignment 4: Client-Facing Prototype and Documentation
The course culminates in a lightweight prototype accompanied by a structured documentation package for the SME partner. Prototypes demonstrate feasibility and may take the form of a functional proof-of-concept, data workflow, or analytic tool. The documentation package includes:

- A system overview and key assumptions
- Stakeholder needs and prioritized requirements
- Updated traceability artifacts
- Documented design decisions with justification
- A brief deployment guide outlining dependencies and risks

This final assignment integrates systems rigor, stakeholder alignment, and design justification into a coherent deliverable suitable for a resource-constrained organization.

*Implementation Considerations*

To support replicability, the course relies on commonly available tools and platforms rather than specialized infrastructure. AI support is provided through widely accessible generative AI and data analysis tools, while documentation and collaboration are managed using shared workspaces and template-based artifacts. Data governance, privacy, and security expectations are established early through partner agreements and university policies and IRB requirements and will be revisited throughout the course to support responsible practice.

Overall, the course structure emphasizes disciplined yet lightweight systems engineering practice grounded in authentic client engagement and reinforced through iterative review and feedback. A small set of tightly aligned assignments and shared artifacts anchors the course, supporting both systems rigor and responsible AI use without imposing excessive overhead. This structure is intended to support transferability to other institutions seeking to integrate AI-enabled systems engineering into undergraduate curricula; instructional materials and assignment templates are available from the authors upon request. As a proposed course model, implementation would involve limited enrollment and a small number of SME partners to preserve instructional quality and feasibility; variation in partner readiness and data availability should be anticipated.

Table 2: Mapping of learning objectives to core assignments, assessment methods, and AI-supported activities

| Learning Objective | Core Assignment(s) | Evaluation Approach | Role of AI |
|---|---|---|---|
| Apply systems engineering rigor through requirements traceability and auditable design decisions | AI-assisted requirements audit; traceability automation sprint | Rubric-based analysis of requirements quality, traceability completeness, and decision documentation; instructor review during design reviews | Supports detection of ambiguous or inconsistent requirements; assists with maintaining and checking traceability links |
| Develop systems thinking across the lifecycle and evaluate design alternatives | Traceability automation sprint; client-facing prototype and documentation package | Artifact review assessing lifecycle coverage, alignment between requirements and design choices, and coherence of system rationale | Helps surface missing lifecycle links, supports consistency checking, and visualizes dependencies |
| Demonstrate responsible AI literacy and professional judgment | AI-assisted requirements audit; anticipatory foresight memo | Review of design rationales documenting AI use, limitations, and | Provides candidate analyses and summaries that students must justify, revise, or reject |

| | | | |
|---|---|---|---|
| | | uncertainty; reflection analysis | |
| Engage stakeholders effectively and translate perspectives into system requirements | AI-assisted requirements audit; client-facing prototype and documentation package | SME partner feedback on clarity, usefulness, and alignment with organizational context; instructor assessment of stakeholder artifacts | Assists with synthesis of stakeholder inputs while requiring human interpretation and validation |
| Apply anticipatory reasoning and ethical judgment to AI-enabled systems | Anticipatory foresight memo | Rubric-based evaluation of foresight depth, consideration of unintended consequences, and ethical reasoning; qualitative analysis of reflections | Supports scenario exploration and impact identification; does not make ethical judgments |
| Communicate system designs clearly to non-expert clients | Client-facing prototype and documentation package | SME partner feedback; instructor assessment of documentation clarity and completeness | Helps generate draft explanations and summaries that students refine and contextualize |
| Develop professional identity and reflective practice in systems engineering | Reflections and design rationale memos (across assignments) | Qualitative analysis of reflections focusing on learning moments, tradeoffs, and accountability | Prompts reflection by exposing gaps, assumptions, and uncertainty in student work |

## Educational Evaluation Approach

In this study, success is defined along four dimensions: improvement in the rigor and quality of systems engineering artifacts; demonstrated responsible and reflective use of AI tools; evidence of iterative design reasoning and documented tradeoffs; and validation from SME partners regarding the usefulness and feasibility of student deliverables. The evaluation framework is intended to guide future implementation and assessment of the course and translates these dimensions into structured evidence.

Evaluation is organized around four complementary lenses aligned with the course learning objectives.

The first lens examines the rigor of students' systems engineering processes. Evidence includes requirements documents, traceability matrices, and written design rationales. Requirements are evaluated against established quality standards, with attention to whether they are clear, testable, and logically consistent. Traceability is assessed by examining how effectively stakeholder needs connect to system requirements and subsequent design decisions. Design rationales are reviewed to determine whether assumptions are made explicit, tradeoffs are justified, and revisions are supported by evidence across iterations. The evaluation also separates AI-suggested material from student-authored reasoning, with rubrics emphasizing justification, evidence, and defensible decision-making.

The second lens examines student learning and professional development. It considers growth in systems thinking, responsible engagement with AI tools, and ethical judgment. Pre- and post-course surveys measure changes in students' confidence and perceived competence when working with AI-enabled systems within structured engineering processes. Reflective memos and design rationales are analyzed to understand how students respond to uncertainty, interpret AI-assisted outputs, and integrate ethical and future-oriented considerations into their design decisions.

The third lens incorporates structured feedback from SME partners to assess external relevance. Partners provide input during midterm and final reviews using short surveys and guided prompts focused on clarity, feasibility, and organizational usefulness. This perspective offers independent validation of whether student work aligns with professional expectations in resource-constrained settings.

The fourth lens examines iteration quality and design reasoning. Version histories of requirements and prototypes, along with review artifacts, are analyzed to identify evidence of problem reframing, alternative exploration, and justified revisions. Rubrics assess whether stakeholder input and testing evidence inform changes to system boundaries, requirements, and design decisions over time.

Responsible AI use is assessed across all lenses. Students are required to document AI limitations, uncertainty, and assumptions within their artifacts. AI-generated outputs must be examined critically and either adopted with justification or revised in light of stakeholder evidence and domain knowledge. Teams are required to document these decisions in their traceability artifacts and design rationale memos. Model and data cards, along with anticipatory foresight memos, provide additional documentation of transparency and appropriate use.

This evaluation approach provides a structured yet lightweight means of assessing whether the course achieves its educational goals. By combining artifact-based assessment, reflective evidence, and stakeholder feedback, the approach aligns with expectations for rigor while remaining appropriate for exploratory, practice-oriented undergraduate education.

**Discussion**

From a DEED perspective, the course offers a model for teaching design within AI-assisted workflows by embedding AI into the representations and critique practices through which students learn to reason about systems. AI is integrated into requirements development, traceability, and design rationale in ways that prompt reflection and expose ambiguity. In doing so, the course strengthens students' ability to justify design decisions under sociotechnical constraints.

The framework positions systems engineering as an inherently socio-technical practice. By incorporating AI-assisted analysis directly into core artifacts such as requirements documents and traceability matrices, the course makes reasoning processes visible. Students must interpret AI-generated outputs, assess their validity, and document their decisions. This approach aligns with prior arguments that AI education is most effective when integrated into disciplinary workflows that emphasize professional judgment and accountability [11, 13].

A key design decision is structuring AI use as a critique mechanism rather than a primary authoring tool. Students draft requirements independently before introducing AI assistance, and they must reconcile discrepancies between their own analysis and AI suggestions. This sequencing reduces passive reliance on automated outputs and reinforces independent reasoning. AI becomes a prompt for scrutiny and revision, not a substitute for decision-making.

The model also advances role-based AI literacy. Students encounter AI as analysts, evaluators, and decision-makers who must determine when its outputs are useful and when they require revision. Ethical judgment is embedded in routine engineering decisions, ensuring that responsibility remains with the human designer.

The SME context further sharpens these learning experiences. In smaller organizations, documentation and traceability often function as practical coordination tools rather than formal compliance artifacts. Students operate within realistic constraints related to time, data availability, and organizational capacity. “Real-world” in this setting refers to authentic stakeholder engagement, evolving requirements, and practical limitations, not to full commercial deployment. These conditions require teams to negotiate scope, prioritize effectively, and justify tradeoffs in ways that mirror professional practice.

The contribution of this work lies in restructuring systems engineering pedagogy around iterative, AI-augmented design reasoning within authentic client contexts. It extends existing scholarship on AI integration and systems thinking by demonstrating how those insights can be operationalized within a single undergraduate course model. Table 3 summarizes the distinctions between traditional undergraduate systems engineering courses and the proposed AI-supported, client-based model.

Table 3: Key Distinctions Between Traditional Systems Engineering Courses and the Proposed Course Model

| Dimension | Traditional SE Course | Proposed Course |
|---|---|---|
| Requirements | Document assignment | Iterative design artifact with AI critique |
| AI | Add-on module or tool demo | Embedded within core SE artifacts |
| Projects | Hypothetical cases | Live SME client engagement |
| Pedagogy | Process-focused, linear | Studio-based, iterative, critique-driven |

## Conclusion

This paper presents a course design framework for integrating artificial intelligence into undergraduate systems engineering education while maintaining rigor, accountability, and human judgment. The proposed model integrates systems engineering discipline with stakeholder engagement, iterative design practice, and anticipatory reasoning to address a gap in current approaches, which often present AI either as an isolated technical topic or as a productivity aid detached from systems thinking. The contribution is a coherent course architecture, grounded in existing scholarship, that institutions can adapt to their own contexts.

At its core, this work reframes AI's role in systems engineering education. The course positions AI as a tool that supports and extends design reasoning, while keeping responsibility for interpretation and decision-making with students. AI is used to surface assumptions, expose tradeoffs, and assist with traceability, but students must evaluate outputs, justify their choices, and document their rationale.

This work suggests that AI integration in systems engineering education is most effective when it is embedded within core design artifacts and decision processes, instead of treated as a separate technical module. The framework encourages educators to move beyond narrow conceptions of AI proficiency and toward learning experiences that connect technical rigor with sociotechnical awareness and professional responsibility. Future efforts will focus on piloting the course, refining its assignments and assessment instruments, and testing the model across different institutional settings. The long-term goal is to contribute a reusable structure for preparing undergraduate engineers to design and govern AI-enabled systems in practice.

**References**


[1] INCOSE, *INCOSE systems engineering handbook*. John Wiley & Sons, 2023.

[2] C. S. Wasson, 'System engineering competency: the missing course in engineering education', in *2012 ASEE Annual Conference & Exposition*, 2012, pp. 25–1227.

[3] A. F. Squires *et al.*, 'Investigating an innovative approach for developing systems engineering curriculum: The Systems Engineering Experience Accelerator', in *2011 ASEE Annual Conference & Exposition*, 2011, pp. 22–970.

[4] F. Robinson-Bryant, 'Developing a systems thinking integration approach for robust learning in undergraduate engineering courses', in *2018 ASEE Annual Conference & Exposition*, 2018.

[5] S. Amershi *et al.*, 'Software engineering for machine learning: A case study', in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 2019, pp. 291–300.

[6] National Institute of Standards and Technology, '2023 Annual Report of the Visiting Committee on Advanced Technology', U.S. Department of Commerce, Washington, DC, Apr. 2024.

[7] Z. D. Asher, N. L. Ramo, and T. H. Bradley, 'The use of systems engineering principles to improve learning outcomes in a multidisciplinary course', in *2018 ASEE Annual Conference & Exposition*, 2018.

[8] D. A. Schön, 'Educating the reflective practitioner', 1987.

[9] C. L. Dym, A. M. Agogino, O. Eris, D. D. Frey, and L. J. Leifer, 'Engineering design thinking, teaching, and learning', *Journal of engineering education*, vol. 94, no. 1, pp. 103–120, 2005.

[10] J. Schleiss, A. Johri, and S. Stober, 'Integrating ai education in disciplinary engineering fields: Towards a system and change perspective', *arXiv preprint arXiv:2410. 12795*, 2024.

[11] J. Schleiss and A. Johri, 'A roles-based competency framework for integrating artificial intelligence (ai) in engineering courses', *arXiv preprint arXiv:2410. 12796*, 2024.

[12] V. V. Poulsen, M. Guertler, B. Eisenbart, and N. Sick, ‘Advancing systems engineering with artificial intelligence: a review on the future potential, challenges and pathways’, *Proceedings of the Design Society*, vol. 5, pp. 359–368, 2025.
[13] O. Levy, I. Dikman, N. Levy, and M. Winokur, ‘Work in Progress: AI-Powered Engineering-Bridging Theory and Practice’, in *2025 IEEE Engineering Education World Conference (EDUNINE)*, 2025, pp. 1–4.
[14] R. Poli, ‘Anticipation: what about turning the human and social sciences upside down?’, *Futures*, vol. 64, pp. 15–18, 2014.
[15] M. Nadin, ‘Anticipation’, *International journal of general systems*, vol. 39, no. 1, pp. 35–133, 2010.
[16] R. Rosen, ‘Anticipatory systems’, in *Anticipatory systems: Philosophical, mathematical, and methodological foundations*, Springer, 2011, pp. 313–370.
[17] D. H. Rhodes and A. M. Ross, ‘Anticipatory capacity: Leveraging model-based approaches to design systems for dynamic futures’, in *2009 International Conference on Model-Based Systems Engineering*, 2009, pp. 46–51.
[18] M. Heidaripour, ‘Integrating Design Futuring into Engineering Education’, in *2025 ASEE Annual Conference & Exposition*, 2025.
[19] R. Von Schomberg, ‘A vision of responsible research and innovation’, *Responsible innovation: Managing the responsible emergence of science and innovation in society*, pp. 51–74, 2013.
[20] M. Brundage *et al.*, ‘Toward trustworthy AI development: mechanisms for supporting verifiable claims’, *arXiv preprint arXiv:2004. 07213*, 2020.
[21] P. B. Checkland, ‘Soft systems methodology’, *Human systems management*, vol. 8, no. 4, pp. 273–289, 1989.