2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION
Paper ID #52834

AI as an Intern: A Curriculum Framework for Supervising Vibe Coding in Data Science Education

Dr. Irene Tsapara, National University

Dr. Irene Tsapara is an academic leader, researcher, and educator specializing in Data Science, Artificial Intelligence, and computational learning theory. She serves as Academic Program Director for the Doctorate in Data Science at National University, where she oversees curriculum design, research supervision, and program accreditation. Dr. Tsapara holds a Ph.D. in Mathematics with specialization in Computational Learning Theory, Machine Learning, and Artificial Intelligence from the University of Illinois. Her current research focuses on the integration of AI-assisted learning frameworks, including vibe coding, into Data Science and interdisciplinary education. She develops scalable curricular models that align AI literacy, statistical reasoning, and ethical governance with foundational mathematical rigor. Her work emphasizes the use of large language models (LLMs) in promoting conceptual understanding, transparency, and reproducibility in computational practice.

Dr. Tsapara has over two decades of teaching and programming experience and has contributed to the development of graduate and doctoral curricula across data science, machine learning, and applied analytics. She also collaborates with the National AI Research Institute (TILOS) and several academic-industry partnerships focusing on responsible AI, data governance, and interdisciplinary education. Her professional mission is to reimagine Data Science education as a connective discipline, one that unites computational precision, mathematical depth, and human-centered AI ethics to prepare graduates for leadership in an intelligent, data-driven world.

Danae Nikki Vassiliadis, University of Chicago

Danae Vassiliadis is a data science practitioner and educator whose work focuses on applied machine learning, data systems, and AI-driven decision-making, with an emphasis on human behavior and interaction with intelligent systems. She serves as a Teaching Assistant in the Master of Science in Applied Data Science program at the University of Chicago, where she supports graduate-level instruction in big data systems, cloud computing, and machine learning.

She is also a Data Science Consultant at Accenture, where she leads generative AI learning and upskilling initiatives and contributes to the development of proprietary GenAI tooling for enterprise applications. Her work examines how individuals and organizations interact with AI systems, including the design of frameworks for the evaluation and deployment of large language model–based solutions, with attention to user behavior, trust, and decision-making.

Danae holds a Master of Science in Applied Data Science from the University of Chicago and dual bachelor's degrees in Industrial Engineering and Psychology from Northwestern University. Her interdisciplinary training informs her approach to the design of AI systems that account for behavioral variability.

Her current research focuses on AI-mediated programming and education, including human–AI collaboration and the integration of large language models into data science training. She is particularly interested in the effects of generative AI on learning behavior, authorship, and problem-solving, as well as in the development of evaluation frameworks that support rigor and transparency. More broadly, her work examines how data science education and practice can incorporate AI while maintaining methodological and analytical foundations.

Andrew D'Amico, Northwestern University

Andrew D'Amico is an entrepreneur, AI systems architect, and applied data science practitioner whose work lies at the intersection of artificial intelligence, systems design, analytics, and engineering education. He is a consultant with Northwestern Analytics Partners, where he contributes to AI and analytics initiatives spanning intelligent decision support, quantitative modeling, and agent-based systems.

He holds a Master of Science in Data Science in Artificial Intelligence from Northwestern University and undergraduate degrees in Philosophy and English Literature from Loyola University Chicago, with

a concentration in Philosophy of Science. He serves as a teaching assistant in Northwestern's Data Science program, supporting instruction in Business Process Analytics, Decision Analytics and Operations Research, Data Engineering, and Quantitative Finance in Data Science. He is a certified Project Management Professional (PMP) and PMI Disciplined Agile Senior Scrum Master (DASSM), with delivery experience across Scrum, Waterfall-Predictive, and Disciplined Agile methodologies.

His current research focuses on two convergent areas. The first is AI-mediated software development, including vibe coding and collaborative programming frameworks that are reshaping how programming is learned, guided, and evaluated. The second is assessment and pedagogy in the age of AI, with particular attention to authorship, process-based evaluation, educational validity, and the preservation of rigorous learning outcomes in environments shaped by generative systems. More broadly, his work examines how institutions can integrate AI in ways that are methodologically sound, educationally credible, and responsive to the changing structure of technical and intellectual labor.

Drawing on both technical and humanistic training, he brings a multidisciplinary perspective to the governance and institutional adoption of AI in engineering and data science education.

AI as an Intern: A Curriculum Framework for Supervising Vibe Coding in Data Science Education

Irene Tsapara[1], Andrew D'Amico[2], Danae Vassiliadis[3]

[1]National University, San Diego, USA
[2]Northwestern University, Evanston, USA
[3]University of Chicago, USA

Abstract

The emergence of what is increasingly referred to as 'vibe coding,' formalized here as an AI-assisted development framework grounded in natural language prompting, iterative reasoning, and contextual evaluation, may represent an important shift in Data Science (DS) education. Instead of emphasizing syntax mastery and procedural execution, vibe coding reframes computational work as an interactive modeling process between human reasoning and intelligent systems. This paradigm can help rebalance the implementation of algorithms, analytical modeling, and theoretical understanding in DS instruction. This article develops a curriculum framework that integrates vibe coding into Data Science programs to support conceptual learning, model interpretability, and quantitative rigor. Automating routine programming tasks may allow faculty to reallocate instructional time toward the mathematical and statistical foundations of the discipline, including probability theory, linear algebra, optimization, and statistical inference. These areas form the computational core of Data Science and underpin reproducibility, verification, and uncertainty quantification. The curriculum structure is organized around three technical components. Conceptual alignment ensures that AI-assisted competencies are mapped to DS learning outcomes in data literacy, algorithmic reasoning, and model transparency. Pedagogical implementation embeds AI-supported programming exercises into courses such as Applied Data Modeling and AI Collaboration in Data Science, emphasizing prompt precision, data validation, and computational reproducibility. Ethical and analytical assurance introduces evaluation protocols for AI-generated code using statistical benchmarking, bias detection, and verification metrics. Integrating vibe coding into DS curricula can enhance accessibility, reduce redundant technical overhead, and strengthen students' ability to connect computational processes with statistical reasoning. This approach offers an instructional model that aligns automation with analytical depth, helping scientists prepare data to design, interpret, and validate AI-driven systems with both technical and mathematical precision. The approach may also be adapted, with discipline-specific modifications, to Computer Science curricula

Keywords: Generative Artificial Intelligence; STEM education; Project Governance; Agile methodology; CRISP-DM; Design Thinking; AI-assisted programming; human-AI collaboration; ethical AI supervision; AI pedagogy; PMBOK-aligned governance.

1. Introduction

Generative AI (GenAI) now drafts text, explanations, and code at a level that invites students to treat it as a collaborator rather than a calculator. The pedagogical challenge is to teach supervisory literacy: treating GenAI as an intern-level contributor, capable but fallible, requiring delegation, validation, and corrective oversight [6, 12, 47]. This perspective aligns with calls to move beyond prohibition toward guided, transparent use in higher education [16, 11]. Despite widespread adoption of generative AI tools [9], current Data Science pedagogy rarely specifies supervisory governance structures that make student reasoning auditable and assessable. This

paper addresses that gap by proposing a governance-centered instructional model that aligns classroom practice with emerging professional expectations for accountable AI supervision.

We distinguish between governance frameworks (PMBOK predictive and hybrid project management), iterative delivery frameworks (Scrum-based Agile), problem-framing methodologies (Design Thinking), and analytics lifecycles (CRISP-DM). Each can be treated as addressing a different, and sometimes overlapping, control layer in AI-supervised project work and can help scaffold iterative prompting, review, and validation when students oversee AI. Building on constructivist and experiential learning [8, 13], metacognition [14, 15], and cognitive load theory [7], we propose a hybrid PMBOK-Agile model that embeds auditability and ethical governance while preserving adaptability to evolving AI outputs. We then map the model to curriculum levels and assessment practices that mirror professional supervision of AI in modern workplaces [2, 17].

This supervisory model aligns with what is increasingly referred to as vibe coding: a human-AI collaborative programming paradigm grounded in natural-language prompting, iterative reasoning, and contextual evaluation. In this article, we use the term to refer to the supervised use of generative AI to draft and iteratively refine code, while the student remains responsible for problem framing, validation, and interpretation. We make three contributions: (1) we define vibe coding as a supervised human-AI programming paradigm, (2) we propose an AI-assisted framework using a hybrid PMBOK-Agile governance model for teaching it, and (3) we illustrate feasibility through preliminary classroom implementation and anecdotal instructional evidence.

2. Background and Theory

2.1. Constructivist and Experiential Foundations

Constructivist theories position learners as active constructors of meaning, forming knowledge through interaction and social mediation. Within this paradigm, GenAI can be understood as a cognitive partner that extends Vygotsky's "zone of proximal development" [8], supporting students in tackling problems slightly beyond their current competence. Kolb's experiential learning cycle (experience, reflection, conceptualization, and experimentation) maps neatly onto iterative student-AI collaboration [13].

2.2. Metacognition and Cognitive Load

Metacognitive awareness, the capacity to plan, monitor, and evaluate one's own thinking, is foundational to effective supervision of generative AI systems [14, 15]. In the AI-as-Intern model, students are required to deliberate on which tasks to delegate to GenAI, how to frame those tasks, and how to interpret and verify the resulting outputs. This process shifts student engagement from passive consumption toward active oversight, requiring continuous monitoring of relevance, accuracy, and alignment with disciplinary standards. Rather than assessing whether students can produce an answer, the instructional emphasis shifts to evaluating how students reason through AI-generated content, identify errors or gaps, and justify revisions. Cognitive load considerations further motivate the use of structured scaffolding and staged responsibility, ensuring that learners are not overwhelmed by simultaneous demands of technical execution, conceptual reasoning, and ethical judgment [7, 46]. In Data Science contexts, this form of metacognitive supervision is especially critical for interpreting models, validating assumptions, detecting spurious patterns, and reasoning under uncertainty, all of which require human judgment that cannot be delegated to automated systems.

2.3. Metacognition, Assessment, and the Shift from Production to Reasoning

The widespread availability of generative AI fundamentally alters the role of assessment in Data Science education [12, 25]. When students can readily generate code, explanations, or analytical summaries using AI tools, assessment strategies that prioritize procedural execution or syntactic correctness lose their discriminatory power [28]. Under these conditions, learning outcomes must shift from measuring production to evaluating reasoning, judgment, and critical engagement with computational outputs [24, 15]. In AI-augmented learning environments, the central pedagogical question is no longer whether a student can produce a model or script, but whether they can explain why a particular approach is appropriate, how assumptions influence results, and where AI-generated outputs may be incomplete, biased, or incorrect [29]. This reframing positions critical thinking and metacognitive awareness as primary assessment targets rather than ancillary skills [14, 48]. Students must demonstrate the ability to interrogate AI responses, identify logical or statistical flaws, and articulate corrective actions grounded in disciplinary knowledge [24].

From a Data Science perspective, this shift is especially consequential. Model outputs are probabilistic rather than definitive, and errors often arise not from coding mistakes but from violated assumptions, data leakage, multicollinearity, or misinterpretation of uncertainty [23].

AI-generated code may execute correctly while still producing misleading or invalid conclusions [27]. Metacognitive supervision, therefore, becomes a necessary condition for responsible practice, requiring students to reflect on modeling choices, validate results against theoretical expectations, and communicate limitations transparently [26]. Cognitive load theory further supports this instructional redesign. By allowing AI systems to handle routine implementation details, educators can reduce extraneous cognitive load and re-allocate instructional effort toward germane load associated with reasoning, interpretation, and decision-making [7]. Structured scaffolds-such as validation checklists, reflection prompts, and audit trails-enable students to gradually assume greater supervisory responsibility without sacrificing conceptual depth [15]. Assessment artifacts thus evolve from final products to documented reasoning processes, aligning evaluation with the competencies required for trustworthy human-AI collaboration [28].

2.4. From Prohibition to Transparent Integration

Earlier instructional responses to GenAI were dominated by restrictions or prohibition, driven by concerns about academic integrity, authorship, and assessment validity [16]. However, such bans are increasingly unsustainable as GenAI tools integrate into professional workflows [12]. Best practices have therefore shifted toward structured transparency, emphasizing disclosure, documentation of human oversight (e.g., auditability), and explicit accountability for the final work product [11]. Graduates are now expected not merely to use AI-enabled tools but to exercise judgment in supervising, validating, and interpreting their outputs. For that reason, current best practice emphasizes transparency, disclosure, and auditability: students should document when AI was used, what human oversight was applied, and who remains accountable for final decisions [11].

2.5. Literature Synthesis and Gap

Across these strands, a central instructional requirement is to make human supervision visible and assessable when students work with AI tools. We operationalize that requirement by treating governance artifacts (plans, risk registers, validation evidence, and revision logs) as both process controls and learning artifacts. While existing scholarship has discussed GenAI adoption,

academic integrity, and human-AI collaboration in higher education, less work has specified a governance-centered instructional model for Data Science that makes student supervision visible, assessable, and reproducible. The central gap addressed in this paper is therefore not whether AI can be used in instruction, but how its use can be structured so that reasoning, validation, and accountability remain pedagogically central.

2.6. Definition of Vibe Coding

In the remainder of this paper, vibe coding refers to supervised human-AI programming, often defined as AI-assisted programming grounded in natural-language prompting: students may use generative AI to draft and iteratively refine code, but they remain accountable for problem framing, verification, and interpretation. The pedagogical value of this approach lies not in replacing disciplinary reasoning, but in redistributing effort away from syntax-only work and toward conceptual understanding, model interpretation, and analytical judgment.

AI-assisted coding offers a structural response to this imbalance by leveraging AI to automate routine programming tasks, including boilerplate code generation and syntax troubleshooting [12]. By reducing the cognitive and instructional overhead associated with low-level implementation details, instructional time can be reallocated toward analytical reasoning, model interpretation, and validation [7]. This shift enables students to engage more deeply with the mathematical and statistical foundations that govern data-driven decision making [23]. In this way, vibe coding functions not as a shortcut, but as a pedagogical mechanism for restoring theoretical depth within applied Data Science education.

3. Frameworks for Teaching AI Supervision

3.1. PMBOK-Aligned Predictive Governance and Project Control

The Project Management Institute's *PMBOK Guide* describes a *predictive project management* approach organized around five process groups- Initiating, Planning, Executing, Monitoring and Controlling, and Closing- that provide explicit governance checkpoints for ethical supervision and documentation [10]; these groups are useful in education because they provide explicit governance checkpoints that make expectations, evidence, and accountability visible, which is essential when students are working with generative AI. In our educational adaptation, *Initiating* involves students defining learning goals, clarifying the scope of permitted AI use, setting ethical boundaries, and specifying success criteria. *Planning* entails translating that framing into a project charter, identifying validation sources, anticipating risks, and establishing a data strategy. *Executing* is the phase in which they do the work, while practicing disciplined delegation to AI, meaning tasks are bounded, prompts and outputs are documented, and decisions are recorded. *Monitoring and Controlling* is the repeated cycle of validation, peer review, bias detection, and corrective action. Closing is the stage at which students formalize their results with a reproducible package and a reflection memo that demonstrates both technical learning and responsible judgment. Predictive governance is most effective when requirements are sufficiently knowable early. In innovation and software development contexts, PMI frameworks such as the PMBOK Guide's predictive framework and the Disciplined Agile framework, as well as agile methodologies in general, recognize rolling-wave planning, in which near-term work is elaborated in detail while future requirements are progressively refined. The strength of a predictive lifecycle lies in its emphasis on governance, traceability, and evidence, making it well-suited to accreditation and quality assurance; these phases function as governance gates and checkpoints aligned with model development, diagnostics, and refinement [31, 32, 33].

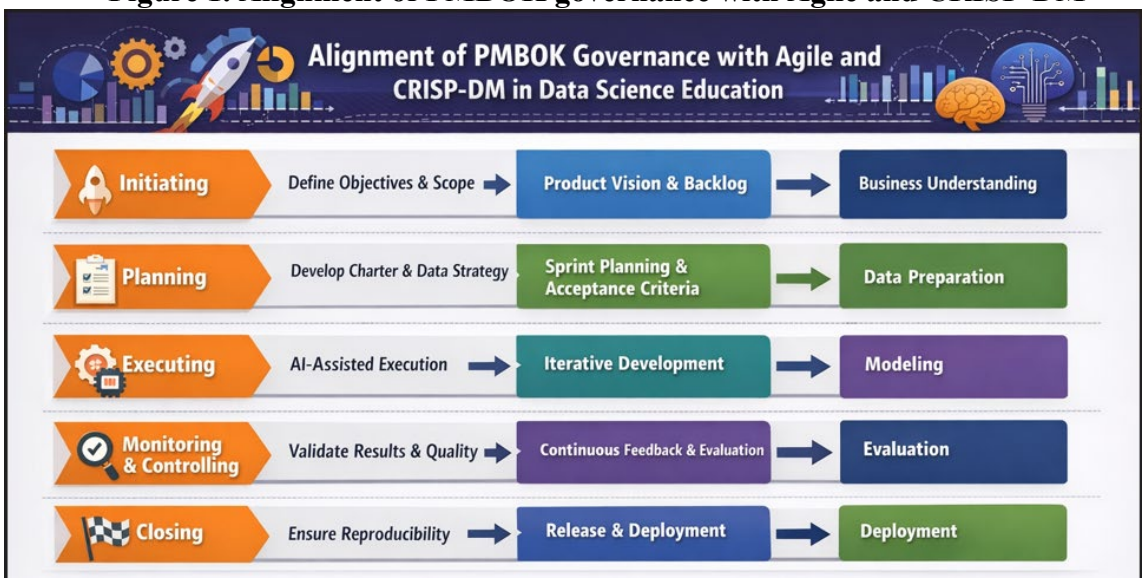
PMBOK-aligned predictive governance is useful in education because it makes planning, scope control, risk management, and evidence of decision-making visible. Predictive governance should not be treated as synonymous with waterfall. PMI distinguishes predictive, hybrid, and adaptive ways of working, and even in predictive settings, rolling-wave planning allows near-term work to be elaborated while later work remains subject to revision as uncertainty is reduced [42].

This distinction matters in Data Science because model development is inherently cyclical. Evaluation frequently sends students back to problem framing, data preparation, or assumptions, so the governance function is not to impose linearity but to make iteration auditable. In this mapping, Monitoring and Controlling becomes the governance loop that records what failed, why it failed, and what was revised, aligning naturally with CRISP-DM's repeated modeling and evaluation cycles [21, 36, 43]. The same logic underpins the AI-as-team-member metaphor. Generative AI can draft code, propose alternatives, and accelerate routine work, but it does not own correctness, evidence, or ethics. Students, therefore, remain accountable for validation, risk management, and transparent documentation. The NIST Generative AI Profile likewise emphasizes oversight, evaluation, and documentation for responsible use of generative systems [40]. Educational research reinforces this emphasis on scaffolded, human-in-the-loop adoption. Learning gains are strongest when AI use is structured through prompts, reflection routines, and validation checkpoints, and when assessment practices make authentic reasoning auditable rather than invisible [45, 44].

Table 1: Alignment of PMBOK predictive governance with Agile practices and CRISP-DM phases in Data Science education

PMBOK Predictive Governance	Primary Alignment	Purpose	Agile	Alignment	CRISP-DM
Initiating	Define objectives, scope, ethical boundaries, and success criteria	Product vision, problem framing, backlog initialization	Business Understanding		
Planning	Develop project charter, validation sources, risks, and data strategy	Sprint planning, task breakdown, acceptance criteria	Data Understanding, Data Preparation		
Executing	Perform planned work with documented AI delegation and controls	Iterative development, AI-assisted task execution	Modeling		
Monitoring and Controlling	Validate outputs, detect bias, manage quality, and apply corrective actions	Continuous feedback, retrospectives, and model evaluation	Evaluation		
	Formalize results, ensure reproducibility, and reflect on outcomes	Release review, documentation, knowledge transfer	Deployment		

Figure 1. Alignment of PMBOK governance with Agile and CRISP-DM



3.2. CRISP-DM: Data Workflow Rigor

The Cross-Industry Standard Process for Data Mining (CRISP-DM) provides a rigorous and well-established workflow that is particularly well suited to Data Science, analytics, and STEM-oriented curricula [20, 21]. Its six stages (Business Understanding, Data Understanding, Data Preparation/Preprocessing, Modeling, Evaluation, and Deployment) map directly onto contemporary AI-augmented analytics pipelines, offering students a structured mental model for navigating complex, data-driven problems. Within this framework, generative AI can be productively employed during the Modeling phase to support tasks such as code generation, feature engineering, and exploratory experimentation. However, CRISP-DM explicitly requires that human judgment dominate the Evaluation stage, where students must assess model performance, validate assumptions, and interpret results against domain-specific criteria.[36] These evaluative decisions are documented in validation ledgers that capture metrics, diagnostics, and reflective justification, reinforcing accountability and reproducibility. In contrast to less-structured exploratory or purely iterative workflows, CRISP-DM enforces explicit checkpoints that prevent the premature deployment of AI-generated results without empirical validation. By embedding AI use within a disciplined, phase-based process, CRISP-DM contributes procedural rigor to otherwise creative workflows, ensuring that AI-assisted analysis remains transparent, empirically justified, and grounded in sound analytical reasoning rather than opportunistic automation. In this paper, CRISP-DM functions as the analytics workflow layer, while the collaborative programming framework specifies how students supervise and document AI use within that workflow.

3.3. Agile: Iteration and Adaptability

Agile principles emphasize flexibility, feedback, and incremental progress [13]. In this paper, "Agile" primarily refers to Scrum-style iterative delivery (timeboxed sprints, product backlogs, sprint reviews, and retrospectives). We note that Agile is an umbrella term: all Scrum is Agile, but not all Agile is Scrum (e.g., Kanban). PMI's Disciplined Agile extends these approaches by providing a toolkit for selecting context-appropriate ways of working. [33, 35, 37]

Within an AI supervision context, each prompt-response cycle functions as a micro-iteration: define a user story (the analytic task), delegate to GenAI, inspect the output, and adapt the prompt based on results. Agile retrospectives parallel structured reflection logs, encouraging learners to document what worked, what failed, and why. This iterative rhythm lowers fear of failure and promotes experimentation. When paired with PMBOK predictive governance, Agile iterations operate within defined governance gates and checkpoints, ensuring that adaptability does not come at the expense of documentation, validation, or ethical oversight. Combined with PMBOK-aligned documentation rigor, Agile injects adaptability and learner agency, aligning with contemporary models of metacognitive self-regulation. [34, 37]

In software engineering terms, this trade-off parallels technical debt: accelerating delivery before the architecture and requirements are fully defined can lead to later rework [38, 39]. This risk is especially relevant in AI-assisted coursework, where repeated local refinements can improve immediate outputs while degrading alignment with project goals, methodological validity, or ethical constraints. In AI-assisted development, the analogous failure mode is agent drift, where local iterations optimize short-horizon outputs while degrading global coherence. Drift is not limited to code. If an AI model repeatedly redraws an image across many generations, small deviations compound, leading the final output to diverge materially from the original. Iterative AI workflows, therefore, require guardrails: explicit constraints and invariants anchoring each iteration to the intended baseline.

In Data Science courses, Agile micro-iterations support iterative model development, diagnostics, and refinement, enabling students to test assumptions, evaluate performance metrics, and revise analytical decisions across successive cycles while maintaining alignment with ethical and methodological constraints. These iterative activities are naturally related to the CRISP-DM Modeling and Evaluation phases, where repeated experimentation and validation are central to responsible analytical practice.

3.4. Design Thinking: Problem Framing and Creativity

Design Thinking complements PMBOK-aligned governance and Agile by focusing on empathy, ideation, prototyping, and testing [18, 19]. When students use GenAI to brainstorm or generate design alternatives, the framework ensures that output remains anchored to human context and stakeholder needs. Instructors can integrate empathy mapping and "what-if" prompts to encourage students to evaluate consequences rather than just efficiency. This approach stimulates creativity and ethical foresight but lacks formal validation and risk management unless coupled with PMBOK-aligned documentation. Embedding Design Thinking early in projects fosters ownership of problems before delegating to AI systems.

4. Comparative Analysis

The four frameworks vary in purpose, rhythm, and degree of structure. PMBOK-aligned predictive governance and methodology provide accountability and traceability; Agile provides adaptability but can drift without guardrails; Design Thinking ffosters creativity and stakeholder empathy; and CRISP-DM contributes analytical rigor. Rather than repeating a full comparison here, the detailed framework-by-framework contrast is developed in Section 5 and synthesized in Table 6. Taken together, these frameworks suggest a layered design rather than a single-method solution. PMBOK-aligned governance supplies structure and traceability, Agile supplies iteration and reflection, Design Thinking improves problem framing, and CRISP-DM secures analytical

soundness. In Data Science education, CRISP-DM organizes the modeling workflow, while the PMBOK-Agile hybrid governs how humans and AI collaborate within that workflow.

Table 2: Comparison of instructional frameworks for AI supervision

Framework	Strengths	Limitations
PMBOK-aligned governance	Governance, accountability, ethics	Overly procedural if static
Agile	Iteration, adaptability, reflection	Can lack audit documentation
Design Thinking	Creativity, problem framing, empathy	Weak on validation rigor
CRISP-DM	Rigor, workflow clarity, data focus	Limited metacognitive scaffolding

4.1. Process Overview

The hybrid merges PMBOK-aligned governance with Agile iteration to create a cyclical supervision loop suitable for AI-integrated coursework. Each cycle includes five stages:

- **Initiation:** Define project scope, intended AI use, and ethical boundaries.
- **Planning:** Draft a charter, outline validation protocols, and identify data or bias risks.
- **Execution:** Delegate tasks to GenAI, record prompts, and collect outputs.
- **Monitoring:** Validate results, note discrepancies, and adjust prompts iteratively.
- **Closing:** Compile an audit trail, reflection memo, and reproducibility package.

The hybrid model is intentionally asymmetric: predictive artifacts define the north-star outcome and architectural guardrails, while Scrum-style iterations operationalize delivery in bounded increments that fit the agent's context window and reduce drift. The result is controlled adaptability: requirements can be discovered through rolling-wave elaboration without losing traceability to intended outcomes.

4.2. Core Artifacts

To operationalize the hybrid, several deliverables anchor the process assessment. These artifacts are organized into three categories reflecting predictive governance, Agile iterative delivery, and hybrid AI supervision.

Table 3 introduces the core artifact set used to operate the hybrid workflow. These artifacts convert otherwise invisible reasoning into assessable evidence and support both accountability and analytical validation across the modeling lifecycle [10, 3].

Table 3: Core artifacts supporting Predictive governance, Agile iteration, and Hybrid AI supervision.

Artifact Category	Artifact	Description
Predictive (PMBOK-aligned) governance artifacts	Project Charter	Defines objectives, success metrics, stakeholder roles, and ethical scope.
	Scope Statement	Specifies in-scope and out-of-scope activities, assumptions, and constraints.

Artifact Category	Artifact	Description
	Requirements Register	Documents, functional, non-functional, and ethical requirements are subject to validation.
	Work Breakdown Structure (WBS)	Decomposes project deliverables into manageable components.
	Work Packages / WBS Dictionary	Provides detailed descriptions, responsibilities, and acceptance criteria for WBS elements.
	Quality Guidelines: Definition	Establishes quality standards, validation thresholds, and completion criteria.
	Risk Register	Identifies anticipated risks, such as bias, data leakage, and misuse, and outlines mitigation strategies.
Agile (Scrum-aligned) iterative delivery artifacts	Product Backlog	Prioritized list of analytical tasks, hypotheses, and modeling objectives.
	Sprint Backlog	Subset of backlog items selected for execution within a sprint.
	Sprint Goal / Objectives	Concise statement of the intended outcome for a sprint iteration.
	Sprint Review Summary	Documentation of completed work, feedback, and demonstrated outputs.
	Retrospective Notes	Structured reflection on process effectiveness, challenges, and improvement actions.
Hybrid AI-supervision artifacts	Prompt Protocol	Structured documentation of prompt design, revisions, and AI responses across iterations.
	Validation Ledger	Evidence of human verification, statistical checks, diagnostics, and corrective actions.
	AI Audit Trail	Chronological log linking inputs, outputs, decisions, and revisions for reproducibility.
Hybrid AI-supervision artifacts	Reflection Memo	Narrative account of learning, challenges, ethical reasoning, and judgment calls.
	Guardrail Specification	Explicit constraints, invariants, and boundaries governing acceptable AI behavior and outputs.
	Developer/Team Capability Profile	Description of human expertise, domain knowledge, and oversight responsibilities in the human-AI system.

Table 4 clarifies why the artifact set matters, both educationally and institutionally, by linking each category to learning outcomes and quality-assurance expectations.

Table 4: Alignment of core artifacts with learning outcomes and accreditation expectations.

Artifact Category	Primary Learning Outcomes Supported	Accreditation / QA Relevance
Predictive (PMBOK-aligned) governance artifacts	Problem definition, requirement analysis, risk awareness, quality assurance, and ethical compliance	Evidence of planning, rigor, traceability, and governance is required for program review and accreditation
Agile (Scrum-aligned) iterative delivery artifacts	Iterative reasoning, feedback incorporation, collaboration, and metacognitive reflection	Demonstrates continuous improvement, formative assessment, and learner engagement
Hybrid AI-supervision artifacts	Critical evaluation of AI outputs, human judgment, validation, reproducibility, and ethical oversight	Provides auditable evidence of responsible AI use, authentic assessment, and transparency

Table 5 summarizes how we conceptualize the interaction of PMBOK predictive governance, Agile iterative delivery, and the CRISP-DM analytics workflow as a layered system for AI-assisted data science work. PMBOK establishes the governance backbone by defining scope boundaries, quality criteria, and stage-gate checkpoints. Agile then operates inside those guardrails through short micro-iterations that support feedback, adaptation, and incremental delivery. CRISP-DM structures the analytical work itself across data understanding, preparation, modeling, and evaluation cycles, producing empirical evidence that informs both Agile iteration decisions and PMBOK gate reviews. Across all layers, hybrid AI-supervision artifacts function as an oversight anchor. They make transparency, traceability, validation, and ethical accountability explicit by connecting decisions, outputs, and revisions into an auditable record that supports responsible AI use and defensible assessment.

Table 5: Conceptual interaction of PMBOK predictive governance, Agile iterative delivery, and CRISP-DM analytics.

Layer	Primary purpose	What it controls	How it interacts with the other layers	Typical outputs	Oversight anchor (Hybrid AI-supervision artifacts)
PMBOK predictive governance	Establishes upfront scope, quality criteria, and governance gates	Scope boundaries, acceptance criteria, risk controls, decision rights, stage-gate approvals	Defines the constraints and guardrails within which Agile micro-iterations proceed; sets "go/no-go" gates that analytics outcomes must satisfy before advancing	Project charter, scope baseline, quality plan, risk register, gate checklists, governance approvals	Governance log, accountability matrix, audit trail of decisions, ethical and compliance checkpoints
Agile iterative delivery	Executes micro-iterations to enable feedback, adaptation, and incremental value delivery	Sprint goals, backlog prioritization, rapid refinement of requirements and artifacts	Operates inside PMBOK-defined gates; uses CRISP-DM outputs to refine user stories, acceptance criteria, and deliverables based on evidence	Sprint backlog, increments/prototypes, demos, retrospectives, updated user stories	Human-in-the-loop review notes, iteration-level validation reports, change rationale tied to gates and criteria
CRISP-DM analytics workflow	Structures the analytical workflow across modeling and evaluation cycles	Data understanding, preparation, modeling, evaluation, and iteration logic	Provides the evidence base that informs Agile iteration decisions; produces evaluation results that feed PMBOK gate reviews and quality criteria	Data documentation, feature engineering logs, model candidates, evaluation metrics, error analysis, model cards	Model governance artifacts: validation checklist, bias/fairness checks, reproducibility package, metric thresholds and sign-off
Cross-layer integration (Hybrid AI-supervision artifacts)	Anchors human oversight, validation, and ethical accountability across all layers	Transparency, traceability, responsible AI controls, and decision documentation	Connects PMBOK gates, Agile iteration evidence, and CRISP-DM evaluation into a unified oversight system	Integrated audit trail, traceable requirements-to-evidence map, risk/ethics register updates, approvals, and exceptions	Central supervision bundle: oversight dashboard, approvals ledger, exception handling, provenance and reproducibility record

5. Agile in Comparative Context

Among the frameworks considered—PMBOK-aligned governance, Design Thinking, and CRISP-DM—Agile provides the most pedagogically coherent rhythm for integrating GenAI into STEM curricula. While each model contributes distinct strengths, Agile's iterative, reflective, and adaptive structure best aligns with the cognitive demands of human-AI collaboration and the

dynamic nature of generative technologies. Its principles of incremental progress, continuous feedback, and collaborative learning parallel the way students must interact with AI systems: prompt, evaluate, refine, and iterate. In this sense, Agile transcends its origins in software engineering and becomes a cognitive-pedagogical framework for managing uncertainty and promoting metacognition [14, 15]. Because Agile encompasses multiple frameworks, Scrum is referenced when discussing sprint mechanics, while Disciplined Agile represents PMI's integrated agile portfolio. Absent explicit guardrails and audit artifacts, "vibe coding" devolves into untraceable experimentation, increasing the risk of hallucination and reducing instructional defensibility.

5.1.1. Agile versus PMBOK-Aligned Governance: Adaptability over Proceduralism

PMBOK-aligned governance, articulated through PMI standards and guides, supports multiple delivery lifecycles, including predictive, hybrid, and adaptive approaches. Its emphasis on scope control, quality standards, risk management, and documentation establishes accountability and traceability that are essential for accreditation and quality assurance [31, 32, 33]. Rather than prescribing a single linear process, it defines governance structures and decision points that can be applied across varying levels of uncertainty.

Within that governance context, Scrum-style Agile delivery emphasizes rapid iteration, empirical feedback, and continuous adaptation. Students working with GenAI benefit from cycles that support experimentation, hypothesis revision, and self-correction—capabilities that are central to Agile practice [12, 34]. PMBOK-aligned governance explicitly accommodates such uncertainty through hybrid and adaptive approaches, including rolling-wave planning, in which near-term work is elaborated in detail while future requirements remain progressively refined [31, 33, 41].

Agile complements PMBOK governance by converting static compliance artifacts into dynamic accountability mechanisms. Where governance requires defined scope, quality criteria, and risk controls, Agile practices such as sprint reviews and retrospectives provide frequent evidence-based reassessment within those constraints [31, 33, 34]. In educational settings, the combined use of Scrum-style iteration and PMBOK-aligned governance preserves oversight while supporting adaptability and deeper learning.

5.1.2. Agile versus Design Thinking: Iteration Meets Empathy

Design Thinking prioritizes human-centered creativity through phases of empathy, ideation, prototyping, and testing [18, 19]. It is particularly effective at fostering problem framing and divergent thinking, two areas in which GenAI can serve as a brainstorming assistant. Yet Design Thinking's lack of temporal discipline and quantifiable checkpoints can limit its pedagogical utility when complex projects require measurable progress and traceable revisions. Agile's sprint-based cadence addresses these limitations by embedding iterative experimentation within defined timeboxes and evaluation criteria. Students can use GenAI in the ideation phase (to generate design alternatives or test assumptions) while applying Agile methods to manage deliverables, feedback loops, and ethical reviews. The integration of Design Thinking and Agile thus supports both creativity and accountability: empathy-driven exploration is balanced by disciplined reflection and documented iteration. This fusion mirrors professional AI development practices, where user needs and system transparency must coexist.

5.1.3. Agile versus CRISP-DM: Managing Uncertainty in Analytical Workflows

The Cross-Industry Standard Process for Data Mining (CRISP-DM) remains the dominant methodology for analytics education and professional data workflows [20, 21]. It enforces a

rigorous sequence-Business Understanding, Data Understanding, Preparation, Modeling, Evaluation, and Deployment-ensuring analytic soundness and reproducibility. However, CRISP-DM's structured linearity can impede rapid adaptation when working with generative systems or exploratory learning tasks. Agile, by contrast, thrives on uncertainty. and iteration. Each sprint functions as a micro-cycle of hypothesis testing: students define objectives, apply AI tools, validate outputs, and adjust methods. Embedding CRISP-DM phases within Agile sprints yields dual benefits: the methodological rigor of CRISP-DM and the flexibility of Agile. For example, Sprint 1 may focus on Data Understanding and Preparation; Sprint 2 on Modeling and Evaluation; Sprint 3 on Deployment and ethical reflection. This integration enables iterative validation of both models and reasoning processes, aligning with evidence-based inquiry central to STEM education.

5.1.4. Agile versus Accreditation and Governance Frameworks

Accrediting agencies increasingly emphasize continuous improvement, transparency, and reflection as markers of educational quality [10]. Agile's cyclical processes naturally satisfy these principles through its built-in retrospectives and documentation loops. Each sprint conclusion offers an opportunity for formative reflection, risk reassessment, and recalibration of ethical guidelines. Unlike one-time audits or end-of-term evaluations, Agile provides embedded accountability that evolves with learner maturity. From a governance standpoint, Agile operationalizes institutional ethics policies into recurring classroom practice. Disclosure statements, sprint logs, and AI audit trails become living documents rather than static compliance artifacts. This alignment ensures that AI integration remains not only pedagogically effective but also institutionally defensible, fulfilling both ethical and regulatory expectations.

5.1.5. Agile as a Cognitive Apprenticeship Model

Beyond project management, Agile functions as a cognitive apprenticeship model for AI supervision. Each iteration allows learners to externalize thought processes, test reasoning, and receive feedback, core mechanisms of metacognitive development [14, 15]. Faculty act as "coaches" who guide reflection, rather than as gatekeepers enforcing compliance. Students progressively assume more autonomy, mirroring the transition from novice to expert supervision. In AI-rich learning environments, this structure teaches how to manage uncertainty, monitor cognitive bias, and evaluate evidence, skills critical for trustworthy AI use. Agile's short feedback cycles cultivate self-regulation and awareness of cognitive load [7], while retrospectives provide explicit space for ethical reasoning and emotional calibration. This integration of technical, reflective, and ethical domains represents the essence of responsible AI education.

5.1.6. Why Agile Excels for Generative AI Supervision

Generative AI introduces three core challenges: the unpredictability of output, the opacity of reasoning, and dependence on contextual prompts. Agile directly addresses these through iterative supervision, collective review, and continuous adaptation. Each sprint becomes a controlled experiment in which learners balance delegating to AI with verifying its results. This process mirrors professional AI workflows, such as reinforcement learning with human feedback (RLHF), in which oversight and correction occur continuously rather than post hoc [12]. Agile thus operationalizes the principles of ethical and reflective AI use: transparency, accountability, and human judgment [57]. It transforms learning from a static transfer of knowledge to a dynamic process of managing intelligent systems, precisely the competence required in the AI-augmented workforce [2, 5]. For these reasons, Agile serves as the pedagogical core of the proposed hybrid model, providing the rhythm and reflective depth that other frameworks alone cannot sustain.

Table 6 compares Agile, PMBOK-aligned governance (labeled "PMI" in the matrix for brevity), Design Thinking, and CRISP-DM through the lens of supervising generative AI in STEM courses, where students must iterate quickly while maintaining rigor and accountability. Agile aligns best with the natural human-AI loop because it supports rapid prompt refinement, short feedback cycles, and metacognitive reflection through sprint reviews and retrospectives [15]. PMBOK-aligned governance is less adaptive in day-to-day iteration, but it is strongest for traceability because it forces explicit scope boundaries, quality criteria, risk controls, and decision checkpoints that make AI use auditable.

Design Thinking supports exploration and problem framing, but it is typically weaker on structured validation unless instructors add formal evidence gates. CRISP-DM provides the most direct analytics structure for data understanding, preparation, modeling, and evaluation, yet it does not automatically enforce prompt discipline or revision transparency without additional supervision artifacts. Taken together, the table motivates a hybrid approach: Agile provides iteration and reflection, PMBOK-aligned governance provides traceability and ethical accountability, and CRISP-DM provides analytical rigor, with AI supervision artifacts bridging all layers to ensure students doubt outputs, validate claims, and document decisions rather than trusting AI-generated results.

Table 6: Comparative analysis of Agile and related frameworks for Generative AI supervision in STEM education.

Criterion	Agile	PMBOK	Design Thinking	CRISP-DM
Adaptability	High supports prompt refinement and model evolution	Low-linear and prescriptive	Medium-creative but less structured	Low-sequential and rigid
Cognitive Scaffolding	Iterative reflection and metacognitive growth [15]	Emphasizes compliance, not reasoning	Encourages empathy and ideation	Focused on procedural rigor
Ethical Accountability	Built into retrospectives and sprint logs	Enforced post-process	Contextual but under-documented	External to workflow
Data and Workflow Rigor	Balanced via iterative validation and CRISP-DM embedding	Moderate through documentation	Weak-lacks data discipline	Strong-excellent reproducibility
Alignment with AI Pedagogy	Mirrors human-AI interaction cycles (prompt-evaluate-revise)	Static oversight without adaptation	Strong for ideation and creativity	Strong for analytics, weak for iteration

5.1.7 Rubric Anchors

The assessment rubric for AI-as-Intern courses should evaluate:

- **Delegation Quality** - clarity and appropriateness of tasks assigned to AI.
- **Validation Rigor** - thoroughness of checks against sources or datasets.
- **Documentation** - completeness and organization of audit materials.
- **Ethical Reasoning** - identification and mitigation of bias or misuse.
- **Reflection Depth** - metacognitive insight and articulation of learning transfer.

By grading supervision rather than output, the framework reinforces that responsible AI integration is an intellectual skill, not a shortcut to automation. Empirical studies in authentic assessment [3] show that such process-oriented rubrics promote deeper engagement, transfer of learning, and ethical awareness.

Taken together, the comparison supports the hybrid design advanced in this paper: Agile supplies iteration and reflection, PMBOK-aligned governance supplies traceability and ethical accountability, and CRISP-DM supplies data-scientific rigor. AI supervision artifacts bridge these layers by ensuring that students question outputs, validate claims, and document decisions rather than trusting AI-generated results.

6. Curriculum Integration

Effective integration of the PMBOK-Agile hybrid framework within STEM curricula requires deliberate scaffolding across academic levels to balance cognitive load, ethical reasoning, and disciplinary depth. Each level introduces progressively greater autonomy while maintaining reflective oversight and documentation requirements.

6.1 Re-centering Mathematical and Statistical Foundations through Vibe Coding

The integration of vibe coding into Data Science curricula creates an opportunity to re-center mathematical and statistical foundations that are often marginalized by syntax-heavy instructional practices. Traditional programming-centered instruction often prioritizes implementation mechanics over analytical reasoning, leaving little time to understand the theoretical assumptions that govern data-driven models. By automating routine coding tasks through AI-assisted collaboration, vibe coding reduces extraneous technical overhead and enables deeper engagement with probability, inference, linear algebra, and optimization, which form the computational core of Data Science [23, 24]. A central benefit of this shift is the increased emphasis on model interpretability. For example, rather than focusing on how to implement a regression model in a particular library, students can focus on interpreting coefficients, understanding their magnitude and direction, and explaining how they relate to the underlying data-generating processes. This conceptual focus reinforces the distinction between statistical association and causal reasoning and encourages students to critically evaluate whether model outputs align with domain knowledge and theoretical expectations [24, 27].

Vibe coding also supports more rigorous treatment of multicollinearity, a common source of misinterpretation in applied modeling. AI-generated code may successfully fit a model while obscuring instability caused by correlated predictors. Under a vibe coding framework, students are expected to diagnose such issues by examining correlation structures, variance inflation indicators, and sensitivity of coefficient estimates across model specifications. The instructional emphasis thus shifts from producing a working model to reasoning about its reliability and limitations, reinforcing linear algebraic concepts underlying matrix conditioning and parameter estimation [23].

Similarly, the bias-variance tradeoff becomes more accessible when coding mechanics are abstracted away. Rather than tuning models through trial-and-error execution, students can reason explicitly about underfitting and overfitting, model complexity, and generalization error. AI-assisted workflows allow learners to explore these tradeoffs conceptually by comparing model behavior across validation contexts and reflecting on how data size, feature selection, and regularization influence performance [29]. This analytical framing promotes principled model selection grounded in statistical reasoning rather than empirical guesswork. Uncertainty communication represents another critical area where vibe coding strengthens foundational

understanding. Data Science models produce probabilistic outputs, yet uncertainty is often underemphasized in implementation-focused instruction. By reallocating instructional effort toward interpretation, students can engage more deeply with confidence intervals, predictive uncertainty, and the limitations of point estimates. Vibe-coding environments require students to articulate uncertainty explicitly, assess the robustness of conclusions, and communicate limitations transparently, aligning instructional practice with professional standards for responsible data analysis [30, 26].

Collectively, these shifts demonstrate that vibe coding does not diminish technical rigor but instead repositions it. By delegating routine implementation to AI systems under human supervision, educators can foreground the mathematical and statistical reasoning that distinguishes Data Science from software development. This re-centering strengthens students' ability to design, interpret, and validate models, ensuring that automation enhances analytical depth rather than replacing it. Despite the growing importance of Data Science across disciplines, many curricula remain disproportionately focused on syntactic proficiency, emphasizing programming libraries, APIs, and procedural execution over conceptual understanding [22]. While coding fluency is necessary, this emphasis often comes at the expense of foundational topics such as probability theory, statistical inference, linear algebra, and optimization, which are essential for interpreting models and reasoning under uncertainty [23]. As a result, students may learn how to implement algorithms without fully understanding their assumptions, limitations, or implications [24].

6.1.1 Introductory Level

At the foundational level, students complete short, faculty-guided exercises that introduce them to GenAI capabilities and limitations. These activities may include critiquing intentionally flawed AI outputs, correcting code snippets, or validating statistical summaries against authoritative datasets. At this stage, the instructional emphasis is not on independent use of AI, but on developing metacognitive awareness of how and why AI-generated content can fail. The assessment focuses on students' ability to identify reasoning errors, articulate why an output is incorrect or incomplete, and justify corrective actions, rather than producing polished solutions.

The instructional goal is to cultivate an understanding that AI-generated content requires human verification rather than blind trust. Faculty act as co-supervisors, explicitly modeling structured reasoning, verification strategies, and reflective questioning [49, 50]. Vibe coding is introduced in a limited and guided form, ensuring that students first develop cognitive schemas for evaluation, bias awareness, and transparency before engaging in more autonomous human-AI collaboration. Early exposure lays the metacognitive foundation for later independent supervision and establishes a baseline ethical literacy in bias, uncertainty, and accountability. The following vignette is illustrative rather than prescriptive; it is one feasible pattern for structuring early supervision practice.

6.1.2 Intermediate Level

Intermediate courses extend supervision practices through structured projects that span multiple Agile sprints and include formal documentation for each iteration. Students develop and maintain prompt protocols, validation ledgers, and reflection logs that explicitly capture planning, monitoring, and evaluation decisions. At this level, assessment criteria shift decisively away from the correctness of outputs toward how students reason about AI-generated results, including how assumptions are tested, errors are detected, and revisions are justified.

Faculty provides rubrics that emphasize metacognitive processes, such as the clarity of planning strategies, the rigor of validation checks, and the depth of reflective analysis, rather than the superficial quality of final deliverables. Tanner's metacognitive principles guide structured reflections in which students analyze their decision paths, justification strategies, and iterative revisions [15]. At the intermediate level, students are ready for partial immersion in vibe coding, in which they assume increasing responsibility for supervising AI outputs while remaining within faculty-defined scaffolds. This stage bridges procedural competence and conceptual understanding, preparing students for full agentic collaboration.

6.1.3 Advanced and Capstone Level

In advanced undergraduate or capstone courses, learners implement full PMBOK-Agile cycles encompassing initiation, planning, execution, monitoring, and closing. Deliverables include comprehensive reproducibility packages, risk registers, and ethics statements documenting how bias, privacy, and fairness considerations were addressed. At this level, assessment prioritizes students' ability to manage complex reasoning processes across extended timelines, integrating technical, ethical, and organizational constraints.

Students are expected to coordinate project teams, delegate tasks among peers and GenAI systems, and maintain audit trails that satisfy institutional and accreditation standards for transparency [10]. The instructional focus shifts from learning AI functionality to demonstrating professional-grade oversight, with metacognitive supervision, decision accountability, and reflective governance at the center. This level represents full immersion in vibe coding, as students possess the conceptual maturity required to supervise AI systems autonomously while maintaining analytical rigor.

6.1.4 Graduate and Research Level

Graduate-level implementation situates the framework within authentic research and applied inquiry contexts. AI tools serve as assistants for literature synthesis, model prototyping, and exploratory hypothesis generation, while students remain fully accountable for methodological rigor, statistical validity, disclosure, and auditability [12]. At this stage, assessment centers on students' ability to justify methodological choices, evaluate uncertainty, and critically assess the epistemic limits of AI-assisted analysis.

Dwivedi et al. emphasize that future professionals must manage AI as a collaborator in data-driven inquiry rather than as a passive computational utility [12]. Accordingly, graduate curricula should require reproducibility packages, detailed code, clear documentation, and explicit statements of AI involvement. Graduate-level work entails sustained research-grade immersion in vibe coding, in which metacognitive regulation, ethical self-governance, and analytical responsibility are fully internalized.

6.2 Examples and vignettes for each instructional level

6.2.1 Illustrative vignette: Introductory Data Science, Class Plan, Introductory Level.

The professor opens class by projecting an AI-generated answer to a homework-style question the students recognize. The response is fluent and confident, but wrong in subtle ways. Before anyone speaks, the professor sets the rule for the day: "We are not here to use AI. We are here to interrogate it. Assume this answer is unreliable until proven otherwise."

She walks through the first paragraph aloud, pausing at moments of uncertainty to flag them. She notes where the language is vague about assumptions, where a confidence interval is

mentioned without specifying conditions, and where a causal claim quietly replaces a descriptive one. She explicitly rejects one sentence, explaining why it violates basic statistical reasoning. She models verification in real time, checking definitions, questioning assumptions, and explicitly rejecting a claim that cannot be supported by the data. Students watch the instructor model how a data scientist thinks in real time, checking definitions, questioning scope, and treating AI as an unverified secondary source rather than a shortcut. This step can also be executed as group exercise.

Students are then given a short worksheet containing three AI artifacts: a flawed explanation, a misleading visualization, and a short code snippet that runs but produces an incorrect summary. Their task is not to improve it, extend it, or optimize it. Their task is to break it. Working in small groups, they annotate each artifact. They are required to name the failure mode they see, whether it is a logical gap, a statistical misinterpretation, a hidden assumption, or a hallucinated reference. No one is allowed to rewrite the output yet. The emphasis is on diagnosis, not repair. They annotate the response, marking logical gaps, incorrect statistical reasoning, unsupported claims, and misleading language. Each issue must be labeled by type of failure, not simply flagged as "wrong."

In the second phase, students propose corrections, but each change must be justified in writing. A student who revises the meaning explains why the original calculation was inappropriate, given the skewed distribution. Another student replaces visualization and cites lecture notes on scale distortion. The professor circulates, asking questions like, "What evidence do you have that this is wrong?" and "How would you convince someone who trusts the AI more than you?" Every correction must be paired with a justification grounded in course material, lecture notes, or an authoritative dataset.

To ground the critique, students validate claims against a provided dataset and a short excerpt from a peer-reviewed article. They use a verification checklist provided by the instructor to confirm variable definitions, sampling assumptions, and boundary conditions. They describe what initially made the AI answer convincing, what signals triggered skepticism, and what judgment calls required human reasoning rather than automation. When discrepancies arise, the professor highlights them as productive moments, reinforcing that uncertainty and error detection are expected outcomes rather than failures.

At the end of class, students submit a brief reflection memo. They respond to prompts about what initially made the AI output seem plausible, what signals triggered skepticism, and what judgment calls required human reasoning rather than automation. The professor emphasizes that these reflections matter more than arriving at a clean final answer.

Assessment for the activity rewards diagnostic reasoning. Students earn credit for accurately identifying errors, clearly explaining why they matter, and supporting their critique with evidence. Students who accept AI outputs uncritically lose points, even if their final answer happens to be correct. Drafts, margin notes, and partial analyses are explicitly encouraged.

Later in the term, limited vibe coding is introduced only after students demonstrate consistent competence in critiquing AI-generated answers and code. Even then, students must read generated code line by line, predict outcomes, and identify potential bias or failure modes before execution. The professor occasionally refuses to use AI during demonstrations to reinforce that human judgment is not optional.

By design, students leave the foundational phase with a habit of skepticism. They understand that AI outputs are starting points for critique, not authorities. They know how to identify mistakes, explain corrections, and justify trust decisions in line with disciplinary standards. Most importantly, they learn that in data science, verification is the work.

6.2.2 Vignette: Faculty framing for an intermediate data science face-to-face class, Intermediate Level

The professor opens Week 2 with a simple statement on the board: "AI is not the analyst. You are."

She informs the class of the rule for the term: any AI output is a draft that must earn trust through evidence. The grade will not be awarded for polished answers. It will reward *students for how they* question, test, and revise. She demonstrates with a live prompt.

- **Initiate/ Build (brainstorm):**
  "I'm going to ask AI for three modeling approaches," she says, "but I am not asking it to choose for me." She types: Propose 3 reasonable modeling approaches for this dataset and 5 risks that could invalidate the results. Do not assume anything you cannot verify. Ask me 3 questions that would change your recommendation."
  AI responds quickly. Students nod. The professor pauses: "Now we doubt it."
- **Plan/Understand:** She highlights two AI assumptions and asks, "Where would we find evidence for these?" Students point to the data dictionary and the target definition. She underlines: the knowledge lives here, not in the AI response.
- **Discuss/ Execute/Model:** She asks students to challenge one AI suggestion each: "What could go wrong? What would mislead us?" They bring up leakage, wrong splits, and misleading metrics.
- **Revise (guide the AI)/ Monitor/Control:** She says, "If the AI is vague, that's on us. We didn't manage it." She types a tighter prompt: "Only generate code to test missingness, leakage indicators, and split validity. Output a checklist of tests and what would count as a failure." The response is better.
- **Validate (look for mistakes):** She runs the checks. One feature is clearly post-outcome. "AI didn't catch that," she says. "And that's normal. AI executes. It doesn't own correctness."
- **Close/ Deploy:** She shows the class what earns credit: *AI-suggested feature X. We tested prediction-time availability. It failed. We rejected it.* Remove the feature, rerun the evaluation, and update the risk register.
  She closes the lesson with the framing she wants students to internalize:

  "You don't use AI to get answers. You use AI to generate candidates, then you prove what is true. Your expertise, judgment, and verification are the product. AI performs only what you supervise."

6.2.3 Discussion on requirements and addressing AI at this level in Assessments, Advanced Level

In this advanced capstone assignment, teams use an appropriate professional platform, typically GitHub, to ensure version control, traceability, and the integrity of the work. All code, documentation, and artifacts are committed through a disciplined workflow (clear commit messages, tagged releases at each sprint, and pull requests or review notes), ensuring a verifiable history of what changed, when, and why. This is not "extra tooling." It is the mechanism that makes supervision real: instructors can audit iterations, confirm reproducibility, and distinguish original student reasoning from AI-generated drafts. Alongside GitHub commits, students are required to maintain a conversational and revision log with AI, which functions like a controlled lab notebook. Each AI interaction is recorded with the prompt, the AI output, what was accepted or rejected, the evidence used to validate it, and the resulting revision (linked to the relevant GitHub commit or

notebook cell). This pairing creates a complete audit trail: prompt output verification and decision versioned changes.

Metacognition is treated as a graded skill, not a reflection add-on. Each sprint includes a short, structured exercise in which students articulate how their thinking evolved: the assumption they initially held, the evidence that challenged it, what they changed, and the risk or quality gate that drove the change. These metacognition prompts are concrete and decision-centered (for example: "Which AI suggestion did you distrust most, what test did you run, and what did the result force you to revise?"). Over time, students internalize the practice of supervising AI like a delegated specialist: they learn to guide it with precise constraints, demand testable outputs, validate every claim, and document reasoning in a way that withstands external review.

6.2.4 Example of a Data Science Assignment for students who have already undergone instruction on the ethical and proper use of AI in their class, Graduate Level

The assignment includes instructions for completing a Classification problem using Random Forest. The assignment comprises 8 stages and requires them to log each interaction with the AI. Students will be evaluated on preprocessing, modeling, hyperparameter tuning, evaluation, and communicating metrics, as well as on submitting the following: the Validation and Metacognition Exercise and the AI Dossier, as described by Tsapara et al. [58].

Validation and Metacognition Revision Logs

Metacognition is assessed through reasoning logs, annotated revisions, and reflections that show planning, monitoring, and evaluation.

In the course design, students complete a Validation and Metacognition Log to make explicit the planning, monitoring, and evaluation of AI-rich analytic work. The log has four parts: (A) a pre-training validation plan that identifies the three most likely failure modes in the pipeline (e.g., leakage, cross-validation errors, metric or encoding mistakes), specifies concrete detection checks for each, and defines what "suspiciously good" performance would look like and what to audit first; (B) exactly three structured, iterative GenAI dialogues focused on leakage and CV integrity, metric critique and threshold reasoning, and method or hyperparameter justification, with each dialogue linked to a corresponding code change or action; (C) at least one independent validation check beyond the AI's claims (e.g., a majority-class baseline, a two-pipeline comparison with controlled CV, or another explicit verification step); and (D) a post-task reflection documenting revisions triggered by evidence, where GenAI was helpful versus misleading, one incorrect or incomplete AI output and the student's correction, and planned improvements for earlier validation in future work. [58]

AI Evaluation Dossier

Students must include an AI Evaluation Dossier entry (with a provided exemplary) that is completed progressively across all stages of the assignment. The dossier documents (1) AI-use disclosure describing where and why AI tools were used, (2) revision evidence (3 to 5 stage-based entries) explaining what changed, why, and what evidence prompted revisions, (3) at least one explicit validation check, (4) a metrics and validation critique identifying which metrics AI suggested, what was missing or inappropriate, and what the student added, (5) function and method analysis detailing the key libraries, functions, and algorithms used, their suitability, and viable alternatives, and (6) student-added contributions beyond AI output (e.g., stronger validation, alternative models, refined interpretation). To support fairness and confirm understanding, students may also complete a brief ownership verification using their minimum viable evidence

artifacts, explaining the preprocessing location and rationale, the chosen metric, any trade-offs made, one AI suggestion rejected or corrected, and the validation check they trust most. Finally, students submit downloaded AI dialogue logs as dossier artifacts. [58]

6.3 Scaffolding and Cognitive Load

This developmental trajectory aligns with Bloom's revised taxonomy, progressing from understanding and applying to analyzing, evaluating, and creating [1]. Cognitive load theory further supports the incremental release of responsibility across instructional levels [7]. Novice learners benefit from tightly structured prompts, exemplars, and validation templates, while advanced learners engage in extended, open-ended supervision cycles that demand independent judgment. As scaffolds gradually fade, extraneous cognitive load decreases, and students allocate cognitive resources to higher-order reasoning, interpretation, and ethical decision-making. This progression ensures that full immersion in vibe coding occurs only after learners have developed the metacognitive capacity required for responsible human-AI collaboration, reinforcing expertise through autonomous reasoning rather than premature reliance on automation.

7 Discussion

The detailed design, implementation, and grading of the assignments and lesson plans, including the required AI Evaluation Dossier components (AI-use disclosure, revision evidence, validation checks, metric critique, method analysis, student-added contributions, ownership verification, and dialog-log artifacts), are outside the scope of this article [58]. This article instead focuses on a framework for teaching interaction and collaboration with Generative AI in coding contexts, in which the learner serves as a project manager and is accountable for quality control, critical thinking, learning, and outcome validation. Nevertheless, the AI-guided instructional approach led to student self-reflections suggesting growth in metacognitive monitoring and evaluation, including attention to specific concepts. We treat these observations as anecdotal and illustrative rather than as formal evaluation. Students reported that AI support was most beneficial when used to question assumptions and stress-test decisions rather than to generate final answers. In our informal review of dossiers and reflections, we observed a few signs of uncritical reuse of AI-generated instructions. Because we did not conduct formal similarity scoring or systematic coding of reflections, we treat this observation as anecdotal rather than quantitative.

Additionally, formal quantitative outcome analysis is still developing; future work should test learning gains, validation accuracy, and transfer relative to traditional programming assignments. A full analysis, including similarity scoring across reflections, sentiment analysis compared with evaluation metrics, correlation analysis, and detailed Exploratory Data Analysis, will be presented in a future article based on the consenting cohort.

7.1 Equity, Access, and Linguistic Mediation in AI-Rich Assessment

Prior work shows that prompt language can materially affect model performance, with English prompts often yielding stronger results than native-language prompts in some settings, thereby disadvantaging multilingual learners or students who are not confident composing iterative, high-precision questions in English [52, 56]. In addition, large language models can encode cultural and linguistic biases stemming from the dominance of the training data, which can shape explanations, examples, and evaluative framing in ways that are not neutral across populations [53, 54]. Consistent with international guidance that highlights the risks of digital inequities in educational GenAI adoption, we treat access, language, and tool choice as core validity considerations rather than peripheral constraints [51].

In future implementations, all participants will have access to institutionally supported tools, most notably Microsoft's Copilot environment, while also retaining the option to use alternative LLMs or free-tier tools that may differ in features, integrations, and support for longer, more technical dialogues [55]. We anticipate that such differences may affect the elaboration and usefulness of AI feedback, with important equity implications when students are evaluated on the quality of reflective dialogue and iterative improvement. To mitigate these concerns, the dossier will require explicit disclosure of tool choice and revision triggers, and the course will further support inclusion by providing multilingual prompt scaffolds, permitting bilingual logging (original language plus a brief English translation), and standardizing minimum-capability expectations, such as baseline context length, availability of chat-history export, and access to the same evaluation rubric, so that performance differences will be less attributable to tool tier than to student learning. Although these measures will support the feasibility of dossier-based AI quality control and metacognitive scaffolding, several limitations will still constrain interpretation and generalizability. The following section will summarize these limitations and outline targeted future work designed to address them.

7.2 Limitations

Although the purpose of this article is to conceptualize and frame an approach to teaching with vibe coding, the present discussion remains preliminary, grounded in a small, consenting cohort; accordingly, claims about effectiveness and equity should be treated as provisional and context-dependent. Rather than centering the framework on dossier-based assessment, the instructional emphasis should be placed on how faculty explicitly teach students to treat AI as an intern-level collaborator: useful, productive, and often insightful, but never autonomous, authoritative, or exempt from supervision. In classroom practice, this means modeling reiterated cycles of questioning, answering, revising, and re-questioning, so that students learn to expand rather than truncate inquiry. Instead of accepting the first plausible response, students are taught to ask follow-up questions, request alternatives, probe assumptions, compare explanations, and test whether an answer remains stable when reframed. This expansive approach is pedagogically valuable because it makes reasoning visible in class discussion and helps students experience AI interaction as a process of guided intellectual supervision rather than answer retrieval.

At the same time, important limitations remain. Tool use is heterogeneous and temporally unstable: even when students have access to institutionally supported systems, model behavior, interfaces, and features evolve over time, and students may use different LLMs or access tiers, creating meaningful variation in the depth, style, and usefulness of AI feedback that is unrelated to learning itself [53]. Equity and accessibility concerns also extend beyond first-language status. Reiterative questioning, extended dialogue, and critical comparison of outputs can impose additional cognitive, linguistic, and time burdens on some students, while differences in digital fluency may further widen participation gaps. Privacy and governance constraints likewise remain significant, because classroom use of AI may involve sensitive prompts, third-party processing, data retention, and the need for informed boundaries around what should and should not be entered into external systems [53].

These limitations are not peripheral because the teaching practices themselves are mediated by language, access, and confidence. Research on prompting suggests that the language students use when interacting with models can shape the quality of responses, potentially disadvantaging those who are less fluent or less comfortable sustaining iterative, high-precision dialogue in English [54]. Cultural and linguistic skews in model outputs may also influence what is presented as clear

reasoning, persuasive evidence, or appropriate interpretation, meaning that classroom discussion of AI outputs cannot be treated as culturally neutral [55]. For this reason, the framework should mitigate inequities not primarily by intensifying assessment requirements, but by strengthening instructional supports: faculty can model how to question AI recursively, provide multilingual prompt scaffolds, normalize bilingual discussion, and create structured in-class conversations about why one answer is more trustworthy than another. In this version of the framework, metacognition is developed through guided classroom discourse rather than treated chiefly as a graded compliance exercise. Students learn to articulate what they asked, why they asked it, what changed in the answer, what they still distrust, and how their own thinking evolved through interaction. International guidance similarly underscores that responsible educational adoption of generative AI requires sustained attention to access, inclusion, and governance, reinforcing that these are central validity issues for teaching and learning, not merely technical implementation details.

8 Future work

Future work should prioritize broader and more representative replication, tighter standardization of the AI treatment, stronger inclusion, governance, and transparency controls, expansion to multiple assignments and disciplines, and fully specified introductory lesson and assessment guidance. This work should be extended to larger cohorts across multiple courses, institutions, and STEM disciplines, with eventual adaptation to other fields [5, 9, 11]. These replications should report disaggregated outcomes, including language background and access constraints, and should explicitly characterize possible consent and participation bias [51]. Future studies should also standardize the AI treatment more carefully by recording model and platform metadata, defining a minimum-capability baseline, and comparing outcomes when students use a common tool versus tools of their own choosing [6, 17, 40]. A major line of future research should focus on faculty workload, especially in relation to the design, validation, and implementation of AI-supported assessments and metacognitive step-by-step guides for students [15, 44, 58]. This includes studying how lesson interventions and assessment structures can be scaled sustainably without overburdening instructors [44, 58]. Future work should also examine equity and accessibility through multilingual scaffolds, such as bilingual prompting templates and bilingual reflection options, while reducing unnecessary writing load by pairing structured checklists with brief justifications [46, 52]. In addition, privacy, governance, and regulatory constraints require deeper analysis, including the implications of transparency laws, accreditation requirements, and other institutional policies that may hinder the timely redesign of courses to align with emerging technologies [10, 40, 51]. In this area, future implementations should include clear data minimization rules for logs, standardized redaction guidance, retention windows, and transparent consent processes aligned with recognized GenAI education guidance [40, 51]. Finally, future research should address both technological uncertainty and scalability through sampling-based review, rubric calibration, and lightweight automated checks that support instructors without turning the dossier into a surveillance system [40, 57]. It should also consider the limitations of predicted GenAI developments and the growing implications of Agentic AI for supervision, assessment design, instructional responsibility, and institutional governance [5, 17, 57].

9 Assessment and Policy

Assessment and policy must evolve together if AI is to be integrated responsibly into teaching and learning. This section argues that when AI is treated as an academic "intern," evaluation should focus on the quality of student supervision, validation, and reflection, while institutional policy should provide the transparency, documentation, and governance structures needed to support that work consistently and ethically.

9.1 Process-Centered Evaluation

Traditional product-based grading fails to capture the quality of human supervision in AI-augmented learning. If AI is treated as an intern in class - capable of producing useful drafts, suggestions, and analyses, but still requiring direction, monitoring, and correction - then the primary evidence of mastery should be process-centered rather than product-centered. In this model, students are evaluated not only on the final artifact, but on how effectively they scope tasks for AI, judge the quality of its outputs, identify errors or bias, revise weak responses, and document responsible decision-making. This aligns with authentic assessment because it measures the kinds of supervisory, evaluative, and reflective practices students will need in real professional settings [3, 4]. It also reflects the growing labor-market reality that value increasingly lies not in avoiding AI, but in knowing how to direct and validate it responsibly [2].

Under this approach, process-centered deliverables should constitute the primary evidence of learning. These include:

- **AI Task Briefs:** Short planning documents in which students define what they are asking the AI to do, why they are delegating that subtask, what counts as an acceptable output, and what must still be checked by a human. This makes visible whether the student can frame a task appropriately rather than outsourcing thinking wholesale.
- **AI Audit Trails:** Structured records of prompts, outputs, follow-up prompts, and decision points, showing how the student interacted with the AI over time. These trails should reveal whether the student iteratively refined instructions, challenged questionable answers, and used the tool strategically rather than uncritically.
- **Annotated Revisions:** Side-by-side comparisons showing where students corrected AI-generated text, code, analyses, or explanations, with annotations explaining why changes were necessary. These revisions are especially valuable because they expose the student's capacity to improve accuracy, discipline-specific reasoning, tone, and ethical compliance [50, 58].
- **Validation Ledgers:** Tabular or checklist-based evidence of verification steps, such as source triangulation, dataset tests, recalculations, cross-checking with class materials, or model-output verification. In the "AI as intern" framing, this is the equivalent of demonstrating how a supervisor checks the work of a junior assistant before approving it [40, 51, 57].
- **Escalation Logs:** Brief records of moments when students determined that AI support was insufficient, misleading, or risky and that direct human judgment was required. These logs help distinguish responsible use from overreliance by rewarding students for recognizing when the "intern" should not be trusted without intervention.
- **Reflective Analyses:** Written commentaries in which students explain what the AI did well, where it failed, what biases or inaccuracies emerged, how they resolved uncertainty, and what they learned about their own reasoning in the process. These artifacts operationalize metacognitive awareness by asking students to monitor both the tool and themselves [14, 15].

This framework is particularly important because AI-supported learning is not simply about efficiency; it is about developing habits of oversight, judgment, and accountability. A student who can produce a polished final product with AI assistance may still have learned very little if they cannot explain how the output was generated, why it was revised, or what risks were involved. By

contrast, a student who documents careful prompting, verification, critique, and revision demonstrates genuine mastery of both the subject matter and the process of responsible AI supervision. In this sense, process-centered evaluation makes visible the human intellectual work that remains essential even when AI acts as an academic intern [44, 51, 57, 58].

9.2 Institutional Policy Implications

Institutional frameworks must evolve from restrictive to transparent integration policies [11]. We recommend establishing governance protocols that clearly define acceptable uses, disclosure requirements, and documentation standards. Related guidance similarly advocates that faculty shift from punitive enforcement to guided oversight [16] and [5]. The following policy components are recommended:

- **Disclosure Requirement:** All AI assistance must be acknowledged, with associated audit trails submitted as part of the delivery.
- **Documentation Standards:** Programs should adopt uniform templates for audit logs, validation records, and ethical statements to ensure comparability across courses.
- **Assessment Alignment:** Rubrics should explicitly measure ethical supervision, reflection, and verification processes, not merely technical outcomes.
- **Data Governance:** Institutions should provide approved, secure AI environments to prevent data misuse and privacy violations.

Such measures align with accreditation expectations for accountability and integrity [10], ensuring that AI-integrated pedagogy remains defensible, auditable, and ethically grounded.

10 Conclusion

GenAI is reshaping educational practice, requiring instructional models that move beyond prohibition and toward structured, transparent integration. This work has presented a supervised, agentic framework for human-AI collaboration that emphasizes accountability, metacognitive oversight, and alignment with established process models. By positioning AI as a collaborator rather than a replacement, the proposed approach supports ethical use while preserving academic rigor and institutional responsibility.

Within Data Science education, the integration of vibe coding represents a deliberate curricular transformation rather than a superficial adoption of new tools. Automating routine programming tasks allows instructional effort to be reallocated toward analytical reasoning, statistical inference, model interpretation, and uncertainty analysis. This alignment of automation with analytical depth strengthens students' ability to evaluate assumptions, validate results, and reason critically about AI-generated outputs, reinforcing the mathematical and statistical foundations that distinguish Data Science as a discipline.While this work focuses on Data Science, the framework can be extended in a bounded manner to Computer Science education, where the emphasis shifts from inference and uncertainty toward correctness, efficiency, and formal verification. In this context, human-AI collaboration would prioritize algorithmic validation, performance constraints, and correctness guarantees, illustrating that agentic AI frameworks must be adapted to disciplinary objectives rather than applied uniformly across domains. Future research will empirically evaluate this framework through mixed-methods studies and controlled comparisons measuring conceptual understanding, reasoning quality, and validation accuracy. These studies will provide stronger evidence for the pedagogical impact of governance-based AI supervision and clarify which foundational coding skills must remain explicitly scaffolded.

References

[1] Anderson, L. W., and Krathwohl, D. R. 2001.A Taxonomy for Learning, Teaching, & Assessing. Longman

[2] Bughin, J.; Seong, J.; Manyika, J.; Chui, M.; and Joshi, R. 2018. Notes from the AI frontier: Modeling the impact of AI on the world economy. McKinsey Global Institute.

[3] Gulikers, J. T. M.; Bastiaens, T. J.; and Kirschner, P. A. 2004. A five-dimensional framework for authentic assessment. Educational Technology Research and Development 52(3):67-86.

[4] Herrington, A., and Herrington, J. 2006. Authentic Learning Environments in Higher Education. IGI Global.

[5] Holmes, W.; Porayska-Pomsta, K.; and Rodrigo, M. M. T. 2023. Generative AI in education: Opportunities, challenges, and future directions. Computers and Education: Artificial Intelligence 5:100171.

[6] Kasneci, E.; Sessler, K.; Kuchemann, S.; Bannert, M.; Dementieva, D.; Fischer, F.; and Kasneci, G. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. Learning and Individual Differences 103:102274.

[7] Sweller, J. 1988. Cognitive load during problem solving: Effects on learning. Cognitive Science 12(2):257-285.

[8] Vygotsky, L. S. 1978. Mind in Society: The Development of Higher Psychological Processes. Harvard University Press.

[9] Wang, J.; Li, F.; and Chen, Y. 2024. Artificial intelligence in education: A systematic literature review. Expert Systems with Applications 247:123456.

[10] WASC Senior College and University Commission (WSCUC). 2022. 2022 Handbook of Accreditation. WSCUC.

[11] Zawacki-Richter, O.; Marin, V. I.; Bond, M.; and Gouverneur, F. 2019. Systematic review of research on artificial intelligence applications in higher education. International Journal of Educational Technology in Higher Education 16(1):39.

[12] Dwivedi, Y. K.; Kshetri, N.; Hughes, L.; Slade, E. L.; Jeyaraj, A.; Kar, A. K.; and Wright, R. 2023. So what if ChatGPT wrote it? International Journal of Information Management 71:102642.

[13] Kolb, D. A. 1984. Experiential Learning: Experience as the Source of Learning and Development. Prentice Hall.

[14] Flavell, J. H. 1979. Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry. American Psychologist 34(10):906-911.

[15] Tanner, K. D. 2012. Promoting student metacognition. CBE-Life Sciences Education 11(2):113-120.

[16] Cotton, D. R. E.; Cotton, P. A.; and Shipway, J. R. 2023. Chatting and cheating: Ensuring academic integrity in the era of ChatGPT. Innovations in Education and Teaching International 60(2):117-127.

[17] Jarrahi, M. H.; Newlands, G.; Lee, M.; Wolf, C. T.; Kinder, E.; and Sutherland, W. 2023. Navigating the generative AI revolution. Business Horizons 66(5):503-518.

[18] Brown, T. 2008. Design Thinking. Harvard Business Review 86(6):84-92.

[19] Cross, N. 2011. Design Thinking: Understanding How Designers Think and Work. Berg.

[20] Shearer, C. 2000. The CRISP-DM model: The new blueprint for data mining. Journal of Data Warehousing 5(4):13-22.

[21] Wirth, R., and Hipp, J. 2000. CRISP-DM: Towards a standard process model for data mining. In Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining (PKDD/PAKDD Workshop), 29-39.

[22] Wing, J. M. 2006. Computational thinking. Communications of the ACM 49(3):33-35.

[23] Hastie, T.; Tibshirani, R.; and Friedman, J. 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed. Springer.

[24] Shmueli, G. 2010. To explain or to predict? Statistical Science 25(3):289-310.

[25] Bates, T.; Cobo, C.; Marino, O.; and Wheeler, S. 2023. Can artificial intelligence transform higher education? International Journal of Educational Technology in Higher Education 20(1):1-21.

[26] Doshi-Velez, F., and Kim, B. 2017. Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608.

[27] Rudin, C. 2019. Stop explaining black box machine learning models. Nature Machine Intelligence 1(5):206-215.

[28] Luckin, R.; Holmes, W.; Griffiths, M.; and Forcier, L. B. 2016. Intelligence Unleashed: An Argument for AI in Education. Pearson Education.

[29] Geman, S.; Bienenstock, E.; and Doursat, R. 1992. Neural networks and the bias/variance dilemma. Neural Computation 4(1):1-58.

[30] Gelman, A.; Carlin, J. B.; Stern, H. S.; Dunson, D. B.; Vehtari, A.; and Rubin, D. B. 2013. Bayesian Data Analysis. 3rd ed. CRC Press.

[31] Project Management Institute. 2021. A Guide to the Project Management Body of Knowledge (PMBOK Guide). 7th ed. Newtown Square, PA: Project Management Institute.

[32] Project Management Institute. 2017. A Guide to the Project Management Body of Knowledge (PMBOK Guide). 6th ed. Newtown Square, PA: Project Management Institute.

[33] Project Management Institute. 2021. The Standard for Project Management. Newtown Square, PA: Project Management Institute.

[34] Project Management Institute. 2018. Agile Practice Guide. Newtown Square, PA: Project Management Institute.

[35] Project Management Institute. 2020. Disciplined Agile Toolkit. Newtown Square, PA: Project Management Institute.

[36] CRISP-DM Consortium. 1999. CRISP-DM 1.0: Step-by-Step Data Mining Guide. Brussels: SPSS Inc.

[37] Royce, W. W. 1970. Managing the Development of Large Software Systems. In Proceedings of IEEE WESCON, 328-338.

[38] Boehm, B. W. 1988. A Spiral Model of Software Development and Enhancement. Computer 21(5):61-72.

[39] Pressman, R. S., and Maxim, B. R. 2014. Software Engineering: A Practitioner's Approach. 8th ed. McGraw-Hill.

[40] Autio, C.; Schwartz, R.; Dunietz, J.; Jain, S.; Stanley, M.; Tabassi, E.; Hall, P.; and Roberts, K. 2024. Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile (NIST AI 600-1). National Institute of Standards and Technology.

[41] Knutson, J. 1996. Rolling wave approach to project management. PM Network. Project Management Institute.

[42] Project Management Institute. n.d. Process Groups: A Practice Guide.

[43] Shimaoka, A. M.; Ferreira, R. C.; and Goldman, A. 2024. The evolution of CRISP-DM for Data Science: Methods, processes, and frameworks. SBC Computing Reviews 4(1):28-43. https://doi.org/10.5753/reviews.2024.3757

[44] Fajardo-Ramos, D. C., and Mella-Norambuena, J. 2025. Human-in-the-loop assessment with AI: Implications for teacher education in Ibero-American universities. Frontiers in Education. https://doi.org/10.3389/feduc.2025.1710992

[45] Yan, L., et al. 2025. The effects of generative AI agents and scaffolding on enhancing students' comprehension of visual learning analytics. Computers & Education.

[46] Lineman, J. P.; Sweet, M. M.; and Sutton, F. 2025. Beyond content: Leveraging AI and metacognitive strategies for transformative learning in higher education. Transnational Journal of Business.

[47] Chen, K.; Tallant, A. C.; and Selig, I. 2025. Exploring generative AI literacy in higher education. Information and Learning Sciences.

[48] Zhao, Y.; Yue, Y.; Sun, Z.; Jiang, Q.; and Li, G. 2025. Does generative artificial intelligence improve students' higher-order thinking? Journal of Intelligence.

[49] Stanford Center for Teaching and Learning. 2025. AI teaching strategies.

[50] University of Virginia Pressbooks. 2024. Critical evaluation of AI outputs.

[51] UNESCO. 2023. Guidance for generative AI in education and research.

[52] Kmainasi, M. B.; Khan, R.; Shahroor, A. E.; Bendou, B.; Hasanain, M.; and Alam, F. 2024. Native vs non-native language prompting: A comparative analysis. arXiv.

[53] Tao, Y., et al. 2024. Cultural bias and cultural alignment of large language models. PNAS Nexus 3(9):346.

[54] Weidinger, L., et al. 2023. Biases in large language models: Origins, inventory, and mitigation. ACM Computing Surveys.

[55] Warren, T. 2025, February 25. Microsoft makes Copilot Voice and Think Deeper free with unlimited use. The Verge.

[56] Navigli, R.; Conia, S.; and Ross, B. 2023. Biases in large language models: Origins, inventory, and discussion. Journal of Data and Information Quality 15(2):1-21.

[57] Langer, M.; Baum, K.; and Schlicker, N. 2025. Effective human oversight of AI-based systems. Minds and Machines.

[58] Tsapara, I.; D'Amico, A. 2026. Designing Assessments for the New AI-Era. ASEE

Biographies

Irene Tsapara, National University & TILOS NSF-sponsored AI-Institute, Chicago, Illinois

Dr. Irene Tsapara is an academic leader, researcher, and educator specializing in Data Science, Artificial Intelligence, and computational learning theory. She serves as the Academic Program Director for the Doctorate in Data Science at National University, overseeing curriculum design, research supervision, and program accreditation. She holds a faculty position at Northwestern University and is also a faculty member of the TILOS NSF-sponsored AI Institute. Dr. Tsapara holds a Ph.D. in Mathematics, specializing in Computational Learning Theory, Machine Learning, and Artificial Intelligence, from the University of Illinois, and has extensive experience in the Financial and Education Sectors. Her experience includes serving as an Investment Analyst at Hedge Fund Research and a Data Scientist at Goldman Sachs. Her current research focuses on integrating AI-assisted learning frameworks, including vibe coding, into Data Science and interdisciplinary education. She develops scalable curricular models that align AI literacy, statistical reasoning, and ethical governance with a foundation in mathematical rigor. Her work emphasizes the use of large language models (LLMs) in promoting conceptual understanding, transparency, and reproducibility in computational practice.

Danae Vassiliadis, M.S., is a Data & AI Team Lead at Accenture and a Teaching Assistant at the University of Chicago.

She earned her M.S. in Applied Data Science from the University of Chicago and her B.S. in Industrial Engineering and Management Sciences from Northwestern University. Her professional and academic interests span predictive analytics, AI-driven decision-making, the psychology of learning, and neuropsychology, with a particular focus on how data and AI can support human development, performance, and understanding.