2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION
Paper ID #50304

Faculty Governance by Design: A Low-Code Agentic Workflow for Curriculum Drafting

Dr. Xiaoguang Ma, University of Wisconsin - Platteville

Dr. Xiaoguang Ma is an Associate Professor in the Department of Electrical and Computer Engineering at the University of Wisconsin-Platteville, where he has taught undergraduate courses since joining in 2018. He earned his Ph.D. from Florida State University, an M.S. from the Chinese Academy of Sciences, and a B.S. from Tsinghua University. Prior to academia, Dr. Ma worked as a Communication Architect at ABB Inc., specializing in industrial communication protocols and cybersecurity, and holds a 2018 patent for "Parallel Redundancy Protocol over Wide Area Networks." His current research explores smart microgrid cybersecurity and blockchain applications for data security in agriculture, collaborating on projects like a cybersecurity testbed and dairy farm data protection at UW-Platteville's Pioneer Farm. Dr. Ma's work bridges education and innovation, advancing solutions for critical infrastructure and technology security.

Jing Wang

Jing Wang is an Instructional Designer with 19 years of experience developing scalable, data-driven learning programs across higher education and Fortune 500 companies. Currently an Instructional Designer at the Universities of Wisconsin System, she specializes in blending AI, analytics, and instructional systems design to create learner-centric solutions that deliver measurable impact. With a background in economics and a Master's Degree in Instructional Design from Florida State University, Jing bridges the gap between analytical strategy and innovative pedagogy. Her recent work emphasizes using AI automation to elevate digital accessibility and streamline the development of complex STEM courses, ultimately building future-ready learning ecosystems.

©American Society for Engineering Education, 2026

Faculty Governance by Design: A Low-Code Agentic Workflow for Curriculum Drafting

Abstract

Generative AI is increasingly used as an assistant for knowledge work, supporting brainstorming, drafting, revising, coding, and transforming ideas into usable artifacts. In engineering education, these capabilities create an opportunity to reduce faculty time spent producing instructional materials. At the same time, responsible use must satisfy competing constraints: maintaining instructional quality, avoiding hallucinated or biased content, demonstrating responsible AI use to students (e.g., traceable sourcing and explicit review), preserving faculty pedagogical judgment, and operating within real-world limits of time, cost, and privacy requirements.

This paper reports the implementation of the POSED system (Plan, Outline, Sources, Evaluation, Draft), an agentic, human-in-the-loop (HITL) curriculum-development workflow realized as a reproducible low-code system on the n8n platform. POSED uses visual orchestration and deterministic control frames around probabilistic AI generation calls, with pause-and-approve checkpoints that enforce faculty-led governance and preserve editable intermediate artifacts. Rather than treating generative AI as an autonomous content producer, POSED positions generative AI as a constrained collaborator embedded in a process-guided workflow approved by faculty.

POSED encodes multi-stage prompting, artifact handoffs, and governance gates into an executable pipeline so instructors do not need to manually coordinate prompts, intermediate files, and model configurations across tools. A central systems contribution is treating model choice and settings as first-class workflow decisions: POSED supports modular routing across heterogeneous models and tools to balance capability, time efficiency, privacy/IP constraints, and operational (token/API) cost. In the evaluation runs reported here, most generation used cloud LLM APIs, while workflow execution (n8n) and retrieval storage (local Qdrant) were self-hosted; optional local-model execution is supported for privacy-sensitive steps.

This design also addresses a reliability tension rooted in LLM sampling: more exploratory generation can improve idea exploration and fluent drafting while increasing the risk of unverified details; more conservative settings can reduce unsupported elaboration but still do not guarantee correctness. Accordingly, POSED treats reliability as a workflow property enforced by deterministic control logic around probabilistic generation: retrieval-augmented generation (RAG) is grounded in instructor-approved sources, structured critique is applied before final drafting, and HITL approval gates ensure scope, structure, sourcing, and final outputs remain faculty-controlled and traceable. POSED also preserves teaching style through a versioned

professor-persona artifact that can accumulate refinements across runs, reducing homogenization without shifting the instructor into post-hoc "editor only" labor.

We implemented POSED as an n8n-based workflow with more than 200 nodes and evaluated it by generating lecture slides and instructor notes across course contexts in Electrical and Computer Engineering and Computer Science. Within this scope, the instructor's active labor decreased from an estimated 10–20 hours per lecture for manual preparation of new or substantially revised materials to approximately 2 hours of active review and edits, with a marginal computational AI cost under $1.50 per lecture.

We further report a horizontal comparison against three realistic direct-generation baselines (ChatGPT, Gemini Gems, and NotebookLM) and a vertical comparison across Aug. to Dec. 2025 that illustrates capability drift and motivates why workflow-level governance remains necessary even as models improve.

1 Introduction

Curriculum development in engineering education is a high-skill, time-consuming activity that blends content expertise with instructional design and iterative refinement of explanations, examples, visuals, and assessments. Established approaches such as ADDIE [1] and backward design [2] provide a unified structure to align goals, instruction, and assessment. However, instructors still invest substantial efforts to organize materials, curate examples, verify technical accuracy, and keep content current, particularly in fast-evolving areas such as Electrical and Computer Engineering and Computer Science, where tools, standards, and best practices change rapidly. Empirical studies of higher-education teaching and course development repeatedly identify time as a major constraint, especially for technology-mediated courses, where preparation demands can be substantial [3, 4].

Recent advances in generative artificial intelligence (GenAI), particularly large language models (LLMs), have accelerated interest and practice in AI-assisted educational design. In everyday use, faculty can now prompt commercially available systems to produce polished artifacts end-to-end, for example, lecture outlines, slides, instructor notes, rubrics, coding examples, diagrams, and even interactive web-based demonstrations. LLMs can also quickly iterate on different versions, add explanations (novice vs. advanced), generate multiple example sets, refactor materials for different teaching plans, and translate content into alternative modalities. As a result, faculty use of GenAI is increasingly framed not only as workload reduction, but as a way to accelerate iteration, explore alternatives, and prototype instructional assets under real constraints of time and attention [5–7].

At the same time, the tooling landscape is shifting toward integrated, end-to-end experiences that compress multi-step work into a single interface. For example, Google NotebookLM can generate slide decks grounded in notebook sources, and Gemini can generate slide presentations with self-defined Gems. These "one-stop" LLM generations reduce friction for producing high-quality first drafts and increasingly support direct prompt-to-artifact creation for many routine instructional assets.

However, integrating LLMs into curriculum design is not a straightforward automation process.

In high-stakes instructional contexts, the central question is reliability and accountability: can AI assistance be adopted in a way that preserves faculty pedagogical judgment, supports verifiable sourcing, and demonstrates responsible AI use to students?

One widely documented limitation is hallucination. LLMs can generate fluent but ungrounded statements, fabricate citations, or introduce subtle technical errors [8, 9]. In curriculum development, these errors are especially problematic because, even when the overall narrative is pedagogically useful, a small number of ungrounded claims can propagate misconceptions and erode trust in course materials. Hallucinations are also difficult to detect through a quick read; they often appear as confident, well-structured prose. This behavior is a byproduct of probabilistic generation and can persist even under conservative generation settings. In practice, mitigating hallucination typically requires an explicit grounding strategy (e.g., retrieval-augmented generation (RAG) over curated sources), a sourcing discipline that emphasizes vetted references, and deliberate review and critique steps before materials are finalized [10].

The second concern is model bias: LLM output can encode or amplify biases presented in training data, which is especially problematic when generating examples, scenarios, or evaluation criteria that shape student experience and belonging [11].

Thirdly, these technical risks interact with human factors. In curriculum development, LLM tools are often utilized as consulting and drafting systems to propose outlines, examples, explanations, and collect references. Many instructors may be tempted to accept these outputs with insufficient verification, especially under time pressure. This form of overreliance is consistent with broader findings on automation bias, where people defer to automated suggestions even when they are imperfect [12]. More concerningly, empirical evidence suggests that heavy reliance on LLM assistants during writing tasks can produce cognitive offloading and accumulated cognitive debt, including reduced engagement during composition and weaker recall or ownership of self-produced content [13]. While the study is reported as a preprint and should be interpreted with appropriate caution, it reinforces a key educational concern: if instructors adopt GenAI primarily as a convenience layer that collapses planning, sourcing, and synthesis into a single step, they risk shifting from active curriculum design to passive post-hoc editing, potentially weakening the expert judgment they aim to model for students.

A fourth concern is review reliability when faculty are presented with finished-looking AI outputs too early. ADDIE [1] and backward design principles [2] implicitly rely on a routine in which instructors validate instructional intent and alignment before investing in delivery polish. In contrast, highly polished, multimedia-rich drafts can trigger a “Wow” impression and shift attention toward presentation quality and plausible structure, making it easier to overlook deeper issues such as weak objective-assessment alignment, gaps in scaffolding, or unsupported technical claims. For this reason, a safer workflow is to lock high-level intent and structure first, then develop detailed content and examples, and only then add multimedia and presentation polish after the underlying logic and sourcing have been validated.

Together, these concerns argue against a purely chat-first approach where faculty generate finished materials and then attempt to correct problems after the fact. They also motivate workflow designs that keep instructors engaged in the same staged thinking routine used in instructional design, with explicit checkpoints that make review deliberate rather than reactive.

Recent engineering-education applications also suggest a trajectory from AI-assisted content generation toward more governed and discipline-aware implementations. Prior work has explored agentic AI for active learning in computer networks education [14], interdisciplinary integration of generative AI across Computer science, Electrical and Computer Engineering, and Economics instruction [15], and the PROTECT framework for preserving academic integrity in AI-era project-based learning [16]. Taken together, these studies suggest that while AI can support content creation, personalization, and engagement, effective educational use still depends on explicit instructional scaffolds, human oversight, and governance mechanisms. The present paper extends that line of work by focusing specifically on a faculty-controlled workflow for curriculum drafting, with auditable intermediate artifacts, source validation, and retrieval-grounded generation.

Beyond process design, implementation must contend with technical realities that strongly influence reliability, privacy, and cost. A practical complication is that "LLM" is not a single capability. Models differ in reasoning ability, context length, multimodal capability, response time, and cost, and they can exhibit different error profiles depending on the task [9]. This matters operationally because curriculum work mixes creative generation (examples, scenarios, visual prompts) with accuracy-critical synthesis (definitions, citations, and technical correctness). As a result, a single-model approach is rarely optimal; multi-model routing allows different subtasks to be assigned based on capability needs, context limits, error profiles, and operational cost.

In addition, privacy and data governance constraints in education are non-trivial. Course development may include sensitive information (student data, unpublished assessments, proprietary materials), and learning ecosystems can integrate numerous third-party integrations, raising concerns about data harvesting, compliance, and intellectual property [17]. These realities motivate scenarios in which sensitive inputs are processed locally or within institution-controlled infrastructure, while non-sensitive tasks can use cloud services. However, local execution often relies on open-weight models, which can lag frontier cloud models on complex tasks [18].

Multi-model routing helps match subtasks to capability, privacy constraints, and cost, but it does not eliminate the core reliability problem. Hallucination can persist across models and even under conservative generation settings, so improving trustworthiness requires controls that ground outputs and force explicit checking rather than relying on model choice alone. Two families of techniques are especially relevant to improve trustworthiness in AI-assisted curriculum generation. First, RAG grounds outputs in retrieved references by conditioning generation on a curated corpus rather than on parametric memory alone [10]. Second, verification and critique loops (e.g., structured self-checking or multi-agent review patterns) can reduce unforced errors by requiring explicit checks before finalizing artifacts (though they do not eliminate hallucination and can increase cost if applied indiscriminately) [19, 20]. From a systems perspective, these methods are most effective when embedded before irreversible synthesis steps rather than used only as post-hoc cleanup. Lastly, practical limits such as context window size can cause goal and requirement drift across long interactions, which further requires intermediate artifacts (plan, outline, vetted sources, critique notes) that persist state across stages, support targeted human edits, and avoid restarting from the beginning when revisions are needed.

To address the limits of ad hoc prompting and chat-centric tooling, we implement POSED (Plan, Outline, Sources, Evaluation, Draft), a human-in-the-loop workflow for curriculum artifact

generation. POSED frames curriculum development as a systems engineering problem: preserve intent, control risk, and keep decisions auditable through intermediate artifacts and enforced review gates.

In POSED, instructional design intent is captured explicitly in plan and outline artifacts [1, 2]. Drafting is constrained by sourcing-first RAG so outputs are grounded in instructor-approved references. We implement POSED as a low-code workflow to keep the process transparent and modifiable without requiring faculty to build complex software systems [21–23].

The remainder of the paper is organized as follows. Section 2 describes the POSED workflow architecture and low-code implementation. Section 3 evaluates POSED in faculty use (scope-limited to lecture slides and instructor notes), including a stage-by-stage artifact trace, a horizontal comparison against direct-generation tools available in Dec. 2025, and a vertical comparison across Aug.–Dec. 2025 POSED-flow generated presentations. Section 4 summarizes contributions, limitations, and future work.

2 POSED System Implementation

POSED operationalizes curriculum development as a governed workflow rather than a one-shot AI generation step. We implement POSED in a low-code orchestration environment (n8n) not primarily as "easier to build," but as *high-inspectability*: the node-based workflow externalizes the control logic so faculty can see, audit, and modify where sourcing happens, where checks occur, and where human approvals are required. In this sense, the workflow functions as an inspectable "pedagogical circuit" that makes governance decisions explicit without requiring faculty to implement a bespoke software system. In our implementation, execution is self-hosted and paired with a local vector database (Qdrant) for retrieval-grounded drafting, with optional local LLM execution available for privacy-sensitive steps.

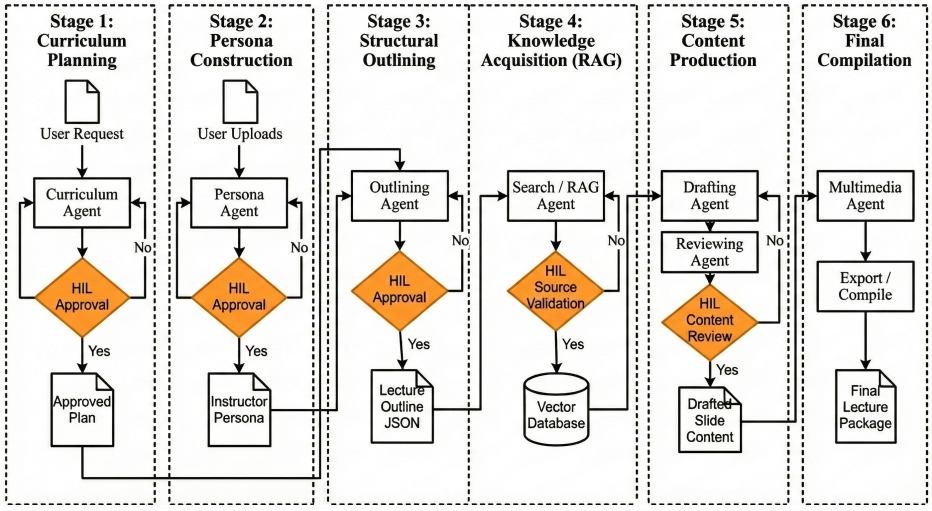


Figure 1: High-level POSED architecture with six stages and mandatory HITL gates.

2.1 Six-stage Workflow Implementation

Operationally, POSED is instantiated as a six-stage pipeline with mandatory pause-and-approve gates. Each stage produces concrete artifacts, such as an approved plan, an instructor persona, an approved outline, an instructor-vetted source set for retrieval, and a reviewed slide deck with speaker notes, that are persisted for traceability, editability, and reuse.

Stages 1–4 establish the holistic curriculum structure at a planning level (including pre-lecture activities, in-class activities, and assessment/project elements captured in the plan/outline artifacts), while in this paper, Stages 5–6 implement the draft–review loop and final export assembly for the delivered artifacts (slides and speaker notes). The artifact schemas are extensible, so future stages can deepen and operationalize the planned elements as dedicated subroutines (e.g., pre-lecture activities, in-class activities, assessments, or projects) without changing the earlier governance structure. Figure 1 summarizes the stage structure, and Figures 2–8 show the corresponding n8n subflows exported from the working system.

2.1.1 Stage 1: Curriculum Planning (Planner Agent + Scope Gate)

Stage 1 collects the minimum context needed to bound generation and produces a planning artifact (`curriculum_plan.json`). The workflow starts with a structured form (e.g., topic, audience level, prerequisites, lecture length, and module purpose). A Planner agent then performs background retrieval (e.g., web search) and proposes a draft plan that includes a rationale, candidate student learning outcomes (SLOs), and a time-aware list of subtopics.

Table 1: Stage 1 planning artifact (curriculum plan) fields and instructor scope-gate actions.

Field group (artifact)	What it contains (examples)	Instructor action at scope gate
Session context	Topic, audience level, prerequisites, duration, module purpose, run/session metadata	Confirm or edit constraints that bound downstream generation
Learning objectives (SLOs)	AI-proposed objectives plus any faculty-added objectives; each marked selected/not selected	Select, revise wording, add/remove objectives
Subtopics and timing	Candidate subtopics with suggested time allocation, category/role, short description, and justification	Select the set and sequence; adjust timing and emphasis
References per subtopic	Initial links associated with each subtopic (used later for sourcing/RAG)	Keep/replace/remove links; add instructor-preferred sources
Faculty decision prompts	Explicit "decision points" raised by the system (e.g., depth tradeoffs, optional lab choices)	Choose options that affect scope and teaching moves

Stage 1 ends with a mandatory scope gate implemented as a HITL form. The instructor selects, edits, or removes proposed SLOs and subtopics (and can add missing items) before the workflow proceeds. The approved plan is saved as a machine-readable JSON artifact for downstream stages; because it is a standalone file, it can also be revised outside POSED using a text/code editor when needed. This gate implements a backward-design check before any slide drafting begins [2]. Table 1 summarizes the main fields surfaced in this planning artifact and the corresponding instructor actions.

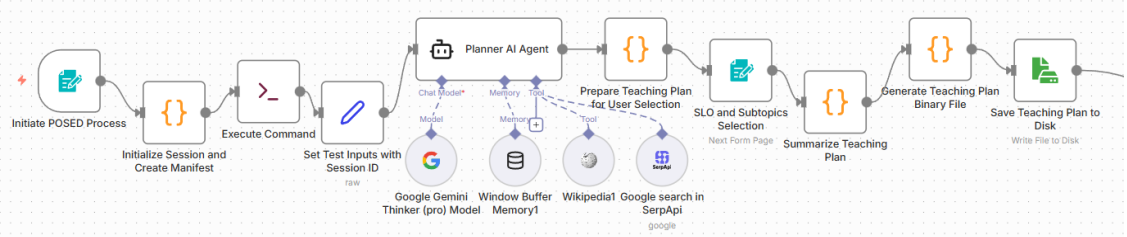


Figure 2: Stage 1 implementation (n8n): session initialization, Planner AI Agent, HITL SLO/subtopic scope gate, and plan persistence.

2.1.2 Stage 2: Instructor Persona Construction (Persona Agent + Identity Gate)

Stage 2 produces a versioned `professor_persona.md` artifact that captures teaching style constraints (tone, pacing, preferred examples, and classroom interaction patterns) for downstream drafting. The workflow optionally ingests instructor-provided materials (e.g., prior lecture notes, exemplar slides/notes, or other teaching documents). It then selects the best available starting profile using a simple precedence rule: if a professor-specific persona already exists, it becomes the base and only new evidence is merged in; otherwise the workflow falls back to a department/college "common profile" template; if neither is available, it defaults to an internal template.

The Persona Agent synthesizes a draft persona that is consistent with the approved curriculum plan and the uploaded evidence. A mandatory HITL Identity Gate (form-based) then requires the instructor to approve, edit, or regenerate the persona before it can be used by later stages. The approved persona is saved as a new version of `professor_persona.md`, enabling cumulative refinement across runs: future executions can incorporate additional uploaded exemplars and instructor edits through the version history, improving personalization through richer conditioning inputs rather than any model retraining or parameter updates.

2.1.3 Stage 3: Structural Outlining (Outlining Agent + Outline Gate)

Stage 3 produces the teaching outline as a structured artifact, `lecture_outline.json`. We use JSON for two reasons. First, it remains human-readable for offline instructor review and edits. Second, downstream code nodes can reliably consume stable fields (e.g., ordered subtopics, ordered slides within each subtopic, and per-slide metadata such as slide number, title, required key points, and optional visual placeholders). In this implementation, the outline also captures non-slide instructional elements (e.g., pre-lecture activities, in-class activities, assessments, and projects) as part of the planned lesson structure; what remains future work is implementing dedicated subroutines that expand these high-level items into fully developed artifacts.

A mandatory HITL Outline Review gate supports approve-as-is, targeted revision, regeneration, or an instructor-uploaded override (replacing the JSON artifact). This keeps sequencing and cognitive-load decisions instructor-controlled while ensuring the approved structure propagates cleanly into Stage 4 and 5 [1, 24].

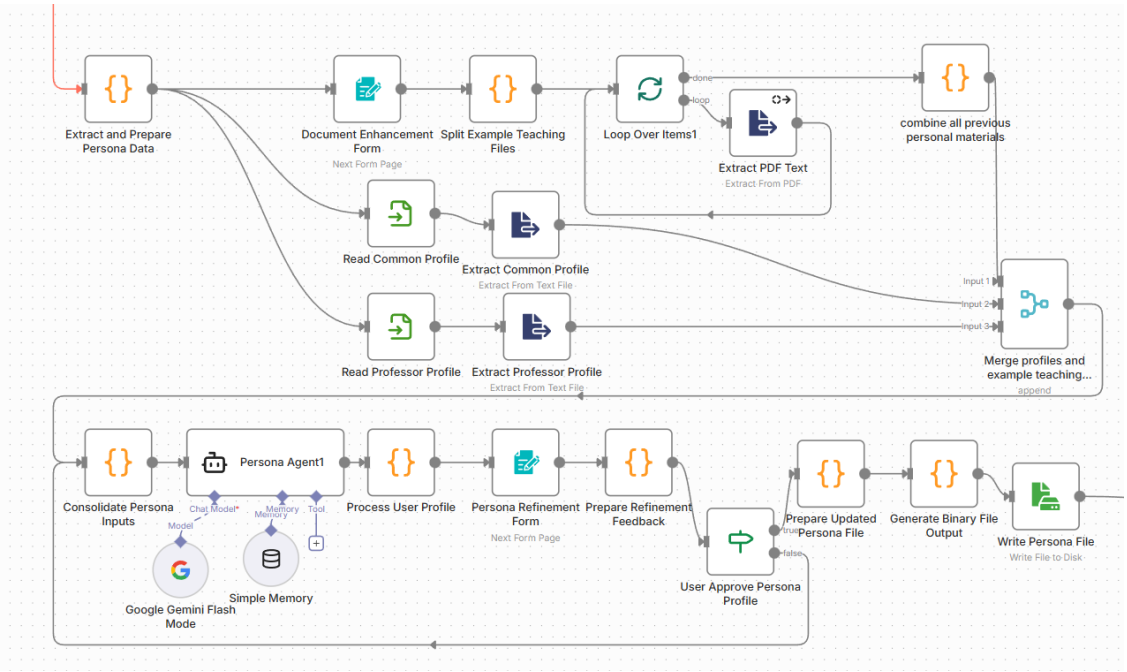


Figure 3: Stage 2 implementation: document ingestion, persona synthesis, and HITL persona refinement/approval gate.

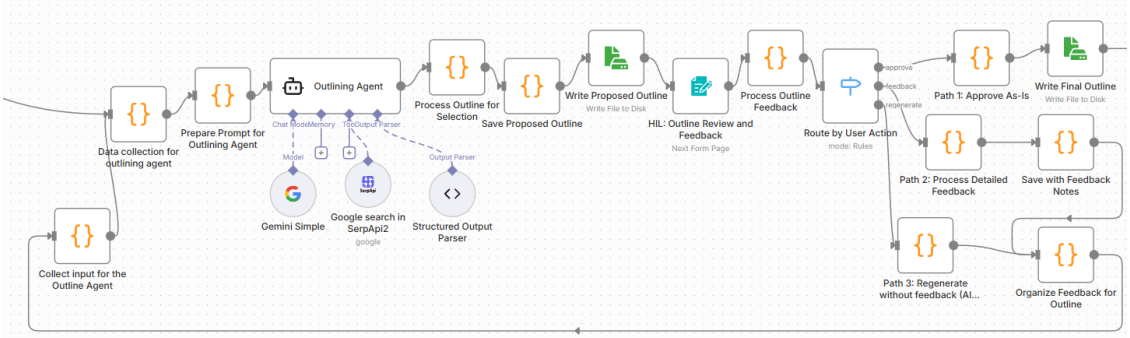


Figure 4: Stage 3 implementation: outlining agent, output parsing to Json, and HITL review.

2.1.4 Stage 4: Knowledge Acquisition and Retrieval Indexing (Resource Search + Source Validation + Qdrant)

Stage 4 constructs a lecture-specific knowledge base to ground drafting. A Resource Search Agent generates targeted queries from the approved outline and collects candidate URLs and documents. The Resource Review form is a mandatory HITL gate: instructors validate relevance and reject unsuitable sources before ingestion.

Approved sources are then scraped/downloaded with a content router (PDF vs. HTML), analyzed for extraction quality, chunked, embedded, and stored in a local vector database (Qdrant) [10]. In this implementation, retrieval grounding is not an optional enhancement. Drafting is constrained to retrieve only from the vetted corpus, improving traceability and reducing the risk of ungrounded claims [10].

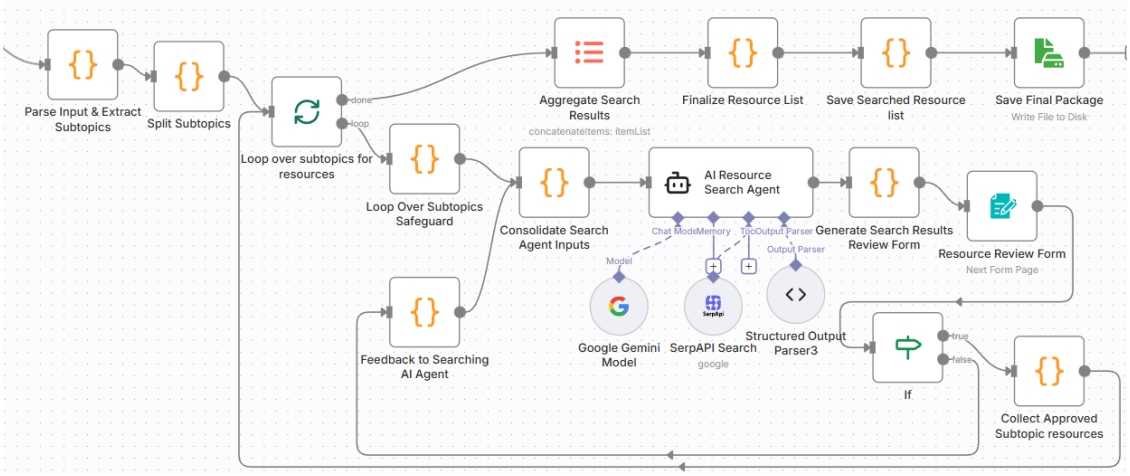


Figure 5: Stage 4A: outline-to-query generation, Source search, and HITL source validation gate.

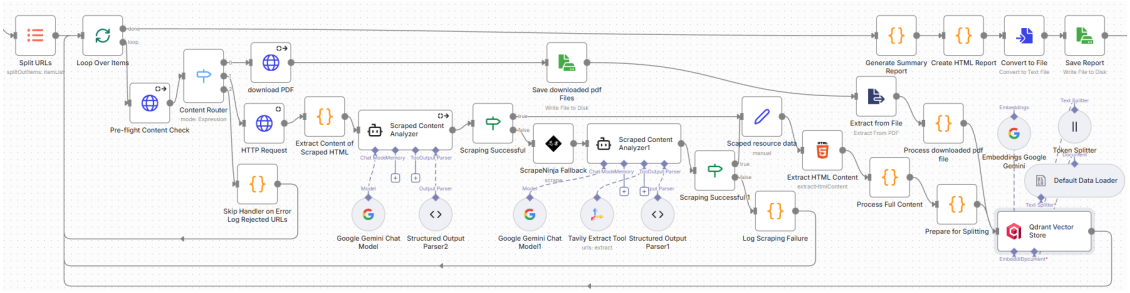


Figure 6: Stage 4B: approved-source scraping/download routing, extraction analysis, chunking/embedding, and Qdrant ingestion for lecture-specific retrieval.

2.1.5 Stages 5 and 6: Draft/Review Loop and Export Assembly (Reviewer + HITL Content Approval + Multimedia + Compile)

Stages 5 and 6 implement the production loop and final artifact assembly. In Stage 5, the workflow iterates through outline items, retrieves context from Qdrant, drafts slide content and

speaker notes, and invokes an Editor/Reviewer agent for rubric-like critique and consistency checks. The workflow then pauses at a HITL content approval gate with bounded iteration (e.g., stop at approval or a maximum number of revisions), reducing the risk of automation bias by requiring explicit faculty judgment before progression [12].

Stage 6 assembles the final deliverables. The HTML presentation assembly subflow reads the generated slide/note drafts, integrates multimedia outputs, and compiles downloadable artifacts. Multimedia generation is implemented as a distinct agentic subflow that decides whether to download images, generate charts from retrieved data, or generate an illustration prompt for an image model, and then saves and embeds assets during final HTML construction.

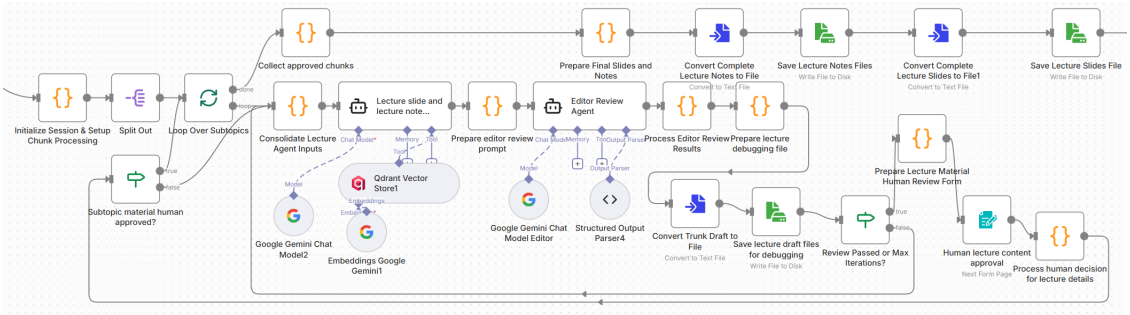


Figure 7: Stage 5 implementation: retrieve (Qdrant) → draft slides/notes → reviewer agent → HITL content approval with bounded iterations.

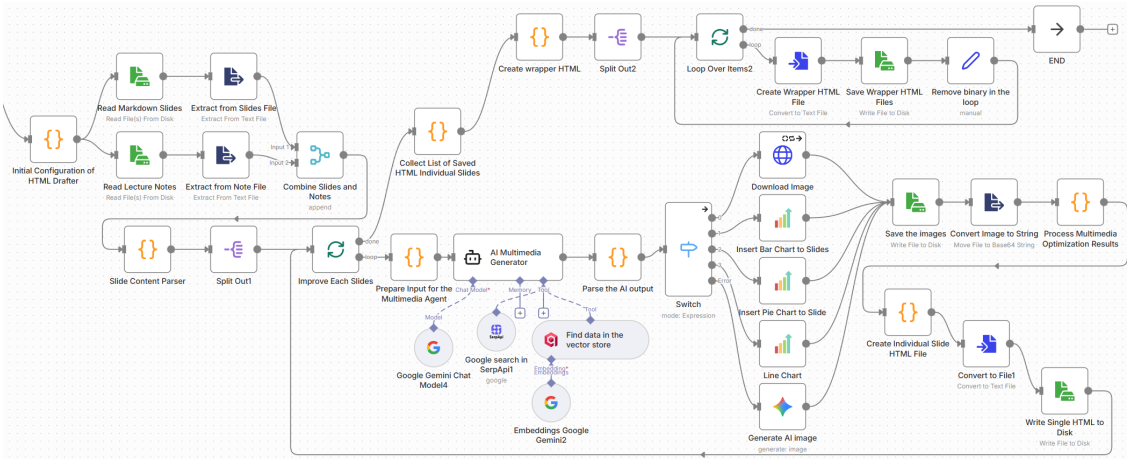


Figure 8: Stage 6 implementation: multimedia decision logic (download/charts/image generation), asset embedding, HTML wrapper creation, and export packaging.

2.2 Low-code Orchestration and Platform Selection

We implemented POSED as a low-code workflow to make the process executable, inspectable, and modifiable by faculty without requiring full-stack software development. The n8n platform was selected primarily because it supports self-hosting and complex branching/iteration in a visual workflow environment. Beyond engineering convenience, the visual workflow externalizes

governance decisions (where sourcing happens, where checks occur, and where instructors approve), making the system less of a black box and supporting reflective teaching practice and auditability.

At the workflow level, POSED's HITL gates map cleanly to n8n's form node. n8n also supports traceability through execution history: past runs and intermediate node outputs can be inspected to debug errors and audit decision points. Sub-workflows can be invoked as reusable components, which helps keep POSED stages (e.g., retrieval, critique loops, export assembly) as separate, testable modules.

For comparison, widely-used automation platforms differ substantially in deployment and governance affordances. We selected n8n because it meets three requirements that matter for governed curriculum generation and are rare in SaaS-first tools: (i) institution-controlled execution (self-hosting via Docker, with local network access), (ii) native support for long, branching workflows with iteration (needed for stage gates, retries, and bounded review loops), and (iii) inspectability and audit trails at the level of individual steps (node-level inputs/outputs and execution history), which makes the workflow debuggable and reviewable rather than a black box. These capabilities are what allow POSED to implement pause-and-approve gates, enforce sourcing-before-drafting, and preserve intermediate artifacts without requiring custom full-stack development.

In comparison with other popular agentic AI platforms, Zapier is positioned as a cloud-first automation platform and is not designed for consistent on-premises or hybrid orchestration; Make provides an enterprise on-premises agent that enables access to on-network systems while execution remains orchestrated by the platform; and Power Automate supports hybrid integration through the on-premises data gateway to connect cloud flows to on-premises resources. In our implementation context, none of these platforms satisfied the primary requirements of this project as effectively as n8n.

2.3 Cross-cutting Implementation Choices

Given the selected orchestration platform, POSED uses four recurring implementation patterns: (i) explicit HITL gates, (ii) retrieval grounding via a lecture-specific vector store, (iii) heterogeneous model routing under privacy constraints, and (iv) model sampling treated as workflow-level controls.

2.3.1 HITL Gates Ensure Faculty Governance

In n8n, HITL review can be implemented through multiple channels, including messaging platforms (e.g., Telegram or Discord) or email-based review, where the workflow notifies the instructor when an artifact is ready. In this implementation, we use n8n Forms for HITL gates because they provide immediate, in-context feedback: the instructor can review the generated artifact and respond without switching applications. This choice also fits a developer-and-user workflow, where the author monitors runs closely and uses the gate as a natural breakpoint for debugging when an artifact is produced.

The Form-based approach has limitations. Because the workflow waits for form interaction, it

does not provide the same asynchronous notification that email or messaging channels can offer. In addition, rich HTML rendering is limited in form displays, which constrains how much formatting can be used in review screens. For future deployments with multiple instructors or longer-running jobs, these gates can be moved to email or messaging without changing POSED stage logic; the core requirement is that each stage produces a reviewable and editable artifact, and the workflow halts until the instructor explicitly approves or revises it.

2.3.2 RAG-Based Draft Generation

Retrieval grounding is implemented with a lecture-specific corpus stored in Qdrant. In POSED, the RAG corpus is not a fixed background collection. For each lecture module, the system first searches online for candidate webpages and PDF documents aligned with the approved outline. The instructor then reviews that candidate list and may approve, reject, replace, or manually add sources. Stage 4B then performs acquisition, extraction, chunking, embedding, and ingestion of only the instructor-approved sources into the vector database [10]. Stage 5 retrieves relevant chunks for each outline item and conditions drafting on the retrieved context. This split makes evidence acquisition auditable and keeps drafting dependent on the vetted corpus rather than on open-ended model recall [10].

2.3.3 Heterogeneous Model Routing and Data Sovereignty

A practical lesson from implementation is that "one model for everything" is rarely optimal. Models vary in reasoning capability, latency/cost, multimodal support, and the tendency to introduce unsupported details [9]. POSED therefore supports heterogeneous routing so that sensitive inputs and internal documents can be handled locally or within institution-controlled infrastructure, while capability-intensive subtasks can be routed to cloud models when appropriate. In the evaluation runs reported in this paper, POSED used cloud LLM APIs (Google Gemini API) for most generation steps, while the workflow execution (n8n) and retrieval store (local Qdrant) were self-hosted. Local LLM execution, through ollama, is supported as an optional path for privacy-sensitive steps.

Reliability is treated as a workflow property rather than a model property. Drafting remains grounded through retrieval from instructor-approved sources [10], and HITL approval gates constrain where and how outputs can advance. This partitioning aligns with privacy and governance concerns in educational ecosystems [17, 25]. It also acknowledges that open-weight/local deployments may require additional engineering to approach frontier performance on complex tasks, which further motivates workflow-level controls rather than assuming any single model will perform well across all subtasks [18].

2.3.4 Model Settings as Workflow Controls

Within n8n, model sampling settings (e.g., temperature and related parameters) are configured at the LLM node used by each AI agent. In POSED, these settings are treated as workflow controls rather than global defaults. Stages that prioritize reliability and tight adherence to constraints (e.g., outlining, critique, or citation-sensitive drafting) use more conservative settings, while stages that benefit from variation (e.g., brainstorming examples or illustration prompts) can use

more exploratory settings. This design reinforces the broader principle that deterministic workflow control should manage how and where probabilistic generation is used.

Table 2: Illustrative model-setting intent by subtask in POSED. Ranges are practical starting points and should be tuned per model or vendor; reliability-critical stages still rely on RAG grounding and HITL review.

Subtask	**Primary goal**	**Sampling intent**	**Temperature**	**Top-p**
Planning / idea expansion	Broaden options and generate alternatives	Exploratory	0.7–1.0	0.9–1.0
Outlining (structure)	Stable sequencing under consistent constraints	Conservative	0.2–0.5	0.7–0.9
Source query formulation	Focused and precise retrieval targets	Highly conservative	0.0–0.3	0.6–0.9
Drafting with retrieved evidence	Faithful synthesis from vetted sources	Conservative	0.2–0.5	0.7–0.9
Reviewer / critique	Detect inconsistencies and enforce rubric checks	Highly conservative	0.0–0.3	0.6–0.9
Multimedia prompt generation	Creative variation within explicit constraints	Exploratory	0.7–1.0	0.9–1.0

Temperature primarily controls how much randomness the model injects when choosing the next token, while top-p limits sampling to the smallest set of high-probability tokens whose cumulative probability exceeds p. In practice, lowering these values tends to produce more consistent outputs, but still does not guarantee correctness. For Gemini 3 models in our implementation, temperature was set to 1 in accordance with the current Google API requirement.

3 Evaluation

This section evaluates POSED as a faculty-controlled workflow for drafting teachable lecture artifacts (slides and instructor notes). The focus is on whether the POSED system can generate usable materials while preserving instructor control through HITL approval gates and editable intermediate artifacts.

We first document the intermediate outputs produced at each POSED stage for a lecture module on the topic of "AI hardware," developed for a sophomore-level Logic and Digital Design course in Dec. 2025. The unit of analysis for this lecture module includes (1) a teaching plan, (2) the professor persona file, (3) a lecture outline, (4) an instructor-approved reference list, (5) a slide deck in markdown format, (6) presentation speaker notes in markdown format, and (7) the complete presentation deck in HTML format. We then conduct a horizontal comparison between the materials generated by the POSED flow and three direct-prompt-generation baselines via different LLMs available in Dec. 2025, i.e., ChatGPT, Gemini Gems, and NotebookLM. To demonstrate the rapid advance of LLM capability, we present a vertical comparison using an earlier set of generated lecture modules on the topic of "Agentic AI," prepared for an AI literacy course in Aug. 2025, highlighting how outputs changed over only four months. Finally, we

summarize operational cost and model routing from provider billing logs to show the usage of different models.

Beyond documenting artifacts, baseline outputs, and cost traces, this evaluation also makes the review criteria explicit. Because POSED is designed as a governed instructional workflow rather than a one-shot content generator, the assessment should account not only for final output quality, but also for pedagogical usefulness, source traceability, and the degree of instructor control preserved throughout the process.

Accordingly, in addition to the artifact trace, horizontal comparison, and provider-cost analysis, our evaluation uses structured expert review of the generated instructional materials. Specifically, the second author, an experienced instructional designer, assessed the retained Dec. 2025 outputs across the comparison dimensions summarized in Table 3, including material completeness, pedagogical specificity, readiness-to-teach, traceability/governance artifacts, visual support, and instructor control surface. This expert review is complemented by POSED's internal AI review log, which serves as a pre-instructor screening artifact by flagging unsupported claims and likely hallucinations and by critiquing organization, pacing, and identifying missing explanations. We treat the AI review log as a workflow aid rather than as ground truth.

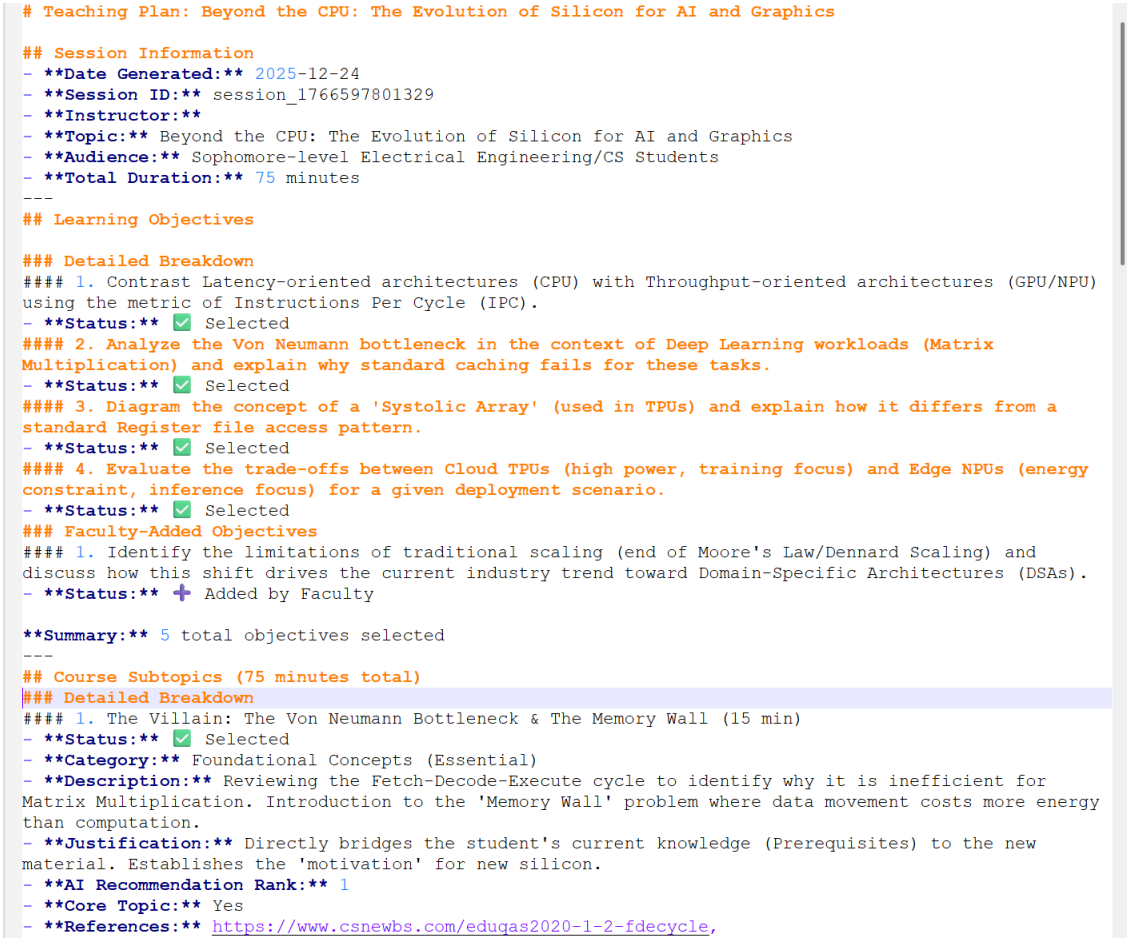
Prior work on the quality evaluation of AI-generated educational resources has emphasized dimensions such as authenticity, accuracy, legitimacy, and relevance [26]. However, we do not adopt that framework directly here. Instead, we use an expanded rubric that is more suitable for POSED's governed, artifact-centered workflow. Our evaluation extends beyond content quality alone to include source traceability, pedagogical specificity, readiness-to-teach, instructor control surface, and revision burden. This broader framing is appropriate because POSED already embeds a preliminary fact-checking and critique layer through instructor-approved retrieval grounding and an AI review log that screens for unsupported claims, likely hallucinations, and pedagogical weaknesses before instructor review.

3.1 POSED Artifacts by Workflow Stage

A core design intent of POSED is that teaching artifacts are generated through a governed sequence rather than a single-shot "make me slides" prompt. In this subsection, we show representative artifacts produced by POSED at each stage for the lecture materials on the topic of "AI hardware," generated in Dec. 2025.

Plan (teaching plan before persona). In POSED, planning comes first. The Plan stage produces a teaching plan that is primarily about disciplinary intent and scope—topic framing, prerequisites, learning objectives, and a proposed module structure. At this point, the goal is to lock down what students should learn and what constraints the lecture must satisfy, independent of any particular instructor style. Figure 9 shows the resulting teaching plan artifact reviewed and approved before outlining. At the end of the learning objectives section, a faculty-added learning objective is inserted during review (marked "Added by Faculty"), illustrating that the Plan gate supports instructor modification rather than only approve/reject decisions.

After the plan is approved, POSED generates a separate instructor persona artifact that captures delivery preferences (tone, pacing, example style, and interaction cues) to guide how the same



```
# Teaching Plan: Beyond the CPU: The Evolution of Silicon for AI and Graphics

## Session Information
- **Date Generated:** 2025-12-24
- **Session ID:** session_1766597801329
- **Instructor:**
- **Topic:** Beyond the CPU: The Evolution of Silicon for AI and Graphics
- **Audience:** Sophomore-level Electrical Engineering/CS Students
- **Total Duration:** 75 minutes
---
## Learning Objectives

### Detailed Breakdown
#### 1. Contrast Latency-oriented architectures (CPU) with Throughput-oriented architectures (GPU/NPU)
using the metric of Instructions Per Cycle (IPC).
- **Status:** ✅ Selected
#### 2. Analyze the Von Neumann bottleneck in the context of Deep Learning workloads (Matrix
Multiplication) and explain why standard caching fails for these tasks.
- **Status:** ✅ Selected
#### 3. Diagram the concept of a 'Systolic Array' (used in TPUs) and explain how it differs from a
standard Register file access pattern.
- **Status:** ✅ Selected
#### 4. Evaluate the trade-offs between Cloud TPUs (high power, training focus) and Edge NPUs (energy
constraint, inference focus) for a given deployment scenario.
- **Status:** ✅ Selected
### Faculty-Added Objectives
#### 1. Identify the limitations of traditional scaling (end of Moore's Law/Dennard Scaling) and
discuss how this shift drives the current industry trend toward Domain-Specific Architectures (DSAs).
- **Status:** ➕ Added by Faculty

**Summary:** 5 total objectives selected
---
## Course Subtopics (75 minutes total)
### Detailed Breakdown
#### 1. The Villain: The Von Neumann Bottleneck & The Memory Wall (15 min)
- **Status:** ✅ Selected
- **Category:** Foundational Concepts (Essential)
- **Description:** Reviewing the Fetch-Decode-Execute cycle to identify why it is inefficient for
Matrix Multiplication. Introduction to the 'Memory Wall' problem where data movement costs more energy
than computation.
- **Justification:** Directly bridges the student's current knowledge (Prerequisites) to the new
material. Establishes the 'motivation' for new silicon.
- **AI Recommendation Rank:** 1
- **Core Topic:** Yes
- **References:** https://www.csnewbs.com/eduqas2020-1-2-fdecycle,
```

Figure 9: POSED Plan artifact: teaching plan with prerequisites, learning objectives, and module structure approved before outlining and drafting. The artifact also records a faculty-added objective ("Added by Faculty"), illustrating instructor modification at the Plan gate.

content is taught, rather than what the content is. Figure 10 shows the persona artifact used downstream to shape narrative choices and classroom moves (with the instructor name removed for review).

```
Version: 1

### **Comprehensive Persona Profile: Dr. xxxx**

#### **Executive Summary**
Dr. xxxx is a former industry Lead Engineer and Communication Architect who bridges the gap between hardware and software through a
pragmatic, "binary-world" pedagogy. He focuses on technical mastery, industry-aligned proficiency, and the responsible integration of
 AI to develop career-ready engineers.

#### **Synthesized Pedagogical Style:**
Dr. xxx employs a "Technology Tree" approach, contextualizing digital logic as the essential bridge between analog circuits and
high-level software engineering. His teaching is intensely hands-on and iterative, utilizing the "7 Times Rule" and structured
frameworks for content creation. He transitions students from theoretical Boolean algebra to practical application—such as soldering
and circuit debugging—using AI tools (Gemini, ChatGPT, Copilot) as controlled utilities. This methodology emphasizes
"learning-by-doing," where students move from elementary logic gates to complex sequential state machines through constant coding and
 physical prototyping.

#### **Synthesized Assessment Methods:**
Assessment is holistic and performance-driven, prioritizing both theoretical rigor and functional implementation. A strict
requirement exists where students must maintain a high exam average (typically >70%) to pass, ensuring fundamental competency. The
grading structure balances high-stakes exams (45%) with a significant final project (15%) and intensive lab work (20%). Evaluations
frequently require students to demonstrate technical autonomy, involving oral defenses and live demonstrations of physical systems,
such as a sensor-driven robotic car.

#### **Core Educational Values:**
*   **Industrial Relevance & Technical Mastery:** Prioritizing "Lead Engineer" standards, including soldering, debugging, and
hardware-software synthesis.
*   **Responsible AI Integration:** Cultivating critical AI literacy and prompt engineering while upholding strict academic
integrity.
*   **Intellectual & Creative Autonomy:** Empowering students to move from passive learners to active creators who can direct AI and
hardware for specific real-world outcomes.
*   **Practical Experiential Learning:** Ensuring every theoretical concept is reinforced through lab-intensive application and
iterative project development.
```

Figure 10: POSED Persona artifact: instructor delivery preferences used downstream to shape pacing, examples, and teaching style (faculty name removed for review purpose).

Outline (structure that becomes the slide map). The Outline stage translates the approved teaching plan into a structured lecture blueprint that drives downstream generation. In POSED, the outline is the instructor's primary control surface for sequencing, cognitive load, and narrative flow. Importantly, it is not limited to slide structure: the same outline artifact already captures planned elements of a more holistic module (e.g., pre-lecture tasks, in-class activities, and assessment/project intentions) as part of the instructional design record.

In the current implementation, our evaluation focuses on the slide and speaker-notes artifacts produced from the approved outline. Figure 11 shows a representative outline artifact, and Figure 12 shows the table-of-contents/navigation view of the final presentation that mirrors the approved outline and supports lecture flow.

While the outline already specifies broader learning elements at a planning level, we have not yet implemented dedicated subroutines that expand each non-slide element into a fully realized standalone artifact (e.g., ready-to-deploy pre-lecture worksheets, in-class activity sheets, or assessment items). This extension is planned future work.

Sources (traceable grounding before drafting). In the Sources stage, POSED turns an approved outline into a traceable retrieval database before any drafting occurs. The workflow first searches each outline topic and returns a short list of candidate links (e.g., top results per subtopic) for instructor review. The instructor then approves, rejects, or replaces links, and may add additional sources manually to reflect course expectations and local context. After approval, the POSED flow scrapes the selected sources, extracts text from webpages or downloaded PDF

```
{
  "status": "proposed",
  "generated_at": "2025-12-25T17:42:41.178Z",
  "instructor": "xxxxx",
  "session_info": {
    "session_id": "session_1766684239257",
    "topic": "Beyond the CPU: The Evolution of Silicon for AI and Graphics",
    "audience": "Sophomore-level Electrical Engineering/CS Students"
  },
  "outline": {
    "module_title": "Beyond the CPU: The Evolution of Silicon for AI and Graphics",
    "pre_lecture_activity": {
      "activity_title": "The Energy Audit: Compute vs. Memory",
      "description": "Students are tasked with researching and comparing the energy cost (in Picojoules) of a single 32-bit Floating Point Addition versus a single DRAM memory access. They will submit a brief calculation showing why moving data across a motherboard is more 'expensive' than the actual computation, setting the stage for the 'Memory Wall' discussion. Estimated time: 30 minutes.",
      "alignment_with_persona": "This activity uses the 'Binary-World' approach to ground students in the physical reality of hardware before discussing high-level AI concepts."
    },
    "presentation_outline": [
      {
        "subtopic_title": "The Villain: The Von Neumann Bottleneck & The Memory Wall",
        "slides": [
          {
            "slide_number": 1,
            "title": "The Classical Architecture: A Review",
            "key_points": [
              "Review of the classical Von Neumann model (CPU/Memory separation).",
              "The 'Shared Bus' limitation: One instruction/data at a time.",
              "How this architecture served general computing for 50 years."
            ],
            "visual_idea": "Diagram showing the CPU and Memory connected by a narrow bottleneck bus."
          },
```


Figure 11: POSED Outline artifact: structured module outline that includes the slide map (titles, sequencing, and planned teaching moves), pre-lecture activities, and other planned instructional elements.

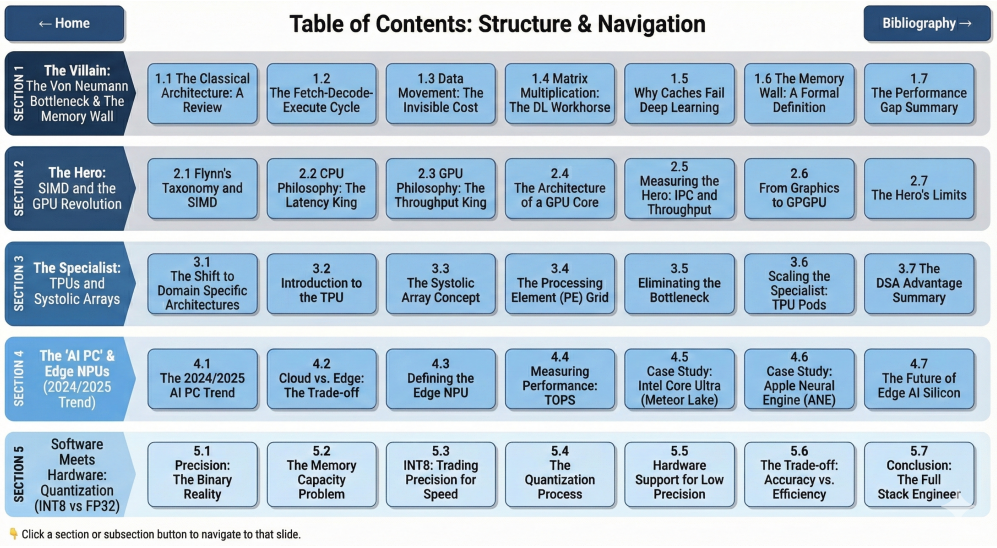


Figure 12: POSED navigation artifact: table of contents of the final slides mirroring the approved outline to support lecture flow.

files, and vectorizes the resulting chunks in a local Qdrant vector database. Subsequent drafting uses RAG against this instructor-approved Qdrant collection, so slide and note generation is grounded in a verifiable set of sources rather than ad hoc model recall. Figure 13 shows the first page of the sourcing report artifact produced at this stage.

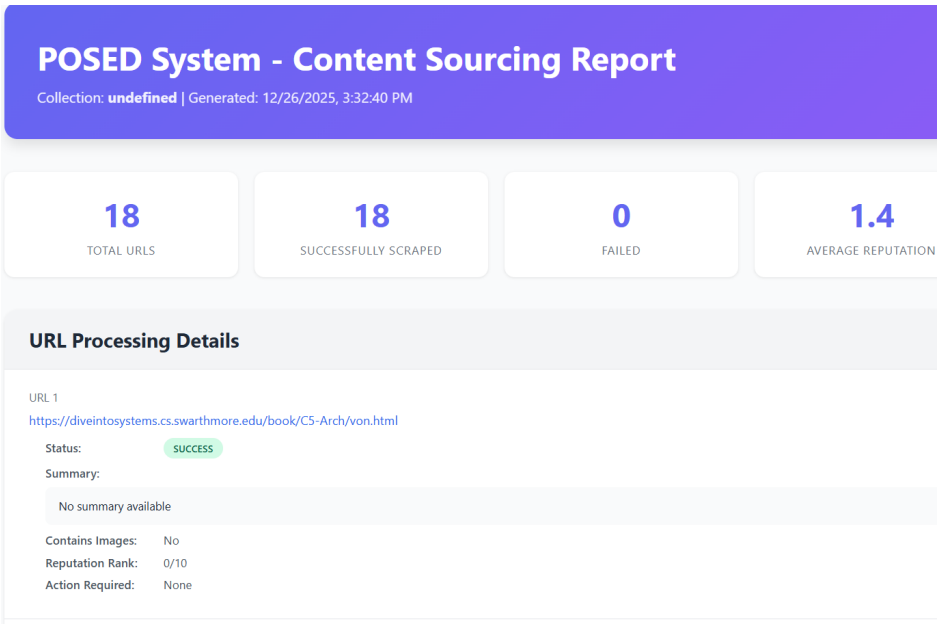


Figure 13: POSED Sources artifact: traceable source trail surfaced for instructor review prior to drafting. In this iteration, instructor verification is treated as the primary reliability mechanism.

Draft (slides + instructor notes + AI review log gate). The Draft stage produces the slide deck and speaker notes in markdown format, and then assembles the final HTML presentation. A distinguishing feature in the POSED output is that instructor notes are treated as first-class artifacts (not an afterthought), supporting teachability rather than only visual polish.

During drafting, POSED also generates an AI review log as a system artifact (Figure 14). This review log serves as a gatekeeper and as feedback for the instructor. First, it automatically checks for reliability issues (e.g., unsupported claims or likely hallucinations) before the material reaches the instructor. Second, it critiques the drafted slides and notes from a pedagogical perspective (organization, structure, pacing, missing explanations). In the current implementation, the workflow only passes the materials to the instructor's review when the review score exceeds 85/100. The log is not treated as ground truth; it is a structured, traceable critique intended to reduce superficial review and keep the instructor's attention on high-impact issues.

After the instructor approves the drafted artifacts, the workflow produces a complete HTML presentation with navigation, and supports charts/multimedia as the last assembly step. The retained markdown artifacts (slides and notes) are also reusable: they can be fed into other emerging presentation tools, e.g., NotebookLM, as model capabilities evolve, without redoing the earlier plan/outline/sourcing work. Figure 15 shows a representative POSED generated "AI hardware" slide with integrated instructor notes access and navigation buttons.

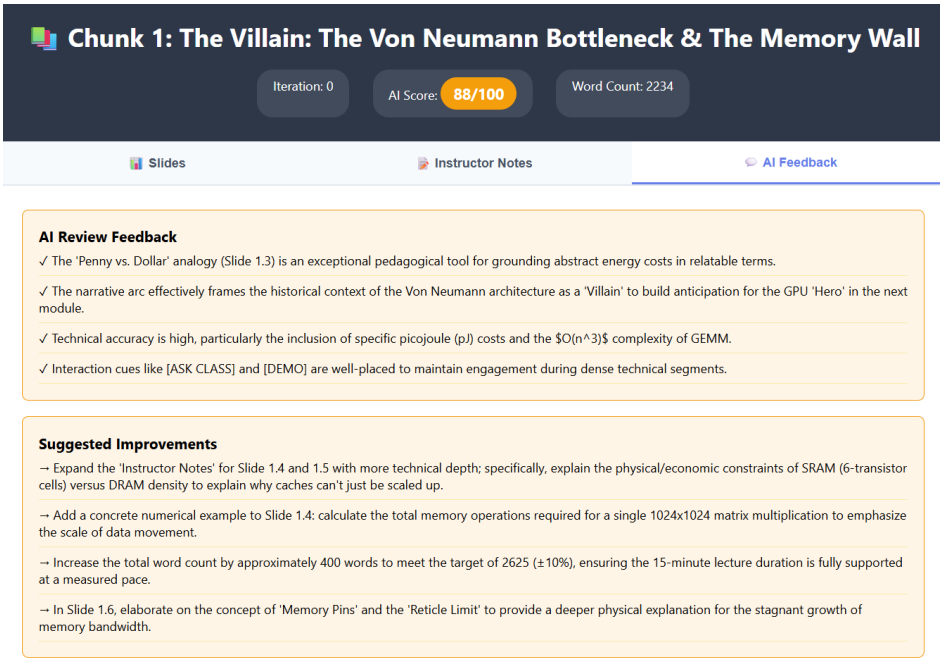


Figure 14: POSED Draft-stage system log: AI review artifact used as an automatic gatekeeper and as structured feedback on reliability and teachability before instructor review.

3.2 POSED vs. Direct Generation Tools Horizontal Comparison: Creating "AI Hardware" Lecture

We compare POSED outputs to three direct-generation baselines that reflect realistic faculty "fast path" workflows: (1) ChatGPT presentation generation producing a PPTX slide deck; (2) Gemini Gems SlideDeck producing an HTML output that, in our sample, resembled a continuous document/outline rather than a slide-oriented deck; and (3) NotebookLM presentation generation using a Gemini-produced draft as input, producing a PDF slide deck. These alternatives are fast and often visually polished, but they typically expose only the end product and do not preserve POSED-style upstream governance artifacts (e.g., plan/outline approval gates and an instructor-vetted retrieval corpus) as reviewable deliverables.

Figures 16, 17, and 18 show representative outputs from each baseline. In this sample, the tools differed not only in visual appearance, but also in the role they implicitly played for the instructor: POSED functioned as an instructional architect with performance-oriented notes, ChatGPT as a drafter that summarizes content into bullets, Gemini Gems as a topic summarizer with minimal pedagogical scaffolding, and NotebookLM as a visual designer that packages content into a polished format.

A major practical difference across tools is editability: PPTX (by ChatGPT) is straightforward to revise, PDFs (by NotebookLM) are costly to modify (even with conversion tools), and HTML outputs (by Gemini and POSED system) are editable but require comfort with web layout. In this sample, NotebookLM produced the most visually polished deck but with low editability due to PDF output; POSED was next because its pipeline supports figure generation and maintains

Figure 15: Representative POSED Draft output (Dec. 2025 “AI hardware” module): slide styling, integrated navigation, and instructor-notes access designed for “ready-to-teach” use.

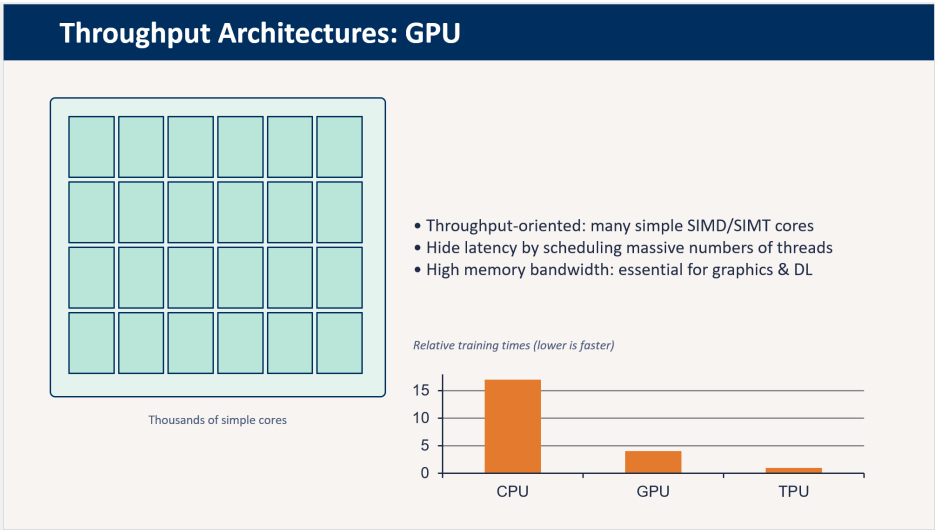


Figure 16: Direct generation baseline (ChatGPT): representative slide from the Dec. 2025 "AI hardware" prompt.

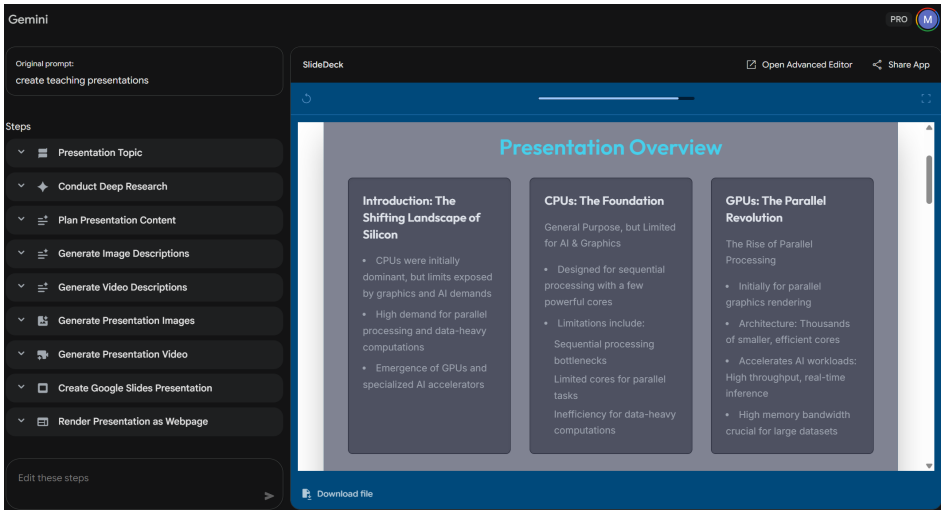


Figure 17: Direct generation baseline (Gemini Gems SlideDeck): representative output from the same Dec. 2025 hardware prompt. In our sample, the output was text-heavy and outline-like (closer to a continuous document than a slide deck), and it lacked meaningful visuals.

Table 3: Horizontal comparison (Dec. 2025 "AI hardware" module): POSED vs. direct-generation baselines.

Criterion	POSED (HTML + notes)	ChatGPT (PPTX)	Gemini Gems (HTML)	NotebookLM (PDF)
Output artifact format(s) and editability	HTML presentation with Markdown source; slides and notes are segmented into modular units for revision; HTML layout edits require web skills	PPTX deck; high editability in standard slide editors; tends toward bullet-point-heavy formatting	HTML/Doc outline; output is a continuous text document rather than a slide deck; structural changes require significant manual reformatting	PDF deck; low editability; modifications typically require conversion tools or manual recreation
Material completeness for a full lecture	High (plan $\rightarrow$ outline $\rightarrow$ slides $\rightarrow$ notes), with explicit figure statements and bibliographic trail	Medium (slides produced quickly, but lacks upstream plan/outline artifacts and separate notes depth)	Low–Medium (high-level topic coverage without slide-by-slide breakdown, examples, or depth)	Medium (visually complete deck with narrative, but lacks POSED-style instructor notes and lesson-plan artifacts)
Pedagogical specificity (checks, cues, pacing)	High; includes delivery cues (e.g., interaction prompts and demos) aligned to instructor persona	Medium; accurate facts but limited scaffolding and few interaction prompts	Low; encyclopedic descriptions with minimal instructional structure or cues	Medium; strong narrative structure, but limited active-learning prompts or classroom delivery cues
Readiness-to-teach (notes + navigation)	High; instructor notes function as a teaching script supporting delivery decisions	Medium; notes often restate slide text and require instructor reconstruction	Low; no instructor notes and substantial rewrite required to be teachable	Medium; visually projection-ready, but lack of a separate teaching script limits immediate use
Traceability and governance artifacts	High; upstream review gates and retrieval grounding occur before drafting; sources are vetted via RAG	Low; primarily end-product only; citations may appear without upstream verification trail	Low; no citations or source links in our sample, limiting verifiability	Low–Medium; includes a bibliography, but lacks an outline gate and upstream artifact trail
Visual support (figures and diagrams)	Medium–High; visuals are defined via figure statements and can be data-grounded with citations	Medium; includes generic chart placeholders without detailed figure logic	Low; no meaningful visuals generated in our sample	High; most visually cohesive diagrams and layout in our sample, though figures may be illustrative rather than data-cited
Instructor control surface (where edits occur)	Upstream; plan/outline gates reduce downstream rework	Downstream; edits happen on the final slides	Downstream; requires rewriting the content structure from scratch	Downstream; revise deck and reconstruct notes
Role assumed by AI tool	Instructional architect that structures process and delivery artifacts	Content drafter that summarizes topics into slides	Topic summarizer that lists facts without slide-level structure	Visual designer that packages content into a polished deck

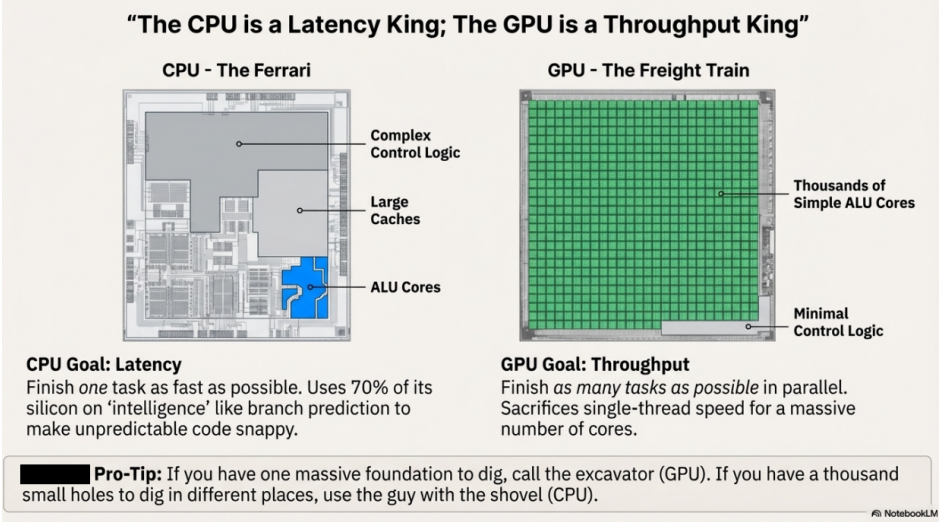


Figure 18: Hybrid baseline (Gemini draft → NotebookLM slides): representative output. In our sample, this pipeline produced the most visually cohesive deck, but it still lacked POSED's upstream governance artifacts and separate instructor notes.

editable Markdown sources for slides and notes; ChatGPT's output was less visually cohesive; and Gemini Gems produced no meaningful visuals in our run. (We report these as observations tied to the specific tool versions and settings used in Dec. 2025 rather than as a universal ranking).

3.3 Vertical Comparison of POSED Outputs: Aug. to Dec. 2025

A practical challenge in Agentic AI workflows is capability drift: the same workflow can yield different artifacts as models and toolchains evolve. To assess domain agility and workflow maturation under this drift, we compare two POSED runs authored by the same instructor: Aug. 2025 ("Agentic AI," a qualitative/conceptual domain) and Dec. 2025 ("AI Hardware & Silicon," a quantitative, physics-grounded domain). Across both runs, POSED's governance scaffold remained unchanged, yet the resulting instructional artifacts differ in ways that reflect maturation of the workflow's outputs rather than a change in oversight structure.

First, the visual specification shifted from concept-oriented illustrations (Fig. 19) toward more data-grounded figures in Dec. 2025 (Fig. 15). In Aug. 2025, visuals were frequently framed as metaphors intended to clarify abstract ideas. By Dec. 2025, the workflow more often produced explicit figure requests that specify chart type, variables, and citation targets (e.g., energy-per-operation comparisons and efficiency benchmarks anchored to instructor-provided references). This does not eliminate hallucination risks, but it increases verifiability by making visual intent and sourcing explicit and reviewable before inclusion.

Second, instructor notes evolved from primarily explanatory text into more delivery-oriented guidance. Aug. 2025 notes tended to read as conversational explanations of slide content, whereas Dec. 2025 notes more often included pacing and classroom-management cues (e.g., prompts to pause, run a short demonstration, or elicit a quick calculation). In effect, the workflow shifted

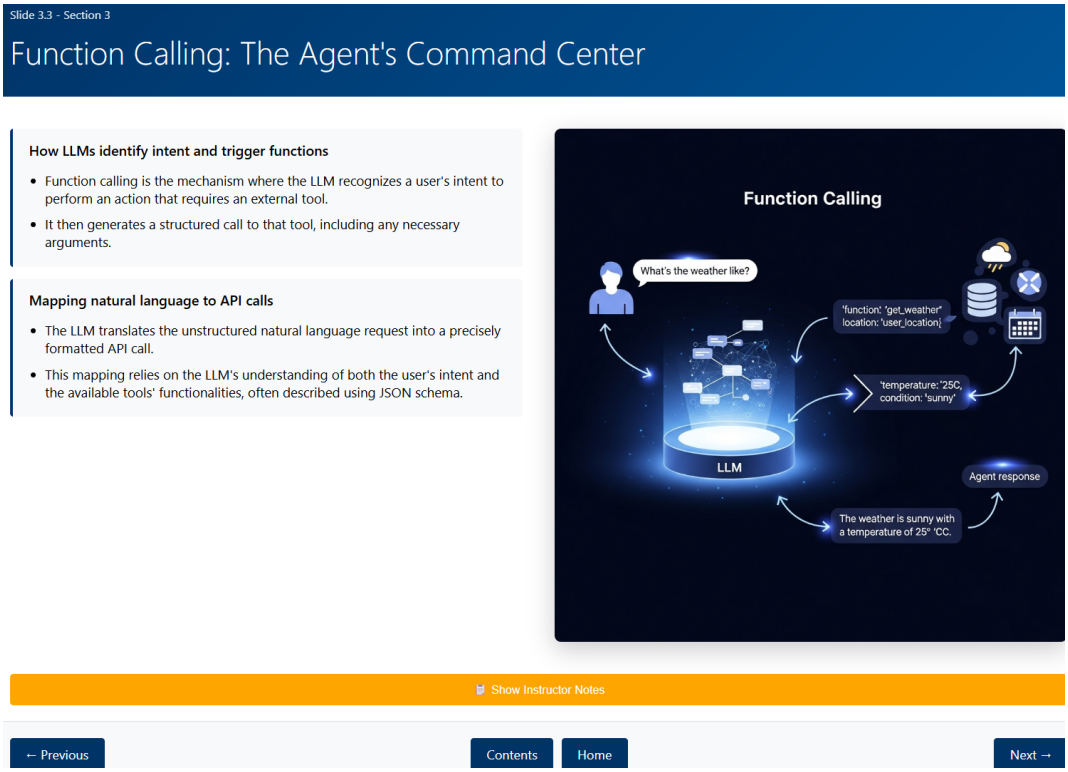


Figure 19: Aug. 2025 POSED output showing concept-oriented visual framing.

from producing only “what to say” toward also supporting “how to teach it”.

Finally, the Dec. 2025 artifacts showed tighter anchoring through domain-appropriate analogies. Hardware constraints and architectural tradeoffs were mapped to concrete student experience (e.g., throughput vs. latency contrasts and memory-movement bottlenecks described through physical-world metaphors), improving accessibility without sacrificing technical rigor.

This longitudinal comparison does not attribute changes to any single model update. Instead, it supports POSED’s design premise: deterministic workflow control and faculty approval gates provide a stable governance scaffold, allowing instructors to benefit from evolving model and tool capabilities while preserving traceability and instructional ownership.

3.4 Operational Cost and Model Routing (Provider Billing Logs)

In addition to artifact quality and governance, we recorded operational AI cost from provider billing dashboards. For the Dec. 2025 “AI hardware” lecture module, provider-billed usage totaled $1.22 over the execution window (Dec. 24–30, 2025); the Aug. 2025 “Agentic AI” module totaled $1.35. We report these as per-module marginal computational costs and exclude fixed infrastructure costs, since POSED ran on existing faculty equipment.

Figure 20 summarizes Dec. 2025 input-token usage by model. To make the billing logs stage-interpretable, we deliberately executed the six POSED stages across three separate days: Day 1 generated the teaching plan and instructor persona; Day 2 ran outlining, sourcing, and the

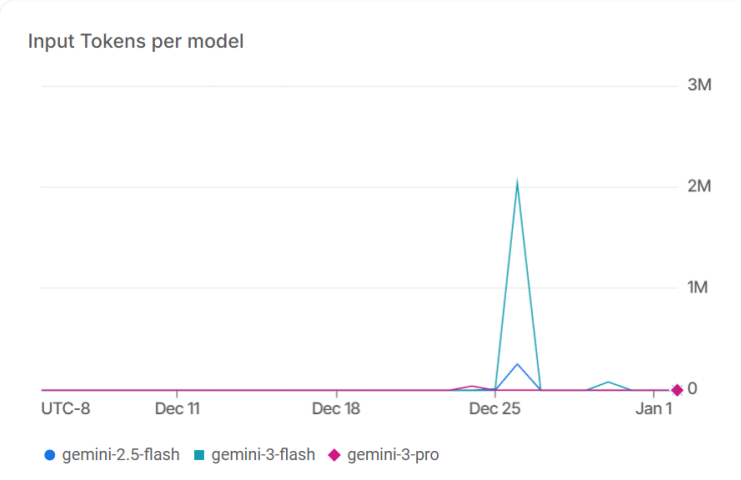


Figure 20: Provider dashboard view of input-token usage by model for the Dec. 2025 AI hardware lecture module. POSED routes subroutines across models to balance capability, faithful drafting from the approved RAG corpus, and cost.

primary drafting passes; and Day 3 executed multimedia generation and final HTML assembly. This staging makes the usage trace easier to read: direction-setting calls concentrate early, bulk drafting dominates the middle window, and the remaining activity reflects final assembly and multimedia steps.

The same plot also reflects deliberate model routing by subroutine intent. Higher-reasoning calls are kept narrow for direction-setting steps (planning and constraint-setting) that are immediately externalized into reviewable artifacts. Most outlining and drafting is routed to faster "direct-answer" models because the goal is faithful realization from the instructor-approved outline and vetted RAG corpus, which helps limit ungrounded elaboration; lower marginal cost is an additional benefit of this routing.

In our billing dashboard for this run, output-token charges were not itemized separately; given the workflow's high input-to-output ratio, any output-token component was small relative to the total and is included in the provider-reported $1.22.

Table 4: Dec. run: input-token totals by model and primary POSED usage role.

Model	**Input tokens**	**Primary role in POSED run**
gemini-3-pro	40.51K	Planning requiring higher reasoning capacity (initial synthesis and constraint-setting).
gemini-2.5-flash	258.38K	Persona analysis and other fast intermediate transformations where "good enough" quality is sufficient.
gemini-3-flash	2.047M	Outlining, drafting, and reviewing.
Total	**2.346M**	Sum of input tokens shown in the dashboard for the Dec. 2025 run.

3.5 Limitations

This evaluation is scope-limited. Results reflect two lecture modules generated by a single instructor across different topics and a descriptive, artifact-centered comparison to direct-generation tools available in Dec. 2025. The comparative assessment summarized in Table 3 adds structured expert review by the second author, but cross-instructor replication was not available during the present study period. Accordingly, the reported results should be interpreted as a single-instructor evaluation with structured expert review, not yet as evidence of broader instructor-to-instructor reproducibility.

In addition, the vertical comparison highlights capability drift: model and platform changes over short time spans can alter drafting behavior and visual quality, which is why this paper emphasizes workflow governance and traceable intermediate artifacts as the stable contribution rather than any fixed ranking of tools. Future work will expand to additional instructors and course contexts, standardize scoring rubrics across tools, collect structured instructor and student feedback, and evaluate automated source-quality signals only after aligning them with faculty trust requirements.

4 Summary and Future Work

This paper reports the implementation and evaluation of the POSED system, a faculty-governed, low-code agentic workflow for curriculum generation. Rather than using a single prompt-to-slides step, POSED treats curriculum development as a staged engineering process: each stage produces an editable artifact and pauses at a HITL gate before the workflow can proceed. In two lecture-module runs, POSED produced ready-to-teach slides and instructor notes while preserving a traceable artifact trail and reducing rework by moving review to earlier, lower-cost decision points (plan, outline, and sourcing) instead of post-hoc slide repair.

Our horizontal comparison with direct-generation tools shows a consistent trade-off: prompt-to-deck systems can be fast and visually polished, but they typically collapse the workflow into an end product with limited upstream governance, and their editability depends strongly on output format (PPTX vs. HTML vs. PDF). In contrast, POSED's advantage is not a single "best deck", but controlled convergence—faculty can steer scope and structure early, verify grounding via an instructor-vetted retrieval corpus, and then draft from that evidence with review logs and notes that support delivery.

The vertical comparison across Aug. 2025 and Dec. 2025 illustrates capability drift over short time spans. As models and image pipelines improved, the resulting artifacts became more visually coherent and better aligned to engineering expectations (e.g., more data-grounded figures and more delivery-oriented notes). However, these improvements do not remove the need for workflow-level controls: faculty-owned sourcing, structure, and approval remain the mechanisms that make the process reliable and accountable as tool behavior changes.

Future work extends POSED in three directions. First, we will complete the curriculum scope already present in the plan and outline artifacts by adding dedicated subroutines that expand non-slide elements into standalone, ready-to-deploy materials (pre-lecture activities, in-class activity sheets, assessments, and project scaffolds) under the same artifact-and-gate governance.

Second, we will collect structured feedback from instructors and students to refine agent behaviors and critique rubrics, so improvements are driven by authentic classroom use rather than ad hoc prompt tuning. Third, to support personalization without sacrificing privacy, we will add student-profile artifacts and route privacy-sensitive personalization through local models, while keeping non-sensitive generation modular and auditable.

Several engineering refinements are also immediate. We will introduce explicit alignment artifacts that map learning outcomes to slides, activities, and assessments to make review more consistent with instructional design practice. We will also formalize model-routing policy as an editable artifact (capability targets, local vs. cloud constraints, and cost limits), and expand export paths from POSED's Markdown-first sources so new presentation generators (e.g., NotebookLM-style tools) can be adopted without losing the approval gates and audit trail that make the workflow accountable.

References

[1] R. M. Branch, *Instructional Design: The ADDIE Approach*. Springer, 2009.

[2] G. Wiggins and J. McTighe, *Understanding by Design*, 2nd ed. Association for Supervision and Curriculum Development (ASCD), 2005.

[3] B. J. Mandernach, S. W. Hudson, and S. Wise, "Where has the time gone? faculty activities and time commitments in the online classroom," *Journal of Educators Online*, vol. 10, no. 2, 2013. [Online]. Available: https://files.eric.ed.gov/fulltext/EJ1020180.pdf

[4] R. Van de Vord and K. Pogue, "Teaching time investment: Does online really take more time than face-to-face?" *The International Review of Research in Open and Distributed Learning*, vol. 13, no. 3, 2012. [Online]. Available: https://www.irrodl.org/index.php/irrodl/article/view/1190

[5] E. Kasneci, K. Sessler, S. Küchemann *et al.*, "Chatgpt for good? opportunities and challenges of large language models for education," *Learning and Individual Differences*, vol. 103, p. 102274, 2023.

[6] UNESCO, "Guidance for generative AI in education and research," United Nations Educational, Scientific and Cultural Organization, 2023. [Online]. Available: https://unesdoc.unesco.org/ark:/48223/pf0000386693/PDF/386693eng.pdf.multi

[7] U.S. Department of Education, Office of Educational Technology, "Artificial intelligence and the future of teaching and learning: Insights and recommendations," U.S. Department of Education, Tech. Rep., 2023. [Online]. Available: https://tech.ed.gov/ai-future-of-teaching-and-learning/

[8] Z. Ji, N. Lee, R. Frieske *et al.*, "Survey of hallucination in natural language generation," arXiv preprint, 2022. [Online]. Available: https://arxiv.org/abs/2202.03629

[9] L. Huang, W. Yu, W. Ma, W. Zhong *et al.*, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," arXiv preprint, 2023. [Online]. Available: https://arxiv.org/abs/2311.05232

[10] P. Lewis, E. Perez, A. Piktus *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems*, 2020. [Online]. Available: https://arxiv.org/abs/2005.11401

[11] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Computing Surveys*, 2021.

[12] R. Parasuraman and D. H. Manzey, "Complacency and bias in human use of automation: An attentional integration," *Human Factors*, vol. 52, no. 3, pp. 381–410, 2010.

[13] N. Kosmyna *et al.*, "Your brain on ChatGPT: Accumulation of cognitive debt when using an AI assistant for essay writing task," arXiv preprint, 2025. [Online]. Available: https://arxiv.org/abs/2506.08872

[14] X. Ma and J. Wang, "Wip: Active learning through prompt engineering and agentic ai simulation-a pilot project in computer networks education," in *Proceedings of the 2024 IEEE Frontiers in Education Conference (FIE)*, 2024.

[15] X. Ma, Y. Wu, X. Liu, and J. Wang, "Integrative ai practices across disciplines: Innovations in computer science, electrical engineering, and economics education," in *Proceedings of the 2025 IEEE Frontiers in Education Conference (FIE)*, 2025.

[16] X. Ma, Y. Wu, M. Roopaei, and J. Wang, "Wip: Protect: A framework for preserving project-based learning integrity in the ai era," in *Proceedings of the 2025 IEEE Frontiers in Education Conference (FIE)*, 2025.

[17] M. R. Sanfilippo, N. Apthorpe, K. Brehm, and Y. Shvartzshnaider, "Privacy governance not included: analysis of third parties in learning management systems," *Information and Learning Sciences*, vol. 124, no. 9-10, pp. 326–348, 2023.

[18] G. Zhang, Q. Jin, Y. Zhou, S. Wang, B. R. Idnay, Y. Luo, E. Park, J. G. Nestor, M. E. Spotnitz, A. Soroush, T. R. Campion, Z. Lu, C. Weng, and Y. Peng, "Closing the gap between open source and commercial large language models for medical evidence summarization," *npj Digital Medicine*, vol. 7, no. 1, p. 239, 2024.

[19] S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, and J. Weston, "Chain-of-verification reduces hallucination in large language models," in *Findings of the Association for Computational Linguistics: ACL 2024*. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 3563–3578. [Online]. Available: https://aclanthology.org/2024.findings-acl.212/

[20] N. Shinn, B. Labash, A. Gopinath *et al.*, "Reflexion: Language agents with verbal reinforcement learning," arXiv preprint, 2023. [Online]. Available: https://arxiv.org/abs/2303.11366

[21] D. Pinho, A. Aguiar, and V. Amaral, "What about the usability in low-code platforms? a systematic literature review," *Journal of Computer Languages*, vol. 74, p. 101185, 2023.

[22] M. O. Ajimati, N. Carroll, and M. Maher, "Adoption of low-code and no-code development: A systematic literature review and future research agenda," *Journal of Systems and Software*, vol. 222, p. 112300, 2025.

[23] J. T. Sodano and J. F. DeFranco, "Citizen development, low-code/no-code platforms, and the evolution of generative ai in software development," *Computer*, vol. 58, no. 5, pp. 101–104, 2025.

[24] M. T. H. Chi and R. Wylie, "The ICAP framework: Linking cognitive engagement to active learning outcomes," *Educational Psychologist*, vol. 49, no. 4, pp. 219–243, 2014.

[25] R. Marachi and L. Quill, "The case of canvas: Longitudinal datafication through learning management systems," *Teaching in Higher Education*, pp. 418–434, 2020.

[26] Q. Huang, C. Lv, L. Lu, and S. Tu, "Evaluating the quality of ai-generated digital educational resources for university teaching and learning," *Systems*, vol. 13, no. 3, p. 174, 2025.