2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #50248

Designing Story-Driven and AI-Augmented Instruction: A Pedagogical Approach to Contextualizing Engineering Education. Work in Progress

Enas Aref, Bowling Green State University

Enas Aref is an Assistant Teaching Professor in the College of Engineering and Innovation at Bowling Green State University. Her work focuses on integrating artificial intelligence into engineering and management education. She develops structured, human-centered AI pedagogies that support critical thinking, student engagement, and ethical AI use. Her research spans STEM education, ergonomics, and AI-augmented instructional design.

Baqer Aljabr, The University of Toledo

https://sites.google.com/view/baqeraljabr/home?authuser=2

Dr. Muteb Mohammad Aljasem, Bowling Green State University
Christopher Kluse, Bowling Green State University

©American Society for Engineering Education, 2026

Designing Story-Driven and AI-Augmented Instruction: A Pedagogical Approach to Contextualizing Engineering Education

Abstract

This paper addresses the prevalence of decontextualized problem-solving in undergraduate engineering by integrating structured storytelling with generative artificial intelligence (AI). While narrative pedagogy fosters deep conceptual understanding, its implementation is often unstructured and labor-intensive. This paper solves these challenges by identifying a set of pedagogical characteristics of story-driven examples to ensure narrative rigor. To operationalize these characteristics, the study introduces a custom AI artifact designed to assist educators in generating high-fidelity, story-driven problems. The artifact utilizes a "Educator-in-the-Loop" architecture, allowing educators to input specific learning outcomes and personalization hooks to produce scaffolded, transfer-oriented scenarios. By embedding abstract concepts specifically within Electrical and Computer Engineering (ECE) into structured narratives, the intervention aims to enhance student engagement and stimulate higher-order thinking. This paper documents the design of the AI agent and the pedagogical framework, offering a scalable model for meaningful AI integration across STEM disciplines. This is a work in progress paper.

Introduction

Engineering students frequently struggle to bridge the gap between theoretical classroom instruction and real-world application, leading to a superficial grasp of foundational concepts. This disconnect represents a significant gap created by the dominance of decontextualized approaches that prioritize abstract theory over practical relevance. As the global workforce evolves, the necessity for contextualized, engaging, and student-centric learning has become a critical priority for engineering programs [1]. To address this, educators are increasingly exploring narrative-based pedagogies to explain technical content, thereby fostering deeper conceptual understanding and long-term retention.

The use of Artificial Intelligence (AI) combined with structured storytelling offers a scalable solution to the challenge of student disengagement [2], [3], [4]. By embedding technical concepts into structured narratives, educators can strengthen the connection between students and complex content. While personalized storytelling is a powerful cognitive engine, it has historically been labor-intensive for faculty. However, generative AI now allows for the tailoring of these examples to individual learner profiles, contextualizing problems based on specific student needs and experiences in real-time [5], [6].

Despite the potential of AI-enhanced storytelling, the integration of these technologies into higher education requires more than mere technical implementation. Unregulated AI use risks degrading academic rigor or introducing "shortcut learning" if the narrative is not strictly aligned with specific learning outcomes. There is a pressing need for a structured design framework that constrains AI behavior within established pedagogical theories to ensure that the resulting instructional materials are ethically sound, technically rigorous, and promote learning [5], [7].

While existing literature on narrative pedagogy often emphasizes multimedia delivery [8], [9], it lacks a rigorous structural for developing the narrative for storytelling problems [10]. This study addresses this gap by identifying fourteen primary and secondary pedagogical characteristics designed for engineering education. To operationalize these standards, we introduce a Custom AI Agent utilizing an "Educator-in-the-Loop" (EITL) architecture. This system constrains generative AI output through a structured intake interface, ensuring the instructor remains a "Pedagogical Engineer" who maintains technical rigor and ethical grounding. By mitigating risks like "shortcut learning," this approach offers a scalable pathway toward personalized STEM learning [11] and serves as a foundation for future empirical evaluation.

This work is positioned within a design-based research (DBR) paradigm, which aims to understand the nature of interventions through intended design [12]. This work represents an initial design and theorization phase. The goal of this phase is not to evaluate learning outcomes, but to articulate, ground, and formalize a pedagogically rigorous design framework for story-driven, AI-augmented engineering instruction. In DBR, early phases focus on identifying design principles, articulating theoretical grounding, and producing reusable design knowledge that can inform subsequent empirical cycles.

This paper contributes to engineering education research in three ways. First, it formalizes narrative pedagogy in engineering through a consolidated, research-grounded, and theory-aligned set of instructional characteristics that articulate how storytelling can be created without sacrificing technical rigor. Second, it introduces an Educator-in-the-Loop AI design architecture that reframes generative AI as a constrained pedagogical instrument rather than an autonomous content generator, positioning educator intent and judgment as central to instructional quality. Third, it offers a reusable design rubric that enables faculty to audit and refine story-driven problems prior to classroom deployment; taken together, these contributions function as testable design hypotheses that can inform future classroom-based and empirical studies.

The remainder of this paper is organized as follows: we first explore the role of story-driven problems in STEM and establish the theoretical foundations supporting this approach. We then define the characteristics that distinguish rigorous story-driven problems and describe the design of the custom AI agent. Finally, we provide a case-study dissection of a story generated for an Electrical and Computer Engineering (ECE) context, followed by a discussion of instructional integration, future research directions, and concluding remarks.

Story-Driven Problems in STEM Education

Over recent years, storytelling in education has attracted the attention of the scientific community as it can be used to motivate students, make them more interested, and facilitate the process of knowledge transfer. The organization of using this approach is especially useful in STEM education [13], where abstract ideas can be very abstract, and real-life use is not always apparent. Problems using stories in STEM will help to facilitate stronger learning by settling technical data within the story contexts that students can relate to and are interested in. According to Mou (2024), the use of storytelling can go a long way in enhancing the capacity of students to memorize and practice knowledge. As an example, the implementation of story-based scenarios in physics or engineering classes can result in better student learning and engagement.

Inquiry-based science education (IBSE) strategies are regarded as an active learning approach that enables students to acquire skills and knowledge necessary for meeting the challenges of the 21st-century world. Several factors can contribute to improving the success of IBSE, and the subsequent acquisition of inquiry skills in particular. This research focuses on the synergistic role of applying IBSE through well-constructed inquiry tasks in STEM-oriented subjects, complemented by digital technologies and formative assessment instruments, and administered by educators trained in this area.

However, when it comes to technical classes such as engineering, it is difficult to use storytelling. Engineering problems, unlike humanities topics or social sciences, need some degree of abstraction and technicality that is hard to humanize. Use of storytelling in engineering undergraduate education will be viewed as less rigorous than the material content taught, particularly when it is used in lower grades, such as circuits or programming [14]. In addition, to design meaningful and effective story-driven problems, it will be necessary to be cautious in designing the problems so as not to simplify or distort technical principles too much.

The gaps in the existing areas of implementation of these methods into undergraduate engineering learning encompass the absence of empirical research on long-term outcomes of storytelling on conceptual learning and problem-solving skills in engineering [15]. Though there are isolated instances of successful narrative-based teaching in particular situations, they are not well organized and incorporated into the curriculum[10]. Therefore, additional studies will be required to investigate how storytelling can be scaled and integrated into engineering classes in order to achieve the most significant benefits.

Despite the growing recognition of storytelling as an effective pedagogical approach in STEM education, there remains a notable gap in the literature regarding a unified list of characteristics that define effective story-driven examples. While several studies and articles address different aspects of storytelling [16], [17], [18], they often lack systematic integration or a consolidated framework. This lack of coherence complicates the application of storytelling techniques in educational settings. Also, a cohesive framework that describes how the characteristics of storytelling can align with assessment methods and educational strategies remains absent. This gap complicates efforts to measure the effectiveness of storytelling approaches in enhancing STEM learning.

Theoretical and Instructional Design Foundation

The integration of storytelling and Artificial Intelligence in STEM education requires a robust multi-lens framework to ensure pedagogical rigor and student success [4]. To move beyond a purely technological implementation, this study synthesizes established theories from cognitive science, educational psychology, and instructional design. By grounding our approach in these diverse fields, we transition from viewing storytelling as a mere engagement tool to a structured cognitive engine capable of fostering deep conceptual understanding. The following sections detail the specific roles of Constructivism, Experiential Learning, Cognitive Load, Self-Determination, Zone of Proximal Development, Constructive Alignment, and Adaptive Learning, illustrating how they collectively form the scaffolding for story-driven engineering instruction.

Kolb's Theory of Experiential Learning

According to the Theory of Experiential Learning, the learning process is vital in terms of experience [19]. Moreover, learning can be characterized as a process that includes four distinct phases, namely, concrete experience, reflective observation, abstract conceptualization and active experimentation. The theory of Kolb could be used in the engineering education process by designing courses that enable students to contact the problems directly, reflect on their experiences, theorize about the experiences, and subsequently test their theories through further experimentation. Experiential Learning involves knowledge building through four learning modes, represented by the four stages of the learning cycle. This process is cyclic in that the students are not merely learning theoretical knowledge but can also be able to use that knowledge in practical situations, which is critical in solving problems in the field of engineering.

Constructivist Theories of Learning

The constructivist learning theories focus much on the idea that knowledge is built as people interact with the environment, not passively accepted [20]. Based on the previous knowledge, learners develop new knowledge through the process of experiences and interactions. Constructivism, in its turn, enables the students to work with difficult engineering concepts by integrating them into a story. These stories offer a significant background, which assists the learners in building knowledge in a manner that is relevant to them. Through engaging in problems that are story-based, students get to think critically and synthesize information and gain a better understanding of the material.

Cognitive Load Theory

The Cognitive Load Theory is an idea put forward by Sweller that proposes that the working memory capacity of learners is limited [21]. According to the theory, in the case of learning complex content, the instructional design must reduce the extraneous cognitive load and maximize the intrinsic cognitive load to make the understanding easier. In engineering education, this theory will be utilized through simplifying complex concepts with the help of a story.

Self-Determination Theory

The theory Self-Determination Theory (SDT) by Deci and Ryan involves motivation and the contribution of autonomy, competence and relatedness towards intrinsic motivation[22] [23]. SDT in educational contexts implies that the students will feel more motivated and engaged when they have control over what they learn (autonomy), feel able to master the material (competence), and perceive the relationship of the learning material with their lives (relatedness) [22], [24], [25]. Engineering education can be improved through the use of AI technologies and personalized problems that are based on narratives to improve student autonomy and competence. This is done by making the learning experiences student-oriented, which means that the student guides the learning process in their own interests, as well as prior knowledge and still makes them feel adequate by being given challenges progressively.

Adaptive Learning Theory

The Adaptive Learning Theory is about developing a customized learning experience that adapts dynamically to the needs of the learner [26].In this method, the AI tools are implemented to study the progress of a learner and adjust the content so that it is possible to make sure that the material is not too simple or too challenging. Adaptive learning may be applied in the field of engineering education by introducing AI-driven systems in the creation of customized story-based learning scenarios. These AI systems understand the performance of students and give each student specific challenges or assistance so that they are given a specific level of difficulty. This individualization can maximize the learning process and increase the interest and motivation of the student body.

The Zone of Proximal Development (ZPD)

Rooted in Vygotsky's (1978) socio-cultural theory, the Zone of Proximal Development (ZPD) identifies the cognitive space between what a learner can accomplish independently and what they can achieve with strategic guidance [27]. In the context of story-driven engineering problems, the ZPD provides the logical 'location' for narrative intervention. By situating problems within this zone, the instruction ensures that the challenge is sufficient to promote growth without triggering the frustration of cognitive overload.

Constructive alignment

Constructive Alignment is an instructional design framework that emphasizes the deliberate alignment of learning outcomes, instructional activities, and assessment to ensure pedagogical coherence [28], [29]. This framework positions instructional activities as the primary mechanisms through which learning outcomes are achieved and demonstrated [30]. Within this perspective, narrative-based instruction is pedagogically legitimate when the story itself is intentionally designed to elicit the specific reasoning, decisions, and problem-solving processes targeted by course outcomes. By embedding assessment-relevant variables, constraints, and decision points directly within the narrative, story-driven problems generate observable and measurable evidence of learning.

While each of these theories provides a distinct perspective on the learning process, their true value lies in their convergence within the design of engineering problems. Together, they transition the learner from a passive consumer of information to an active protagonist in a scaffolded, relational, and ethically situated narrative. This theoretical synthesis serves as the foundation for the Characteristics of Story-Driven Engineering Problems identified in this study. In the subsequent section, we operationalize these theories into specific, actionable design principles that guide the creation of AI-augmented content, ensuring that every narrative remains academically rigorous, personally relevant, and cognitively accessible

Defining Characteristics of Story-Driven Engineering Problems

To effectively operationalize storytelling within STEM curricula, we are proposing a dual layered framework of characteristics. These are categorized into primary and secondary attributes based on their instructional proximity to core learning objectives. The primary characteristics represent the foundational pedagogical pillars necessary to catalyze learner engagement, provide cognitive scaffolding, and ensure the mastery of technical outcomes. Conversely, the secondary characteristics serve as integrative layers; while they may not be mandatory for every technical exercise, they enrich the learning experience by fostering multidisciplinary literacy, ethical reasoning, and collaborative proficiency. These skills situates engineering within its broader societal and systemic contexts [31].

Primary Characteristics:

P1. Connected to learning outcomes: The narrative arc is strictly mapped to specific educational competencies. The story acts as the vehicle for the "need to know," ensuring the plot directly facilitates the mastery of core concepts [28]

P2**.** Assessment-Integrated: The narrative provides the necessary data points, constraints, and variables for formal evaluation. The problem-solving within the story serves as measurable evidence of student performance.[32], [33]

P3. Relational: Humanizes abstract data by introducing roles, agency, and consequences. By situating the learner within a professional or social relationship, it fosters a sense of accountability and relevance which establish and sustain motivation for learning [34, p. 69]

P4. Culturally and socially responsive: Grounds the problem in the learners' diverse social contexts and lived experiences. This validates the student's identity and increases relevance through Culturally Sustaining Pedagogy [35]**.**

P5. Engaging & Motivating: Uses relatable narrative framing to sustain curiosity. By fostering autonomy and competence, the design satisfies psychological needs for intrinsic motivation and facilitates deep interest in complex problem-solving [24], [25].

P6. Scaffolded: Decomposes complexity into sequential narrative milestones. By pacing the introduction of variables, the design maintains the learner within the Zone of Proximal Development (ZPD) and ensures cognitive load remains within processing capacities [27], [36]

P7. Simplifies Complex Concepts: Employs metaphors and analogies to translate abstract principles into intuitive narrative constructs. By leveraging familiar mental models, the design reduces extraneous cognitive load and facilitates schema acquisition, allowing learners to prioritize higher-order synthesis over basic decoding [37]

P8. Reflective and Promotes Critical Thinking: Prompts learners to evaluate their reasoning, assumptions, and ethical justifications. By requiring students to defend technical decisions and analyze consequences, the design facilitates "reflection-in-action," transforming problem-solving into a metacognitive process that deepens professional judgment [19].

P9. Student-centered: Positions learners as active decision-makers rather than passive consumers. By granting agency over the narrative trajectory and solution paths, the design shifts the "locus of control" to the student, fostering ownership and deep engagement through constructivist principles [20].

P10. Transfer-Oriented Facilitates the generalization of core concepts by applying knowledge to novel narrative scenarios. By cycling through reflective observation and abstract conceptualization, the design enables "Active Experimentation," ensuring technical principles transfer effectively to diverse engineering challenges [19]

Secondary Characteristics:

S1. Ethical, Moral, and Societal Relevance: Situates technical problems within real-world dilemmas to illuminate the consequences of engineering work. This "social constructivist" lens ensures technical knowledge is developed by negotiating the interplay between engineering solutions and their impact on human communities [38]

S2. Collaborative: Facilitates group problem-solving and peer integration through shared narrative goals. This design ensures compliance with accreditation and industry recommendations, requiring students to function effectively on teams that establish goals, plan tasks, and maintain a collaborative, inclusive environment [5], [38]

S3. Adaptive and Personalized: Tailors narrative complexity and technical constraints to the learner's interests and prior experience. By dynamically adjusting scaffolding based on individual profiles, the design optimizes the cognitive path to ensure content remains challenging yet accessible within the learner's mastery level [26].

S4. Multidisciplinary: Connects engineering principles to domains such as economics, environment, and public health to situate problems within broader global systems. By integrating these constraints, students are required to make informed judgments that account for the societal, economic, and environmental impacts of technical solutions [5], [38] By establishing this definitive set of ten primary and four secondary characteristics, the framework moves beyond general creative writing into the realm of structured instructional design. Each characteristic acts as a pedagogical filter to guide the creation of story-based examples or problems. Table 1 delineates the alignment between these characteristics and the theoretical frameworks established in the previous section. Whether the educator will be drafting the story manually or by using an AI agent, ensuring that narrative elements never come at the expense of technical accuracy or learning outcomes. Critically, this list fills a significant gap in engineering education by consolidating disparate narrative strategies into a unified set of characteristics rooted in established learning theories. While storytelling has long been recognized for its potential to increase engagement, its implementation in STEM often remains anecdotal or unstructured. This framework formalizes the process, bridging the gap between "creative storytelling" and "engineering rigor" by grounding every narrative choice in evidence-based principles

Table 1: Mapping Story-Driven Problem Characteristics to Learning Theories

Characteristic ID	Theoretical Grounding	Rationale
P1, P2	Constructive Alignment	Ensures that narrative activities and embedded problem solving are directly aligned with intended learning outcomes and formal assessment.
P3, P5	Self-Determination Theory	Supports motivation and engagement by fostering relatedness, relevance, and sustained learner interest through narrative context.
P4, P9, S1, S2, S4	Constructivism	Grounds learning in students' prior knowledge, social context, agency, and collaborative meaning-making.
P6	Zone of Proximal Development	Structures narrative scaffolding to maintain challenge within learners' proximal developmental range.
P7	Cognitive Load Theory	Reduces extraneous cognitive load by using narrative metaphors to make abstract concepts more cognitively accessible.
P8, P10	Experiential Learning Theory	Promotes reflection and transfer by cycling learners through experience, abstraction, and application.
S3	Adaptive learning Theory	Enables personalization of narrative context and challenge based on learner characteristics and needs.

Designing AI-Augmented Story Problems

The Prepare–Create–Test–Improve–Integrate (PCTII) design cycle is a domain-agnostic framework developed to transition generative AI from an autonomous agent to a highly constrained instructional design instrument [39], [40]. By prioritizing pedagogical intent over technological affordance, the cycle ensures that story-driven engineering problems are grounded in explicit learning outcomes rather than emergent AI behavior [5], [41]. This systematic approach treats the AI as a pedagogical design assistant that operationalizes complex instructional design characteristics under the strict supervision of the educator. This interaction provides the foundational layer of the EITL architecture (see Figure 1a). At its core, PCTII serves as a model for professional accountability in AI customization and integration. Through the iterative Testing and Improvement phases, the educator, in this case the developer of the custom GPT, audits the output of the custom GPT for technical inaccuracies, undesirable behavior, or misalignments with the intended curriculum. By treating AI outputs as provisional until validated by the educator, the framework reinforces the educator's role as the final arbiter of quality. This custom design process ensures that the resulting story-driven problems are rigorous educational tools rather than merely entertaining narratives.

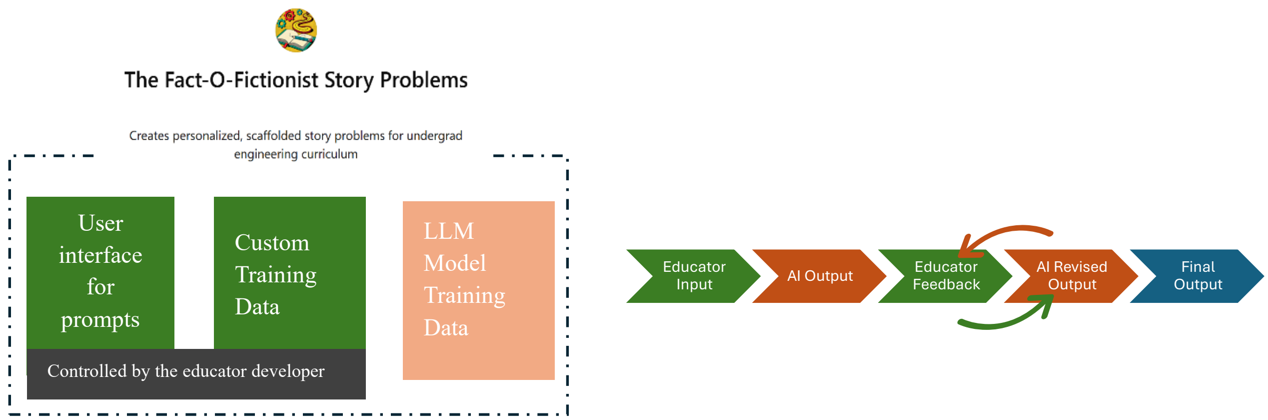


Figure 1: a) Pedagogical Engineering & Artifact Design (EITL- Layer 1), b) Instructional input & Contextual Refinement (EITL-Layer 2)

The educator, in this case the user, interacts with the custom GPT "The Fact-O-Fictionist Story Problems" through a specialized Educator-Facing Interface designed to preserve instructional authorship and academic rigor. In this configuration mode, the custom GPT acts as a pedagogical design assistant, requiring the educator to provide "Ground Truth" inputs in order for it to function. As shown in Figure 1b, the educator continues to engage with the system in a cycle of input-feedback-output until an optimum; error-free result is achieved.

The interface (see Appendix A) serves as the primary EITL control mechanism. Rather than a blank chat prompt, the system utilizes a structured **"Instructor Intake" checklist (A–L)** that operationalizes pedagogical intent. For instance, by requiring "Boundaries & Exclusions" (Item I), the educator prevents the AI from introducing advanced methods not yet covered in the curriculum, thereby managing the students' cognitive load. Furthermore, "Personalization Hooks" (Item H) allow the educator to scale **Culturally Sustaining Pedagogy** without the time-intensive burden of manually researching diverse student interests for every problem set. Table 2 illustrates how these specific intake-questions map directly to the storytelling characteristics identified in the previous section.

This dual layer of EITL architecture ensures that the AI never operates in a vacuum; instead, it generates structured cognitive contexts only after the educator has established the design code and narrative boundaries. Rather than treating generative AI risk as a purely technical problem, this work conceptualizes risk mitigation as a pedagogical design responsibility. The EITL architecture functions as a buffer that constrains AI behavior through instructional intent, curricular boundaries, and educator judgment. By requiring explicit educator input and iterative validation, the system shifts accountability away from the AI system and back to the instructor, mitigating risks such as hallucinated content, misaligned difficulty, and shortcut learning. In this sense, EITL is not merely a usability feature, but a pedagogical governance mechanism.

Table 2 Mapping the Pedagogical Intake Interface to Storytelling Characteristics

Intake Parameter (AI Interface Question)	Characteristics Addressed
A. **Course & context** (course title, department, term)	P1, S4
B. **Topic focus** (specific topic/subtopics & target skills; list common misconceptions to address)	P7, P10, S1
C. **Learner profile** (student level; prerequisites, etc.)	P6, S3
D. **Learning outcomes** (action-verb outcomes + success criteria / evidence expected)	P1, P2
E. **Activity setting & time** (in-class or asynchronous; time on task)	P6. S2
F. **Deliverable & format** (what students will submit: worked solution, code, memo, diagram, slides, etc.)	P2, P9, S2
G. **Tools & constraints** (allowed tools/software, calculators, formula sheets, provided data)	P2, P6
H. **Personalization hooks** (student interests or local/industry context to weave in)	P4, P5, S1, S3, S4
I. **Boundaries & exclusions** (methods not yet covered or topics to avoid)	P6
J. **Materials to integrate (optional)** (syllabus, lecture notes, datasets). Upload for tight alignment	P1
K. **Difficulty control** (target Bloom level; scaffolding depth; partial-credit approach)	P6, P8
L. **Solution preferences** (Do you want an answer key / stepwise rubric now, or only upon request?)	P2

Case Study: The Stack Garage

The development of "The Stack Garage" began as a collaborative interaction between the educator and the custom AI agent. The educator utilized the pedagogical intake interface to define the core technical concepts and learning objectives. The AI agent then proposed a narrative featuring Alex, a curious junior technician, and Casey, a senior technician-in-training, to frame the instruction as professional mentorship. This resulted in a comic-style narrative where the characters troubleshoot an LED control routine on a vehicle to contextualize the "need to know" within a relational and culturally responsive setting (see Appendix B for the full case study). Following this initial draft, the educator intervened by providing feedback to ensure all narrative elements met strict technical rigor. Although the AI provided the initial story structure, the educator adjusted the technical dialogue and corrected errors and typos. This was achieved through iterative re-prompting of the AI, a step that proved crucial for correcting potential hallucinations and errors while emphasizing the indispensable role of the educator in the loop. Once the output was verified to be free of error and met all specified pedagogical requirements, the educator deemed the artifact ready for classroom implementation.

The systematic mapping of the "Stack Garage" case study to the established pedagogical framework is detailed in Table 3, providing an evaluation of how the AI-generated content aligns with each narrative requirement. This mapping demonstrates that the artifact is not merely a creative exercise but a theoretically grounded instructional tool where every dialogue choice and technical variable serves a specific educational function.

Table 3 Mapping the "Stack Garage" Case Study to Storytelling Characteristics

Characteristic ID	Evidence from "The Stack Garage" Case Study
P1	The narrative arc is strictly mapped to mastering JSR and RTS behaviors within the CPU.
P2	Tasks 1 and 2 provide assessment points for formal evaluation.
P3	The mentorship between Alex (junior tech) and Casey (senior tech) creates a professional context that fosters accountability.
P4	The car garage setting grounds abstract logic in a universal social context familiar to almost all college students.
P5	The comic-book framing and car "troubleshooting" plot sustains curiosity through playful storytelling.
P6	The activity breaks down complexity by teaching the Program Counter before moving to the Stack and Subroutines.
P7	Employs the "bookmark" metaphor to make the function of the Program Counter intuitive for the learner.
P8	Metacognitive Questioning: Characters like Alex model the critical thinking process by asking clarifying questions—"So, it remembers where it left off?"—which encourages learners to reflect on the underlying mechanics of memory registers
P9	The learner is positioned as an active participant/problem-solver who must "fix the subroutine."
P10	The transition from the garage story to a formal technical table facilitates the generalization of concepts to novel scenarios.
S3	Narrative tech-roles are tailored to the practical, hands-on career interests typical of engineering students.
S4	Connects ECE logic (JSR/RTS) to automotive maintenance, demonstrating how engineering exists within broader systems.

This analysis is insightful because it transforms abstract pedagogical theories into a concrete, observable rubric for engineering education. Educators can utilize this table as a comprehensive checklist to audit their own story-based examples, whether drafted manually or via an AI agent, ensuring that every problem maintains the necessary balance of technical rigor, relational engagement, and culturally responsive context before reaching the classroom.

Classroom Integration and Pedagogical Flow

The instructional implementation of the "Stack Garage" followed a structured four-stage design that transitioned learners from theoretical concepts to applied engineering practice. The process began with a traditional lecture covering foundational concepts.

This was followed by the storytelling in-class example, students then completed homework assignments featuring problems with identical logical structures, requiring them to engage in the same processes to solve problems within the established narrative framework. The cycle concluded with a formal assessment, where the narrative scaffolding was removed in favor of professional engineering deliverables, requiring students to demonstrate mastery of the foundational concepts and their applications.

Conclusion

This paper presents a consolidated framework of 14 research-grounded storytelling characteristics designed to empower STEM educators in the creation of rigorous, story-driven problems and examples. To operationalize these characteristics, we developed a custom GPT assistant governed by the PCTII design cycle. This process transitions generative AI from an autonomous agent into a precise instructional design tool. Central to this approach is a two-fold EITL structure: the first layer involves the educator's foundational design of the custom GPT to ensure pedagogical rigor, while the second layer involves an active usage cycle where the educator iterates with the AI through feedback and re-prompting until an optimum, error-free story problem is achieved.

By providing educators with a scalable method to personalize instruction without sacrificing academic rigor, this framework offers a viable path toward human-centric, AI-augmented engineering pedagogy. While this work was demonstrated within an ECE context, the storytelling characteristics and the custom GPT are inherently domain-agnostic, making them applicable across all engineering disciplines and adaptable for broader expansion into various STEM majors.

As a work-in-progress, the next phase of this research will involve a classroom study to measure the impact of these story-driven problems on student retention, conceptual mastery, and self-efficacy in Electrical and Computer Engineering (ECE).

References

[1] A. Bhardwaj, A. Sergeeva, J. Mahoney, and J. Nickerson, "Theorizing as Problem Solving: A Pragmatist Perspective on the Logic of Pursuit," *Strategy Science*, vol. 10, no. 4, pp. 338–359, Dec. 2025, doi: 10.1287/stsc.2023.0075.

[2] M. Alaidaros and S. H. Mohamed, "Integrating PBL, Agile, and AI-Enhanced Storytelling for Entrepreneurial Teamwork in Saudi Higher Education," *IJARPED*, vol. 14, no. 3, p. Pages 1985-2006, Sep. 2025, doi: 10.6007/IJARPED/v14-i3/26452.

[3] J. Belda-Medina and M. B. Goddard, "AI-Driven Digital Storytelling: A Strategy for Creating English as a Foreign Language (EFL) Materials," *International Journal of Linguistics Studies*, vol. 4, no. 1, pp. 40–49, Feb. 2024, doi: 10.32996/ijls.2024.4.1.4.

[4] S. Kahouf, M. AbouHashesh, and S. Abdelrhman, "THEORETICAL PERSPECTIVES ON AI-ENHANCED DIGITAL STORYTELLING: IMPLICATIONS FOR LEARNER ENGAGEMENT AND MOTIVATION," *Scientific Culture*, vol. 11, no. 4, pp. 799–817, 2025.

[5] E. Aref, "Team dynamics and conflict resolution: Integrating Gen AI in project-based learning to support students' performance," *Western Michigan University*, 2024.

[6] R. Martínez-Peláez *et al.*, "Can Large Language Models Foster Critical Thinking, Teamwork, and Problem-Solving Skills in Higher Education?: A Literature Review," *Systems*, vol. 13, no. 11, p. 1013, Nov. 2025, doi: 10.3390/systems13111013.

[7] I. M. García-López and L. Trujillo-Liñán, "Ethical and regulatory challenges of Generative AI in education: a systematic review," *Front. Educ.*, vol. 10, p. 1565938, Jun. 2025, doi: 10.3389/feduc.2025.1565938.

[8] H. Hussain and N. Shiratuddin, "A Digital Storytelling Process Guide for Designers," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 8, no. 8, pp. 13–17, Nov. 2016.

[9] P. Yee and E. Stevents, "Digital Storytelling," in *The Culture of Digital Scholarship in Academic Libraries*, R. C. Roemer and V. Kern, Eds., Chicago: American Library Association, 2019, pp. 83–100. Accessed: Jan. 21, 2026. [Online]. Available: https://alastore.ala.org/content/culture-digital-scholarship-academic-libraries

[10] B. R. Robin, "Digital Storytelling: A Powerful Technology Tool for the 21st Century Classroom," *Theory Into Practice*, vol. 47, no. 3, pp. 220–228, Jul. 2008, doi: 10.1080/00405840802153916.

[11] R. Baltà-Salvador, I. El-Madafri, E. Brasó-Vives, and M. Peña, "Empowering Engineering Students Through Artificial Intelligence (AI): Blended Human–AI Creative Ideation Processes With ChatGPT," *Computer Applications in Engineering Education*, vol. 33, no. 1, p. e22817, 2025, doi: 10.1002/cae.22817.

[12] C. Hoadley and F. C. Campos, "Design-based research: What it is and why it matters to studying online learning," *Educational Psychologist*, vol. 57, no. 3, pp. 207–220, Jul. 2022, doi: 10.1080/00461520.2022.2079128.

[13] R. Cabrera, A. Carrión, C. Carrión, and G. Romero, "Storytelling and stem: a scoping review," in *International Conference on Technological Innovation and AI Research (ICTIAIR 2025)*, Mar. 2025, pp. 72–77. doi: 10.1049/icp.2025.1247.

[14] M. F. Dahlstrom, "Using narratives and storytelling to communicate science with nonexpert audiences," in *,Proceedings of the national academy of sciences, 111(supplement_4),* 2014, pp. 13614-13620. doi: 10.1073/pnas.1320645111.

[15] S. Klassen, “Introduction – Stories in Science Teaching,” *Interchange*, vol. 41, no. 4, pp. 311–313, Oct. 2010, doi: 10.1007/s10780-010-9131-2.
[16] L. Chen *et al.*, “Embedding Digital Data Storytelling in Introductory Data Science Course: An Inter-Institutional Transdisciplinary Pilot Study,” *JPBLHE*, vol. 11, no. 2, pp. 126–152, Oct. 2023, doi: 10.54337/ojs.jpblhe.v11i2.7767.
[17] A. L. Molder, B. Anderton, E. Howell, S. Goodwin, S. Behrman, and M. Ellefson, “Use of personal storytelling in educational videos promotes student engagement and science identity in undergraduate biology courses,” *J Microbiol Biol Educ.*, pp. e00084-25, Oct. 2025, doi: 10.1128/jmbe.00084-25.
[18] L. Zhang and D. A. Bowman, “Exploring Effect of Level of Storytelling Richness on Science Learning in Interactive and Immersive Virtual Reality,” in *ACM International Conference on Interactive Media Experiences*, Aveiro JB Portugal: ACM, Jun. 2022, pp. 19–32. doi: 10.1145/3505284.3529960.
[19] C. Kingkaew *et al.*, “A Learning Environment to Promote Awareness of the Experiential Learning Processes with Reflective Writing Support,” *Education Sciences*, vol. 13, no. 1, p. 64, Jan. 2023, doi: 10.3390/educsci13010064.
[20] S. Chuang, “The Applications of Constructivist Learning Theory and Social Learning Theory on Adult Continuous Development,” *Performance Improvement*, vol. 60, no. 3, 2021, doi: 10.1002/pfi.21963.
[21] E. Gkintoni, H. Antonopoulou, A. Sortwell, and C. Halkiopoulos, “Challenging Cognitive Load Theory: The Role of Educational Neuroscience and Artificial Intelligence in Redefining Learning Efficacy,” *Brain Sciences*, vol. 15, no. 2, p. 203, Feb. 2025, doi: 10.3390/brainsci15020203.
[22] T. K. F. Chiu, “Using self-determination theory (SDT) to explain student STEM interest and identity development,” *Instructional Science*, vol. 52, no. 1, pp. 89–107, Feb. 2024, doi: 10.1007/s11251-023-09642-8.
[23] C. E. A. MA, “Self Determination Theory and How It Explains Motivation,” PositivePsychology.com. Accessed: Jan. 21, 2026. [Online]. Available: https://positivepsychology.com/self-determination-theory/
[24] Y. Lei, “Unlocking the secrets of STEM success: Exploring the interplay of motivation to learn science, self-regulation, and emotional intelligence from a perspective of self-determination theory,” *Learning and Motivation*, vol. 87, p. 102012, Aug. 2024, doi: 10.1016/j.lmot.2024.102012.
[25] R. M. Ryan and E. L. Deci, “Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being,” *American Psychologist*, vol. 55, no. 1, pp. 68–78, 2000, doi: 10.1037/0003-066X.55.1.68.
[26] W. Strielkowski, V. Grebennikova, A. Lisovskiy, G. Rakhimova, and T. Vasileva, “AI-driven adaptive learning for sustainable educational transformation,” *Sustainable Development*, vol. 33, no. 2, pp. 1921–1947, 2025, doi: 10.1002/sd.3221.
[27] G. Salomon, “Cognitive Effects With and Of Computer Technology,” *Communication Research*, vol. 17, no. 1, pp. 26–44, Feb. 1990, doi: 10.1177/009365090017001002.
[28] J. Biggs, “Aligning teaching for constructing learning,” *Higher Education Academy*, vol. 1, no. 4, pp. 1–4, 2003.
[29] J. Biggs, “Constructive alignment in university teaching,” *HEDSA Review of Higher Education*, vol. 1, pp. 5–22, 2014.

[30] L. Newby and L. Cornelissen, "A neutral toolkit? For a fundamental critique of constructive alignment," *Teaching in Higher Education*, vol. 30, no. 7, pp. 1687–1702, Oct. 2025, doi: 10.1080/13562517.2025.2465987.

[31] D. H. Jonassen and W. Hung, "All Problems are Not Equal: Implications for Problem-Based Learning," *Interdisciplinary Journal of Problem-Based Learning*, vol. 2, no. 2, Oct. 2008, doi: 10.7771/1541-5015.1080.

[32] E. E. Khoury, E. Nasr, and E. Aref, "(Re)Defining Ethical Assessment with the Advent of GenAI," Sep. 28, 2025, *EdArXiv*: zp458_v1. doi: 10.35542/osf.io/zp458_v1.

[33] M. S. Trevisan *et al.*, "Integrated Design Engineering Assessment and Learning System (IDEALS):," *2012 ASEE Annual Conference &Amp; Exposition Proceedings*, 2012, doi: 10.18260/1-2--21546.

[34] S. A. Ambrose, M. W. Bridges, M. DiPietro, M. C. Lovett, and M. K. Norman, *How learning works: Seven research-based principles for smart teaching*. John Wiley & Sons, 2010.

[35] D. Paris, "Culturally Sustaining Pedagogy: A Needed Change in Stance, Terminology, and Practice," *Educational Researcher*, vol. 41, no. 3, pp. 93–97, Apr. 2012, doi: 10.3102/0013189X12441244.

[36] J. Sweller, "Cognitive Load During Problem Solving: Effects on Learning," *Cognitive Science*, vol. 12, no. 2, pp. 257–285, 1988, doi: 10.1207/s15516709cog1202_4.

[37] J. Sweller, J. J. G. van Merriënboer, and F. Paas, "Cognitive Architecture and Instructional Design: 20 Years Later.," *Educational Psychology Review*, vol. 31, no. 2, pp. 261–292, Jun. 2019, doi: 10.1007/s10648-019-09465-5.

[38] ABET, "CRITERIA FOR ACCREDITING ENGINEERING PROGRAMS. Effective for Reviews during the 2025-2026 Accreditation Cycle," 2024.

[39] E. Aref, "Empowering Engineering Educators: Build and Implement a Custom AI Assistant for Student Learning and Engagement," presented at the ASEE NCS Conference 2025, Marshall University, Huntington, WV, Mar. 2025. [Online]. Available: DOI: 10.13140/RG.2.2.33418.99522

[40] E. Aref, "Empowering Educators: Build and Implement a Custom AI Assistant for Student Learning & Engagement," presented at the Mid-Western Educational Research Association Annual Meeting 2025, Milwaukee, Wisconsin, Nov. 2025.

[41] S. K. Shankar, G. Pothancheri, D. Sasi, and S. Mishra, "Bringing Teachers in the Loop: Exploring Perspectives on Integrating Generative AI in Technology-Enhanced Learning," *Int J Artif Intell Educ*, vol. 35, no. 1, pp. 155–180, Mar. 2025, doi: 10.1007/s40593-024-00428-8.

Appendix-A

Custom AI : The Fact-O-Fictionist Story Problems (access link)

Design and Interface

The Fact-O-Fictionist Story Problems 5 Thinking mini

Thought for 5s ›

Great — let's build a story problem! 🎬 📐

I'll follow your instructor intake so the problem fits your course and students. I can align the story to your uploaded materials (Mr. BNE, sample story examples, Stack Register Hotel).

Please fill in this concise checklist (A–L). you can paste short answers — keep each item 1–2 lines if you like.

A. Course & context — course title and discipline (e.g., "ECET3490 — Intro to Assembly / Computer Org")
B. Topic focus — topic / subtopics / target skills / common misconceptions to address
C. Learner profile — student level and prerequisites (e.g., "sophomore; know hex, registers, basic loops")
D. Learning outcomes — 1–3 action-verb outcomes with success criteria (what evidence you expect)
E. Activity setting & time — in-class or asynchronous; time-on-task; individual or group (size)
F. Deliverable & format — what students submit (worked solution, code, diagram, brief memo, etc.)
G. Tools & constraints — allowed tools/software, calculators, formula sheets, provided data
H. Personalization hooks — student interests to weave in (sports, music, local industry, etc.)
I. Boundaries & exclusions — methods not yet covered or sensitive topics to avoid
J. Materials to integrate (optional) — which uploaded files/lectures/datasets to use (or "none")
K. Difficulty control — Bloom level (apply/analyze/create?), desired scaffolding/partial credit policy
L. Solution preferences — include full solution / stepwise rubric / only hints? (default: no full solution)

Also choose the **story style** (pick one number, *just the number*):
1 — Comic style
2 — Text-based story problem
3 — Textbook style

When you send these, I'll draft a scaffolded, culturally-responsive story problem that matches your choices.

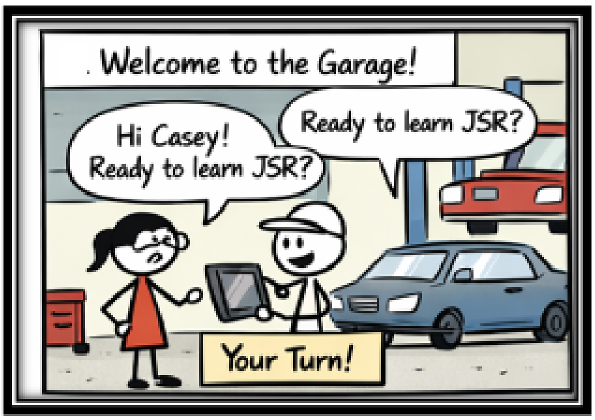
Appendix-B

The JSR @ RTS Garage Case study

The JSR @ RTS Garage

A short comic-style activity to teach JSR / RTS behavior.

Characters: Alex (the curious junior tech) and Casey (the careful senior-in-training).



Alex: Casey, Ooh, subroutine is very confusing.
Casey: Well, I would be happy to help you, Alex. Let's break it down together.

Casey: Before jump into SBR, you must understand the work of the **Program Counter**.
Alex: It shows what runs next?
Casey: Exactly—like a bookmark.
PC → next instruction

Alex: So, after one instruction being executed…
Casey: …the Program Counter (PC) automatically moves to the address of the next instruction, which tells the CPU what operation to execute next.

PC = PC + 1

Your Turn (Task 1 — trace the PC):

Casey: The CPU is currently executing **instruction1** at address **$1010**. The next instruction (**instruction2**) is at address **$100A**.

The PC (Program Counter) will point to:

________.

Remember, the PC always points to the **next instruction.**

Alex: Understood!

Casey: Now suppose that you have a code repeat frequently, Time for **a subroutine**, a reusable helper block of code.

Alex: So, we call it whenever we need it instead of writing it over and over?
Casey: That's exactly right.

JSR helper; CALL helper

Alex: Okay, but how does the CPU actually jump to the subroutine?

Casey: Good question. That's where JSR, (*Jump to Subroutine)*, comes in. When the CPU sees a JSR instruction, it does two things. First, it saves the address of the *next instruction, stored in PC,* so it knows where to come back after executing the code in subroutine. Then it jumps to the subroutine.

Alex: So, it remembers where it left off?

Casey: Exactly. Let's look at an example.

The JSR @ RTS Garage

```
JSR TurnLight   ; jump to subroutine
MOV X, RESULT ; continue after


TurnOn
        LDAA #$01 ;
        STAA LIGHT
        RTS          ; return from subroutine
```

Casey: Let's walk through it step by step. When the CPU executes JSR TurnLight, it *jumps* to the subroutine called TurnLight. But before it jumps, the CPU saves the address of the next instruction MOV X, RESULT into Stack Register so it knows where to return.

Alex: So, the program doesn't lose its place?

Casey: Exactly. Now the CPU starts executing the subroutine. Inside TurnLight, the CPU loads the value 01 and stores it in LIGHT, which turns the light on. When it reaches RTS, the RTS pulls out the saved return address from Stack Register and stores it into PC register so that the CPU goes back to MOV X, RESULT.

Alex: So JSR jumps away, RTS brings us back, and the program keeps running.

Casey: That's the key idea behind subroutines, reuse without repeating code."

Your Turn (finish Task 1): Complete the table below:

Address	Instruction	Comment
$1002	JSR TurnOn	Jump to subroutine TurnOn
$1003	NOP	Continue here after return
TurnOn:		
$2000	LDAA #$01	Load 1 into register A
$2001	RTS	Return to instruction after JSR

Table to Fill:

Step	Instruction	PC Before	PC After
1	**JSR 0x104**		
2	**LDAA #$01**		
3	**RTS**		
4	**NOP**