# Pedagogical Debt: Formalizing and Auditing Accumulated Instructional Shortcuts in STEM Course Materials

Project 2 of 3 — Research Proposal and Paper Draft

Henry Fan
San Jose State University McNair Scholar
Foothill College Science Learning Institute Research Lead
henry.fan@sjsu.edu

Draft: March 6, 2026

**Abstract**

Software engineering has *technical debt*: the accumulated cost of shortcuts taken during development that make future work harder. We introduce an analogous concept for education: *pedagogical debt*, the accumulated cost to student understanding of instructional shortcuts taken during curriculum design. Common forms include introducing notation before intuition, stating definitions before establishing need, sequencing for coverage rather than comprehension, and omitting verification opportunities that would allow students to self-assess. We provide a formal definition of pedagogical debt grounded in Harel's necessity principle and Jeff Anderson's twelve criteria for hands-on modeling activities. We then build PedDebt, an open-source static analysis tool that takes course materials as input—lecture notes, problem sets, textbook chapters—and produces a structured pedagogical debt report. We validate the tool against expert instructor assessments across fifteen STEM courses and show that our automated debt scores agree with expert judgment at a Cohen's $\kappa$ of 0.71. We find that courses with high pedagogical debt scores show significantly lower student self-efficacy ratings even after controlling for course difficulty. The tool, training data, and all course material datasets are released with no paywall.

## 1 Introduction

Ward Cunningham introduced the metaphor of *technical debt* in 1992 to describe a pattern that every software engineer recognizes: the accumulated cost of expedient choices that solve an immediate problem while making future work harder [1]. The metaphor was immediately productive because it gave software teams a shared vocabulary for reasoning about tradeoffs they had always navigated but never been able to name precisely.

We argue that an analogous accumulation occurs in instructional design, and that it is equally pervasive and equally undiscussed.

Consider a common pattern in introductory calculus: the formal definition of the derivative is presented before students have encountered any problem that cannot be solved without it. Students are asked to memorize and manipulate a limit expression whose purpose is opaque. The notation is introduced before the intuition. This is a shortcut—it is faster for the instructor and easier to grade—but it costs something. Students who do not understand why the definition is needed are unable to use it generatively, cannot self-verify their work, and are more likely to abandon the subject when difficulty increases.

Jeff Anderson has identified this pattern precisely in his necessity principle: *"if math is the medicine, what is the headache?"* [2]. The headache—the intellectual need—must precede the medicine. When it does not, the instructor has incurred pedagogical debt: a cost that will be paid later, by the student, often invisibly, in the form of shallow understanding, low persistence, and inability to transfer knowledge.

Unlike technical debt, pedagogical debt is currently invisible. There is no tool that audits a set of lecture notes and says: "You introduced the Fourier transform before establishing any problem that requires it. This is a motivational shortcut. The cost is estimated at N points of student self-efficacy." This paper builds that tool.

### 1.1 Why a Static Analysis Tool?

The deliberate choice of the phrase *static analysis*—borrowed from programming language theory, where it refers to analysis of source code without executing it—is not accidental. We want to make a precise analogy legible to CS audiences: just as a type checker examines a program's source and flags type inconsistencies without running the program, PedDebt examines a course's materials and flags structural inconsistencies without requiring data on student outcomes.

This matters for Jeff Anderson's verification principle: *students should be able to verify their models without needing to ask the teacher for validation* [2]. We extend this principle to the instructor level: an instructor should be able to audit their own materials without needing external experts or outcome data. PedDebt makes that possible.

### 1.2 Scope and Ethical Position

This tool analyzes *instructional materials*, not students. This is a deliberate ethical choice. NLP-based systems that monitor student discourse or predict individual departure risk have significant privacy and equity concerns [5]. PedDebt audits the designed artifact—the course itself—not the people navigating it. The implicit theory of change is systemic: if we can identify and reduce pedagogical debt in course materials, the environment improves for all students, with no surveillance required.

This aligns with the scholar-activist tradition described by McAlevey and Lawler [6]: the researcher's role is to help practitioners systematize knowledge they already have. Every instructor who has taught long enough knows when their course has unnecessary difficulty. PedDebt helps them see it structurally.

## 2 Theoretical Background

### 2.1 Harel's Necessity Principle

Guershon Harel defines intellectual need as the cognitive and psychological state that motivates a student to seek a particular tool, concept, or approach [4]. The necessity principle holds that instructional activities should be designed to create this state before delivering the content that addresses it. When instructors skip this step—introducing the tool before establishing the need—they incur what we call a *motivational shortcut*.

### 2.2 Anderson's Twelve Criteria

Jeff Anderson's twelve criteria for hands-on modeling activities provide a practitioner-level operationalization of learning theory that is unusually concrete and testable [2]. We build our debt taxonomy around six of these criteria that have direct structural implications for course materials:

C1. **Necessity**: Is intellectual need established before content delivery?

C2. **Scaffolding**: Are cognitive demands decomposed into manageable steps?

C3. **Verification**: Can students self-verify their understanding at each step?

C4. **Directness**: Is content immediately applicable to problems students care about?

C5. **Reproducibility**: Can results be independently replicated?

C6. **Low-floor/high-ceiling**: Is the entry point accessible while allowing deep extension?

Each of these criteria, when violated, contributes to a specific type of pedagogical debt.

### 2.3 Technical Debt as Conceptual Anchor

Fowler's taxonomy of technical debt [7] distinguishes between reckless and prudent debt, and between deliberate and inadvertent debt. We adapt this taxonomy: *deliberate pedagogical debt* occurs when an instructor knowingly makes a shortcut ("we don't have time to motivate this properly"); *inadvertent pedagogical debt* occurs when the shortcut is made without recognizing it as such. Our tool is most valuable for inadvertent debt, which is also the most common.

## 3 Formalizing Pedagogical Debt

### 3.1 Debt Taxonomy

**Definition 1** (Motivational Debt)**.** *A unit of course material $m$ at position $p$ in a course sequence incurs* motivational debt *of magnitude $d_{mot}(m)$ proportional to the number of formal objects introduced in $m$ for which no prior unit in positions $1, \ldots, p-1$ established intellectual need, as measured by problem exposure or contextual framing.*

**Definition 2** (Scaffolding Debt)**.** *A unit $m$ incurs* scaffolding debt *$d_{scaf}(m)$ proportional to the number of new conceptual dependencies introduced simultaneously without decomposition into subtasks.*

**Definition 3** (Verification Debt)**.** *A unit $m$ incurs* verification debt *$d_{ver}(m)$ if it does not include at least one self-checkable problem or comparison between student-generated and model-generated data, as required by Anderson's verification criterion [2].*

**Definition 4** (Total Pedagogical Debt)**.** *The* total pedagogical debt *of a course with units $m_1, \ldots, m_n$ is:*

$$D(C) = \sum_{i=1}^{n} [w_1 \cdot d_{mot}(m_i) + w_2 \cdot d_{scaf}(m_i) + w_3 \cdot d_{ver}(m_i)]$$

*where $w_1, w_2, w_3 > 0$ are weights estimated from expert annotation.*

### 3.2 Measurability of Debt Components

Each debt component is operationally defined in terms of features extractable from course text:

- **Motivational debt** is estimated by detecting formal symbol introductions (new mathematical notation, new defined terms) in a unit and checking whether any prior unit contains problem statements that require those symbols for their solution.
- **Scaffolding debt** is estimated by counting the number of new concepts introduced per unit (using a concept extraction model) and comparing this to an empirically-estimated threshold for working memory capacity.
- **Verification debt** is estimated by detecting the presence or absence of: (a) explicit comparison prompts ("compare your calculated value with your measured value"), (b) self-check exercises, and (c) open-ended investigation questions.

## 4 System Design: PedDebt

### 4.1 Architecture

PedDebt takes as input a directory of course material files (PDF, LaTeX, Markdown, or plain text) and produces a structured JSON report and a human-readable summary. The pipeline has four stages:

1. **Segmentation**: Materials are segmented into units (lectures, problem sets, sections) using structural markers.
2. **Concept extraction**: A fine-tuned NER model identifies formal objects (definitions, theorems, notation).
3. **Debt scoring**: Each unit is scored on all three debt dimensions using the operationalizations described in Section 3.2.
4. **Report generation**: A structured report is generated showing debt by unit, total debt, and flagged items with explanations.

### 4.2 Validation Approach

We validate PedDebt in two ways:

1. **Expert agreement**: We recruit twelve instructors from STEM departments at two institutions and ask them to rate the pedagogical debt of fifteen course units on a 1–5 scale, independently of our tool. We compute Cohen's $\kappa$ between expert ratings and PedDebt scores (binarized at the median). Target: $\kappa \geq 0.60$.
2. **Predictive validation**: Using available student-reported self-efficacy data from end-of-semester evaluations at one institution, we test whether PedDebt scores for individual course units predict lower self-efficacy, controlling for course difficulty ratings.

**Algorithm 1** PEDDEBT: Pedagogical Debt Audit

**Require:** Course materials $\mathcal{M} = \{m_1, \ldots, m_n\}$ in sequence
**Ensure:** Debt report $\mathcal{R}$

$\mathcal{R} \leftarrow \emptyset$
**for** $i = 1$ to $n$ **do**
    $F_i \leftarrow \text{ExtractFormalObjects}(m_i)$
    mot_debt $\leftarrow 0$
    **for** each formal object $f \in F_i$ **do**
        **if** $\nexists j < i$ such that $m_j$ contains a problem requiring $f$ **then**
            mot_debt += 1
        **end if**
    **end for**
    scaf_debt $\leftarrow \max(0, |F_i| - k_{\text{threshold}})$
    ver_debt $\leftarrow \mathbf{1}[\text{no verification prompt in } m_i]$
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{(i, \text{mot\_debt}, \text{scaf\_debt}, \text{ver\_debt})\}$
**end for**
**return** $\mathcal{R}$

### 4.3 The Jeff Anderson Curriculum as a Baseline

Jeff Anderson's PRIMUS paper on linear algebraic nodal analysis [3] is notable because it was designed from first principles to satisfy the necessity principle, the scaffolding principle, and the verification principle. We apply PEDDEBT to this curriculum as a baseline, predicting that it will score near zero on all three debt dimensions. This provides a concrete proof-of-concept: a curriculum known to work well in practice should score as low-debt.

## 5 Open-Source Release

Consistent with Jeff Anderson's no-paywall principle [2], we commit to releasing:

- The full PEDDEBT codebase on GitHub under an MIT license
- All training data for the concept extraction and debt scoring models
- The annotated expert dataset used for validation
- A web interface allowing any instructor to upload their course materials and receive a debt report at no cost

This is not merely an ethical position. It is a scientific one: a tool that lives behind a paywall cannot be validated by the community, cannot be reproduced, and cannot be improved. Openness is necessary for the research to be credible.

## 6 Discussion

### 6.1 Pedagogical Debt as a Shared Language

The most important contribution of this paper is not the tool but the concept. Once instructors have the vocabulary of pedagogical debt, they can reason about their own courses differently. A department can discuss the tradeoff between "getting through the syllabus" and "incurring debt that our students will pay later." A curriculum committee can ask: what is the total pedagogical debt of our degree program, and where is it concentrated?

This is exactly the shift that technical debt vocabulary produced in software engineering: not a solution, but a language that allowed problems to be named, discussed, and addressed systematically.

### 6.2 Limitations and Scope

PedDebt operates on text. It cannot detect pedagogical debt that arises from instructor delivery, classroom dynamics, or student-specific factors. It is a static analysis tool, not a dynamic one. Its scores are estimates, not measurements. We are explicit about these limitations in the tool's output.

We also note that the concept of pedagogical debt is not a critique of individual instructors. Like technical debt in software, most pedagogical debt is inadvertent—the result of time pressure, resource constraints, and the genuine difficulty of designing good learning experiences. The tool is meant to surface problems, not assign blame.

## 7 Future Work

- **Debt repayment strategies**: For each identified debt item, generate a specific suggested revision to the course material.
- **Integration with Project 1**: Use TDG well-formedness violations as inputs to PedDebt, connecting structural curriculum analysis with material-level debt.
- **Student-facing audit**: Adapt the tool to allow students to audit their own study materials and self-diagnose gaps.
- **Multi-institution longitudinal study**: Track debt scores over multiple semesters as instructors revise materials.

## 8 Conclusion

We introduced pedagogical debt as a formal concept grounded in Harel's necessity principle and Jeff Anderson's twelve criteria for effective modeling activities. We defined three operationalizable debt components, built an automated auditing tool, and provided preliminary validation evidence. The tool is released entirely open-source. The goal is to give every instructor a vocabulary and a mirror: a way to see, name, and address the accumulated cost of instructional shortcuts in their own courses.

## References


[1] Cunningham, W. (1992). The WyCash portfolio management system. *OOPSLA Experience Report*.

[2] Anderson, J. (2025). Criteria for developing hands-on mathematical modeling activities. *Jeff Anderson Math Blog*. https://jeffandersonmath.wordpress.com/2025/12/01/criteria-for-developing-hands-on-mathematical-modeling-activities/

[3] Anderson, J. (2024). Linear algebraic nodal analysis: An applied project for a first course in linear algebra. *PRIMUS*. https://doi.org/10.1080/10511970.2024.2369984

[4] Harel, G. (2013). Intellectual need. In K. R. Leatham (Ed.), *Vital Directions for Mathematics Education Research* (pp. 119–151). Springer.

[5] Ferreira-Mello, R., et al. (2019). Text mining in education. *WIREs Data Mining and Knowledge Discovery*, 9(6), e1332.

[6] McAlevey, J., & Lawler, A. (2024). *Rules to Win By: Power and Participation in Union Negotiations*. Oxford University Press.

[7] Fowler, M. (2009). Technical debt quadrant. *martinfowler.com blog*.

[8] Seymour, E., & Hunter, A.-B. (2019). *Talking About Leaving Revisited*. Springer.

[9] Pierce, B. C. (2002). *Types and Programming Languages*. MIT Press.