NO PREP

"Vibe Coders" and Surface Workers

Why Real Skills Still Matter

Includes: Student reading, case study, differentiated questions, teacher key

Name: Period: Date:

"Vibe Coders" and Surface Workers: Why Real Skills Still Matter

A New Term for a Familiar Problem

In early 2025, the term "vibe coding" began circulating in software development communities. It referred to a practice enabled by the emergence of highly capable AI coding assistants: writing code by feel and

Name: **Period:** **Date:**

"Vibe Coders" and Surface Workers: Why Real Skills Still Matter

A New Term for a Familiar Problem

In early 2025, the term "vibe coding" began circulating in software development communities. It referred to a practice enabled by the emergence of highly capable AI coding assistants: writing code by feel and iteration rather than through deep understanding of what the code actually does. A vibe coder prompts an AI tool to generate a function, tests whether it seems to work in obvious cases, and ships it without understanding the underlying logic, edge cases, or failure modes. The term was initially used with affectionate irony by experienced developers who found AI tools dramatically accelerating their work. It quickly became a more serious concern when it became clear that many of the people doing this were not experienced developers using AI to go faster, but beginners using AI to skip the learning that would have made them developers at all. The concept translates beyond coding. A "surface worker" is someone who can produce outputs that look like the result of skilled work, because AI tools handle the underlying complexity, but who has not developed the foundational understanding that would allow them to evaluate, modify, troubleshoot, or defend what they have produced. A surface writer produces polished prose without having developed an argument. A surface analyst produces formatted reports without understanding the statistical methods used. A surface lawyer produces formatted legal briefs without understanding the case law being cited. The output looks professional. The capability that would make the person a genuine professional is absent. The emergence of surface work is a natural consequence of tool capability outrunning skill development. When tools can produce the visible output of skilled work without the underlying skill, there is a persistent temptation to produce the output and present it as evidence of skill. In the short term, this often works. Grades improve. Initial impressions are positive. But surface work is fragile in ways that genuine skilled work is not. It holds up in familiar,

routine situations and fails unpredictably when the situation requires genuine understanding.

What Breaks When You Cannot Go Deeper

The fragility of surface work becomes visible in several predictable situations. The first is when something goes wrong and the practitioner must diagnose the problem. A developer who vibe-coded a feature and did not understand what they wrote cannot debug a failure in that feature without starting over from the AI. A paralegal who submitted an AI-generated research memo cannot defend the citations if a senior attorney questions one. A student who submitted an AI-generated analysis cannot answer follow-up questions about the methodology they supposedly used. In each case, the surface of the work provided no protection against the scrutiny that the situation required. The second is when the situation is novel. AI tools perform best on problems that resemble their training data. When a vibe coder encounters an unusual architecture constraint, an undocumented API, or a system interaction that AI tools have not seen modeled before, they have no resources beyond prompting the AI and hoping it knows. When it does not know, the developer has no analytical foundation to fall back on. This is the 'brittleness in novel situations' that the reading on junior employees discussed. Surface work survives routine situations and collapses in exactly the situations where resilience matters most. The third is when the stakes require accountability. In professional contexts, practitioners are not just expected to produce work. They are expected to stand behind it. An attorney who signs a brief is responsible for the legal reasoning it contains. A financial analyst who submits a valuation is responsible for the methodology. A doctor who orders a treatment is responsible for the clinical reasoning that supports it. AI tools generate output, but they do not provide accountability. The practitioner who submits that output is accountable for understanding it, which means they need to actually understand it. Surface work and professional accountability are fundamentally incompatible.

The Vibe Coding Case in Detail

Software development is worth examining in more detail because it is both an early and a well-documented case of AI tool capability creating surface worker dynamics. Between 2023 and 2025, the percentage of professional code written with significant AI assistance increased dramatically. GitHub Copilot, ChatGPT, and

purpose-built coding assistants became standard tools in development environments, and developer productivity by certain metrics increased substantially. Companies reported faster shipping times, higher code coverage, and reduced time-to-feature for many development tasks. The problems began emerging in code review and in production failures. Senior developers reviewing AI-assisted code found repeated patterns: code that handled the specified requirements but failed to account for edge cases, security vulnerabilities that an understanding developer would have caught, and architectural decisions that worked for the immediate use case but created costly technical debt for the future. The AI had solved the stated problem without the system-level understanding that would have identified the adjacent problems the solution created. A senior developer who understood the system could catch these issues. A vibe coder who did not understand what they had written could not. Several prominent security researchers published analyses in 2024 documenting AI-generated code security patterns in production systems. They found that AI coding assistants consistently generated code with certain categories of vulnerabilities, including SQL injection vulnerabilities, buffer overflow risks, and authentication bypass patterns, at rates that security-conscious manual coding would have avoided. The vulnerabilities were not the AI's fault in any meaningful sense: the AI generated what it was asked to generate. The fault was in deployment by practitioners who lacked the foundational knowledge to evaluate what they were shipping. Vibe coding in security-sensitive applications is not just suboptimal. It is actively dangerous.

Surface Work Across Professions

The surface worker dynamic is not limited to coding. It is visible across every field where AI tools have become capable enough to produce the visible output of skilled work. In marketing and communications, AI can produce fluent, well-formatted copy that sounds authoritative but contains factual errors, misrepresents sources, or makes claims the brand cannot substantiate. A surface-level communications professional submits the copy without the domain knowledge to catch these problems, and the organization is exposed. A genuine professional who understands the domain catches the errors before they cause damage. In academic research, the ability of AI tools to generate coherent literature review sections and statistical write-ups creates opportunities for surface research that mimics the form of

genuine scholarship without the substance. Researchers who use AI to generate sections they do not understand cannot catch the errors AI introduces, cannot engage meaningfully with reviewers who question their methodology, and cannot build on their own work in future studies because they do not understand what they supposedly produced. Several high-profile retractions in 2023 and 2024 were attributed to AI-generated content that contained fabricated citations or incorrect statistical claims that authors had not verified. In financial analysis, AI tools can generate formatted valuations, discounted cash flow models, and market analysis reports. The appearance of rigor is built in. But a surface-level analyst who submits an AI-generated valuation without understanding the assumptions embedded in the model is making claims they cannot defend when a sophisticated investor or a due diligence team asks them to explain their methodology. The professional who cannot go below the surface of their own work is professionally vulnerable in every high-stakes interaction their work encounters.

Why Real Skills Remain the Foundation

The answer to the surface work problem is not to avoid AI tools. It is to ensure that genuine foundational skill is developed before or alongside AI tool use, so that the practitioner can evaluate, modify, and be accountable for AI output rather than simply presenting it. This is the difference between an experienced developer using AI to code faster and a vibe coder using AI to code without learning. The experienced developer uses the tool to amplify documented capability. The vibe coder uses the tool to simulate capability they do not possess. Foundational skill also provides resilience that surface work cannot. A developer who understands code can debug AI-generated code, catch its security failures, and adapt it when the situation requires something the AI has not seen. A researcher who understands statistical methodology can evaluate AI-generated analysis, identify incorrect assumptions, and defend their findings. A writer who understands argument construction can use AI to polish drafts while maintaining accountability for the ideas. In each case, the foundational skill is what makes AI assistance genuinely useful rather than a liability. The students and workers who will navigate the AI transition most successfully are those who treat foundational skill development as non-negotiable, regardless of what AI tools can do. They understand that the tool is only as valuable as the judgment directing it, that surface work is professionally

fragile, and that the situations in which skill actually matters are exactly the situations that cannot be offloaded to AI without accountability. Real skills remain the foundation not because AI tools will go away, but because the quality of what AI tools can help you do depends entirely on the depth of understanding you bring to directing them.

Key Information: Surface Work and Real Skills

Detail	Information
Vibe Coding	Writing code by feel and AI iteration without understanding the underlying logic or failure modes
Surface Worker	Produces AI-assisted output that looks skilled but lacks foundational understanding to evaluate or defend it
When Surface Work Fails	Diagnosis (debugging), novel situations, and high-stakes accountability contexts
GitHub Copilot / AI Coding Tools	Increased developer productivity metrics while creating new code review and security challenges
2024 Security Research	AI coding assistants generated SQL injection, buffer overflow, and auth bypass vulnerabilities at concerning rates
Surface Work in Research	AI-generated lit reviews with fabricated citations; several retractions in 2023-2024
Surface Work in Finance	Formatted AI valuations with unverifiable assumptions, vulnerable under due diligence scrutiny
Foundation vs. Surface	Foundational skill allows evaluation, debugging, modification, and accountability for AI output
Key Principle	AI tools amplify genuine capability; they simulate capability that does not exist yet

Case Study: Dev, 23, Junior Software Developer at a Startup

Dev had taught himself to code primarily through AI assistance, using ChatGPT and GitHub Copilot to build portfolio projects that landed him a junior developer role at a five-person startup. Three months in, a production bug brought down a key feature for six hours. The CTO asked Dev to diagnose it. Dev spent two hours prompting AI tools with error logs, getting back suggestions, trying them, and getting new errors. He could not fix it. The CTO sat with him for thirty minutes, traced the problem to a database query Dev had vibe-coded six weeks earlier, and fixed it in ten minutes. 'Do you understand what was wrong?' the CTO asked. Dev understood the fix as described. He did not understand why the original code had failed or what the fix actually changed. 'I'm not angry,' the CTO told him. 'But you need to actually learn how this works. The AI can help you build things fast. It can't debug for you under pressure, and it can't care about what breaks.' Dev enrolled in a structured computer science course that night.

1. What does the CTO's statement 'the AI can help you build things fast, but it can't debug for you under pressure' reveal about the fundamental limitation of surface coding?

2. The reading says surface work 'holds up in familiar, routine situations and fails unpredictably when the situation requires genuine understanding.' How does Dev's story illustrate this pattern?

Multiple Choice

1. According to the reading, what was the original, ironic use of the term 'vibe coding,' and how did that change?

a) It was used critically by educators, then adopted proudly by developers who disagreed

b) It was a marketing term coined by AI tool companies to describe their newest productivity features, and developers later picked it up

c) It was used by beginners to describe confusion, then adopted by experts as a compliment

d) It was used affectionately by experienced developers going fast, then became a serious concern as beginners used it to skip learning

2. What does the 2024 security research finding demonstrate about AI-generated code?

a) AI assistants generated certain security vulnerability patterns at concerning rates when used by practitioners without foundational knowledge to catch them

b) Security vulnerabilities in AI-generated code are always obvious to an experienced reviewer and are easy to identify and remove during an ordinary code review

c) AI tools refuse to generate code with known security vulnerabilities

d) AI-generated code is more secure than manually written code on average

3. According to the reading, what is the key difference between an experienced developer using AI and a vibe coder using AI?

a) The experienced developer uses AI only for simple, repetitive tasks, while the vibe coder uses it for the genuinely complex ones

b) The experienced developer uses AI to amplify documented capability, while the vibe coder simulates capability they do not have

c) The experienced developer spends more time per task to ensure AI output quality

d) The experienced developer uses more expensive AI tools with better security features

Matching

Write the letter of the correct definition on the line next to each term.

Term	Definition
1. ___ Vibe coding	A) The systematic examination of code by another developer to identify errors, security issues, and design problems
2. ___ Technical debt	B) Writing code through AI iteration and feel without understanding the underlying logic or failure modes
3. ___ SQL injection	C) A thorough investigation of a company, deal, or work product before making a major decision or commitment
4. ___ Due diligence	D) A security vulnerability where an attacker can manipulate database queries through unvalidated user input
5. ___ Code review	E) Hidden future costs created by shortcuts in code or system design that must eventually be addressed

Fill-in-the-Blank

1. A surface worker can produce outputs that look like the result of skilled work but has not developed the _____ understanding that would allow them to evaluate, modify, or defend what they have produced.

2. AI tools generate output, but they do not provide _____, which means the practitioner who submits that output must actually understand it.

3. The experienced developer uses AI as a tool to _____ documented capability, while the vibe coder uses it to simulate capability they do not yet possess.

Short Answer

Answer each question in 2-4 complete sentences.

1. The reading identifies three specific situations in which surface work breaks down. Describe two of them and explain why surface work is particularly vulnerable in each.

_____

_____

_____

2. The reading says that foundational skill 'provides resilience that surface work cannot.' Using one specific example from the reading, explain what that resilience looks like in practice.

_____

_____

_____

3. The reading ends by saying real skills remain the foundation 'not because AI tools will go away, but because the quality of what AI tools can help you do depends entirely on the depth of understanding you bring to directing them.' Explain that argument in your own words.