

Beyond the Algorithm: Ethical Risk Assessment in Tech Startups

Devyn Miller
Fowler School of Engineering
Chapman University
Orange, United States
devmiller@chapman.edu

Keywords—artificial intelligence, cloud computing, ethical considerations, risk management, security and privacy

I. THE RACE FOR INNOVATION: AT WHAT COST?

AI-powered tools and cloud-based development environments have completely changed how software engineers work. They make coding faster, remove friction in setting up development environments, and, in some cases, even introduce automation into fields that historically required human intuition. But with these advancements come risks—some obvious, some more subtle, and all requiring careful thought. The reality is that the rush to innovate often means that growth takes a higher priority than ethical concerns. By the time risks are recognized, they're either too deeply embedded to fix or ignored in favor of maintaining momentum.

In this paper, I will discuss three startups—Cursor, StackBlitz, and Figure.ai—that are pushing the boundaries of AI-assisted development, cloud-based programming environments, and robotics, respectively. Each one presents real ethical challenges: Cursor risks making developers dependent on AI-generated code while introducing copyright and bias issues, StackBlitz expands security vulnerabilities in ways that are poorly understood, and Figure.ai is working on humanoid robots that could pose serious safety and labor concerns. If I were working at any of these companies, I would have clear ethical lines I wouldn't want to cross—points where I would have to raise concerns, escalate internally, and, if necessary, take those concerns outside the company.

II. THE ETHICAL PRICE OF TECHNOLOGICAL ADVANCEMENT

Every major advancement in software engineering comes with trade-offs. Cloud computing has made development more flexible but has also introduced a whole new level of data privacy concerns. AI-generated code promises faster development cycles but raises questions about bias, intellectual property, and quality degradation. Humanoid robots can automate tasks in dangerous environments, but they also threaten human jobs and introduce safety risks.

Startups often operate in a way that makes ethical regulation difficult. The culture is usually built around rapid iteration and a “go big or go home” fast-paced mindset, where the priority is proving technical feasibility and market fit rather than pausing to assess long-term consequences. The ISO 31000 risk management framework [3] provides a structure for ethical assessment and breaks risk down into identification (what could go wrong?), analysis (how bad could it be?), and evaluation (how much risk is acceptable?). However, in practice, startups tend to cut corners on this process until a crisis forces them to pay attention.

III. THE HIDDEN DANGERS OF AI-ASSISTED CODING

Cursor is an AI-powered coding assistant that integrates directly into development environments, predicting and generating code based on what a developer is working on. It's built on a fork of Visual Studio Code and trained on large datasets of existing code. This allows it to autocomplete functions, suggest optimizations, and even write entire files and blocks of logic. The company has received \$8 million in funding from OpenAI and is positioning itself as a way to make coding faster and more efficient. However, with this comes serious risks that aren't easy to mitigate [1][2][6].

A key issue is over-reliance on tech, leading to the loss of our own skills. If a developer (especially a junior one) starts leaning too heavily on Cursor's suggestions, they may never fully develop their ability to think through problems or write secure, optimized code on their own. A scenario where companies begin expecting engineers to rely on AI rather than understanding the logic themselves isn't far-fetched. This could easily result in software filled with security vulnerabilities because developers never actually understood the code they deployed.

There's also the issue of copyright and licensing ambiguity. Because Cursor is trained on vast amounts of existing code, there's no clear way to ensure that it isn't just regurgitating proprietary or GPL-licensed snippets. If a company unknowingly ships code that Cursor generated from a copyrighted source, they could face severe legal and financial repercussions—except, at that point, it would be

nearly impossible to prove where the issue started.

Bias is another concern. AI-generated code is only as good as the dataset it was trained on. If Cursor's training data favors certain programming paradigms or styles over others, it could create a self-reinforcing loop where alternative approaches get drowned out. If, for example, it overwhelmingly suggests Python code because of Python's dominance in open-source repositories, languages with less representation could slowly be pushed further to the margins.

If I worked at Cursor, I would feel ethically obligated to speak up if the company refused to implement any kind of tracking for where code was sourced from or if there was clear evidence that security vulnerabilities caused by AI-generated code were becoming common but dismissed. For example, if the junior devs started struggling to debug AI-generated code which led to production bugs or security flaws, I would push to investigate and assess potential skill degradation. I might also suggest that the junior devs (and the entire dev team, for that matter) take annual assessments to ensure that they are staying up-to-date on the latest security and best practices needed to succeed at Cursor. Now, let's say security researchers traced the flaw back to Cursor's AI-generated code, but there was no way to determine whether that code was copied from an open-source project under a restrictive license or if the AI introduced the error itself. If leadership continued to ignore these concerns, I would escalate externally—potentially to open-source oversight organizations or software engineering ethics committees. It is important that AI-generated code is held to the same accountability standards as human-written code.

IV. DATA SOVEREIGNTY AND SECURITY GAPS IN CLOUD-BASED IDEs

StackBlitz is working to change how developers write and run code by allowing full Node.js environments to be executed directly in a browser using WebAssembly-based virtualization. This means that developers can start writing code instantly without needing to install dependencies or configure local environments. While this removes considerable friction, it also introduces serious security risks that haven't been fully explored [4][7][9].

One of the biggest risks is browser-based security vulnerabilities. StackBlitz promotes its architecture as "unmatched security" because everything runs in the browser. In reality, this execution model introduces entirely new attack vectors. For example, a bad actor could exploit WebAssembly memory isolation flaws to pull sensitive data from a running project, such as API keys or database credentials. Another issue is automated package installations, which could allow supply chain attacks to slip through unnoticed. Data sovereignty is another critical issue. Since StackBlitz enables real-time collaborative coding, project data could easily be routed through different jurisdictions,

potentially violating data privacy laws like GDPR [10]. If, for instance, a healthcare startup in the US builds a patient management system using StackBlitz, but the session data is processed in a non-compliant region, they could be held legally responsible (even though StackBlitz doesn't provide a clear way for developers to control where their data is stored).

If I worked at StackBlitz, I would have serious ethical concerns if I found out that the company was aware of data residency issues but failed to address them, or if security vulnerabilities were continuously downplayed due to business pressures. For example, imagine a fintech company in France using StackBlitz to build a banking app, but they don't realize that their project data is being processed on servers outside the EU and thereby be a potential violation of GDPR regulations. Then, imagine a few months down the road, the company faces fines and legal scrutiny after customers' financial data is mishandled. And say that StackBlitz knew about this risk but continued marketing itself as a "secure" solution without providing clear data handling/storage options for developers. In the event of such an issue, I would first reach out to the leaders at StackBlitz and, at the very least, make sure that developers are being informed on where their data is processed; however, if leadership ignored these concerns internally, I would escalate things and reach out to independent security organizations such as CERT/CC for responsible disclosure.

I would not be satisfied by a mere legal fix; I would push for StackBlitz to create user-controlled data region settings to allow companies to choose where their code and data are stored. Without these safeguards, StackBlitz isn't just innovating; it's putting companies at legal risk without their knowledge.

V. THE HUMAN COST OF INTELLIGENT ROBOTICS: SAFETY AND LABOR CONCERN

Figure.ai is developing AI-powered humanoid robots designed for industrial and healthcare applications with the goal of automating labor-intensive tasks. The company has a valuation of \$2.6 billion and is pushing hard to get functional prototypes into real-world environments. While this company has great potential, there is also significant risk[8][5].

One of the most immediate concerns is physical safety in shared workspaces. Early demonstrations show Figure.ai robots working alongside humans, but there aren't obvious proximity sensors or safeguards in place. A miscalculation or sudden hardware failure could lead to serious injuries, especially in industrial settings where robots handle heavy objects. If safety testing isn't of top priority, the company could end up deploying robots that threaten the safety of its workers.

Labor displacement is another major concern. While Figure.ai frames its work as enhancing human capabilities,

rather than replacing it, the reality is that its technology is designed to replace human labor. If the company doesn't invest in transition programs or worker retraining, it could contribute to mass unemployment, especially in industries that rely on manual labor.

If I worked at Figure.ai, I would find it ethically unacceptable if safety engineers' recommendations were ignored in favor of hitting deployment deadlines or if the company attempted to misrepresent its impact on labor markets. I would first advocate for a mandatory safety testing phase before any large-scale rollout. If those concerns weren't taken seriously internally (e.g., the team still rushed deployment), I would escalate to OSHA or IEEE ethics committees. Similar to the Cursor data security example, I believe that a responsible resolution shouldn't simply "band-aid" the issue; it should involve creating strict updates on safety auditing requirements to ensure that robots can't be deployed without extensive real-world testing.

VI. ETHICS IN AI-POWERED STARTUPS: A CALL TO ACTION

Cursor, StackBlitz, and Figure.ai all represent major innovations in software and AI, but each one comes with ethical challenges that can't be ignored. If engineers don't push for responsible innovation early on, these risks will only grow harder to contain. Being in a fast-moving startup doesn't excuse ignoring long-term consequences. If companies won't self-regulate, it's up to the engineers working on these technologies to step in before the problems become too big to fix.

For AI-powered coding tools like Cursor, this means embedding source tracking and security warnings to prevent legal and cybersecurity issues. StackBlitz should introduce clear data residency controls and proactive security audits so that developers aren't unknowingly exposed to compliance risks. Figure.ai should require meticulous third-party safety testing and labor impact assessments to ensure its robots enhance, rather than replace, human workers.

If companies do not act, the people building these technologies (e.g., engineers) must be the ones to demand better. The cost of inaction isn't just theoretical. It's real, and it's happening now.

REFERENCES

- [1] A. B. Allen, "Startup breakdown: Cursor AI," Substack.com, Jul. 16, 2024. Available: <https://trailruncapital.substack.com/p/startup-breakdown-cursor-ai> (accessed Feb. 22, 2025).
- [2] "Cursor - the AI code editor," Cursor.com, 2023. Available: <https://www.cursor.com/en> (accessed Feb. 22, 2025).
- [3] International Organization for Standardization, "ISO 31000:2018," ISO, 2018. Available: <https://www.iso.org/standard/65694.html> (accessed Feb. 21, 2025).
- [4] "What is StackBlitz?," Stackblitz.com, Mar. 19, 2024. Available: <https://developer.stackblitz.com/guides/user-guide/what-is-stackblitz> (accessed Feb. 22, 2025).
- [5] T. Underdog.io Team, "Deep dive into careers and working at Figure.ai," Underdog.io, Feb. 14, 2025. Available: <https://underdog.io/blog/figure-ai-careers> (accessed Feb. 22, 2025).
- [6] "Series A and magic — Cursor - the AI code editor," Cursor.com, 2024. Available: <https://www.cursor.com/en/blog/series-a> (accessed Feb. 22, 2025).
- [7] "StackBlitz — founders, business model, growth competitors," The Brand Hopper, Jun. 03, 2023. Available: <https://thebrandhopper.com/2023/06/03/stackblitz-story-founders-business-model-growth-competitors/> (accessed Feb. 21, 2025).
- [8] "Figure," FigureAI, 2024. Available: <https://figure.ai> (accessed Feb. 22, 2025).
- [9] "StackBlitz — instant dev environments — click. code. done. - StackBlitz," Stackblitz.com. Available: <https://stackblitz.com/> (accessed Feb. 21, 2025).
- [10] Wikipedia Contributors, "General data protection regulation," Wikipedia, Apr. 15, 2019. Available: https://en.wikipedia.org/wiki/General_Data_Protection_Regulation (accessed Feb. 21, 2025).