

Steven Nguyen's Ethical Framework for Technology (Alpha)

Steven Nguyen

Date: 3rd, December 2025

CSED 430 : CS Teaching Practices
Professor Christine Liebe

When I think about the study of computer science today, it's not just code and computers, but tools that have the amazing ability to shape how people around the world live, learn, and connect. It is empowering, but also dangerous when the developers creating the software ignore the side of humanity. During my journey through this course my own experiences as a junior developer, I've reinforced my ethical framework that I want to carry into my career. My personal view rotates around three priorities: care and human impact over just optimization, committing to equality and inclusion in data and deployment, as well as designing software with long-term responsibility to the planet and those who live on it. With these core values, I want to make the decisions about what I create, what I refuse to build, and how I can educate others to think about computing.

To begin, my ethics start with care and human impact, not just the technical success. The majority of tech culture prioritizes optimization with the goal of faster models, higher engagement, and more automation. But when we take a look at real examples of algorithmic decision-making or social media recommendation systems [Hill 1], optimizing the metric becomes ignoring the people behind the data. I believe that CS ethics framework must flip that order. Before asking "Can we improve accuracy by 2%?" a question we should ask is, "Who gets hurt or helped if this system works as intended?" This means treating harm prevention and user well-being as non negotiables, not it would be nice to have. It also means saying no to some uses of technology, even if it is technically and genuinely impressive. When anything is built on manipulation and/or exploitation, as a developer, it shows up in the small decisions. The features we log, how to design defaults, and whether developers assume people are just "users" or actual humans with lives far beyond the software.

The next step is justice and inclusion need to be built into the entire lifecycle of a system, not only checked at the end. The COMPAS recidivism example [Larson, J., Mattu, S., Kirchner, L., & Angwin, J. 2] and the algorithmic fairness work showed how easily models can reproduce and amplify the existing inequalities when trained on biased data. It's not enough to say "we don't use race as a feature" if everything in the dataset is a proxy for race, class, or geography. We should treat these questions as "Who was left out of this dataset?" and "Who carries the risk if this fails?" as core test questions. This may mean being skeptical of being neutral or objective claims, especially when the outputs harm disproportionately to everybody. It also means valuing transparency as in explaining what a system is doing in a way that non-technical people can understand as well as making room for those people to challenge it. Inclusion isn't just a checkbox, but it is a need that shapes how data is collected, how models are evaluated, and who is at the table when the models are deployed.

Another important piece is responsible computing and to extend beyond immediate users and into long-term social and environmental impacts. It's easy to say that digital systems are "clean"

due to the fact that they live on screens, but with the energy cost of large-scale computing, e-waste, and how optimization culture pushes toward endless growth with no real sense of limits [Wattsense 3]. My framework challenges that by asking how a system will age, “Will it lock institutions into patterns that are hard to reverse?”, “Will it encourage more consumption or more extractive behavior?” or “Can it help share knowledge and reduce harm over time?” I prefer simple, more explainable systems when they are able to complete the task and question whether every problem really needs an AI-powered solution. Many times the ethical choice is to improve a low-tech process, not to build a faster model.

My framework didn’t come overnight. It came from learning about uncomfortable examples in class from biased risk scores, addictive feeds, and environmental costs as well as from my own daily life as someone learning to build software. I’ve felt the pull of ‘making it work’ and the pressure to chase performance or optimization. When I take a step back, the same realization hits me, if I don’t consciously choose my values as a developer and instead go with someone else’s values, usually profit and growth, it will never align with my beliefs. Reflecting on my background and communities that has made me much less interested in “neutral” technology and much interested in technology that keeps us accountable to real people, in real contexts.

Because this class is about teaching practices as well as personal ethics, I also think a lot about how to bring these ideas into K-12 classrooms [Brooks 4]. If students only view CS as problem sets and syntax, that idea will narrow their view into the software they build later on. Ethics should show up as questions in everyday lessons and lectures, not just the one-off “ethics unit.” For example, when teaching a sorting algorithm, we can ask, “What data are we sorting?”, “Who decided the labels?” and “What happens if the labels are unjust?” When we design applications, they can talk about best-case and worst-case outcomes and write a short reflection about who might be excluded or harmed. K-12 ethics teaching should also highlight student agency, not just being careful online, but you will one day have power, so how do you want to use it? If students practice thinking critically like that early, they are more likely to grow into developers who see ethics as part of the work, not an optional add-on.

In the end my ethical framework for technology comes down to a simple idea that computer science studies and the outcomes should serve people, not the other way around. Personally it means prioritizing care and human impact, committing to justice and inclusion, and honoring long-term responsibility to communities and the planet! I want them to act like filters I run my future work through and if a product can’t pass those values, then no amount of clever code will make it truly “good.”

References

- [1] Hill, C. (2025, June 26). The social media metrics to track in 2025 (and why). Sprout Social.
<https://sproutsocial.com/insights/social-media-metrics/>
- [2] Larson, J., Mattu, S., Kirchner, L., & Angwin, J. (2016, May 23). How We Analyzed the COMPAS Recidivism Algorithm. ProPublica.
<https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm>
- [3] Energy optimization of buildings: challenges and solutions - Wattsense. (2025). Wattsense.
<http://wattsense.com/blog/energy-management/energy-optimization-of-buildings-challenges-and-solutions/>
- [4] A. Brooks, “Agile Ethics: A Low Stakes, Skills-based Framework for Teaching CS Ethics,” pp. 492–498, Jul. 2024, doi: <https://doi.org/10.1145/3649217.3653539>.
- [5] Computer Society, “Code of Ethics | IEEE Computer Society,” *Computer.org*, 2017.
<https://www.computer.org/education/code-of-ethics>