# CLOSING THE AI RESPONSIBILITY GAP WITH THE CODE OF ETHICS

**Don Gotterbarn, Marty J. Wolf**

East Tennessee State University (United States), Bemidji State University (United States)

gotterba@gotterbarn.com; mjwolf@bemidjistate.edu

**EXTENDED ABSTRACT**

Like many types of technology, early computing technology was developed, at least in part, to support and advance military goals. For some, this put the technology on morally shaky grounds. As computing technology advanced and became more generally accessible, it increasingly was used by some to intentionally cause harm outside the military realm. Others used the technology for seemingly innocuous purposes and caused harm that was unseen by them, but felt by others. These situations sometimes led to an "ethical hysteria" where people used ethical expressions in unhelpful ways or developed tools that only offered partial help or, worse, went down the wrong path.

An early attempt to address harm caused by computing was the introduction of the notion of "Software Engineering" in 1968 at the first NATO Software Engineering Conferences. The goal was to indicate the professional technical skills that were needed to solve the "software crisis" of failed software (Naur & Randell 1968). The solutions tended to be technical and centered on the processes used to develop software; that software development process should follow an engineering model. This led to a report defining "software engineering techniques" to resolve the "software crisis." Over time this foundation was added to by the development of various software life cycles and developing software case tools were supposed to help create better, less harmful software. Yet, it wasn't until the 1990s that the IEEE-CS and the ACM publicly addressed the need for software engineering ethics standards in the IEEE/ACM Software Engineering Code of Ethics (Gotterbarn et al. 1997).

This is the start of a pattern (described in the full paper) whereby each new computing ethical awakening repeats mistakes from earlier eras. There is a pattern of concerns present in many cases reaching back to the first software crisis and now to AI. The most pressing concern stemming from these "crises" for computing ethics is: what structural support can best help those who want to use computing in positive ways and yet have difficulty doing so?

In this paper we focus on AI. There are a number of things that have changed since the last major ethical awakening in computing. First, and importantly, there is a broad range of people who are at the table discussing the ethics and social implications of AI. The different perspectives brought by philosophers, ethicists, linguists, sociologists, computer scientists, mathematicians, data scientists, humanities scholars and so many more, all come to bear on identifying actual and potential harms of computing technology. Further, they offer suggestions on different ways to prioritize those harms.

The other major change is that there is a greater misalignment between corporate interests and "good AI" (AI that does good for society) than with previous ethical awakenings. Developing software that met specification was good for business. Having developers understand the same software lifecycles added efficiencies to software development and contributed to the

corporate bottom line. These days, while products like ChatGPT may be good for OpenAI's bottom line, there is clear harm being caused to other businesses, to education, and even to democracy itself, and this is on top of harms caused by its development (see Bender et al. 2021).

With experts from these different fields coming to bear on AI, significant new problems have been identified, including biased decisions, misclassification, overgeneralizations, lack of contextual understanding, adversarial attacks, and unintended consequences. When a person is responsible for these kinds of problems, they are held accountable or blamed as the cause. When these judgments are left to an AI system there is a difficulty assigning responsibility for problems or bad decisions. Adreas Matthias has called this situation where no human can be morally responsible or liable for a machine's behavior the "responsibility gap" (2004).

Many such as (Goetze 2022, Kiener 2022, Rubel 2019, Santoni de Sio and Mecacci 2021, Tigard 2021) have addressed facets of the responsibility gap, including when and whether it exists. We consider one of them here and the others in the full paper. Munch et al. (2022) argue that there are times when the responsibility gap is good even in the absence of psychological dilemmas. They argue that holding a person responsible for wrong-doing causes that person some amount of harm. In situations where an automated system is equally as effective as a person in making decisions of consequence, no person comes to bear the harm associated with wrong-decision making when there is a responsibility gap. Our contention is that this position and other arguments surrounding the notion of responsibility gaps misdirect discussions about responsibility.

A major problem is that some of the discussion about the responsibility gap has focused on a limited sense of responsibility, one related to blame in some form. This same sense of responsibility was used during the software crisis of the 1960s to blame developers for the failure to develop reliable systems. The result was a system that emphasized finding a program's errors rather than people learning how to take action to decrease the risk of similar errors in future programs. Further, blame would frequently be passed to the client for inadequately specifying requirements. Responsibility for the moral issues surrounding the requirements and the way a system developed were not considered. There was significant effort to develop precise technical requirements as a problem solution.

John Ladd called this responsibility "negative responsibility," a responsibility assigned after the fact. It primarily tries to excuse people from moral responsibility, a legal search for extenuating circumstances, for example. He champions a positive sense of responsibility for what ought to be done. Unlike negative responsibility which tends to be direct, positive responsibility can be indirect. Ladd argues that pointing to the technology does not remove this sense of positive responsibility. Positive responsibility engages with the prospect that things might happen. Guidance from Principle

2.2 of the ACM Code of Ethics and Professional conduct is clear: "Professional competence starts with technical knowledge and with awareness of the social context in which their work may be deployed." Computer professionals are responsible for applying standards within their profession and attempting to avoid anticipatable negative ethical impacts of their work.

The AI responsibility gap discussion misses this opportunity to engage with positive responsibility. Underlying the AI responsibility gap is an assumption of a causal chain looking for a particular event. Implicitly, this makes a standard responsibility denial move, appealing to the complexity of the system, much easier and misses an opportunity to change the behavior of the system's developer. For AI systems, an appeal to the responsibility gap can be used to justify the

development of systems that cause harm. (Google initially did this when Safiya Noble pointed out how their search completion algorithm reinforced racist stereotypes.) The advocacy/acceptance of such positions are inconsistent with ethical computing.

Professional responsibility also includes premeditated concern for the consequences of one's actions on others. This kind of approach is anticipated by the ACM Code of Ethics and Professional Conduct and advocated for by Goterbarn et al. (2022). We highlight another approach next that is applicable to AI and has AI developers focus on how the AI systems are built and how decisions are made by AI developers.

One of the authors participated in the development of several international police criminal intelligence systems. Ethical issues were considered and mitigated in the design of the projects rather than after the systems were built. Implicit and explicit values in design choices and the intentional and unintentional value choices made in technology development were made explicit.

The project set up design guidelines so that solutions to issues were designed before the system was implemented. This helped to change the focus from an overview of ethics (e.g. informed consent) to a deeper focus on the technologies, their impact on society, and the ethical issues that the different technologies may raise.

For example, in order to identify bias in decisions and inferences made from the data, the design process included transparency tools such as understandable process logs and logging mechanisms that tied change details to a user. To ensure the integrity of the data and increase the reliability of inferences made from it, the design included a 'reliability tag' attached to all data.

Not all AI systems make bad decisions, and many are designed to mitigate risks through rigorous testing, validation, and ongoing monitoring. While these after the fact tools are important, positive responsibility calls for more. Using tools like the ACM Code of Ethics and Proactive CARE (Gotterbarn et al. 2022) is essential to ensure responsible and ethical AI deployment. The Code of Ethics provides guiding principles that lead to better design decisions and help developers use positive responsibility to reduce or even eliminate the AI responsibility gap.

**KEYWORDS:** Responsibility, Responsibility Gap, Artificial Intelligence Responsibility, ACM Code of Ethics.

**REFERENCES**

ACM Code of Ethics and Professional Conduct (2018). https://www.acm.org/code-of-ethics

Bender, Emily M, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. (2021). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT '21). Association for Computing Machinery, New York, NY, USA, 610–623. https://doi.org/10.1145/3442188.3445922

Goetze, T. (2022). Mind the Gap: Autonomous Systems, the Responsibility Gap, and Moral Entanglement. FAccT '22.

Gotterbarn, D. (2001). Informatics and professional responsibility. Science and Engineering Ethics, 7, 221–230.

Gotterbarn, D., M.S. Kirkpatrick, and M.J. Wolf. (July, 2022). ”From the page to practice: Support for computing professionals using a code of ethics,” ETHICOMP 2022.

Don Gotterbarn, Keith Miller, and Simon Rogerson. (1997). Software engineering code of ethics.

Commun. ACM 40, 11 (November 1997), 110-118. http://doi.org/10.1145/265684.265699

Kiener, M. (2022). Can we Bridge AI’s responsibility gap at Will?. Ethic Theory Moral Prac 25, 575–593. https://doi.org/10.1007/s10677-022-10313-9

Ladd, J. (1988). Computers and Moral Responsibility: A Framework for an Ethical Analysis, in: Gould, Carol (ed.) The Information Web: Ethical and Social Implications of Computer Networking, Westview Press.

Matthias, A. (2004). The responsibility gap: Ascribing responsibility for the actions of learning automata. Ethics Inf Technol 6, 175–183. https://doi.org/10.1007/s10676-004-3422-1

Munch, L., Mainz, J. & Bjerring, J.C. The value of responsibility gaps in algorithmic decision-making. Ethics Inf Technol 25, 21 (2023). https://doi.org/10.1007/s10676-023-09699-6

Naur, P. & Randell, B., eds. (1968). Software Engineering: Report on a conference sponsored by the NATO Science Committee. http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF

Rubel, A., Castro, C. & Pham, A. (2019). Agency Laundering and Information Technologies. Ethic Theory Moral Prac 22, 1017–1041. https://doi.org/10.1007/s10677-019-10030-w

Santoni de Sio, F., Mecacci, G. (2021). Four Responsibility Gaps with Artificial Intelligence: Why they Matter and How to Address them. Philos. Technol. 34, 1057–1084. https://doi.org/10.1007/s13347-021-00450-x

Tigard, D.W. (2021). There Is No Techno-Responsibility Gap. Philos. Technol. 34, 589–607. https://doi.org/10.1007/s13347-020-00414-7