Ethical issues are not exclusive to AI systems

Otávio Santos Gomes[1], **Gláucia Braga e Silva**[1], **Érica Ferreira de Souza**[2]

[1] Federal University of Viçosa (UFV)
Florestal, MG, Brasil

[2]Federal University of Technology - Paraná (UTFPR)
Cornélio Procópio, PR, Brasil

{otavio.s.gomes, glaucia}@ufv.br, ericasouza@utfpr.edu.br

Abstract. *The increasing integration of software into daily life underscores the urgency of addressing ethical issues in software development. While significant attention has been given to Artificial Intelligence (AI), ethical concerns are equally relevant to other types of software due to their widespread presence and societal impact. This position paper highlights the importance of considering the social, human, and ethical aspects of software systems already embedded in our daily lives. To achieve this, we examine ethics from a philosophical perspective and argue why software developers must adopt an ethical perspective throughout the development process. Additionally, we explore the mirage of software neutrality and discuss perspectives of the field.*

1. Introduction

Technology increasingly impacts human relations and actions, extending well beyond the online sphere. The use of technological tools in diverse human activities has placed computers and their software at the center of human life.

A significant challenge in this context is addressing software's human and social dimensions during its development [Gustavsson and Penzenstadler 2020]. Incorporating this social dimension requires substantial effort from both industry [Mitchell et al. 2022] and academia [Stahl et al. 2016] to emphasize the importance of considering biases, impacts, and harms that a software system may cause, beyond its technical characteristics, such as efficiency and functionality. Human and social aspects, including privacy, trust, sustainability, accessibility, and others, should be regarded as equally important as the technical issues traditionally prioritized. These and some technical issues can be grouped into a category of ethical issues, representing the software's ethical dimension.

Currently, the research on the ethical dimension of software development frequently focuses on issues related to Artificial Intelligence (AI), especially because of the increasing impact this technology has on society. While numerous valid and important discussions are occurring within this domain, it is equally critical to consider ethics in all types of software development, not only in systems that use AI techniques. Although AI introduces new approaches to solving problems and demands distinct development strategies, non-AI software remains the norm and continues to have the most significant impact on people's lives today. Many software applications developed in the past, such as medical and financial applications, have caused tangible harm, including discriminatory practices, environmental damage, or exclusionary experiences.

In this context, is it reasonable to accept that ethical issues are exclusive to AI? Throughout history, ethical concerns have arisen wherever humans exist in society. In the realm of software, these concerns are even more pressing because unresolved ethical issues can affect far more people than a simple human error otherwise would.

This position paper aims to highlight the importance of discussing the ethical aspects of all types of software systems that are already embedded in our daily lives. To achieve this, we discuss ethics from a philosophical perspective and why software developers must approach the development process with an ethical perspective. Furthermore, we present a discussion about the mirage of software neutrality.

The following sections discuss the concept of ethics (Section 2), its relationship to software (Section 3), and perspectives on the topic (Section 4).

2. What is Ethics?

Ethics encompasses a complex framework of moral principles defined by society to facilitate harmonious coexistence within a given context and era [Johnson and Smith 2021]. As such, ethics is inherently contextual and subject to evolution and change, making it challenging to be fully captured and mapped [Shafer-Landau 2018, Aberkane 2018].

Within the realm of philosophy, ethics is a multifaceted discipline that can be further categorized into three main branches: metaethics, normative ethics, and applied ethics. *Metaethics* concerns itself with examining semantic and logical notions in ethics. *Normative ethics* focuses on developing theories and frameworks that enable the evaluation of whether an action is morally acceptable. Lastly, *Applied ethics* refers to applying normative theories within specific contexts [Marturano 2002].

The focus of this paper lies in the application of ethics within the domain of software development, aligning it with the branch of applied ethics. As applied ethics draws upon normative ethics, it is pertinent to delve into prominent ethical theories. These theories provide distinct perspectives through which individuals can discern and evaluate the ethical nature of actions. The well-known ethical theories that warrant discussion in this context include deontology, teleology or consequentialism, and virtue ethics [Stahl et al. 2016].

Deontology posits that the morality of an action is determined by the agent's intention [Stahl et al. 2016]. According to deontology, ethicality is governed by a set of rules or duties, and action is deemed ethical when it adheres to these established rules. Conversely, if these rules are violated, the action is considered unethical [Ananny 2016].

Teleology or Consequentialism asserts that an action's ethical nature is determined by its consequences. Among the various branches of this theory, utilitarianism is particularly prominent. Utilitarianism holds that the ethicality of an action can be assessed by calculating the overall happiness generated and subtracting any resulting pain or suffering [Stahl et al. 2016].

Virtue Ethics, in contrast to deontology and consequentialism, focuses on evaluating the ethicality of actions based on the agent's character. According to this ethical theory, it is not sufficient to assess individual actions in isolation; instead, the individual's character as a whole is taken into account. This includes considering their idiosyncrasies, actions, and non-rational impulses that contribute to shaping their character

[Ananny 2016].

These theories help us better comprehend what ethics is and how philosophy breaks this term down to make it better understood. We hold a teleological perspective concerning the software development lifecycle, recognizing that decisions made during this process carry significant consequences for which developers bear responsibility. This is discussed in depth in the following section.

3. Ethics, Computing, and the Social Aspect of Software

Algorithms are not neutral. Software is not neutral. Systems are not neutral [Mowshowitz 1984, Gotterbarn 2001, Oriogun et al. 2012, Rosenbaum 2020]. They are created by humans and, therefore, express the collective or personal understanding of reality by the person or group that created them [Gustavsson and Penzenstadler 2020]. While this fact is not inherently negative, as numerous algorithms are devised with ethical intentions, acknowledging the non-neutrality of algorithms entails assuming responsibility for the societal implications resulting from design decisions made during software development.

According to Almeida et al. [Almeida et al. 2024], "Algorithms are not just lines of codes in systems. They are architectures that organize complex systems of interactions involving machines and humans". This makes developers and software owners responsible for the possible harmful consequences of the software they develop. Therefore, when developing any software system, it is crucial to observe the ethicality of such a system not only to avoid causing harm and discrimination but also to avoid being legally held accountable.

This is not simply a theoretical analysis exercise. A prominent case of unethical software development is the 'Dieselgate' scandal, in which Volkswagen implemented software designed to manipulate CO_2 emission readings during regulatory tests. Specifically, the system allowed vehicles to operate in two distinct modes: one that exceeded legal emission limits during regular driving and another that complied only under test conditions. Despite objections raised by software engineers within the company, these concerns were not reported to the appropriate regulatory authorities [McNamara et al. 2018].

Another notable case arose during the COVID-19 pandemic's early stages when individuals and organizations rapidly shifted to remote work and virtual communication. Zoom, a widely used video conferencing platform, gained popularity due to its accessibility and user-friendly interface. However, the platform's sudden surge in users exposed critical privacy and security vulnerabilities. Unprepared for the unprecedented demand, Zoom experienced a series of disruptions in which uninvited participants infiltrated meetings — a phenomenon named 'Zoombombing'. These intrusions often involved inappropriate content, including pornography and racist remarks, underscoring the platform's insufficient safeguards and raising serious concerns about user privacy and data security [Young 2021].

A third significant case is the Cambridge Analytica/Facebook scandal, which exposed profound ethical and privacy concerns regarding the misuse of user data for political manipulation. The controversy erupted when it was revealed that Cambridge Analytica, a political consulting firm, had harvested personal data from millions of Facebook users

without their explicit consent. This was facilitated by a seemingly harmless personality quiz app, which not only collected data from participants but also accessed information from their Facebook friends, ultimately extracting data from over 87 million users without authorization. Cambridge Analytica then used this data to construct detailed psychological profiles, enabling the firm to deliver highly targeted political advertisements during key events such as the 2016 U.S. presidential election and the Brexit referendum [Arora and Zinolabedini 2019].

These cases illustrate how software systems, even those with minimal reliance on AI, can cause significant societal harm and raise critical ethical concerns across various contexts. Other notable examples include algorithmic bias in Google's search ranking systems, which reinforced discriminatory stereotypes and systemic inequalities; Google Photos' misclassification of images depicting Black individuals as 'monkeys', highlighting racial bias in computer vision; Apple's deliberate throttling of older iPhone models through software updates under the pretense of battery management, a practice later admitted to being misleading; and Uber's Greyball tool, which exploited geolocation and user data to evade regulatory oversight by identifying and denying service to law enforcement in unauthorized operating areas. These cases underscore the pervasive ethical risks embedded in software design and deployment, even in systems not explicitly labeled as 'AI-driven'.

Although these cases are emblematic, many instances of ethical negligence may occur and go unaddressed by developers, either because they are not obligated to do so or because they fail to recognize these issues as ethical problems with concrete consequences for themselves and others. Consequently, ethical concerns must be integrated into the development of every software system, particularly those intended for widespread use.

4. Perspectives and Progress Evaluation

The debate surrounding ethics in software development and usage is expected to grow exponentially in the coming years, driven by the ubiquitous presence of software in people's lives. The depth and impact of this debate will be determined not only by those actively engaged with the topic but also by the broader forces influencing software development. Emerging regulations and inevitable new problems caused by software systems will highlight the importance of this theme and propel the field forward.

Numerous advancements are anticipated in the field over the coming years, with regulation increasingly influencing software development. Recently, several regulations have been established to promote ethical practices. In the European Union (EU), the General Data Protection Regulation (GDPR), implemented in 2018, set rigorous standards for data protection, transparency, and accountability. Similarly, Brazil introduced the Lei Geral de Proteção de Dados (LGPD) to enforce data protection measures. More recently, the EU enacted the Artificial Intelligence Act, marking a significant development in AI regulation. While these regulations have enhanced our understanding of ethical software practices, they remain insufficient for addressing emerging technological innovations. Consequently, more targeted regulations are expected to be enacted and enforced in the coming years.

In addition, as the demand for ethically sound software grows, comprehensive guidelines for implementing codes of ethics, alongside specialized methods, tools, and

frameworks, are likely to emerge. Given that robust strategies to ensure ethical software development remain scarce in the current literature, further research is crucial. Future efforts might benefit from interdisciplinary approaches that integrate insights from the regulations that are emerging.

Furthermore, implementing strategies to ensure the traceability of ethical requirements throughout the software development lifecycle can facilitate the specification, modeling, development, and testing of these ethical aspects. As Mitchell et al. [Mitchell et al. 2022] asserts, "To create and deploy ethical software, it is vital that ethical concerns of software systems are reflected in their artifacts, such as requirements, software architecture". In this context, research that deepens our understanding of software's ethical dimensions and integrates these considerations into the development lifecycle will be crucial. Achieving this goal requires the development of robust taxonomies and ontologies that offer a comprehensive view of ethical issues, which represents a critical initial step.

For these advancements to materialize, the computer science community must intensify research into the social dimensions of software in collaboration with other fields. In the short term, studies examining how the industry addresses ethical challenges, the development of a broad comprehensive taxonomy of ethical considerations, and the formulation of methodologies that comply with both current and forthcoming legislation will be highly valued. Moreover, integrating discussions of software ethics into academic curricula is essential for raising professionals' awareness of their ethical responsibilities.

Finally, evaluating the progress of this discussion will be a complex task, as it relies not only on quantitative metrics but also on the subjective analyses of professionals advocating for the debate. However, one observable metric could be the number of incidents involving faulty systems reported annually. If this number remains stable or declines, it could be considered a positive outcome, especially given the ever-increasing number of software systems in use. Furthermore, the number of studies that are published on the topic or bridge the gap between academia and industry is also an important metric for the development of the field.

Thus, we hope that research promoting a more human-centric approach to the debate on software systems will thrive in a technological field that remains predominantly focused on numerical and exact analyses while often overlooking the value of more subjective evaluations.

Acknowledgments

We thank CAPES and the Graduate Program in Computer Science at Federal University of Viçosa (PPGCC/UFV) for their institutional support.

References

Aberkane, A.-J. (2018). Exploring ethics in requirements engineering. Master's thesis, Utrecht University.

Almeida, V., Mendonça, R. F., and Filgueiras, F. (2024). Thinking of algorithms as institutions. *Commun. ACM*, 68(1):20–23.

Ananny, M. (2016). Toward an ethics of algorithms: Convening, observation, probability, and timeliness. *Science, Technology, & Human Values*, 41(1):93–117.

Arora, N. and Zinolabedini, D. (2019). The ethical implications of the 2018 facebook-cambridge analytica data scandal. *Journal or Repository Name*.

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

Gustavsson, J. L. and Penzenstadler, B. (2020). Blinded by simplicity: Locating the social dimension in software development process literature. In *Proceedings of the 7th International Conference on ICT for Sustainability*, ICT4S2020, page 116–127, New York, NY, USA. Association for Computing Machinery.

Johnson, B. and Smith, J. (2021). Towards ethical data-driven software: Filling the gaps in ethics research practice. In *2021 IEEE/ACM 2nd International Workshop on Ethics in Software Engineering Research and Practice (SEthics)*, pages 18–25.

Marturano, A. (2002). The role of metaethics and the future of computer ethics. *Ethics and Information Technology*, 4:71–78.

McNamara, A., Smith, J., and Murphy-Hill, E. (2018). Does acm's code of ethics change ethical decision making in software development? In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2018, page 729–733, New York, NY, USA. Association for Computing Machinery.

Mitchell, A., Balasubramaniam, D., and Fletcher, J. (2022). Incorporating ethics in software engineering: Challenges and opportunities. In *2022 29th Asia-Pacific Software Engineering Conference (APSEC)*, pages 90–98.

Mowshowitz, A. (1984). Computers and the myth of neutrality. In *Proceedings of the ACM 12th Annual Computer Science Conference on SIGCSE Symposium*, CSC '84, page 85–92, New York, NY, USA. Association for Computing Machinery.

Oriogun, P., Akinbule, O., Ibecheozor, C., and Nyako, Z. (2012). Software engineering ethical decision making and professional responsibility. In *2012 African Conference for Sofware Engineering and Applied Computing*, pages 7–14.

Rosenbaum, H. (2020). Algorithmic neutrality, algorithmic assemblages, and the life-world. *26th Americas Conference on Information Systems, AMCIS 2020*.

Shafer-Landau, R. (2018). *The Fundamentals of Ethics, 4Th Edition*. Oxford University Press.

Stahl, B. C., Timmermans, J., and Mittelstadt, B. D. (2016). The ethics of computing: A survey of the computing-oriented literature. *ACM Comput. Surv.*, 48(4).

Young, S. (2021). Zoombombing your toddler: User experience and the communication of zoom's privacy crisis. *Journal of Business and Technical Communication*, 35(1):147–153.