**Strict Kantianism**

Chioma Chance

Colorado State University Global

CSC502: Ethical Leadership in Software Development

Dr. Pubali Banerjee

10/20/2024

**Introduction**

"Strict" Kantianism, based on the ethical philosophy of Immanuel Kant, focuses on universal moral laws and the concept of duty. Kantianism emphasizes that actions must align with moral imperatives, regardless of the consequences. In software engineering, where ethical considerations are crucial, this approach can bring structure but also present challenges. In this paper, I will discuss two examples of how strict Kantian ethics can be applied to software engineering, as well as my opinion on its effectiveness.

**First Example: Privacy in User Data**

One of the core tenets of Kantian ethics is that individuals should never be treated merely as a means to an end. In software engineering, this means respecting user autonomy and privacy. For example, consider an application that collects user data. Under a Kantian perspective, developers have an absolute duty to respect user consent and ensure transparency about data collection. This respect for user autonomy is aligned with Kant’s idea of treating people as ends in themselves (Kant, 1785). Even if collecting data without explicit consent might benefit a larger audience or lead to financial profit, the act would be unethical according to Kantianism because it fails to respect the users’ right to informed consent.

From my perspective, this strict adherence to user privacy is vital, especially in today’s world where data breaches and unauthorized data usage are rampant. While it may not always be the most efficient path for companies seeking rapid growth, it ensures the rights and dignity of users are protected, which I believe is a fundamental aspect of ethical software engineering.

**Second Example: Ethical AI Development**

Another example of Kantianism in software engineering is the development of AI algorithms. Under strict Kantianism, developers would be obliged to ensure their AI systems do not produce harm, even unintentionally. The categorical imperative would demand that any algorithm developed must be universally acceptable. For instance, if an AI algorithm makes decisions that may result in biased outcomes against certain demographics, this would be inherently unethical under Kantian principles, as it would fail the universality test (Johnson, 2021). If a biased AI model negatively affects someone’s ability to secure a job, it treats that person unfairly and undermines their value as an individual—something Kantian ethics would not permit.

In my opinion, adhering strictly to Kantian ethics in AI development is challenging but necessary to prevent discriminatory practices. As a student learning about AI, I've realized how easy it is for unintentional biases to be embedded in algorithms. Taking a Kantian approach requires us to be proactive in evaluating potential harm, regardless of whether such evaluations slow down the development process. This insistence on universal fairness, while demanding, can help prevent some of the real-world harm we see today with biased AI systems.

**Opinion on Kantianism in Software Engineering**

Strict Kantianism, in my view, provides a solid ethical foundation for software engineering, but it isn’t without its challenges. Its emphasis on universal rules ensures a high standard of ethics, especially in areas like user privacy and unbiased AI. However, I also see how strict Kantianism can sometimes be inflexible. The world of software development often involves complex trade-offs, and Kantianism doesn’t always provide room for nuance when different ethical duties conflict.

For instance, the duty to protect user data might conflict with the duty to provide critical information during a public health crisis. In such cases, strict adherence to rules might lead to ethically gray areas. While Kantianism serves as a helpful guideline, I think a bit of flexibility, perhaps integrating aspects of other ethical theories like utilitarianism, could make ethical decision-making in software development more practical.

**Conclusion**

In conclusion, strict Kantianism emphasizes universal moral duties that respect individual rights, which is essential in ensuring ethical practices in software engineering. Whether it’s respecting user privacy or ensuring fairness in AI, Kantian principles hold developers to a high ethical standard. While this strictness ensures that human dignity is never compromised, it can be challenging to apply in scenarios where competing ethical duties exist. Ultimately, I think strict Kantian ethics should serve as a foundation, but there should be some flexibility to address the nuanced ethical challenges of software development.

**References**

Kant, I. (1785). *Groundwork of the Metaphysics of Morals*.

Johnson, D. G. (2021). *Computer Ethics* (5th ed.). Pearson.