

Marina Skegro

Dr. Shean Hayes

CSCI 315

February 2, 2023

Ethics in Software Development

In the area of software development, the balance between availability, reliability, and affordability presents significant ethical dilemmas. Therac-25 radiation treatment equipment is an example of how crucial software reliability is for life-critical systems. Navigating the complex connection between cost, accessibility, and ethical requirements is important to determine the appropriate quantity of testing.

To start, prioritizing the well-being of individuals is based on the ethical principle of beneficence. Strict testing procedures are required to reduce the risk to human life caused by software failures, particularly in settings such as healthcare, aviation, and finance. Therac-25 catastrophe offers an eye-opening example of the disastrous consequences that can arise from insufficient oversight and testing. While it could be tempting to cut back on testing to save expenses and increase accessibility to technology, doing so puts human life at serious risk. It is morally unacceptable to sell a less dependable, potentially fatal device to reach more people. Any immediate advantages of greater accessibility are outweighed by the possible harm brought about by system malfunctions.

Other approaches should be investigated to overcome accessibility issues rather than compromising reliability. To reduce the cost of life-saving technology without sacrificing safety requirements, this may entail creative funding mechanisms, funding from the government, or cooperative efforts between industry stakeholders. One potential solution lies in new approaches to testing and development methods. The testing process can be simplified without compromising dependability with the use of automated testing, continuous

integration, and agile approaches. Furthermore, to make sure that safety concerns are not neglected, it is crucial to promote an environment of responsibility and openness inside organizations.

Regarding certification processes for software engineers, I strongly advocate that software developers should go through an equivalent process to prove their competency and accountability, just like certification does for non-software engineers. Certification can assist in establishing minimal standards of skill and ethical conduct in the specialized field of software engineering, which brings distinct problems and risks. Software developers would benefit from certification regarding professionalism, responsibility, and following best practices in software development. In the end, it would benefit society as a whole by increasing public confidence in software systems.

Recognizing that technology is a dual tool that may be used for both good and damage is important when considering how technology affects society and the moral duties of computer professionals. Technology has greatly advanced humankind and changed many facets of our lives, but it has also brought out risks and ethical dilemmas. Computer professionals have a moral duty to protect ethical norms, promote responsible technology usage, and give users health and safety priority. This requires demanding examinations, observance of moral standards, and training to stay current with new developments and best practices (IEEE, 2017).

In my view, the safety and well-being of customers should always come first when developing software. It is not justifiable to sell a less dependable, possibly dangerous system to improve accessibility. The biblical concept of stewardship emphasizes our obligation to apply our knowledge and assets to advance societal well-being. Software developers' moral responsibility is to put user safety and welfare first. Ephesians 4:29 emphasizes the value of

ethical behavior in every aspect of our profession and cautions us to say words that uplift others (The Holy Bible).

In conclusion, the moral dilemmas raised by software accessibility and dependability demand a sophisticated strategy that balances conflicting interests and respects basic moral standards. Although accessibility and cost are significant factors, they cannot be used as an excuse to sacrifice dependability and safety. By implementing strict testing procedures, encouraging professional responsibility, and placing user welfare first, we may effectively manage the moral dilemmas that arise in software development and positively impact a more secure environment.

Works cited:

“The Holy Bible - King James Version.” www.o-bible.com/kjv.html. Accessed 2 Feb. 2024.

“IEEE Code of Ethics.” *IEEE*, 2017, www.ieee.org/about/corporate/governance/p7-8.html.
Accessed 2 Feb. 2024.