Kory Singleton

The Ethical Implications of Software Testing and Certification: A Christian Perspective

In the case of Therac-25, there exists a case of faulty software testing in modern devices with catastrophic results. In a Christian worldview, it is believable that ethical engineering should strive to balance equal access, cost and safety while upholding holy principles such as justice, stewardship and the love for one's neighbor (Mark 12:31).

The Necessity of Testing and Its Limits

Given their positions as stewards of technology (Genesis 1:28), engineers are generally trusted to create systems with the intent of benefiting humanity without fear of harm. Proverbs 11:1 can be related, stating "A false balance is an abomination to the Lord, but a just weight is His delight." In the context of software engineering, a "just weight" can be sought in the form of rigorous testing as reliability in pursuit of public safety is the ultimate goal. However, excessive testing at the risk of inflating costs for treatments may limit the reach of technology. In solving this, I believe a reasonable solution is risk-based testing. The IEEE Code of Ethics places importance on ensuring public welfare while assuring testing is sufficient to limit harm without inflation.

Should a Less Reliable System Be Sold?

I believe that releasing a system known to be less consistent, especially in the context of life-saving or deathly applications, serves as a direct contradiction to biblical values. That is to say that the ends don't justify the means. A system that is flawed in a way that causes harm simply cannot be rationalized as good even though it provides broader availability of goods: this is something that Romans 3:8 warns against, spitting evil for the sake of achieving better outcomes

The ACM Code of Ethics instructs software engineers to "avoid harm" and uphold integrity, affirming that knowingly selling a flawed system is unethical. Investment in better software engineering practices, including peer reviews and formal verification methods, can improve reliability without excessive costs. Prioritizing human lives over profits aligns with Christ's teachings on loving and serving others (Luke 10:27).

Certification for Software Engineers

Many U.S. states require certifications for traditional engineers to ensure public safety. Given the increasing dependence on software in critical systems, a similar process for software engineers is justifiable. Proverbs 15:22 states, "Plans fail for lack of counsel, but with many advisers, they

succeed," supporting the need for formal training and certification to uphold technical and ethical standards.

Ethics and Professional Responsibility in Computing by Loui and Miller emphasizes professional accountability, arguing that software engineers should adhere to ethical and safety standards similar to traditional engineers. The ACM and IEEE codes of ethics advocate for continuous learning and adherence to best practices. Certification would ensure that engineers understand ethical considerations, safety protocols, and best coding practices before working on life-critical systems.

Conclusion

Balancing cost, accessibility, and safety in software engineering is a complex ethical issue. However, from a Christian perspective, ensuring that systems are as safe as reasonably possible aligns with biblical principles of stewardship, justice, and love. Selling a knowingly flawed medical device cannot be justified, and alternative testing strategies should be explored. Furthermore, implementing certification for software engineers would enhance accountability and reliability in critical systems, ultimately serving the greater good. By following these principles, engineers can act ethically while promoting both innovation and accessibility.

References

- Ethics and Professional Responsibility in Computing, M. C. Loui & K. W. Miller
- ACM Code of Ethics and Professional Conduct
- IEEE Code of Ethics
- The Holy Bible (ESV)