The Ethics of Software Testing

By: Janiyah Cutchin-Brown

Overview

The paper this presentation is based on explores the critical ethical responsibilities of software engineers in ensuring their programs are safe, reliable, and thoroughly tested–especially when software impacts human life and wellbeing. It emphasizes the importance of balancing rigorous testing with considerations of cos and accessibility, highlighting how insufficient testing can lead to serious harm, as seen in historical cases like the Therac-25 incident.

Key Points

Risk-Based Testing: The amount and depth of testing should correspond to the potential risk and impact on users. Life-critical, such as those used in healthcare, require higher testing standards.

Avoiding Harm: Following ethical frameworks like the ACM Code of Ethics and Christian principles (e.g., Philippians 2:4), software engineers must prioritize the safety and wellbeing of all users, sometimes investing more resources in testing to prevent harm.

Key Points (cont.)

Accessibility v. Safety: While it is important to make technology affordable and accessible, safety must be never be compromised. Developing efficient, strategic testing methods (such as automated and risk-based testing) can help balance these goals.

Certification and Accountability: The paper advocates for professional certification processes for software engineers, similar to other high-responsibility professions, to ensure they have the skills and accountability needed to protect users and society.

Key Points (cont.)

Ethical Responsibility: Drawing on a spiritual and professional ethics, the paper argues that software engineers have a duty to protect human life through meticulous testing and responsible development practices.

Conclusion

By upholding both professional standards and ethical principles, software engineers can create reliable, safe software that meets societal needs without compromising human wellbeing. Rigorous testing and certification are key tools in fulfilling this responsibility.