Victor Delaplaine

Professor Humel

CPE329

11/19/19

Ethics of Toyota

Should Toyota's firmware quality be considered a violation of safety standards? If so, how could this be prevented or detected sooner in the future?

Toyota Company lost a case presented to the Oklahoma court by Michael Barr. It involved poor quality codes that lead to an accident caused by the unintended acceleration that claimed a life and left one hospitalized. Stakeholders included the Toyota Company, Michael Barr, and the accident victims.

Beyond automobiles, what microcontroller devices pose risk to human life with poor quality code? How can those devices (including their firmware) be vetted for safety?

From the case, violations of the safety standards occurred since Toyota had violated the MISRA-C rule in coding. Acquiring a CPU lacking protection against stack overflow from an uncertified vendor violated the rule. To prevent this, programmers should strictly adhere to the MISRA-C law and contract certified vendors only. Additionally, independent oversight is necessary to safeguard quality (Dunn 2). Finally, the firmware companies should engage other firms to check their systems.

Should coders be held responsible for poorly written code? Should project managers, who may not be coders or look at the software, be held responsible?

Apart from automobiles, other microcontroller devices that pose risks to human life include traffic control devices and microwave devices. According to the accident theory, microcontroller devices pose a threat to human beings if poorly coded. Towards guaranteeing their safety, there is a need to engage independent oversight for vetting the devices and ensuring their compliance with coding rules (Dunn 4). Additionally, coders should be held responsible for poorly written codes since they have a good understanding of MISRA-C rules and overlook them. Project managers without coding knowledge should not be held accountable for wrong coding, but they should be responsible for negligence (Safetyresearch.Net 3).

Is it feasible to create a software equivalent of a professional engineer and requiring software to be "stamped" or "signed off" by a licensed engineer.

It is feasible to create a software equivalent of a professional engineer that requires stamping by a licensed engineer as it will enhance adherence to coding rules. As revealed in the case, human error contributes to risks since it showed that Toyota's programmers overlooked the coding rules. Requiring a licensed engineer to stamp the software enhances accountability and traceability in case of defects.

Works Cited

Cummings, David. 2017. "Why Every Embedded Software Developer Should Care About the

    Toyota Verdict." *Embedded.com*, 3 Dec. 2017, www.embedded.com/why-every-

    embedded-software-developer-should-care-about-the-toyota-verdict/

Dunn, Michael. 2013. Toyota's killer firmware: Bad design and its consequences.

"Toyota Unintended Acceleration and the Big Bowl of "Spaghetti" Code." *Welcome | Safety

    Research & Strategies, Inc*, www.safetyresearch.net/blog/articles/toyota-unintended-

    acceleration-and-big-bowl-%E2%80%9Cspaghetti%E2%80%9D-code