

Research Proposal

Title: Software design and medical technology; investigating awareness and risk

Author: Ash Tyndall

Unit: CITS5502 Software Processes

Background

Medical software systems exist in a unique area of the computing spectrum.

Frequently, they are trusted to correctly administer doses of radiation, chemotherapy, and other substances to a patient who cannot easily determine if a malfunction is occurring.

Similarly, while the medical professionals responsible for the oversight of the machine's operation have knowledge as to what is acceptable/non-acceptable dosage, they do not necessarily have the skills to determine if the machine is correctly executing this knowledge, or if it has suffered a non-obvious fault.

Frequently the potential range of administration of these machines exceeds the safe dosage range, which has lead to past incidents in which patients have recieved orders of magnitude higher dosages than intended, and have suffered permanent damage or death as a result.

A common case study in this area is that of Therac-25; one of the first radiation therapy devices that used primarily software safeguards, and caused six incidents of death or serious injury as a result of poor user interface design and poor software development processes. [2] This and incidents like it lead to increased government and industry regulation in the area, including the institution of IEC 62304, a software life cycle process specifically for medical device software. [1]

Research Topic

This project proposes to investigate the evolution of software design, development processes and user interfaces in medical equipment from pre-software to the current day, and how these factors contribute to increasing operator awareness of error and malfunction, as well as the implication of and potential to override fault indications.

This is of critical importance to the medical industry, as poor application of software design and presentation of error has the potential to cause (and has indeed already lead to) death or serious disability through the misadministration of treatment though a combination of unclear error/data presentation, software edge-cases and ability for operator override in these circumstances.

References

- [1] JORDAN, P. Standard iec 62304-medical device software-software lifecycle processes. In *Software for Medical Devices, 2006. The Institution of Engineering and Technology Seminar on* (2006), IET, pp. 41–47.
- [2] LEVESON, N. G., AND TURNER, C. S. An investigation of the therac-25 accidents. *Computer* 26, 7 (1993), 18–41.