Trustworthy Medical Device Software

Author: Kevin Fu

Presenter: Lily Shpak

Software Medical Devices

- Devices that are used to treat a medical issue
- Usually implanted within the patient
- Allows the doctor or nurse to interface with the device and make periodic updates
- “Computers provide a level of power, speed, and control not otherwise possible”

Main Problem with Software Medical Devices

- Medical devices are now being built with software critical to the device's function
- "2002-2010, software-based devices resulted in over 537 recalls affecting more than 1,527,311 devices"
- A lot of these software malfunctions have had fatal outcomes

Eng. Stage	Adverse Event	Contributing Factor
Requirements Specification	Linear accelerator: Patients died from massive overdoses of radiation.	An FDA memo regarding the Corrective Action Plan (CAP) notes that, "Unfortunately, the AECL response also seems to point out an apparent lack of documentation on software specifications and a software test plan [30, p. 539]."
Design	Pacemakers/Implantable defibrillators: Implant can be wirelessly tricked into inducing a fatal heart rhythm [21].	Security and privacy need to be part of the early design process.
Human Factors	Infusion pump: Patients injured or killed by drug overdoses.	Software that did not prevent key bounce misinterpreted key presses of 20 mL as 200 mL [17].
Implementation	Infusion pump: Underdosed patient experienced increased intracranial pressure followed by brain death.	Buffer overflow (programming error) shut down pump [14].
Testing	Ambulance dispatch: Lost emergency calls.	An earlier system for the London Ambulance Service failed two major tests and was scuttled [20]. Ambulance workers later accused the computer system of losing calls and that "the number of deaths in north London became so acute that the computer system was withdrawn [53]." The ambulance company attributed the problems to "teething troubles" with a new computer system [53].
Maintenance	Health Information Technology (HIT) devices: Computers systems globally rendered unavailable.	An anti-virus update misclassified a core Windows operating system component as malware and quarantined the file, causing a continuous reboot cycle for any system that accepted the software update [32]. Numerous hospitals were affected. At Upstate University Hospital in New York, 2,500 of the 6,000 computers were affected [52]. In Rhode Island, a third of the hospitals were forced "to postpone elective surg-

There are issues at every stage of creating these products.

Software does not have safety checks to prevent mistakes that are easy to make.

As shown by this diagram, human factors have caused issues in the design process. The software did not prevent the doctor from overdosing the patient.

Drivers of the Problem

- Not clear safety requirements from the FDA
- Technology companies are not incentivized to create products to a certain specification
- “We keep running at an ever-faster pace to develop or use increasingly complex software systems that we do not fully understand, and we place such software in systems that are more and more critical”
- Companies have proprietary rights to the devices which make it hard for academic researchers to be involved in the innovation of medical devices

Flawed From the Beginning

- A majority of these devices are being built without talking to software engineers or doctors
- Issues in the specification phase of building the devices
- Systems engineering design standards are not used in design process, so there is no standard approach to building medical devices
- Linear accelerator: Patients died from massive overdoses of radiation, there was an apparent lack of documentation which lead to a lack in a testing plan

Flawed From the Beginning

- Many of the devices work fine on their own
  - First pacemaker used in 1958
- The issue is when we integrate the hardware with the new software components
- There is not clear path to follow for integrating the software and the hardware

Steps to Solve Issues of Current Medical Software

- Adopt modern software engineering techniques
- Meaningful specification requirements
- Apply a system engineering approach
- Mitigate risks due to human factors
- Mitigate low probability, high-consequence risks
- New regulatory policy

Adopt Modern Software Engineering Techniques

- Currently most systems use languages that do not support software fault detection
  - C
- Suggests using Ada - provides extensive support for software fault detection
- Use mechanical compliance software that checks the specification of the software

Meaningful specification requirements

- Safety failures stem from issues during specification of requirements
- If there is not a clear specification it is harder to test
    - False sense of safety
- “Leading software engineers believe that many medical device manufacturers have an opportunity to significantly improve specification of requirements”

Apply a System Engineering Approach

- Devices have been treated as isolated components
- With the introduction of software, need to treat them as full systems
- Evaluation of the devices should be done by third-party experts
    - Should not be connected to the manufacturer

Mitigate risks due to human factors

- Devices are designed without thinking of who is interfacing with them
- User interfaces are ambiguous
- Manufacturers need to meet with doctors and nurses who use the devices

Mitigate Low-Probability, High-Consequence Risks

- Manufacturers, health care professionals, and users are putting too much confidence in medical device software
- Difficult to reproduce software bugs
- Security and privacy risks are low-probability, high-consequences
- Security issues range from a device being down to someone maliciously acting on the device
- Even if security vulnerabilities cannot be fixed, doctors deserve to know all risks

New Regulatory Policy

- Because medical devices have a wide range of operations they perform, they should be regulated by outcome not specific technologies
- A push toward uniform standards lead to an oversimplification because the standards have to apply to all devices
- FDA advises to “update your operating system and medical device software”
    - Software updates carry risks

New Regulatory Policy

- Statistic collection for medical devices need to be standardized
- Currently there is a lack of data and inadequate record keeping
    - Hard to find issues
- FDA does have databases in place to record data, but they are not being used
- Data on devices should be available to public so that new innovations can be made

New Regulatory Policy

- In complex system, it is hard to pinpoint who is responsible for a bug
- FDA acknowledges that a key challenge is a shared responsibility for failures in software
- This makes it hard to solve the issue because it is unclear which party is in charge of that portion of the device
- In the specification, it should be noted which party is responsible for each part of the system

New Regulatory Policy

- When FDA does testing and research on these products, they use biomedical engineers
- FDA has fellowships for experts to come work on testing products
    - Target BMEs
- Software engineers should be used because they can look at the code and understand software risks

Will New Regulatory Policy Actually Affect Change?

- This was a question many of you asked.
- The FDA has laid some of the groundwork to regulate medical devices, but there is not much enforcement done
- Leads to the Question: What is the right amount of regulations so people are safe but products still come out in a timely fashion and at an affordable cost?

About the Author

- Kevin Fu is an assistant professor at University of Massachusetts, Amherst
- In 2009, he was appointed to the National Institute of Standards and Technology (NIST) Information Security and Privacy Advisory Board (ISPAB) for four years [1]
    - Focused on security and privacy of medical devices
- He is credited with establishing the field of medical device security[2]
- Professor Fu went to Capitol Hill to talk about an increasing amount of security threats to medical devices [3]

Critiques

- This is a major issue that has caused terrible outcomes
  - Preventable issues lead to patients dying
- Suggest new policy but acknowledge that is quite hard to standardize it for all devices without over simplifying the regulations
- I think they could have suggested making a board at the FDA with software engineers, biomedical engineers, doctors, nurses, and manufacturers.
- Many of you noted that this paper lacked technical explanations, I think the point of this paper was to bring awareness to issues surrounding medical device software

Summary

- Medical devices need to be trustworthy because they are usually implanted in someone
- In the design phase, there needs to be a clear outline of specifications
- Manufacturers and designers need to treat these devices as full systems
- The FDA needs to put into place policy that has clear outcomes for these devices
- FDA also needs to include a wide range of experts when reviewing devices

Sources

1. https://www.umass.edu/newsoffice/article/kevin-fu-umass-amherst-cyber-security-researcher-named-mit-technology-review%E2%80%99s-2009
2. https://www.secure-medicine.org/about
3. https://www.ansi.org/news_publications/news_story?menuid=7&articleid=0a903ae3-76fc-4f70-9a22-0d56a800293d