STATEMENT OF PROF. KEVIN FU, PH.D.

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF MASSACHUSETTS AMHERST
AMHERST, MA

SOFTWARE ISSUES FOR
THE MEDICAL DEVICE APPROVAL PROCESS

SUBMITTED TO THE
SPECIAL COMMITTEE ON AGING
UNITED STATES SENATE
HEARING ON

A DELICATE BALANCE: FDA AND THE REFORM OF THE
MEDICAL DEVICE APPROVAL PROCESS

WEDNESDAY, APRIL 13, 2011

Introduction

Chairman Kohl, Ranking Member Corker, and the distinguished members of the Special Committee on Aging, I would like to thank you for the invitation to submit a statement for the record regarding the impact of software-related issues on reform of the medical device approval process. My comments below are based on work supported in part by the Institute of Medicine, a Sloan Research Fellowship, the National Science Foundation Directorate for Computer and Information Science and Engineering, and the Office of the National Coordinator for Health Information Technology. However, all opinions, findings, and conclusions are my own and do not necessarily reflect the views of the IOM, Sloan Foundation, NSF, ONC, or my past or present employers.

My name is Kevin Fu. I am a faculty member in Computer Science at the University of Massachusetts Amherst where my research pertains to trustworthy computing for medical devices. I have interacted several times with FDA with multiple presentations on medical device software at FDA's Center for Devices and Radiological Health. My educational qualifications include a Ph.D., master's degree, and bachelor's degree from M.I.T.'s Department of Electrical Engineering and Computer Science. My industrial experience in software systems includes past employment at Cisco Systems, Microsoft, Hewlett-Packard, and the Information Systems department at Holland Community Hospital. In my nearly two decades of experience in software related to health care, I have observed both the risks and benefits of software for medical devices. Highlights include participation in the roll out of a hospital information system to improve patient care at a community hospital and the security analysis of a medical device showing that an implantable cardiac defibrillator could be wirelessly tricked into inducing a fatal heart rhythm.

I attach for the record my presentation and report on *Trustworthy Medical Device Software* commissioned by the Institute of Medicine at the National Academies for the "Public Health Effectiveness of the FDA 510(k) Clearance Process" committee, which will shortly render its recommendations to FDA. My statement summarizes these findings and suggests several questions to ask about the role of software in the medical device approval process.

Findings on Software Issues for Medical Devices

Despite the lessons learned by tragic accidents, such as the radiation injuries and deaths caused by the Therac-25 linear accelerator two decades ago (Leveson, 1993), medical devices that depend on software continue to injure or kill patients in preventable ways. Problems in medical device software result largely from a failure to apply well-known systems engineering techniques, especially during specification of requirements and analysis of human factors. Problems ranging from poor user interfaces to overconfidence in software have led to accidents such as fatally incorrect dosages on infusion pumps and in radiation therapy. A common trait for adverse events in medical device software is that the problems are often set in place before any implementation begins.

Illustrative Examples to Motivate Software Questions

Insufficient number of software experts at FDA. It was explained to me by a former FDA CDRH director that seldom does an FDA inspector assigned to review a 510(k) application have experience in software engineering—even though the majority of medical devices today rely on software. Over half the medical devices on the US market now involve software (Faris, 2006).

Opting to forgo a wireless pacemaker. Karen Sandler, General Counsel for the Software Freedom Law Center, was concerned about the safety of her software-controlled pacemaker because it would have a long-range, wireless interface[1]. She selected an older pacemaker without a wireless component:

> I [was prescribed] a pacemaker. My first question was, "Could I take a look at the [computer] code?" I offered to sign an NDA.... No one would take this concern seriously. I was at risk of sudden cardiac death.
> ...
> After talking to many doctors who didn't really understand why I would be concerned about the safety of the software of the device, finally I found a doctor....who [suggested], "What if we find you an old device without a wireless component?"

Personally, if I were prescribed a medical device by a well informed physician, I would accept the device for my health. However, with increased software complexity, patients and physicians cannot make informed decisions without access to better information about software risks.

Malware on medical devices. An Information Technology (IT) professional from a VA medical center sought my advice at the RSA Security Conference on how to recover from malware that had infected her hospital computer systems. She explained that her medical systems were routinely infected with malicious software because health care professionals like to check email on the same machines used for patient care. However, email is just one potential vector for malware to infect medical devices. Medical devices are exposed to persistent software-based threats when pathways exist to the Internet. I asked why not have separate computers, and she replied that there was not enough desk space. I asked if any of the computers were connected to radiation-emitting machines, and she declined to comment.

[1] http://www.usenix.org/events/healthsec10/tech/

Severe underreporting. Users are not incentivized to report software security problems. At the USENIX Workshop on Health Security and Privacy[1], John F. Murray Jr. (Software Compliance Expert, FDA CDRH/Office of Compliance) speaking for himself commented that:

> We actually know that cybersecurity and viruses are huge problems for medical devices—for networked medical devices. We know that because I get phone calls all the time. We know that because people complain all the time. But unfortunately, the users aren't complaining in any formalized way.
> ...
> If you [discover] some problems, some issues, you need to get into the mode of reporting these kinds of issues. And making it known to the FDA and the authorities that this is a really big issue. Now this is going to become extremely—a hundred times more important when we start using electronic health records.
> ...
> What does the law require you to report? If you're a user facility, the law requires you to report any deaths involving medical devices, or any serious injuries involving medical devices. These things have actually had to occur [to require reporting].

Scott Bolte of GE Healthcare emphasizes that for security problems, formal reporting is especially lacking [2]. This advice was given to FDA six years ago:

> Although there is a lot of anecdotal evidence that malicious software has compromised medical devices, there is a notable lack of formal evidence. So without this formal reporting, the FDA is limited in its ability to act or intervene. Reporting is something providers and arguably the manufacturers themselves can and should start doing immediately.

Inconsistency within FDA on substantial equivalence. At the IOM workshops on the 510(k) clearance process, former FDA officials presented contradictory advice on "substantial equivalence." This concept is important because a manufacturer that demonstrates substantial equivalence to an old predicate device may choose a less thorough regulatory pathway for clearance of a new device.

Dr. Christy Foreman's presentation[3] includes a flowchart that effectively asks, "Does the new technical characteristics raise new types of safety or effectiveness questions." My examples in this section should demonstrate that medical device software can raise new types of safety or effectiveness questions.

However, the slides from Heather Rosencrans (former Director, 510(k) Staff, Office of Device Evaluation, FDA CRDH) seem to indicate that there is never a reason to question the substantial equivalence of a new digital control linked with a previous analog control as a predicate[3]. Then Dr. David Feigel (former Director, CDRH) said during a June 14, 2010 IOM workshop that:

> One of the interesting classes is radiation equipment...even the software, which I wonder where they got the first predicate for software.
> -David Feigal, Fmr. Director, FDA CDRH

[2] http://www.fda.gov/MedicalDevices/Safety/MedSunMedicalProductSafetyNetwork/ucm127816.htm
[3] IOM Workshop on Public Health Effectiveness of the FDA 510(k) Clearance Process, March 1, 2010.

The misconception that software poses risks no different from hardware is illustrated by the tragedies of the Therac-25 (Leveson, 1993), a radiation-emitting device that modified its systems to rely on digital software controls rather than analog hardware controls in the 1980s. This change combined with other human elements raised new safety risks and ultimately injured and killed a number of patients. Unlike a mechanical component, software is not exhaustively testable or interpolatable. In the case of the Therac-25, one result of this software issue was improper switching between therapies that resulted in massive radiation overdoses. Massive overdoses on software-controlled, radiation-emitting machines continue to happen today (Bogdanich, 2010).

Questions

Based on the examples above, I suggest several questions for the Committee to consider.

1. Questions on reporting and statistics
    (a) How many cleared and approved medical devices currently use radio communication (e.g., wireless, MICS)?
    (b) What percentage of medical devices currently cleared or approved involve software, radio or wireless communication, or Internet connectivity?
    (c) What percentage of medical device applications currently under review involve software, radio or wireless communication, or Internet connectivity? To what degree is this amount increasing or decreasing?
    (d) To what degree are critical device functions being performed by software (vs. hardware)? Is the amount increasing? Decreasing?
2. Questions on risk/benefit analysis
    (a) How does a manufacturer demonstrate that wireless communication or Internet connectivity leads to overall better patient outcomes?
    (b) How does a patient or physician learn what programming languages were used in the creation of medical device software? Different programming languages carry different risks.
    (c) What effect does software have on reliability? Availability? Maintainability? Ease of use?
    (d) How do these software characteristics compare with similar implementations in hardware? Does the software make the device safer or more effective?

3. Questions on substantial equivalence

   (a) Why do past senior administrators in CDRH have conflicting definitions of what it means for software-based medical devices to have "substantial equivalence to a predicate" in the context of the 510(k) process? Some claim perhaps unintentionally that software and hardware are no different. On the other hand, some have stated that they do not know how any software could be substantially equivalent to a hardware-based predicate. What is FDA CDRH's current position and how does CDRH plan to develop a more consistent definition across all its scientists and engineers?

   (b) What does data from the predicate device reveal about the new device? Does predicate data save time in specification of the new device? Does predicate data save time in testing of the new device?

4. Question on informed consent for disclosing risks to patients

   What are manufacturers required to disclose to physicians about medical device software risks? How does the process differ for devices that are higher risk or consequence?

5. Questions on security of medical device software

   (a) To what extent will existing recall processes be effective against zero-day software vulnerabilities[4]?

   (b) What contingency plans are in place should the software equivalent of the 1982 Chicago Tylenol cyanide poisonings take place? How much time would it take a manufacturer to address a software vulnerability?

   (c) How does FDA balance the benefits of software with the risks of low-probability, high-consequence problems in software that may result in significant injuries or deaths?

6. Questions on education and FDA personnel

   (a) What is the recommended training for health care professionals for reporting software problems? Which hospital administrator is responsible for this reporting, and how are they incentivized to look for potential problems? That is, who takes ownership of the risks?

   (b) What special training do FDA reviewers receive specific to software engineering, requirements specification, system engineering, hazard analysis, dependable computing, and trustworthy computing?

   (c) Why are there no software experts among any of the past fellows of the FDA Commissioner's Fellowship Program?

   (d) What does FDA do to attract and retain top talent for handling software issues? What employers are the prime competitors for talent?

[4] A zero-day vulnerability is a security problem where the time between discovery of the flaw and exploitation of the flaw is less than a day.

Recommendations

Recommendations to increase the trustworthiness of medical device software include (1) regulatory policies that specify outcome measures rather than technology, (2) collection of statistics on the role of software in medical devices, (3) establishment of open-research platforms for innovation, (4) clearer roles and responsibility for the shared burden of software, (5) clarification of the meaning of substantial equivalence for software, and (6) an increase in Food and Drug Administration (FDA) access to outside experts in software.

Conclusion

There is no question that software provides significant benefits for the function of medical devices. However, software also presents risks qualitatively different from risks of hardware and mechanical components. Many risks of medical device software could be mitigated by applying well-known systems engineering techniques, especially during specification of requirements and analysis of human factors.

Today, the frequency of news reports on tragic, preventable accidents involving software-based medical devices falls somewhere between that of automobile accidents and airplane accidents. Event reporting on tragic medical device accidents is likely headed toward the frequency of the former given the continued increase in system complexity of medical device software and present-day regulatory policies that do not adequately encourage use of modern software engineering and system engineering practices. Left unbalanced, poorly engineered software may become the O-ring of medical devices.

I hope that my statement provides helpful context for the Committee to ask important questions such that FDA can better balance the risks and benefits of medical device software.