Ethics

Paul Jackson

School of Informatics
University of Edinburgh

Required reading

from Lecture 1 of this course was

Compulsory: Read the ACM/IEEE Software Engineering Code of Ethics:
`https://ethics.acm.org/code-of-ethics/software-engineering-code/`
and think about cases where the principles might conflict.

If you didn't do it then, or if you did and have forgotten it, consider this a reminder.

The short version starts:

> *Software engineers shall commit themselves to making the analysis, specification, design, development, testing and maintenance of software a beneficial and respected profession.*
> *In accordance with their commitment to the health, safety and welfare of the public, software engineers shall adhere to the following Eight Principles:*

Principles from the short version of the Code:

1. PUBLIC - Software engineers shall act consistently with the public interest.

2. CLIENT AND EMPLOYER - Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.

3. PRODUCT - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.

4. JUDGMENT - Software engineers shall maintain integrity and independence in their professional judgment.

5. MANAGEMENT - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.

6. PROFESSION - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.

7. COLLEAGUES - Software engineers shall be fair to and supportive of their colleagues.

8. SELF - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

Why? Therac-25

Generally when software contributes to bad outcomes the situation is complex, and allocation of blame is difficult. E.g. something else also had to go wrong to cause the bad outcome (Ariane 5) or it isn't clear what role software played (Chinook FADEC).

The Therac-25 incident is one of relatively few cases where it is clear that software errors – failures of the software engineering process – caused people's deaths.

What happened I

Therac-25 was a medical linear accelerator – it could produce beams of electrons or X-rays to treat tumours.

Earlier versions used software as a convenience, on top of hardware that could stand alone. In particular, they had hardware safety controls.

Key danger: the machine's "raw" electron beam is harmful. It must be treated to produce either a "safe" electron beam or a "safe" X-ray beam. This involves a turntable being correctly positioned. It wasn't.

What happened II

1. Original safety feature: the machine's settings were made manually, then reentered at a terminal. The computer checked for a match.

   Operators thought it inefficient to reenter the data. Replaced by a "hitting return" version.
2. 2.1 Reports of errors were frequent.
   2.2 Operators were told in training that the machine was safe.
   2.3 Error messages were cryptic.
   2.4 It was possible to respond to an error with a "proceed".

   So operators did ignore and override error reports.
3. Unprotected shared data – race conditions. Result: malfunctions that appeared only on particular, fast editing sequences involving the up-arrow key.

The consequences

Between 1985 and 1987 at least 6 people were seriously over-treated.

At least 3 died.

Root causes

- Software was regarded as safe, compared to hardware.
- The manufacturer made unjustified and wildly optimistic claims about how unlikely errors were, which misled clinicians.
- Poor software specification.
- Lack of defensive design.
- Poor software testing.
- Incident reporting was not systematic

Improving practice

Technical fixes, e.g.: there should have been an independent, simple, verifiable double-check of the safety of what the machine was about to do.

Process fix, e.g.: software should have been regarded as risky, and specified and verified accordingly.

Ethical angle

The ACM/IEEE code of ethics was violated, for example because:

- 1. PUBLIC - AECL employees failed “1.06. Be fair and avoid deception in all statements, particularly public ones, concerning software or related documents, methods and tools.” e.g. by claiming it was impossible for Therac-25 to overdose patients.
- 3. PRODUCT - they failed “3.10. Ensure adequate testing, debugging, and review of software and related documents on which they work.”; testing was severely inadequate.
- 6. PROFESSION - they failed “6.07. Be accurate in stating the characteristics of software on which they work, avoiding not only false claims but also claims that might reasonably be supposed to be speculative, vacuous, deceptive, misleading, or doubtful.”

We could go on - exercise.

Reading

Required: obvious, the ACM/IEEE Code of Ethics if you've forgotten it - see Lecture 1.

Suggested: Nancy Leveson's paper on the Therac-25 accidents.