Grand Valley State University
www.gvsu.edu

Software Engineering Ethics

Erik Fredericks
Adapted from Byron DeVries, Jagadeesh Nandigam

Outline

What shall we do today?

- What are **ethics?**
- Why are **ethics?**
- Professional vs general **ethics**
- SE code of **ethics**
- Case Studies (of **ethics**)

What are **ethics**?

(Dictionary) Defined as:

- The discipline dealing with what is good and bad and with moral duty and obligation,

- A set of moral principles,

- The principles of conduct governing an individual (personal ethics) or a group (professional ethics).

What are ethics?

Ethics is the study of what it means to “**do the right thing**”

When faced with difficult ethical situations, it helps to restate an ethical issue, question or dilemma in the form:

- **Is it right to …”?**

Why ethics?

“Good ethical behavior usually **leads to good consequences**, both for ourselves and for society at large.”

“**Happiness** comes from reasoning through a complex moral puzzle, choosing a good course of action, and following through.”

“Scientists and engineers make decisions **crucial to society at large**, and therefore shoulder an enormous burden of public trust.”

Source: “Fundamentals of Ethics for Scientists and Engineers” by Seebauer & Barry

Professional **vs** general/personal ethics

Two ways professional ethics differ from general ethics:

- Professional is an **expert** in a field and therefore has special responsibilities.
- Products of many professionals profoundly **affect large number of people**.

A professional can cause great harm through dishonesty, carelessness, or incompetence.

SE code of ethics and professional practices

Developed by IEEE-CS and ACM Joint Task Force
Eight principles (with several clauses for each principle)

- Public
- Client and Employer
- Product
- Judgement
- Management
- Profession
- Colleagues
- Self

Software engineers should strive to act in a manner that is consistent with the spirit of the SE code.

The SE Code should not be viewed as a simple ethical algorithm for generating ethical decisions.

SE Code of Ethics and Professional Practices

Public – Software Engineers shall act consistently with the public interest

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

Product – Software Engineers shall ensure their products and related modifications meet the highest professional standards possible.

Judgement – Software Engineers shall maintain integrity and independence in their professional judgement.

SE Code of Ethics and Professional Practices

Management – Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.

Profession – Software Engineers shall advance the integrity and reputation of the profession consistent with the public interest.

Colleagues – Software Engineers shall be fair to and supportive of their colleagues.

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.

Ethics Case Studies

Therac-25 refresher

Let's talk about the **Therac-25**

- Radiation therapy machine (chemotherapy)
- Iteration over the well-used Therac-6
  - Now completely software-controlled
    - .... Including safety mechanisms
    - Hey, its probably cheaper to have everything be software-controlled, right?

- Machine was extensively tested and verified over several years
  - A single programmer wrote all the code
  - Unit testing and general SW testing was minimally done

Effectively this problem results from **race conditions** and **concurrency errors**

Case study: Therac-25

Institutional Culture Causes:

- No independent review of source code.
- No assessment how the software might fail.
- Poor UI design: System displayed “MALFUNCTION: XX” when something was wrong and allowed override.
- Initial complaints were not believed.
- No integration testing with hardware and software.

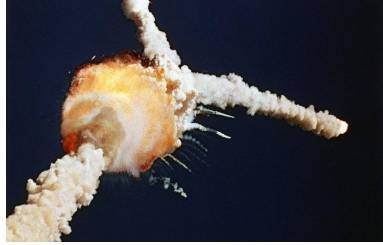
Case study: Therac-25

Engineering Causes:

- Did not take into account unexpected inputs (e.g., key sequences).
- No hardware interlocks to prevent failure due to software.
- Wholesale reuse of previous source code without due diligence even though hardware changed.
- No hardware backups to verify sensors.
- UI and Controller race conditions.
- Incrementing a fault flag that could be overrun (reset to 0).

Split off into groups and come up with a response

Case study: Challenger



The 25th flight of the space shuttle Challenger, STS-51L lasted just 73 seconds on Jan. 28, 1986. Shortly after liftoff, smoke briefly appeared at a joint on the right rocket booster. After a minute, flames appeared near the same location and spread. Seventy-two seconds after launch, the shuttle was obscured by white smoke, followed by an explosive fireball. The seven crewmembers aboard Challenger were killed.

An investigation into the accident, the Rogers Commission Report, determined that the O-rings used in a joint seal on the rockets were inappropriate for the ambient temperature at the time of launch: 36 degrees Fahrenheit (2.2 degrees Celsius). In the cold, the rings reacted differently to the compressive forces of liftoff. The commission found that some engineers were aware of the issue, and had advised against launches in ambient air temperatures below 53 degrees Fahrenheit (12 degrees Celsius).

https://www.livescience.com/55619-engineering-disasters.html

Case study: Challenger

1) Who is at fault?

2) Why are they at fault?

3) What should have been done differently?

NEXT TIME

Demonstration of GitHub for term project

- Note, I'll release the next homework (parts 1 and 2) over the weekend
  - Part 1 - learning Git
  - Part 2 - GitHub setup (among other things)

Intro to SE

- Not necessarily in this order!