Managing Risks in Software Design

Embedded EthiCS, Spring 2023
About Me

I am Michael Pope

Embedded EthiCS @ Harvard
Embedded EthiCS

**Identify** ethical and social issues

**Reason** through ethical and social issues

**Communicate** reasoned positions

**Design** responsible systems
Agenda

1. Trade-offs and Risks in System Design
2. Managing *ex ante* Risks
3. Engagement with Stakeholders
4. Applications and Balancing Goals
Understanding Risk
Ford’s Pinto

• Ford’s first subcompact car

• Available from 1971-1980 in North America

• Ford produced over 3.1 million units
Ford’s Pinto
Computer controlled two modes:
- High-current indirect
- Low-current direct

Malfunction problem:
- Code error
- Opaque interface
- Removal of safety measures

Therac-25
Risks Before and After Harms

- \textit{Ex post}: Recognition of risks after harm occurs

- \textit{Ex Ante}: Recognition of risks before harm occurs
“In our private lives, most low-probability risks we run will never ripen into harm . . . Not so in the public policy realm, where the rules we choose often govern millions or hundreds of millions of events over the long term . . . We can often reduce those risks by greater precautions. But at a certain point such precautions become prohibitively costly . . .”

– Barbara Fried
In an uncertain world, what factors should influence *ex ante* risk assessments?
Managing Risk
Two Possible Approaches

Rights-based Approach

Expected Utility Maximization
Goal Setting: Stakeholder needs and interests

- Customers, employees, administrators, bystanders
- “What product are we making?”
- “What features should that product have?”
Improving Software Design

Goal Setting: Stakeholder needs and interests

Testing: How should we balance risks?

- Validation: “Does the product meet requirements?”
- Verification: “Does the product meet the requirements correctly?”
Why include stakeholder interests and needs in design?

Inductive Risk: What if we’re wrong?

1. Software developers must decide whether a piece of software is ready for deployment.
2. Deployed systems can positively or negatively impact stakeholders’ values, needs, and interests.
3. Decisions about whether software is ready for deployment depend in part on judgments about the risks of error.
4. Therefore, stakeholder values should influence whether a piece of software is judged ready for deployment.
Striking a balance between risk aversion and risk tolerance can fall short of consensus.
Prudent Vigilance

Precautionary Principle

Proactionary Principle

Testing: Stewardship Model
(1) Be explicit about goals and justifications

(2) Maintain independent oversight

(3) Publicly report results (including unintended consequences)

(4) Include mechanisms for “democratic oversight” in collaboration with stakeholders

Prudential Vigilance
When is it safe enough?

- Full Self-driving 2023 Voluntary Recall
- Coding errors, resulting in: erratic braking and lane changes, rolling stops, etc.
When are formal methods required?

1. Tax filing software that calculates how much you should pay on your taxes.
2. Presentation app (e.g., Powerpoint or Keynote).
3. Game app where children ages 3+ can design the make-up and dresses of their favorite princesses.
4. Online data profile platform (e.g., OKCupid), which houses sensitive information, such as users’ sexual orientation and history of (in)fidelity.
5. Mobile payment service (e.g. ApplePay, Venmo).
6. Home security alarm system.
7. Wearable tech (e.g. a FitBit or AppleWatch), which tracks and logs a users’ heart rate.

Why are formal methods required?
THANKS!

Exit Survey