Software Engineering Ethics: Intro

What is “ethics”?  
- Broadly refers to the concern humans have for figuring out “how best to live”  
- Socrates (399 BC): “the most important thing is not life, but the good life”  
- No one wants to live a “bad” life (e.g., shame, sadness, etc.)  
- But what is the best way to achieve a “good” life (not only acceptable, but excellent and worthy of admiration)?  
- Which is what the study of ethics attempts to answer

In work and personal life ...  
- On a personal level, ethics expressed through self reflection, striving to become a better person  
- In work, expressed through formal codes and standards that all members of a profession are held (e.g., medical or legal ethics)

Software engineering ethics  
- Engineering ethics is a well-developed area of professional ethics  
- 1st codes circa 1912, Nat. Soc. of Prof. Engineers (NSPE) 1st formal canons 1946  
- Engineers expected to learn and live up to ethical standards (e.g., ABET)  
- Software Engineering is a relatively young practice, but has its own code of ethics (ACM/IEEE, first version early-to-mid 90’s)

Software engineering ethics can be complex ...  
- Engineering focus primarily on catastrophes:  
  - loss of life or injury from ethical lapses (e.g., Challenger explosion, Ford Pinto fires, Union Carbide/Bhopal disaster, ...)  
  - Engineer’s most basic ethical duty is to “hold paramount the safety, health, and welfare of the public”  
- Software engineers don’t build cars, rockets, bridges, dams, ...  
  - however, these depend upon critical software for their operation and/or design  
  - where software failure/bugs could lead to catastrophic loss

---

1 Adapted from “An Introduction to Software Engineering Ethics”, Vallor and Narayanan.
Exercise 1: Why else should software engineers take professional ethics seriously?

Software engineering has increasingly shortened delivery lifecycles

- instead of software review by management or legal teams
- individual or teams of developers directly release code to end users
- often technical, but no ethical oversight ... who is thinking about ethics?
- engineering projects have many-year lifecycles and multiple layers of oversight
- very hard for malicious (or careless) engineers to sneak past standards / safety checks

The scale of software markets

- software can be used / marketed to the entire world (unlike a bridge)
- through this scale can come great good or great harm

"On June 9, 2011, Google released a "doodle" honoring Les Paul which users found addictive to play with. This is a type of project that's typically done by an individual engineer on their "20% time" in a day or two. A third party, RescueTime, estimated that 5.3 million hours were spent playing this game. Let us pause to consider that 5.3 million hours equates to about eight lifetimes." (Vallor & Narayanan)

- Some issues ...
  - Did the doodle make a positive contribution to the world?
  - Do engineers at Google have an obligation to consider this question before releasing features?

- Also note the disproportionality between amount of time spent creating the feature (a few person days) and the amount of time users spend on it (several lifetimes)

- Engineer’s today grapple with these issues instead of management or anyone else

Lack of geographic constraints

- software engineers often culturally unfamiliar with many of their users
- potential for privacy violations, cultural offenses, etc

"People in many countries are notoriously sensitive to the representation of disputed border territories on maps. In one recent example, an error in Google maps led to Nicaragua dispatching forces to its border with Costa Rica. Google then worked with the US State Department officials to correct the error." (Vallor & Narayanan)
Software engineers are people too (personal ethics)

• like everyone, we have a desire to flourish and do well in life and work

• imagine being faced with a project that presents serious risks to users
  – will you act in a way you’d be comfortable with if it later became public knowledge?
  – would it matter to you whether your family was shamed by your actions?
  – would it matter to you if you saw this later as one of your better or worst moments?
  – could you trust anyone to whom these outcomes didn’t matter?

• thus, professional ethics has a personal component

But personal ethics alone aren’t enough

• professional ethics help to see how personal standards/values apply to your work

• wanting to have integrity is great, but what does integrity look like in a software engineer?

• what coding practices demonstrate integrity? or a lack of it?
Moral Harms

Failures of critical software systems can result in catastrophic losses

- if these are a result of software engineers’ choices to ignore their professional obligations
- then these harms are clearly consequences of unethical professional behavior

But what other harms do software engineers have an ethical duty to consider and prevent?

Case Study 1

Case Study 2