CS5030

Ethics in Software Engineering

Learning outcomes

- On completing this lecture and associated reading, you should be
  - able to identify ethical issues that are applicable to software engineering
  - aware of the Code of Ethics and Professional Practice issued by ACM / IEEE

Ethics

- A system or set of moral principles
  - helps us differentiate right from wrong

Why do we need ethics in SE?

- Software everywhere
  - Critical to the functioning of our society and the wellbeing of its people
  - More than half the world’s population uses the internet
    - Software literacy?

- Professional and moral responsibility of software engineers

Informally, …

- What do we engineer?
- How do we engineer it?
- There may not be straightforward answers to ethical questions

Debt in software engineering

- Technical debt
  - Incurring additional effort or cost in the future by deliberately choosing short term goals over quality

- Ethical debt
  - Potential negative impact on society, including unintended consequences

Taking responsibility

- Michael Armstrong, the former CEO of AT&T, once observed that

“the ancient Romans had a tradition: whenever one of their engineers constructed an arch, as the capstone was hoisted into place, the engineer assumed accountability for his work in the most profound way possible: he stood under the arch.”

Morality and the Software Architect, Grady Booch, 2008

ACM / IEEE code of ethics

- Public interest
- Client and employer interests
- Products meeting highest standards
- Professional judgment
- Ethical management of software projects
- Integrity and reputation of profession
- Being fair and supportive to colleagues
- Self-improvement through life-long learning

Example: principles for ethical AI [Floridi & Cowls]

- Beneficence
  - Promoting well-being, preserving dignity, and sustaining the planet
- Non-Maleficence
  - Privacy, security and ‘capability caution’
- Autonomy
  - The power to decide (to decide)
- Justice
  - Promoting prosperity, preserving solidarity, avoiding unfairness
- Explicability
  - Enabling the other principles through intelligibility and accountability

From: https://hdsr.mitpress.mit.edu/pub/l0jsh9d1/release/7

Some aspects of software ethics

- Data security and privacy
- Fairness of algorithms – avoiding bias
- Consideration of unintended uses / consequences
- Quality of product
- Fairness of business / working practices
- Sustainability
- …

Risks of being unethical [Floridi]

- Ethics shopping
  - Choosing ethical principles that justify current behaviour
- Ethics bluewashing
  - Appearing to be more ethical than they are
- Ethics lobbying
  - Arguing against or attempting to weaken legal enforcement citing self-regulation
- Ethics dumping
  - Moving unethical practices to / using outcomes from regions with weaker ethical enforcement
- Ethics shirking
  - Neglecting ethics where it does not bring other benefits

From: https://link.springer.com/article/10.1007/s13347-019-00354-x

Some challenges of software ethics

- Codes of ethics are usually principles
  - Need to be interpreted and translated into more precise rules that can be applied in practice

- There may not straightforward answers or solutions
  - Usually involve trade-offs

- Lack of process and tool support
  - Ethical deliberation is not usually built into the software lifecycle or organisational culture

- Deciding where the responsibility lies

Key points

- Software engineers have responsibilities to society and their profession
- They should be concerned with ethical as well as technical considerations
- Ethical and professional standards expected of software engineers can be found in codes of conduct published by professional bodies
- There are challenges in translating ethical principles into practice
- Software can be ethical and innovative