Moral Responsibility and Technology

- **Agent**: The entity that performs the action and causes something to happen
- **Patient**: The entity that is affected by the action
- **Moral responsibility**: deals with the link between the agent and the patient. Circumstances for ascribing moral responsibility is not always clear with regard to technology especially when humans and technology interact and affect each other
- When assigning moral responsibility to some person or group, what are characteristics we should look for in their actions and/or outcomes?
  - Causality: a causal connection between the person/group and the outcome of their actions
  - Free will: the ability to freely choose how to act
  - Knowledge: the ability to consider consequences of their actions
  - Constraints: e.g. time constraints – unable to react in time
- What methods are available to us to hold people and groups accountable for their actions? How effective are they and what are their drawbacks?
  - Employees of a company can walk out / protest. Effectiveness depends on the company and also on the number of people
  - Government intervention e.g. prosecution or introduce regulation. Effective after the fact but could also be slow to take effect
  - Public shaming e.g. on social media. Effectiveness depends on the number of people and who is doing the shaming
  - Boycott. Effectiveness depends on the number of people and the impact it has on the company
- How do we deal with the **many hands problem**, i.e. the fact that it is difficult to determine who was responsible when many individual people contribute to the outcome?
  - Third party testing
o Testing in general – module testing, integration testing, regression testing

o Divide responsibility – could do it by modules or could have a hierarchy. Drawback: pointing figures at other parties

o Audit trail – who touched which piece of code last, last person in the authorization trail

o Deployment study – have the software engineers work with the end users to find bugs / issues

o Holding the entire group responsible?
  ▪ Pros: accountability
  ▪ Cons: hard to figure out who worked on it, fairness

o Whistleblowers: mechanism to allow reporting of violations, need to protect whistleblowers

• How can we deal with technologies that can make it hard to understand or consider the outcomes?
  o Software testing
  o Proactive considerations of outcomes
  o Being proactive about updating the technology
  o Corporate culture
  o Standards, best practices

• What about cases where we make decisions based on the outputs of technologies that we don’t even fully understand?
  o Explain it (GDPR)
  o Authority-in-the-loop

Therac-25

• Radiation machine that killed several people in the 80s by delivering a severe radiation overdose, mainly due to software problems

• Who was morally responsible?
  o Project lead for software? Had the final say
  o Doctor / technicians? They should have seen the error and understood the risks (but error was not even in the manual)
  o Testers? Should have done a better job of software testing. Integration testers could have done a better job.
- User Interface? Clearly not effective
- Management? Shouldn't put one programmer on the critical module
- Hardware safeguards removed – designers at fault?

- What actions should have been taken to hold them accountable for that harm?
  - Quality assurance
  - Government oversight
  - Criminal charges? Firings?
  - Civil lawsuits

- What lessons can we take away from Therac-25?
  - Overconfidence in software is bad
  - Reliability is not the same as safety
  - Lack of defensive design
  - Unrealistic risk assessments
  - Inadequate investigations/followups
  - Software reuse
  - Better user interfaces that are safe (not just friendly)
  - Govt oversight needed