Karlsruhe Institute of Technology

Software Engineering II

Prof. Dr. Ralf H. Reussner

Topic 19

Ethos Software Engineer

DSIS – DEPENDABILITY OF SOFTWARE-INTENSIVE SYSTEMS
KASTEL – INSTITUTE OF INFORMATION SECURITY AND DEPENDABILITY

dsis.kastel.kit.edu

Ethos

Ethos is (according to the Duden)

- “A mindset shaped by an awareness of moral values”
- “Overall attitude”
- “Ethical consciousness”
- It pertains to individuals or a community (e.g., a professional group).

Ethic is

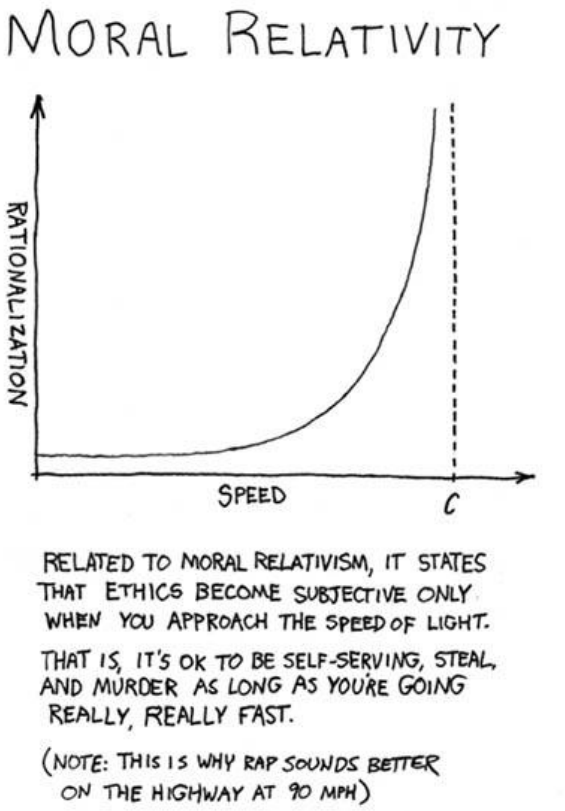
- Subfield of practical philosophy (philosophy of action)
- Deals with the conditions and evaluation of human actions
- Methodical reflection on morality. [Wikipedia]

Ethos for Software Engineers: “Conscious reflection and evaluation of the potential consequences of using the developed software.”

Ethos SE

- This awareness is more or less pronounced depending on the field.
  - Example of a field of software usage with a stronger awareness that requires reflection on actions:

    Software for weapons.
- As software enables new applications and increasingly “digitizes” our lives, new ethical questions and challenges continue to arise.
- New “software impacts” that need to be reflected upon and evaluated continue to emerge.
- <u>Examples:</u>
  - Data Analysis Software
  - Social Media
  - ...



Analogy to the Diesel Scandal

- **Caveat:** Ethical reflection is much more complex than just technical questions (management, public relations, etc.), but we are focusing on the role of the engineer.
- Software modes required for specific test bench situations.
  - Electronic engine management is now linked with ABS, ESP, etc.
- Regulations for emissions measurement **do not** explicitly prohibit this.
- Deviations between emissions values from roller test benches and those in real-world operation were known and accepted.
- Filter systems did not achieve the expected (and already marketed) success.
- **Software mode for the test bench controlled the engine** in such a way that **power and torque decreased**, but **nitrogen oxide emissions were also reduced** compared to real-world operating mode.
- In the last point, a line was crossed in retrospect that should have been critically evaluated from an ethical perspective.

Cambridge Analytica as the Diesel Scandal of Computer Science

Just a quick reminder:

- *Cambridge Analytica*: “was a *data analysis* company founded in 2014 by the British *SCL Group*, which filed for *bankruptcy* in May 2018. It was headquartered in *New York City* and collected and analyzed large-scale data on potential voters with the goal of influencing voter behavior through individually tailored messages (*microtargeting*).” (*Wikipedia*)

- *Microtargeting* requires data about voters in order to categorize them according to psychological models.

Cambridge Analytica as the Diesel Scandal of Computer Science

- *Micromarketing* was first used in Barack Obama's 2008 campaign to specifically target swing voters in swing states.
- Cambridge Analytica used Facebook profiles of users who had *"consented"* (to receive an *"incentive"* for completing a survey) as well as those of their *"friends."*
- This complies with Facebook's terms of use (Data from friends is considered part of one's own data, to which one consents for use.)
- Microtargeting is the digital advancement of *"targeted"* advertising and is legal.
- ***In the Trump election campaign, fake news was deliberately sent to targeted voters.***
- In retrospect, a boundary was crossed at the last point that should have been critically evaluated from an ethical perspective.
  - (Although the dissemination of *fake news* is covered by freedom of speech and has occurred widely on social media both before and after.)

Is regulation the solution?

- Software-based innovations almost inevitably exist in areas of our lives that are not regulated.
  - → “I am allowed to do anything with a clear conscience as long as it is not prohibited by law” is even less suitable as a maxim than usual.
- Promote regulation (== enforce democratically legitimized values and the value of democracy through laws).
- **Difficult:**
  - Drafting the laws: Balancing values, with dividing lines difficult to define universally.
    - Example: Freedom of speech versus insulting a person or damaging democratic institutions.
  - Enforcement: can be difficult on the internet/in software development.
    - Example: Automated detection of violations is difficult and would ultimately mean constant surveillance.
    - But possible: Software approval (similar to type approval by TÜV for vehicle designs).
- Regulation will never be the complete solution!
  - Too slow, too vague, too difficult to enforce.

Ethos of the Individual

Because laws are not in place and regulation will never be a complete solution:

- We must see it as the responsibility of the **individual**:
  - Decide whether the software you're currently developing violates values in its operation.
  - Decide whether you can take responsibility for the development yourself.

Counterarguments

- *"You can't really predict the effects of the software every time."*
  - Example: social media (→ violent videos, deadly "challenges," hate speech, fake news, …)
  - That's true, but often over time, the effects become known, and then it's still good to respond to them.
- *"As an individual, you can't make a difference anyway."*
  - "That's true, but the argument 'if I don't do it, someone else will' is also not a good maxim for ethical behavior, as it would justify everything."
- *"I am just following instructions; someone else is responsible."*
  - That might be true, but we've heard that argument in Germany before (e.g., as a justification for participation in the Holocaust). Does it hold up there?
- *"If I don't implement this as expected, I will lose my job."*
  - It may be true, but with the next employer, I might earn more as a software engineer (certainly not true for all professions, but we are in this "luxury situation"). However, this also means even more responsibility for one's own actions!

What should I do, then?

Critically evaluate the consequences of one's own actions.

"I didn't mean for it to happen this way" is not a good excuse if it could have realistically been foreseen.

- This requires engagement with the world outside the technical system.
- This requires considerations and assessments:
  - How likely is it that a scenario will occur that leads my actions to negative consequences?
  - Which scenario has worse outcomes?
  - What is the effect of "not doing it"?
  - Not having prevented something when you could have is also culpability.

Thought Tasks

… unfortunately current and unfortunately real☹

- Should I participate in the armament?
  - What could happen if we arm? How likely are scenarios where this armament is used differently than I currently think is acceptable? What damage could it cause?
  - What could happen if we don't arm?
  - How likely are scenarios where a lack of armament leads to war? How severe would the damage be?
- Should I contribute to systems for controlling social media?
  - Under what conditions could such systems be abused? What would the damage be?
  - How great would the damage be if social media could operate uncontrollably? How likely would the damage be?

In Summary

- Software-based innovations are almost inevitably also present in unregulated areas of our lives.
  - → “I am allowed to do anything in good conscience that is not prohibited by law” is even less suitable as a principle than usual.
- Examples of the scandals show that it was always a small step,
  - which, in isolation, seemed to require little reflection, but in context, made up the scandal.
- Regulation will never be a complete solution
  - → We must see it as the responsibility of the individual:
- Decide whether the software you are currently developing violates values in operation.
- Fortunately, software engineers are in such high demand that we can have an impact!

Research Ethics by Randall Munroe (CC BY-NC 2.5) xkcd.com/1390/