HARVARD
John A. Paulson
School of Engineering
and Applied Sciences

CS153: Compilers
Lecture 23:
Embedded EthiCS module

Meica Magnani

https://www.seas.harvard.edu/courses/cs153

Announcements

- HW6: Optimization and Data Analysis
  - Due: Tue Dec 3

Embedded EthiCS Module

- Embedded EthiCS
  - Collaboration with Philosophy Dept
  - Integrated into many CS courses
    - https://embeddedethics.seas.harvard.edu/
- Today: Ethics of Open Source
  - Guest lecturer Dr. Meica Magnani

Ethics of Free/Open Source Software Development

Who are you? Why are we talking about this?

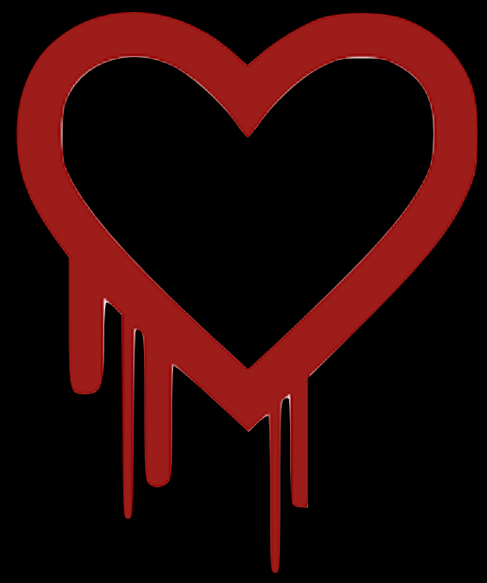
Meica Magnani, mmagnani@fas.harvard.edu

Embedded Ethics, Philosophy Postdoctoral Fellow

Office Hour: November 25, 12-1pm, Emerson 303

Assignment

- Pick one of the principles listed in the Open Source Initiative definition of open source software
- One paragraph: provide and explain an ethical consideration in support of the principle
- One paragraph: provide and explain an ethical consideration against the principle
- Due: Friday 29th by 11:59pm



- OpenSSL: secures substantial part of the web
- Was maintained by a small handful of core developers (a codebase of nearly a half million lines of code!)
- Failed to catch trivial bug in Dec 2011: **HEARTBLEED**
- Cause: maintainer burn out, lack of funding

Agenda

- Free/Open Source Software
- Short History
- Argument from Freedom
- Economic Arguments
- Identifying Possible Ethical Concerns
- Philosophical Tools: Compensation of Maintainers

Free/Open Source Software

- “Free as in speech, not free as in beer”
- Not Freeware
- Free to:

1. run the software
2. see how it works
3. redistribute the software
4. modify and improve software

History

1950s/60s: academics and corporates collaborate; software shared as public-domain

1974: CONTU — computer programs subject to copyright

1983: IBM adopts policy of not distributing source code, results in larger trend (informal communities continue to share)

1985: Stallman forms the Free Software Foundation (FSF); GNU Manifesto; goal to build operating system free of source code constraints

1989: releases GNU Public License

1991: Linux kernel integrated with GNU; finally full operating system

1997: Eric Raymond publishes “the Cathedral and the Bazaar”; inspires Netscape to release Netscape Communicator as free software

1998: “open source” rebranding; strategy to make more palatable to corporate world; Open Source Initiative founded

Richard Stallman

Eric Raymond

Linus Torvalds

Free/Open Source Software Development

- Free Software Federation: supports the development of free software, “free software is the software that grants the user the freedom to share, study, and modify it”
- Open Source Initiative: “educate and advocate for the superiority of an open development process”
- Values (Freedom, Collaboration, Generosity, Openness) and a corresponding Economic Model
- Set of Licensing Practices

Argument from Freedom

“What does society need? It needs information that is truly available to its citizens — for example, programs that people can read, fix, adapt, and improve, not just operate. But what software owners typically deliver is a black box that we can’t study or change.

Society also needs freedom. When a program has an owner, the users lose freedom to control part of their own lives.

And, above all, society needs to encourage the spirit of voluntary cooperation in its citizens. When software owners tell us that helping our neighbors in a natural way is “piracy”, they pollute our society’s civic spirit.

This is why we say that free software is a matter of freedom, not price.”

–Richard Stallman

Identify three different forms of freedom in this argument.

In what sense, then, is“free software [...] a matter of freedom”?

Freedom

1. Freedom to “read, fix, adapt, and improve”, to “study and change” (free software)

2. Freedom as control over one’s own life

3. Freedom as a spirit of “voluntary cooperation” and “helping our neighbors”

Freedom

How is free software a matter of freedom?

Free software itself basically consists of a set of freedoms.

Free software is an enabling condition (makes possible) for control over one's life.

Free software promotes the spirit of voluntary collaboration.

But whose freedom?

Programmers. In their capacity as programmers.

Economic Arguments (Common Good*)

Open Source Economics

- Decentralized social production: collaboration versus institutions; throw tasks to the crowds (users, volunteers, and hobbyists)

- Shift from External to Intrinsic Motivations: altruism, generosity, social concern, innovation for innovation's sake, pride in craftsmanship

- Not Property-Based

Pros

- Captures contributions that are unprofitable for institutions
- Eliminates cost of running an institution
- Fosters a culture of altruism; social concern; generosity
- No need to rely on whoever has the means of production
- Massive sharing of resources
- Turns cognitive supply into something productive

Cons

- Free rider problems
- Eliminates paid positions, leaving individuals vulnerable
- Projects, including critical infrastructure, can be left vulnerable (example: Heartbleed)
- Code can be used for malicious ends (e.g. Seth Vargo and ICE)
- Tragedy of the Commons

Tragedy of the Commons

Situation in which there is a shared resource and individuals, when acting independently according to their own self-interest, behave contrary to the common good (benefit of all) by depleting or spoiling the shared resource through overconsumption or under investment.

Corporations: Solutions but also More Problems

- Potential source of funding! Contribute by paying for developers and maintainers.
- Certain projects still left vulnerable.

- Power differential, huge influence.
- Potential employees, in order to be competitive, must build resume by contributing to open source.
- Corporations use code and software that was developed with labor they did not pay for.

Ethical Concerns

- Well-being of individuals
- Freedoms of individuals
- Social welfare
- Fairness

Ethical Concerns

- Vulnerability of critical infrastructure and software
- Compensation of maintainers, developers
- Corporations profiting from unpaid labor
- Coding/software industry has unfair entrance expectations
- Corporations having too much power (determining which projects are vulnerable, which are supported)
- Use of code for malicious ends

Claim: Maintainers ought to be adequately funded.

Exploitation

- “Maintainers are being exploited”
- Exploitation: asymmetry of power relationship between worker and employee; to use the vulnerability of another for one’s own benefit
- Question to ask: Are maintainers in a position of vulnerability? Is that vulnerability being taken advantage of?

Fairness

- Maintainers are producing a good. "They deserve *fair compensation*."
- Justice in pay: 'agreement view' versus 'contribution view'
    - Public Good: contribute to well-being of society (example: childrearing, caretaking)

Common Good

- Society's security depends on maintainers. Maintainers can only do their job well if adequately funded. The well-being of society, thus, requires that maintainers be adequately funded.
- Common good: shared and beneficial for most members of society
- Note: we are no longer concerned with what is owed to to the maintainer, but what is owed to society

Conclusion

- Free/Open Source Software Development is driven by both a set of values and economic considerations
- Both are good targets for ethical scrutiny!
- General rule of thumb for identifying potential ethical concerns (well-being, freedom, social welfare, fairness); more specific ethical concepts: exploitation, fair compensation, common good
- Final discussion question: Is the value of freedom supported by the open source economic model?

http://bit.ly/cs153ethics