Ethics in Software Engineering

CS350 Introduction to Software Engineering

Shin Yoo

Ethical Issues in Software Development

(a random selection that is by no means exhaustive…)

- Professional Conduct
- Whistleblowing
- Code Provenance
- (Mostly excerpts from CS489 Computer Ethic and Social Issues)

Engineer as a Profession

- profession (n): a paid occupation, especially one that involves prolonged training and a formal qualification (OED)
- Engineers in other fields need to get formal qualification (license) to practice
  - Architects, civil engineers, electrical engineers, etc…
- Why are software engineers not licensed?
  - Too young a discipline? Cultural reasons? Too easy? Too hard? No consensus on how exactly to qualify?

Requirements of a Professional

- Highly developed skills and deep domain knowledge
- Autonomy: you are supposed to know better (than the client) and make the right decision
- Observance of a code of conduct
  - Professional / personal / institutional / community

ACM Code of Ethics and Professional Conduct

- Association for Computing Machinery: established in 1947, the largest scientific and educational computing society (currently over 100,000 student/professional members)
- Executive Council voted to adopt a Code of Ethics in 1992: 24 imperatives that define the personal responsibilities of computing professionals.
- The latest version was created in 2018: https://www.acm.org/code-of-ethics

1. General Ethical Principles:

“A computing professional should…”

1. Contribute to society and to human well-being, acknowledging that all people are stakeholders in computing.
2. Avoid harm.
3. Be honest and trustworthy.
4. Be fair and take action not to discriminate.
5. Respect the work required to produce new ideas, inventions, creative works, and computing artifacts.
6. Respect privacy.
7. Honor confidentiality.

2. Professional Responsibilities:

“A computing professional should…”

1. Strive to achieve high quality in both the processes and products of professional work.
2. Maintain high standards of professional competence, conduct, and ethical practice.
3. Know and respect existing rules pertaining to professional work.
4. Accept and provide appropriate professional review.
5. Give comprehensive and thorough evaluations of computer systems and their impacts, including analysis of possible risks.

2. Professional Responsibilities:

“A computing professional should…”

6. Perform work only in areas of competence.
7. Foster public awareness and understanding of computing, related technologies, and their consequences.
8. Access computing and communication resources only when authorized or when compelled by the public good.
9. Design and implement systems that are robustly and usably secure.

3. Professional Leadership:

“A computing professional, especially one acting as a leader, should…”

1. Ensure that the public good is the central concern during all professional computing work.

2. Articulate, encourage acceptance of, and evaluate fulfillment of social responsibilities by members of the organization or group.

3. Manage personnel and resources to enhance the quality of working life.

4. Articulate, apply, and support policies and processes that reflect the principles of the Code.

3. Professional Leadership:

“A computing professional, especially one acting as a leader, should…”

5. Create opportunities for members of the organization or group to grow as professionals.

6. Use care when modifying or retiring systems.

7. Recognize and take special care of systems that become integrated into the infrastructure of society.

4. Compliance with the Code:

“A computing professional should…”

1. Uphold, promote, and respect the principles of the Code.
2. Treat violations of the Code as inconsistent with membership in the ACM.

2015 San Bernardino Attack and Encryption Row

- On 2 December 2015, there was a mass shooting in San Bernardino, California: 14 were killed, 22 seriously injured (see https://en.wikipedia.org/wiki/2015_San_Bernardino_attack#Motive for details)

- This incident has put the tension between governments and commercial encryption technology in the highlight.

Phone Decryption

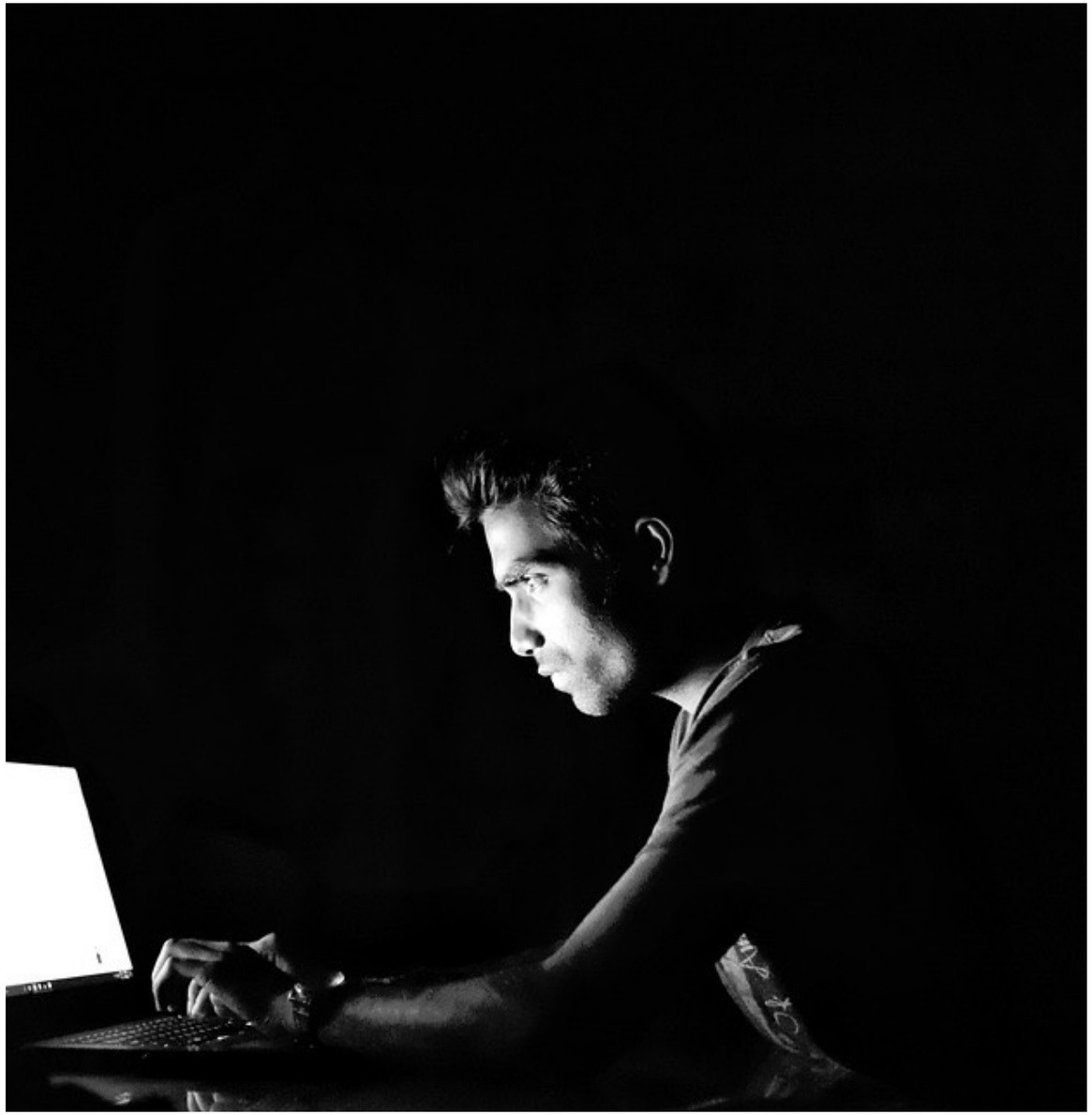
- On 9 February 2016, FBI accounted that it cannot unlock the phone used by one of the shooters (iPhone 5C), and asked Apple to create a special version of iOS that opens a back-door
- Apple declined.
- FBI successfully issued a court order, with the deadline of 26 February 2016.
- Apple still declined.
- On 19 February 2016, DoJ asked Apple to install a malware inside Apple's campus, to allow FBI to remotely hack the phone.
- Apple declined, and announced that, while the company initially cooperated with FBI, one of the promising methods has been rendered useless due to an earlier mistake.

Phone Decryption

- On 28 March 2016, DoJ announced that it unlocked the iPhone, and dropped the suit against Apple.
  - Some claim that an Israeli company, Cellebrite, helped FBI. There are reports that FBI worked with hackers who exploited a zero-day vulnerability.
- In March 2018, LA Times reported that there was nothing useful for investigation in the phone.

Should Apple have complied?

- Back in 2016, 45% of Americans supported Apple’s stance, while 50% supported FBI.
- Do you support Apple, or the US Government?
- Does ACM Code of Ethics have a relevant point here?
- #discussions



Whistle-blower

- *n.* a person who informs on a person or organization regarded as engaging in an unlawful or immoral activity.
- Internal conflict: a whistle-blower often knows that his/her alarms pose a threat to anyone who benefits from the ongoing practice
- External conflict: common ethics require loyalty to your profession, but formal code of professional ethics stress responsibility to the public

Three Elements of Whistle-Blowing (S. Bok, 1982)

- **Dissent**: whistle-blower publicly disagrees with an authority, or a majority view, usually to highlight a negligence or abuse
- **Breach of Loyalty**: whistle-blower goes against his/her own team, violating the obligation to colleagues
- **Accusation:** whistle-blower is effectively singling out a person or a group to call foul

Secrets: On the Ethics of Concealment and Revelation, Sissela Bok, 1982

Individual Moral Choices

- Certain issues are so outrageous that anyone in the position to warn the public almost have to do so, while other matters are so minor that whistle-blowing may be a disproportionate reaction.
- In the middle lies the wide spectrum of matters that will trouble the whistle-blower.
- The three elements provide guildelines.

Individual Moral Choices

- Of dissent, you need accuracy: can you justify your action with sufficient evidence or expertise? Or do you simply have suspicion? Have you considered the damage a false alarm can do?
- Of breach of loyalty, you need careful consideration of alternatives: have you tried to resolve the issue internally? If resolved internally, you remain loyal to both your profession and the public.
- Of accusation, you need fairness: are you really pointing your finger at the right person?

License

- Copyright license is a form of contract (“permissions agreement”) between the holder of the copyright and someone who wants to use the protected work.
- Public copyright license: blanket license that grants permissions to anyone in the general public. The holder can decide to apply public copyright license.

User (Licensee) Right	Berne (1886)	CC0
0. free access, and freedom to use the work as you wish ("use" includes to run a program or to execute a music score)	Partial	Yes
1. freedom to access the "source-code" and use it as you wish, for study or change it for personal use.	No	Yes
2. freedom to redistribute copies	No	Yes
2.1 right to quote (freedom to redistribute copies of fragments)	Yes (small amount)	Yes (any amount)
3. freedom to distribute copies of your modified versions to others	No	Yes

CC

- American Non-profit Organisation, devoted to expanding the range of creative works available for others to build upon legally and to share (https://creativecommons.org/faq/)
- Provides template licenses that people can easily adopt. Templates have evolved over time.

CC Conditions

- Attribution (by): All CC licenses require that others who use your work in any way must give you credit the way you request, but not in a way that suggests you endorse them or their use. If they want to use your work without giving you credit or for endorsement purposes, they must get your permission first.
- ShareAlike (sa): You let others copy, distribute, display, perform, and modify your work, as long as they distribute any modified work on the same terms. If they want to distribute modified works under other terms, they must get your permission first.

CC Conditions

- NonCommercial (nc): You let others copy, distribute, display, perform, and (unless you have chosen NoDerivatives) modify and use your work for any purpose other than commercially unless they get your permission first.
- NoDerivatives (nd): You let others copy, distribute, display and perform only original copies of your work. If they want to modify your work, they must get your permission first.

CC Licenses

- CC0
- Attribution (CC BY)
- Attribution ShareAlike (CC BY-SA)
- Attribution-NoDerives (CC BY-ND)
- Attribution-NoCommercial (CC BY-NC)
- Attribution-NoCommercial-ShareAlike (CC BY-NC-SA)
- Attribution-NoCommercial-NoDerives (CC BY-NC-ND)

CopyLeft

- The simplest way to make a program free software is to put it in the public domain, uncopyrighted. This allows people to share the program and their improvements, if they are so minded. But it also allows uncooperative people to convert the program into proprietary software. They can make changes, many or few, and distribute the result as a proprietary product.

- To copyleft a program, we first state that it is copyrighted; then we add distribution terms, which are a legal instrument that gives everyone the rights to use, modify, and redistribute the program's code, or any program derived from it, but only if the distribution terms are unchanged. Thus, the code and the freedoms become legally inseparable.

- Copyleft is a way of using the copyright on the program. It doesn't mean abandoning the copyright; in fact, doing so would make copyleft impossible. The “left” in “copyleft” is not a reference to the verb “to leave”—only to the direction which is the mirror image of “right”.

https://www.gnu.org/copyleft/

Famous Open Source Licences

License	Linking	Distribution	Modification	Sublicensing
GPL v3	Only GPL v3	Copylefted	Copylefted	Copylefted
GPL Lesser				Copylefted
BSD				Permissive
Apache License				Permissive
MIT	Permissive	Permissive	Permissive	Permissive
Mozilla Public License	Permissive	Copylefted	Copylefted	Copylefted
Beerware	Permissive	Permissive	Permissive	Permissive

```
/*
 * ----------------------------------------------------------------------------
 * "THE BEER-WARE LICENSE" (Revision 42):
 * <phk@FreeBSD.ORG> wrote this file.  As long as you retain this notice you
 * can do whatever you want with this stuff. If we meet some day, and you think
 * this stuff is worth it, you can buy me a beer in return.   Poul-Henning Kamp
 * ----------------------------------------------------------------------------
 */
```

Code Provenance

- Provenance: *n.* the place of origin or earliest known history of something
- In many cases, IP decisions boil down to where code came from, at the lowest level.
- “Have A copied the code X from B?”
- Copied code are called *code clones*.

Clone Detection

- Clones have been studied for multiple reasons.
  - Provenance in the legal context
  - Plagiarism as a specific context of provenance, as well as in the educational context
  - Productivity as there are various views that clones affect software development lifecycle

Types of Clones

- Type 1: exact copy without modifications (except for whitespaces and comments)
- Type 2: syntactically identical copy, with variable names, types, and/or function identifiers changed
- Type 3: copy with further modifications such as swapped line order, etc
- Type 4: semantically identical computation but written in a different logic

Comparison Methods

- Textual Comparison: simply compare line by line - brittle but also language agnostic.
- Token Comparison: compare lines as sequences of tokens (concrete values are abstracted)
- Metric Comparison: collect a set of metrics about code, and compare the metric vectors instead of actual code
- Abstract Syntax Tree: partition AST subtrees by hash and use tree matching algorithm
- Program Dependence Graph (PDG): use graph matching algorithm to compare PDGs - approximative matching since graph isomorphism is NP-hard

Conclusion

- Software is everywhere and increasingly more important. With great power comes great responsibility.
- Understand the standard code of conduct.
- Understand different licenses and treat them seriously.