

The Public Is The Priority: Making Decisions Using The Software Engineering Code Of Ethics

Donald Gotterbarn

Keith W. Miller

Year of Publication: 2009

What is Code of Ethics?

- Code of Ethics are often seen as a way to manage the behavior of members of profession.
- Important challenges described by the author in this paper is how to use the code to balance multiple factors when deciding the best course of action.
- The code can help the software developers to make ethical, complex and technical decisions which will be better for the public, profession and the developer as well.
- Now, we will look into 3 reports which will demonstrate how a professional can use the code to make a decision whenever an ethical conflict arises.

Case 1: Design Risk

- Imagine you work for a company that develops software for military and law enforcements agencies.
- You are working on a project which involves TTWI (which is Through-The-Wall-Imaging) that uses impulse radar to see through hard surfaces like wood, plaster, concrete and brick walls.
- Your company developed and marketed TTWI, which has become its most profitable product line.
- Now, your new project is to develop a software which can tackle TTWI, called ANTI-TTWI.
- If the second project is successful, your company will be able to sell TTWI to one company and ANTI-TTWI to its rival company.
- Here, you will act as a middleman, who can develop both type of products and can sell them to rival companies. What will be your ethical obligations as a software professional in this situation?

Resolving the Issue

- The ACM and IEEE Computer Society sponsored the development of a body of knowledge and ethical guidelines documenting the professional responsibilities and obligations of software engineers.
- Representatives including industry, government, education and military compiled a set of guidelines to inspire software engineers. The code went extensive review process before approval.
- The code summarizes the software engineering professional's ethical aspirations and explains how these aspirations can affect the way software engineers act. It also informs the public about the responsibilities that are important to this profession and educates practitioners on the standards that society expects them to meet and what their peers strive for an expect of each other.

Principles of Code

- The Code includes 8 principles that are arranged with the highest priority first:
 1. **Public**
 2. **Client And Employer**
 3. **Product**
 4. **Judgement**
 5. **Management**
 6. **Profession**
 7. **Colleagues**
 8. **Self**

Applying the Code

- It is not always easy to apply a collection of abstract principles to make a solid decision. That's why Code (which is mentioned in this paper) differs from many other code of ethics in two significant ways.
 1. First, many codes are tied to the organization and only applicable to the organizations and members of that organization only. In contrast, the Software Engineering Code of Ethics and Professional Practice is a code of the profession and not a single organization.
 2. Second, in addition to providing abstract principles the Code specifically addresses the problem of conflicting standards, which can help in making the ethical decision. Take an example of that ANTI-TTWI case, The code clearly states that as a computer professional you should be loyal to your employer. It also declares that as a software engineer, you should report any kind of problem that can put public in danger.
 3. So, these two principles are contradictory in ANTI-TTWI case. Because TTWI can endanger the safety of public and ANTI-TTWI which can misdirects weapons. As well as you should be quiet and not to divulge any information about this because you are loyal to the employer.

Code according to the paper

- The preamble to the code states that:
 - Ethical tensions can best be addressed by thoughtful consideration of fundamental principles, rather than blind reliance on detailed regulations. These principles should influence software engineers to consider broadly who is affected by their work; to examine if they and their colleagues are treating other human beings with due respect; to consider how the public, if reasonably well informed, would view their decisions; to analyze how the least empowered will be affected by their decisions; and to consider whether their acts would be judged worthy of the ideal professional working as a software engineer.
- In other words, this Code will not make a decision automatically and on its own. Professional judgement is still necessary. The software engineer is obligated to act in the public's best interest, even if those actions oppose the interests of the company.

Applying Code to the ANTI-TTWI case

- The main focus of the Code is on the welfare of the public. In all the judgements, concern for the health, safety of the public is primary principle.
- The software engineer should act as much as possible in the interest of the company; however, software engineer must act in a way that enhances the public's safety.
- In this case, software engineer has several options like
 1. Consulting with managers.
 2. Seeking a second opinion from advisor or lawyers.
 3. Conferring with executives.
- But sometimes, if there is no other way then the Code clearly states that: The software engineer is obligated to act in public's best interest, even if those actions oppose the interest of the company.
- At this point, ANTI-TTWI Case will become a whistleblower case and the personal consequences for the engineer might be catastrophic in nature.

Case 2: Who is in Control?

- On 7 October 2008, a faulty computer sent a large passenger plane into a steep dive, which injured 51 passengers onboard. Another disaster that happened in 1994 was the crash of Airbus 310 which resulted in death of 75 passengers.
- In the first case autopilot was on, but computer-generated faulty value. These incorrect value led to the flight control computer commanding a nose-down aircraft movement. After the autopilot was turned off, software gave the control to pilots. In the second case, pilot brought his daughter and son into the cockpit, letting them put their hands on the controls as the autopilot flew the plane.
- In both the cases, media pinned the blame on human factors, which are easier to explain and might underreport the importance of technological problems.

Case 3: Disclaiming Responsibility

- In August 2000, medical technicians at National Oncology Institute of Panama City modified the computerized cancer treatment software which will calculate the radiotherapy treatments. But by the late March 2001, 28 patients had been overexposed during the therapy which resulted in death of 17 patients.
- First group of investigators, placed the responsibility on users who had “misused” the treatment planning software. But, additional group of investigators painted a different picture. Their report noted that the software manufacturer included a total disclaimer of responsibility for calculations accuracy.
- In other words, all the data input was correct, but the technicians did not perform all internal test. They just assumed that the software would perform its function correctly.
- The Code provides specific details about software practitioners’ obligations, and if they ignore those obligations, they are not acting in good faith as professionals. Even though it was not intentional but however, their actions contributed to the injuries.

Conclusion

- All 3 cases involved the potential for fatalities. Computers are very bad at doing what we want them to do, but they are very good at doing what we tell them to do.
- The Software Engineering Code of Ethics and Professional Practice fulfills several functions. The code is meant to be inspirational; it encourages software engineers to undertake positive actions and resist pressure to act unethically.



Thank You

Aditya Singh Rajput
(G01326046)