

Review of the SE Code of Ethics

Principle 7: COLLEAGUES

Software engineers shall be fair to and supportive of their colleagues. In particular, software engineers shall, as appropriate:

7.01. Encourage colleagues to adhere to this Code.

7.02. Assist colleagues in professional development.

7.03. Credit fully the work of others and refrain from taking undue credit.

7.04. Review the work of others in an objective, candid, and properly-documented way.

7.05. Give a fair hearing to the opinions, concerns, or complaints of a colleague.

7.06. Assist colleagues in being fully aware of current standard work practices including policies and procedures for protecting passwords, files and other confidential information, and security measures in general.

7.07. Not unfairly intervene in the career of any colleague; however, concern for the employer, the client or public interest may compel software engineers, in good faith, to question the competence of a colleague.

7.08. In situations outside of their own areas of competence, call upon the opinions of other professionals who have competence in that area.

Section 1: Divide the sections of the SE Code of Ethics into ethical standards and professional standards.

Code 7.01 deals with the preservation of the SE Code itself; without this statement, the code eventually is not followed by anybody, as people who believe in it do not encourage others to do the same. This statement is neither ethical or professional by itself; it is as ethical and professional as you believe the entire code to be.

Code 7.02 is a professional code. It is not unethical to ignore your colleagues, but it is unprofessional.

Code 7.03 is an ethical code. Although it is also unprofessional to take credit for others' work, the basis for our choice of ethical code is that plagiarism is more unethical than unprofessional, because while using someone else's work is completely within the limits of a professional's rights (and oftentimes necessary,) not giving credit where credit is due is not right.

Code 7.04 is both ethical and professional. Although there was some disagreement in the group, performing a subjective review (as opposed to an objective one) is considered unethical. Good software could be rejected because of subjective opinions, which is unethical.

Code 7.05 is an ethical code. Professionally, it is up to the engineer to determine what is the best piece of software for the task. "Fair" is a subjective term in its own right; people who use the code each have to evaluate what "fair" means to them. It is not unprofessional to ignore the advice or criticisms of a colleague if that advice is not sound; however, it could be unethical if they are bringing up important points that were forgotten about.

Code 7.06 is a professional code. When the code is broken, it becomes an ethical issue, but security in general is a professional practice. Exposing personal information raises ethical issues like how the exposed information is handled, but the actual protecting of private information is a professional responsibility.

Code 7.07 is an ethical code. "Unfairly intervene" is a subjective term; again, as with code 7.05, it is impossible to determine what "unfairly" means. A colleague may be endangering the lives of many people with problems; reporting his problems may also unfairly progress your career over theirs. It is an ethical responsibility, especially in safety-critical software, to report problems in peers' software to a supervisor.

Code 7.08 is both ethical and professional. Although we had some group disagreement about this issue, we came to a consensus that it is unethical and unprofessional for a software engineer to produce software for professions which he has no knowledge about. It would be both ethical and professional to develop the software alongside someone in that professional field.

Discussion 2: Are there any parts of the code you feel should be removed?

In our discussion, we felt only one section of the SE Code Section 7 should be removed. Section 7.02 deals with professional responsibilities; it does not need to be in a code of ethics. While unprofessional to ignore your colleagues in development, it is not an ethical issue unless the other codes are violated (7.03-7.08), and those are more specific and easier to single out when broken.

Discussion 3: Which papers related to the subsections of section 7 of the SE Code of Ethics? Which schools of thought apply to each subsection?

The papers that we read related to Section 7.02, 7.04, 7.06, 7.07, and 7.08 of the SE Code of Ethics. The *Process of Incompetence* related mostly to 7.02 and 7.06, where colleagues may not be using the proper design process. *Unskilled and Unaware* is another important paper that relates to our section, mostly in 7.06, 7.07, and 7.08, where colleagues may believe themselves to be more skilled than they really are. Code 7.04 has many papers that deal with it: *No Silver Bullet*, *Rational Design Process*, and *Non-Testable Programs* are all papers that dealt with 7.04. Reviewing other programmer's code is always difficult, but it gets even harder if the proper design process is not followed, or the code is in an untestable environment. Code 7.08 has two other papers that deal with it; both the *Chaos Report* and *Software is Different* deal with the problem is interacting with other professions outside of software.

The different schools of thought that deal with section 7 of the SE Code are rule utilitarianism, utilitarianism, totalitarianism, deontological, and Kant. Many of the actions in section 7 deal with colleagues and treating them fairly, a deontological and utilitarian principle. Totalitarianism and rule utilitarianism both deal with 7.01, making sure that people stick to the SE Code of Ethics.