



Specifying the standard—Make it right:

A Software Engineering Code of Ethics and Professional Practice

Don Gotterbarn

Chair, Software Engineering Ethics and Professional Practice Task Force,
Software Engineering Ethics Research Institute
East Tennessee State University
gotterba@etsu.edu

The members of the computing profession have taken a stand. They have decided to make clear that it is not acceptable to do inferior work or do work which damages our basic values. To make clear what it is to be a software professional, the IEEE Computer Society and the ACM established a Joint Steering Committee for the Establishment of Software Engineering as a profession.

The obligations of a profession are to educate its members technically and socially, and to provide tools to help its members reach these standards. Tools that a profession needs to provide or support include Codes of Ethics, Ethics Tools, and Curricula. They also need to provide leadership in formulating and promoting ethical policy and provide leadership in promoting positive activity. Such activity includes the involvement of industry in training, an involvement in higher education, the development of certification standards and support for those who take ethical risks. There are significant marks that software engineering is moving in this direction.

The IEEE-CS/ACM Joint Steering Committee established a task force on Software Engineering Ethics and Professional Practices. The function of this task force was to document the ethical and professional responsibilities and obligations of software engineers. Professionals generally describe their ethical commitment in codes of ethics developed and maintained by a professional society. The IEEE-CS/ACM Joint Steering Committee for the Professionalization of Software Engineering established the Software Engineering Ethics and Professional Practice (SEEPP) task force to document and codify standards of ethical and professional practice. Recently, December 1998, both the Computer Society and the Association for Computing Machinery adopted the Software Engineering Code of Ethics and Professional Practice. The adoption of the Code by multiple societies moves it toward a profession's code rather than the Code of an individual professional society. The development of the Code was an international project with participants from every continent. The participants responded to a call for participation sent to the memberships of both societies, news groups, and other professional societies. Major companies posted early drafts of the Code for comment by their employees on their elec-

tronic bulletin boards. Reviews, re-drafts, and balloting on the Code were conducted in the international arena. It is not unreasonable to say that this Code represents movement toward an international consensus of what software engineers believe to be their professional ethical obligations.

After multiple preliminary drafts were circulated between January and June 1997, drafts of version 2 appeared in the July issues of *Computers and Society* and *ACM Software Engineering Notes* for comment. Incorporating concerns from these and other reviews version 3 was published with a turn around ballot in the *Communications of the ACM* and *IEEE Computer* in November of 1997. Version 4 was developed and underwent several significant reviews including the review process for becoming an IEEE technical standard. Under the auspices of the Computer Society a representative international balloting group was formed. They commented on the Code and it was the responsibility of the task force to satisfactorily address all concerns about the Code. The Code passed that review and then underwent legal review. These processes were complete in September of 1998 and shortly thereafter it was unanimously approved by the governing bodies of the ACM and the Computer Society.

Software engineering code of ethics and professional practice

A code helps to address several of the responsibilities of a profession. A well-designed code will help to educate several communities. If a code is sufficiently detailed it helps to educate clients and society about what can reasonably be expected from software engineers and expected from their products. In some cases the public's knowledge of correct or acceptable practice will reduce the developer's tendency to take short cuts. A detailed code can be used as a foundation for a malpractice suit against developers who intentionally fail to meet the standards specified in a code. A code also serves to educate its membership and potential membership about the standards of the profession.

"The ...code gives the aspirational elements at a high level of abstraction; the clauses that are included in the full version give examples and details of how these aspirations change the way

we act as software engineering professionals. Without the aspirations, the details can become legalistic and tedious; without the details, the aspirations can become high sounding but empty; together, the **aspirations and the details** form a cohesive code.

In addition to providing standards within the profession, an effective code will also help the practitioner make ethical decisions and help others judge the moral character of the decisions made by a the decision making process

The expression "code of ethics" is ambiguous. Some professional societies distinguish between "codes of ethics", "codes of conduct", and "codes of practice". The relationship between these types of code forms a hierarchy. Codes of ethics are different from codes of conduct and codes of practice. Codes of ethics are more aspirational. They are the mission statements of the profession. Codes of conduct are more oriented toward the professional and professional attitudes. They do not describe details about how to carry out a particular action, but they make clear the issues at stake in software engineering. Codes of practice may fix the accepted state of the art (Berleur, 1996) and relate to best current practices within the profession. The Software Engineering Code is both a Code of Ethics and a Code of Practice.

In developing the Code, several functions of a code of ethics were emphasized. For an emerging profession like software engineering, some of these functions are more critical than other functions. The choice of functions will give the code its own character. The SEEPP task force chose to emphasize Inspiration, Guidance, Education, and Support. At the early stages of a profession's development, the disciplinary function is generally taken over by the law. Unless there is a single organization that requires adherence to the code as condition of membership and membership in that society is a condition of practice, there are no realistic sanctions that can be imposed for code violation. The only form of sanction occurs when the Code is adopted as a generally accepted standard of practice, and both society and legislators view the failure to follow the Code as negligence or malpractice. Education of professionals and non-professionals about the professional obligations described in the Code is required for this type of sanction to work.

The primary functions of the Software Engineering Code of Ethics and Professional Practice were to educate and to provide guidance in decision making for the international community of software developers. The Code clearly defines the responsibility of the profession and the professional to promote and protect positive values.

The SEEPP task force had a series of goals, including:

- to help define the profession of "software engineer" as an entity distinct from other computer professionals.
- to include managerial aspects as well as technical aspects of development in the same code.

- to reflect the international scope of the software engineering profession.
- to inform software engineers at multiple levels: the code should be inspirational, but also instructional.

The Code was designed to establish and sustain a set of values for the profession, and also to help practitioners identify practical ethical issues in their work lives.

1.0 Unique elements

The Software Engineering Code of Ethics and Professional Practice is different from other codes in at least four different ways, namely, 1) it addresses three levels of professional obligation, 2) it includes a suggested way to adjudicate between potentially conflicting imperatives, 3) it provides moral principles to use as guidance in using the code for moral decision making, and 4) it presents a strategy for moral decision making.

1.1 Levels of professional obligation

The Code is organized into eight principles with Clauses under each principle. "The Clauses under each Principle consist of three different types of statement corresponding to each level. Level One: Aspire (to be human); statements of aspiration provide vision and objectives and are intended to direct professional behavior. These directives require significant ethical judgement. Level Two: Expect (to be professional); statements of expectation express the obligations of all professionals and professional attitudes. Again, they do not describe the specific behavior details, but they clearly indicate professional responsibilities in computing. Level Three: Demand (to use good practices); statements of demand assert more specific behavioral responsibilities within software engineering, which are more closely related to the current state of the art. The range of statements is from the more general aspirational statement to specific measurable requirements. (Gotterbarn, et al 1997)"

Although all three levels are in the Code, in many cases the precise distinction between the three levels is not clear. For example some may interpret a clause about testing as being at level two while others may argue that it is at level three. Nevertheless, there is a need to include all three levels in a single code. A single Code needs to include both statements of aspiration and detailed guidance.

The inclusion of detail in the Code sets the stage for judgements about the quality of software development practices. Some might even use these details as standards for enforcement practices.

"The Clauses of each Principle are illustrations of some of the obligations included in these relationships. These obligations are founded in the software engineer's humanity, special care owed to people affected by their work, and the unique elements of the practice of software engineering. The Code prescribes

these as obligations of anyone claiming to be or aspiring to be a software engineer."

1.2 Adjudicate conflicting imperatives

One of the common criticisms of codes is that the clauses of a code sometimes lead practitioners in conflicting directions or the does not prioritize the imperatives of a code. The criticism of the code is that all problems are treated equally. For examples, codes may says both "cause no harm" and "support your employer". In some cases, the ethical decision-maker must choose between seemingly contradictory imperatives. In this case the conflict is between loyalty to the employer or client versus loyalty to the public. To help address this problem, the Software Engineering Code includes a paramouncy clause. The final criterion for any ethical decision is its impact on the client and the public. "In all these judgements concern for the health, safety and welfare of the public is primary. That is, the "Public Interest" is central to this Code." (Code). An ordering of imperatives is suggested to aid the ethical software engineer in decision making. The critical issue in the Code is to determine how a decision effects the stakeholder.

The next new characteristic about the code is it extends those who are normally considered stakeholders in an ethical issue. By stakeholder- we mean individuals or groups who may be negatively affected by the project activities. In most software projects the only stakeholders identified are the practitioner and the customer/client. Investigating 16 organizational IS-related projects led [Farbey, Land and Targett, 1993] to conclude that regarding evaluation of IT investment, "... the perception of what needed to be considered was disappointingly narrow, whether it concerned the possible scope and level of use of the system, [or] the range of people who could or should have been involved ...". They discovered, with the exception of vendors, all stakeholders involved in evaluation were internal to the organizations.

In the Code, however, the minimal list of stakeholders includes: managers, customers, suppliers, communities, other professionals, society, and other employees. Once the stakeholders have been identified. They need to be incorporated into the moral judgement making process. The Code asserts that the Code is not simply a check list of agreed upon goals that one can merely pick the imperative deJour to make a decision but the when doing ethics it is often necessary engage in reflective analysis. The Code advocates the use of several fundamental principles— to help make a judgement,

1.3 Provide moral principles to use as guidance in using the code for moral decision making.

The Code embodies several philosophical principles, without their philosophical labels, to offer practical guidance in decision making. The clause below includes consequential,

Kantian deontology, and Rawlsian theory of justice. In making ethical decisions use ...

"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 speculate on 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 [Emphasis added].

1.4 Present a strategy for moral decision making.

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 speculate on 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 all these judgements concern for the health, safety and welfare of the public is primary. That is, the "Public Interest" is central to this Code.

In addition to stating ethical aspirations for the practice of software engineering, the Code identifies particular idiosyncratic needs of software engineering. The success of this effort to articulate the professional responsibilities of software engineers has already been recognized and is raising another bar. The Engineering Professional Licensing Committees of Texas has addressed the subject of professional registration of software engineers, and officially recognized software engineering as a new discipline with its own foundations and as a unique body of knowledge. They have also expressed high regard for the Software Engineering Code of Ethics and Professional Practice as a standard for software engineers.

The Software Engineering Code of Ethics and Professional Practice is a useful tool. Codes serve to educate, both prospective and existing members of a profession about the shared commitment of the members of a profession to, undertake a certain quality work and the responsibility for the well being of the customer and user of the developed product. Ethics codes also serve to educate managers of the groups of professionals about expected behavior. A manager's expectations will have an effect on what is asked of a professional. Directly and indirectly codes educate management about their responsibility for the effects and impacts of the products developed.

The Code indirectly educates the public at large about what software engineers consider to be a minimal acceptable practice, even when a non-professional practices it. Thus the code can be a catalyst to simultaneously raising the internal expectations of a profession and the expectations of the society at large.

The Code encourages the professional to do positive actions. The Code also encourages the professional to resist pressures to act unethically. The professional can appeal to the imperatives of the Code to indicate ethically accepted practice.

Principle 8 says, "Software engineers shall participate in lifelong learning regarding the practice of their profession and promote an ethical approach to it." The commitment to an ethical approach was evidenced by the hundreds of people who contributed to the development of the Code. Like all successful projects, the commitment and concern of the participants made it possible.

The development of a Code of Ethics is only one step in meeting the responsibility of a profession to mature itself.

2.0 An agenda for the profession

An agenda for meeting the responsibility of the software engineering profession should include:

- International, not merely national, support for ethically sensitive practice

- The development of international standards, for example have the Code adopted by multiple professional organizations and become part of an ISO technical standard.
- Enlist industry and government support in professionalization
- Recognize the multi-disciplinary nature of software development and of the application of software.

The profession and its representative societies need to address these issues to mature the profession of software engineering. The work of the SEEP is complete. The Computer Society and the ACM has formed another group – the Software Engineering Professional Ethics Project (SEPEP)—to help nurture the impact of the Code by developing educational material about it. If you are interested in participating in this effort, contact sepep@etsu.edu. ♦

References:

- Berleur, J. et al., "Codes of Ethics or of Conduct within IFIP and in Other computer societies", *Proceedings 13th World Computer Congress*, 1996
- Code Version 5.2 of the Software Engineering Code of Ethics and Professional Practice. Available on <http://www-cs.etsu.edu/seer/secode.htm>
- Farbey B, Land F and Targett D *How to assess your IT investment*, Butterworth Heinemann, 1993
- Gotterbarn, D, Miller, K., Rogerson, S. "Draft Software Engineering Code of Ethics (version 3)." *Communications of the ACM*, November 1997.
- Gotterbarn, D, Miller, K., Rogerson, S. "Software Engineering Code of Ethics and Professional Practice: Approved," *Communications of the ACM*, October 1999, IEEE Computer October 1999.

SIGCAS Chairs and Editors, 1969 – 1999

Chairs

Robert P. Bigelow (c. Winter 69 – May 71)
Robert Ashenurst (c. June 71 – May 73)
Peter Lycos (Aug 73 – Fall 75)
Ellis Horowitz (Winter 75 – Fall 77)
Charles Davidson (Winter 77 – Spring 81)
Doris Lidtke (Summer 81 – c. Fall 84)
Abbe Mowshowitz (c. Summer 85 – c. Summer 87)
Ron Anderson (Jan 88 – Jul 93)
Dianne Martin (Dec 93 – present)

Editors

Grenville R.S. Bingham (c. Winter 69 – Winter 74)
Carolyn Landis (Spring 75 – Winter 77)
Charles Davidson (Winter 78)
David Love (Spring 79 – Fall 80)
Alex Hoffman (Spring 81 – c. Fall 84)
Richard Rosenberg (c. Summer 85 – Oct 92)
Dianne Martin (July 93)
Bruce Jawer (Dec 93 – March 96)
Tom Jewett (June 96 – present)

Note: dates shown above are based on publication dates shown in each issue of *Computers and Society*. In several cases, specific issues were missing from the set that I was able to obtain from ACM archives and the UCI library. In other cases, dates are unclear because a single issue of the publication covered up to a one-year period. I've provided the best guesses that I can. –T.J.