Teaching Ethics in the Software Engineering Curriculum

Elizabeth Towell
Carroll College, 100 N. East Avenue, Waukesha, WI 53186
etowell@cc.edu

Abstract

This paper describes a pilot study conducted to explore the teaching of ethics in Software Engineering programs. The author reviewed existing literature and then constructed a tentative survey, which was sent to 127 educators. Questions involved ethical topics encountered, methods of delivery, and the use of codes of ethics. Results provide only an informal snapshot of trends but responses are being used to create a revised survey, which will be sent to a larger population.

1. Introduction

Webster defines ethics as "1: a branch of philosophy, dealing with what is good and bad and with moral duty and obligation, 2: the principles of moral conduct governing an individual or a group." The study of ethics includes meta-ethics (general principles), moral theory (ethical systems consisting of criteria and procedures) and practical or applied ethics (the application of ethical systems to situations) [6]. The teaching of ethics in professional programs is generally focused on practical or applied ethics.

This manuscript explores the study of ethics in Software Engineering programs. We attempt to address the why, what and how questions of current ethics education. In addition to conducting a literature review, one hundred and twenty-seven previous CSEET conference attendees in academic positions were sent a questionnaire via email regarding teaching ethics. Thirty-six responses were received (28.3%). Undergraduate programs as well as graduate programs were represented, predominantly in Computer Science and Software Engineering fields. School sizes ranged widely but the greatest number of respondents were from larger schools with over 10,000 students. Most respondents were from North America but responses were also received from Asia, Australia, and Europe. The results cannot be construed to reflect any population other than recent CSEET attendees but they may provide an informal snapshot of trends.

The paper is organized in six sections following this introduction. In section two, we distinguish what is meant by ethics for the purposes of this paper and provide basic assumptions made by the author. In section three, specific ethics topics encountered in Software Engineering curricula are addressed. Section four explores methods of delivery for teaching ethical decision-making. Section five outlines the codes of ethics of various professional computing organizations. Section six explores the relationship of a code of ethics and professional conduct. The final section presents questions and conclusions that have been drawn from the study.

Proceedings of the 16th Conference on Software Engineering Education and Training (CSEET'03)
1093-0175/03 $17.00 © 2003 IEEE

2. Basic assumptions

The general goals of teaching ethics in any venue include deterrence, inspiration and guidance, and a shared understanding of ethical norms and expectations. This paper is premised on the specific assumption that facing ethical dilemmas is an integral part of a Software Engineer's career and that academia should prepare Software Engineers for this aspect of their career. Evidence that this is a widely held belief can be found in accreditation standards for Software Engineering programs. In the US, institutions seeking accreditation in engineering related programs by the Accreditation Board for Engineering and Technology (ABET) initially complete a self-study. Applicants are asked to answer this question, "Describe how this program assures the development of an understanding of the ethical, social, and economic considerations in professional practice" [1]. The model accreditation criteria for Software Engineering undergraduate programs established by the Education Task Force of the Joint Steering Committee of the IEEE Computer Society and ACM also specifies, "engineering responsibility and practice must be stressed, which includes conveying ethical, social, legal, economic and safety issues. These concerns must be reinforced in advanced work, as must the appropriate use of Software Engineering standards" [14].

A second assumption of this paper is that the teaching of Software Engineering ethics must necessarily focus on the process involved in the production of a software product. The concept of Software Engineering ethics should be distinguished from the myriad of abuses that occur via the use of a computer. As Gotterbarn [10] indicates, if any use of a computer to commit a crime can be construed as a case involving computer ethics, "then my use of a scalpel to rob someone is a problem of medical ethics." If we start with too big of a concept it becomes everybody's concern and nobody's business. Furthermore, there is a difference between what is ethical and what is illegal or should be made illegal. It may not be illegal to subtly undermine the efforts of programmers on your team but it is clearly unethical. We can break the law for ethical reasons and we can follow the law for reasons other than ethical ones. The following section continues to refine the context for what is meant by ethics topics in the Software Engineering curriculum.

3. Ethics topics covered in the Software Engineering curriculum

A review of books, case studies, and journal articles was used to generate a set of ethics topics that are routinely taught in Computer Science and Software Engineering programs. These topics are briefly defined in **Table 1**. The definitions cannot be considered comprehensive and there is obvious overlap among the topics. The grouping together of topics may also be arguable.

The results from the survey of recent CSEET attendees are shown in **Figure 1**. The survey asked respondents to rate each ethical area as *critical*, *important* or *unimportant* with respect to coverage in their program. The percent of respondents that believed that coverage of a specific topic was *critical* is shown. (It is interesting to note that in only two cases out of the nine, did a majority of the respondents think that coverage of the topic was critical: 1) quality and testing and 2) liability and risks in health and safety critical environments.) The three topics with the largest number of *unimportant* ratings were: encryption and privacy, fairness and discrimination, and whistle-blowing.

Proceedings of the 16th Conference on Software Engineering Education and Training (CSEET'03)
1093-0175/03 $17.00 © 2003 IEEE

Table 1. Ethics topics in the Software Engineering curriculum

Confidentiality involves the control of information about an individual or an organization provided to an entrusted party.
Conflict of interest involves avoiding compromised decision-making due to offers of financial or other considerations.
Encryption and privacy involves communication of secret information and the bounds of governmental access for the public good.
Fairness and discrimination involves conditions of employment and equality of opportunity.
Intellectual property involves ownership of software, data, and ideas as well as the related legal protections such as copyrights, patents, trademarks, and oaths of confidentiality.
Liability and risks in health and safety critical environments involves legal obligations to protection of the quality of life and the environment.
Quality and testing involves the appropriate comparison of expected and observed behavior of software systems.
Unauthorized access and computer security involves the authorized use of computers and communications.
Whistle-blowing involves rights and responsibilities in terms of making revelations which call attention to threats to the public interest.

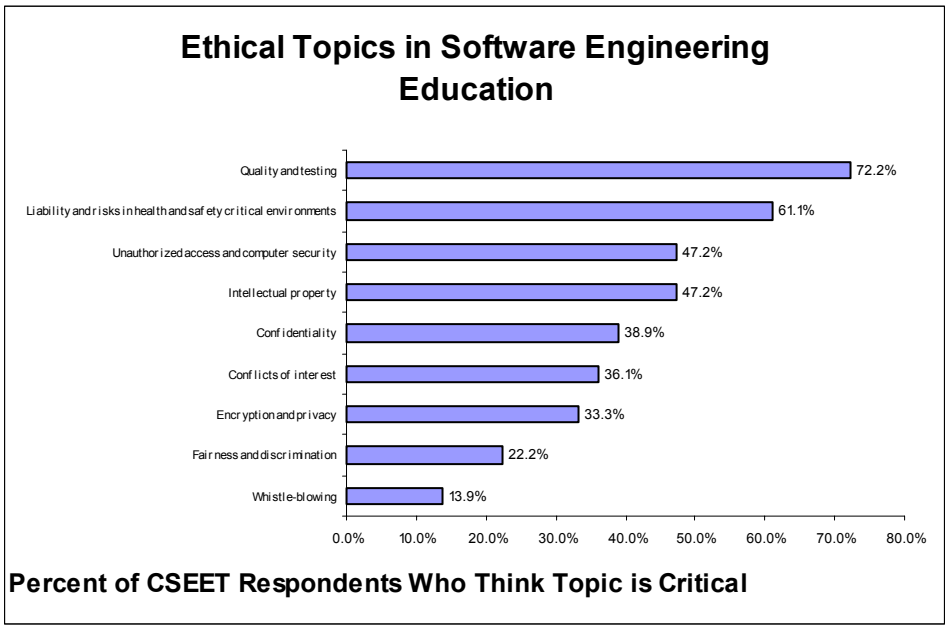

Figure 1. Ethical topics in Software Engineering education and the percent of CSEET respondents that believe the coverage is critical in a software development program

Other topics that were added by respondents included employment agreements, trade secrets, commercial integrity, professional responsibility, research ethics, measurement-related ethics and measurement dysfunction, power relationships, clear thinking about ethical responsibility,

Proceedings of the 16th Conference on Software Engineering Education and Training (CSEET'03)
1093-0175/03 $17.00 © 2003 IEEE

social impact and professional responsibility to society, internationalization, and data protection.

4. Methods for Teaching Ethics in a Professional Program

It is difficult to find definitive advice on how the teaching of ethics is best accomplished. What are the most critical topics (or are they all equally critical)? Curricular models do not address this. How are they best explored? There are limited standards in this area as well. Kevin Bowyer [7] offers this suggestion: ethics should be taught in three ways: discussion of real-world case studies, participation in active learning (exercises), and reading papers by distinguished authors.

The top five methods used from survey results are shown in **Figure 2** below. The most popular techniques were discussing personal experiences and discussing codes of ethics. Other methods employed that were mentioned included guest speakers, news articles, simulations, and risk forums. **Figure 3** gives a general indication of classroom time devoted to teaching ethics. Almost twenty-eight percent of respondents to this survey indicated that less than 1% of classroom time was devoted to teaching ethics. This would equate to less than half an hour total in a three-contact-hour class that meets for 15 weeks.

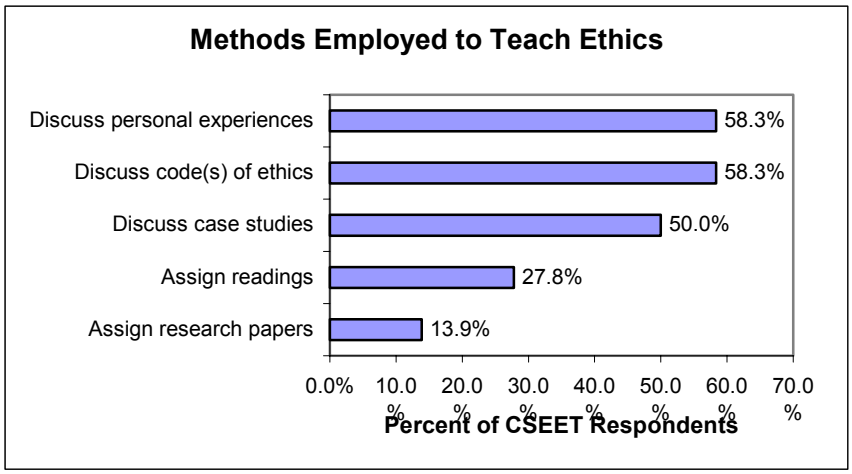

Figure 2. Methods used by CSEET attendees to teach ethics

The survey also queried how ethics instruction should be distributed throughout a program. One could argue that ethical behavior, like "proper documentation" or "logical analysis" is not a topic that can be or should be separated or isolated from the rest of the curriculum. **Figure 4** provides a snapshot of how ethics instruction is distributed throughout the curriculum in programs represented by CSEET attendees responding to the survey. Most respondents indicated that the teaching of ethics is focused in a few courses. (This might provide some explanation for the results in **Figure 3**.) Surprisingly, over 19% of the respondents indicated that the teaching of ethics in their curriculum is largely ignored.

Proceedings of the 16th Conference on Software Engineering Education and Training (CSEET'03)
1093-0175/03 $17.00 © 2003 IEEE

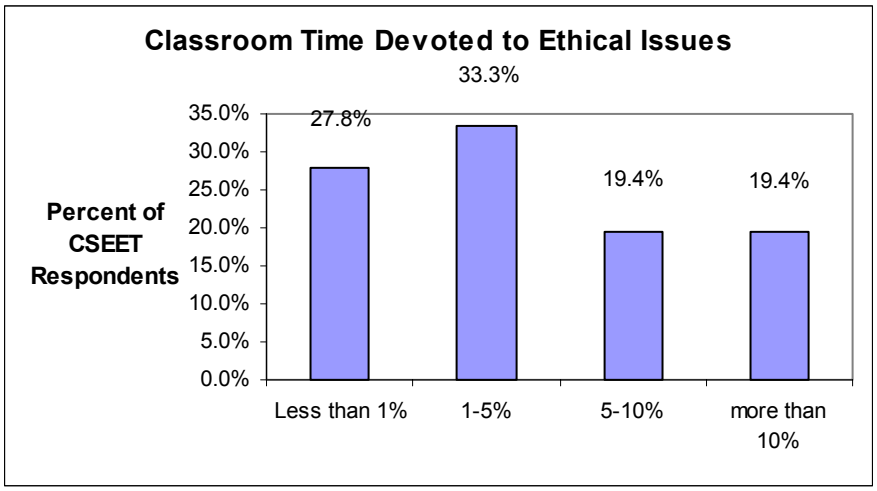

Figure 3. Classroom time devoted to ethical issues

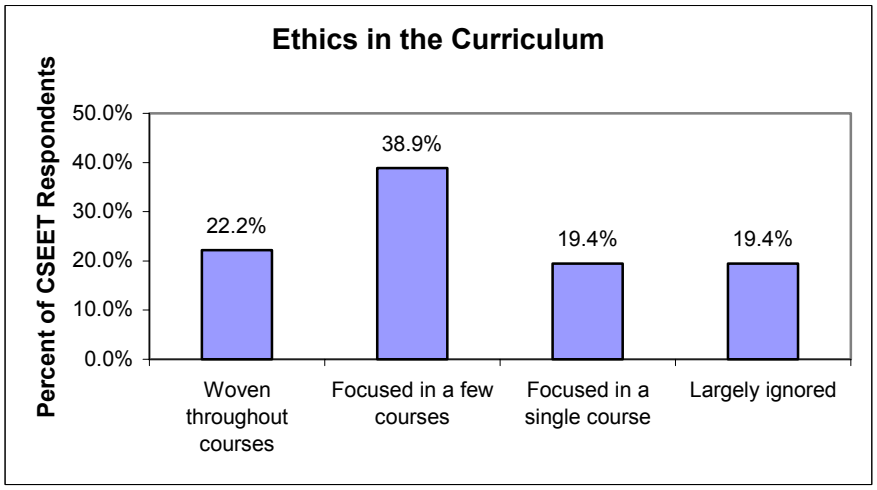

Figure 4. How ethics instruction is distributed throughout the curriculum

5. Codes of Ethics

A very popular technique for teaching ethics involves review of a published code of ethics. Nearly every professional computing organization has published a code of ethics. A brief description of seven prominent codes of ethics in the computing professions is provided in **Table 2**. A code of ethics is used to promote a variety of purposes and goals. Luegenbiehl [13] identifies 12 overlapping functions that can be served by a code of ethics: symbolize professionalism, protect group interests, specify membership etiquette, inspire good conduct, educate members, discipline members, foster external relations, enumerate principles, express ideals, put forth rules, offer guidelines, and codify rights.

CSEET attendees that responded to the survey indicated that they were most familiar with these codes: ACM, IEEE Computer Society, and the Software Engineering Code of Ethics and Professional Practice (adopted by ACM and the IEEE). Other codes named included National

Proceedings of the 16th Conference on Software Engineering Education and Training (CSEET'03)
1093-0175/03 $17.00 © 2003 IEEE

Society of Professional Engineers, Engineering Council, Institution of Electrical Engineers, Southeast Asia Regional Computer Confederation and Computer Society of South Africa.

6. The Relationship of a Code of Ethics and Professional Conduct

Software Engineering is becoming a recognized profession both from societal pressures and within the ranks [11]. ACM and IEEE are working jointly to establish criteria and norms for the professional practice of Software Engineering. Three task forces have been formed to provide a foundation for this daunting goal. They are charged with 1) defining the Software Engineering Body of Knowledge (SWEBOK), 2) defining ethical and professional standards, and 3) defining educational curricula. These three pieces are highly related. The current version of SWEBOK [17] provides many examples of the emerging discipline of Software Engineering including the following:

- Universities throughout the world now offer undergraduate degrees in Software Engineering.
- The Accreditation Board for Engineering and Technology (ABET) and the Computing Science Accreditation Board (CSAB) have reached an agreement to cooperate in the accreditation of Software Engineering programs in the US.
- SEI's Capability Maturity Model for Software and ISO-9000 standards are now used to assess organizational capability for Software Engineering.
- The Software Engineering Code of Ethics and Professional Practice has been adopted by the ACM and the IEEE Computer Society. This code is intended to be one of three pieces that together will form criteria and norms for professional practice of Software Engineering.
- The Texas Board of Professional Engineers, the Association of Professional Engineers and Geoscientists of British Columbia (APEGBC) and the Professional Engineers of Ontario (PEO) have all established requirements for the licensing of Software Engineers.

The general nature of a professional organization is to confer privileges on its members (such as licensing) and to conduct disciplinary action for misconduct (such as disenfranchisement). Does this movement towards Software Engineering as a profession and the cooperation and consolidation of professional organizations signal a movement towards licensing of the profession? This seems to be the case, at least in the United States, where the title of engineer is protected and the practice of other forms of engineering is restricted. It is clear that one purpose of the development of SWEBOK is to establish a body of knowledge that license holders will need to possess. Equally clear is that one purpose of the Software Engineering Code of Ethics is to establish guidelines for disciplinary action involving license holders. The licensing of Software Engineers is a contentious issue. For example, in the survey taken for this study, 44% thought Software Engineering should be a licensed profession, 39% thought it should not be a licensed profession, and the remaining 17% were ambiguous for various reasons or undecided.

Proceedings of the 16th Conference on Software Engineering Education and Training (CSEET'03)
1093-0175/03 $17.00 © 2003 IEEE

TABLE 2. Codes of ethics for various professional computing societies (NOTE: this table has been abbreviated due to the proceedings page limitation)

Association for Computing Machinery (ACM) [2]
Canadian Information Processing Society (CIPS) [8]
New Zealand Computer Society (NZCS) [15]
Software Engineering Code of Ethics and Professional Practice 5.2 [16]
Australian Computer Society (ACS) [3]
British Computer Society (BCS) [4,5]
Institute of Electrical and Electronics Engineers (IEEE) [12]

7. Conclusions and Future Research

We teach ethics in a professional academic program for a variety of reasons including developing a shared understanding of ethical norms and expectations. Accrediting bodies in the Software Engineering field reinforce this notion in specific language calling for ethical consideration in the curriculum. Even if we are not seeking accreditation, we teach ethics because we realize that to do something well a student must not only develop good intellectual habits, he or she must also nurture good ethical habits as well.

There are a variety of topics and a variety of modes employed to cover ethical issues in Software Engineering programs. The data in this study suggests that quality and reliability issues are considered to be of greatest importance. These issues are often explored through discussion of personal experience, case studies, or review of codes of ethics. Three codes that are widely recognized are those of the ACM, the IEEE, and Software Engineering Code of Ethics adopted more recently by both ACM and IEEE. These codes seem aimed at the development of Software Engineering as a profession and the establishment of guidelines for disciplinary action involving license holders.

This paper was written for presentation and discussion at CSEET and is exploratory in nature. The population that was surveyed (CSEET attendees) involves an obvious bias and the number of responses was somewhat limited. For these reasons, more significant analysis of the data at this time is not warranted. Still, much has been learned that will be used as the manuscript is expanded. Future efforts will include:

1) Collection of more data from academicians to study
    a) the relationship between time spent teaching ethics and factors such as size of school and type of program;
    b) the relationship between time spent teaching ethics and factors such as support for the licensing of Software Engineers and accreditation status
2) Collection of data from practitioners to determine ethical areas considered most critical for inclusion
3) Development of a model for testing the effectiveness of different styles of teaching ethical conduct in Software Engineering program

Proceedings of the 16th Conference on Software Engineering Education and Training (CSEET'03)
1093-0175/03 $17.00 © 2003 IEEE

References

[1] ABET, Applied Science Accreditation, 2002. "Volume I Self-Study Questionnaire 2002-3 Edition," Available http://www.abet.org/info_prgs_rac.html downloaded July 3, 2002.

[2] ACM, 2002. "ACM Code of Ethics and Professional Conduct," Available http://www.acm.org/constitution/code.html downloaded July 3, 2002.

[3] ACS, 2002. "Australian Computer Society Code of Ethics," Available http://www.acs.org.au/static/national/pospaper/acs131.htm downloaded July 3, 2002.

[4] BCS, 2002a. "BCS Code of Practice," Available http://www1.bcs.org.uk/homepages/533/ downloaded July 3, 2002.

[5] BCS, 2002b. "BCS Code of Conduct," Available http://www1.bcs.org.uk/homepages/518/ downloaded, July 3, 2002.

[6] Bott, F., Coleman, A., Eaton, J., Rowland, A. 2001. "Professional Issues in Software Engineering," 3rd edition, Taylor and Francis, p. 10.

[7] Bowyer, K. 2001. "Ethics and Computing, Living Responsibly in a Computerized World," IEEE Press.

[8] CIPS, 2002. "Code of Ethics & Standards of Conduct," Available http://www.cips.ca/about/ethics/ downloaded July 3, 2002.

[9] Gotterbarn, D., Miller, K., Rogerson, S. 1999. "Computer Society and ACM Approve Software Engineering Code of Ethics," Computer, October, pp. 84-88.

[10] Gotterbarn, D. 2002a. "Computer Ethics: Responsibility Regained [Online] Available http://www-cs.etsu-tn.edu/gotterbarn/artpp1.htm downloaded June 19, 2002

[11] Gotterbarn, D. 2002b. "An Evolution of Computing's Codes of Ethics and Professional Conduct," Available http://www-cs.etsu.edu/gotterbarn/artge1.htm downloaded June 19, 2002.

[12] IEEE, 2002. "IEEE Code of Ethics," Available http://www.shop.ieee.org/organizations/eab/apc/ceaa/operations/code_of_ethics.htm downloaded July 3, 2002.

[13] Luegenbiehl, H.C. "Computer Professionals: Moral Autonomy and a Code of Ethics," Journal of Systems and Software, Volume 17, 1992, pp. 61-68.

[14] Model Accreditation Criteria, 2002. "Model Accreditation Criteria For Software Engineering Undergraduate Programs," Available from the Education Task Force of the Joint Steering Committee of the IEEE Computer Society and ACM at http://www.computer.org/tab/seprof/Accreditation.html, downloaded July 3, 2002.

[15] NZCS, 2002. "The New Zealand Computer Society, Inc. Code of Ethics & Professional Conduct," Available http://www.nzcs.org.nz/membership/membership.htm downloaded July 3, 2002.

[16] Software Engineering Code, 2002. "Software Engineering Code of Ethics and Professional Practice, Version 5.2" Available http://www.computer.org/tab/seprof/code.htm downloaded July 3, 2002.

[17] SWEBOK, 2001. "Guide to the Software Engineering Body of Knowledge, Trial Version 1.00," Available http://www.swebok.org/ downloaded July 3, 2002.