Workshop on Teaching Ethics in Software Engineering Programmes

J. Barrie Thompson[1] and Elizabeth Towell[2],
[1] *School of Computing and Technology, University of Sunderland*
St. Peter's Way, Sunderland, UK
barrie.thompson@sunderland.ac.uk
[2] *Carroll College*
100 N. East Avenue, Waukesha, WI
etowell@cc.edu

1. Context

To operate effectively in today's volatile software industry, graduates from Software Engineering programmes need to be equipped with knowledge of both real world situations and best Software Engineering practice. This knowledge should include relevant aspects of ethical, professional, and legal issues and there should be recognition that facing ethical dilemmas is highly likely to be an integral part of the graduate's eventual career. Also, professional bodies such as the ACM and IEEE-CS in the United States and the British Computer Society (BCS) in the United Kingdom have always placed a great emphasis on ethical issues. The BCS in particular pays special attention to the way in which ethical topics are addressed when it is accrediting university programmes [2]. There is therefore clear need for ethical issues to be addressed within Software Engineering programmes in an effective and meaningful manner. The importance; of such ethical and professional issues is clearly recognised within the guiding principles for the Software Engineering volume of the Computing Curricula, currently being developed under the auspices of the IEEE Computer Society (IEEE-CS) and the Association for Computer Machinery (ACM) viz:

> *"The education of all Software Engineering students must include student experiences with the professional practice of Software Engineering. The professional practice of Software Engineering encompasses a wide range of issues and activities including problem solving, management, ethical and legal concerns…"* [1]

Another indication that the Software Engineering community recognises the importance of addressing ethical issues within academic programmes occurred at the 2003 CSEE&T conference. The paper by Towell entitled "Teaching Ethics in the Software Engineering Curriculum" [3] resulted in a very lively discussion that continued all the way though the next timetabled slot when the scheduled speaker failed to appear. However, there are difficulties in addressing ethical issues within programmes. There is a lack of consensus regarding whether the topics should be covered in tailor-made modules or whether they should permeate the curriculum. Also there is little information on differing teaching approaches and their effectiveness. This is unlike the situation for technical subjects within the curriculum where there is typically not only a body of knowledge on the subject itself but also on the relevant pedagogy.

2. Workshop Objectives

In this workshop the following objectives will be examined:

1. To consider different approaches that can support the teaching of ethics within Software Engineering and related programmes such as [3, 4]:

- Discussion of an instructor's personal experiences.
- Reviewing various codes of ethics such as the Software Engineering Code of Ethics and Professional Practice produced under the auspices of IEEE-CS and the ACM.

Proceedings of the 17th Conference on Software Engineering Education and Training (CSEET'04)
1093-0175/04 $20.00 © 2004 **IEEE**

- Using case studies such as the Killer Robot to highlight particular ethical considerations.
- Using role-play to engage students in the exploration of ethical situations
- Using games such as Lockheed Martin's "The Ethics Challenge"
- Employing Web-Based Learning Systems such as Walter Maner's Interactive Computer Ethics Explorer.

2. To consider the pros and cons of different approaches and to share participants' experiences especially with regard to possible problematic areas that are best avoided.

3. To identify particular approaches that can be successful within different teaching/learning environments (e.g. traditional lecture-based classes, distance learning at remote centres, and individual web-based learning).

4. To produce recommendations and identify areas for further work that could result in a better understanding of ethics issues by students and a more effective use of what are often limited resources for teaching.

3. Prior Workshop Activities

Position papers/statements relevant to the objectives listed above will be solicited via the conference Web site, on-line publications, and selective mailings. If possible these will be circulated to the delegates prior to the workshop. The Appendix to this paper details two proposed approaches for teaching computer law/ethics that have been supplied by Cem Kaner in response to the call for contributions.

4. Operation of the Workshop

1. Recap the main themes of participants' position papers, and map these against the above objectives.
2. Break out into activity groups. Each group to examine their topic and (i) identify the issues that are important (ii) prioritise these with explicit rationale (iii) provide evidence to justify priorities.
3. Feedback to the full group. Nominated speakers to feedback themes emerging from their group.
4. Comparative analysis of themes from each group. In "mixed" groups analyse the results from the feedback in order to identify the most effective approaches etc.
5. Plenary session to produce recommendations and identify areas for further work.

5. Final Deliverable

A final outcome of the workshop will be the production of a report detailing the major recommendations relevant to the objectives. This will be circulated to participants and a paper based on it will be submitted for journal publication.

References

[1] Computing Curricula Software Engineering (CCSE), - the Software Engineering volume of CC2001, the project's web site is: http://sites.computer.org/ccse/

[2] Thompson J. B. and Edwards H. M. (2001), "Software Engineering in the UK 2001" Forum for Advancing Software Engineering Education (FASE), Vol. 11, No.119, November 15th 2001, pp 1-18, FASE archive is available at: http://www.cs.ttu.edu/fase/archive.htm

[3] Towell, E. (2003). "Teaching Ethics in the Software Engineering Curriculum", Proceedings of the Sixteenth Conference on Software Engineering Education & Training, 20-22 March, Madrid, Spain, .

[4] Towell, E.and Thompson J. B. (2004). "A Further Exploration of Teaching Ethics in the Software Engineering Curriculum", Proceedings of the Seventeenth Conference on Software Engineering Education & Training, March 1-3, Norfolk, Virginia, USA.

Appendix: Two Proposals for Teaching Computer Law/ Ethics

The following proposed approaches for Teaching Computer Law/ Ethics have been supplied, for consideration at the workshop, by Cem Kaner, Department of Computer Sciences, Florida Institute of Technology, 150 West University Blvd. Melbourne, FL 32901.

Proposal 1: Evaluation in the course

At or soon after the start of the semester, I hand out a study guide that includes three types of questions: (a) Definitions, (b) Short Answer, and (c) Long Answer. I supplement the guide with new questions that reflect the direction of the class as it has evolved during the term. The questions cover all of the substantial issues in the course.

The midterm test(s) and final exam questions are all drawn from the study guide questions. Students work on these questions in small teams -I host (but do not actively participate in) some of the class meetings at a local cafe, to encourage students to work together.

The test itself can have a few more questions than a surprise-question test, because I don't have to allow much time for students to read and interpret the questions. I can also ask fairly complex questions because I don't have to be concerned about the time it will take a student whose first language is not English, to puzzle out the meaning of the question. I can also expect (require in grading) a better organized answer that is more responsive to the call of the question than I can reasonably expect with surprise-question tests.

Students in the course spend a lot of time discussing the course issues. Preparation and writing clarity of several students also improves substantially; they can focus on some fairly sophisticated issues.

I've used this approach when teaching courses in Computer Law/Ethics, Software Metrics and Models, Software Testing, and Sensation / Perception (Psychology). It works pretty well in all of these courses but I think it is particularly well suited to a Law/Ethics course. I've only tried this in face-to-face instruction. I think it might work well in distance learning courses if students in the program were used to working in chat sessions.

(At workshop, sample study guide and sample grading standards can be made available).

Proposal 2: Driving the Course With Employment Contracts

This spring, I plan to organize my course in Computer Law & Ethics around a set of employment contracts. The contracts cover full-time employment and consulting. They're directly relevant to the students, because the students will face contracts like these after they graduate--as they gain experience, more of the terms will become negotiable.

As we walk through the contracts, we'll ask, clause by clause, what the clause means, what restrictions it imposes on the programmer who signs it, and what ethical considerations are involved in requiring, enforcing, and breaking the clause. An unenforceable noncompetition clause raises different issues from a copyright assignment, for example.

We'll have to handle some of the key issues in some other way, primarily the issues involving responsibility to the public (consumer protection, liability for defects, privacy concerns) but I think this will be a useful way to introduce the student to intellectual property, service contracts, malpractice, changes in project schedule or scope, dealing with failing projects, racial/gender issues at work, and the guidance and relevance of IEEE and ACM codes of ethics to these issues.

(At the workshop, the strategy can be laid out, some contracts made available, and progress outlined)