Towards Understanding Software Engineering Ethical Frameworks Supporting Professional Bodies Ethical Principles

Peter Oriogun, Faiza Babakano Chike Chiejina, Fatima Chiroma, Malachy Khanoba

American University of Nigeria
School of Information Technology and Communications
Lamido Zubairu Way, Yola Township Bye-Pass
PMB 2250, Yola
Adamawa State, Nigeria

Corresponding Author: peter.oriogun@aun.edu.ng

Abstract

As the basic software engineering structures are built on business backgrounds, the roots of software engineering ethics stern from the normative business framework used by managers in large organizations. But these days, software engineering ethics is being taught as a course within most higher education institutions; but the question is at what stage or level of experience should a training software engineer be taught about ethics? Even after the software engineer begins work in the industry, who is he/she responsible to? There are various software engineering associations and bodies in various parts of the world. We might be tempted to think there should be one body responsible for regulating software engineering professionals in the profession today.

Introduction

Software Engineering ethics is a course that requires some background and prior knowledge and experience on the part of students to actually understand the course content. We try to see the impact of framework for software engineering ethics currently in the literature and evaluating if they are sufficient. The paper also seeks to find if there should be some centralized body to regulate software engineering professionals as a body, despite the existence of many such bodies like ACM/IEEE, Australian Computer society, BCS (British Computing Society) and e.t.c

What are these frameworks currently in the literature?

Smith and Hasnas (1999) suggest that there exist a normative business framework for general business ethics currently in the literature. The framework consists of three theories, namely, the stockholder theory, the stakeholder theory and the social contract theory. The stockholder theory specifies the obligations of between the shareholders and the top management of an organization. The stakeholder theory specifies the obligations between the top management clients and the employees, whilst the social contract stipulates the obligations of the management to the profession and the society (Oz, 1992; Smith and Hasnas, 1999).

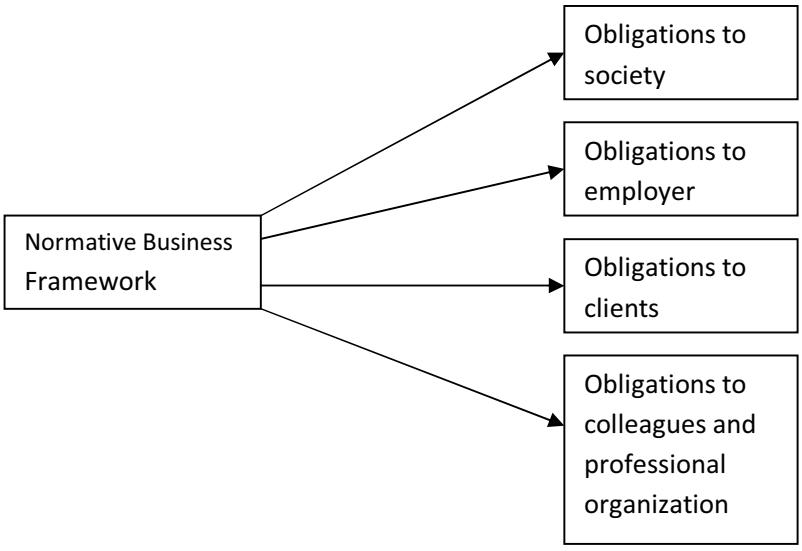

Figure 1: Normative Business Framework

The normative business framework above serves as a way of evaluating and measuring the effectiveness of ethical guidelines within the context of business environments. A number of professional bodies such as the ACM (Association of Computing Machinery), the BCS (British Computer Society), DPMA (Data Processing Management Association), ICCP (Institute for Certification of Computer Professionals), and the CIPS (Canadian Information Processing Society) have codes of conduct that adheres to the above normative business framework. In context of the software engineering profession, the obligations elucidated above is still current and these professional bodies have used the concept of the normative business framework to evaluate, formulate and shape their individual ethical guidelines/principles. According to Moor (1985), the following are the ethical issues that should concern software engineers:

- Privacy
- Logical Malleability
- Speed
- Storage of huge amounts of data
- Copying
- Openness and availability
- Globalization
- Safety
- Power mediation

Bialaszewski and Bialaszewski (2005) argue that we need some experience in software engineering in order to appreciate, comprehend and fully understand the case studies that are used in teaching software engineering ethics courses within higher education institutions. They believe that most Professors are suppose to cover different aspects of software engineering ethics within different course offered on a typical software engineering programmes, however, closer inspection of the literature suggest that this is in fact far from the norm. This inadequacy has lead to the creation of software engineering ethics as a course on its own, offered at student's junior or senior years. At Indiana State University, a course entitled "Ethics and Information Systems" was offered as an elective in the Management Information Systems major. The course covered issues such as free downloading of music, ergonomics, copyright, privacy etc. with the aid of case studies and textbooks during the delivery of the course.

Dodig-Crnkovic and Crnkovic (2005) argues that without guiding principles, case studies in software are difficult to evaluate and analyze; and without case studies, code of ethics are

perplexing. Therefore, he believes that the best way to teach ethics is to apply them in a variety of situations and analyze the results from which the critical evaluation of the codes and relevant cases will result to a well-reasoned learning.

Dodig-Crnkovic and Crnkovic (2005) suggest that software engineering ethics courses is meant to increase the ability of concerned engineers, managers and citizens, to first recognize and then responsibly confront moral issues raised by technological activity. Arguing that this will help foster what they called 'moral autonomy'. What they meant by moral autonomy is basically the skill and habit of engineers to think rationally about ethical issues in their professional activities. Furthermore, they ascertain that the following reasons are the main purpose of software engineering ethics:

- Deal with the true nature of computing as a service to other human beings
- Convey a sense of professional responsibility not covered in other courses
- To sensitize students to Computer Ethics issues
- To provide tools and methods for analyzing cases
- To provide practice in applying the tools and methods to actual or realistic cases
- To develop in the student good judgment and helpful intuitions – ethical autonomy

Dodig-Crnkovic and Crnkovic (2005) concur with Bialaszewski and Bialaszewski (2005), because according to the literature they also believe that the above topics are not currently addressed in software engineering education.

What is the prior study for a software engineering course within higher education institution?

Because these topics are not addressed inside the software engineering education, it is suitable that before a software engineering ethics course is taught, the student must have experience in projects, industry regulations, and must be familiar with the software engineering profession as a whole to fully appreciate a software engineering ethics course.

According to Gotterbarn (1995) the following are the three main objectives for having a course on ethics within the curriculum:

- The ability to identify correctly the potential for an ethical problem in a particular context, and the ability to identify what moral rules are being compromised.
- The ability to identify the cause of these issues, determine several alternate forms of action consistent with morality in that context and for each of these possible actions to determine expected outcomes and reasons for taking or not taking those actions.
- The ability to select a workable solution and work through the situation either technically or morally.

Thus the above objectives justifies the need for students having some practical experience of building software before being taught ethics in class using case studies.

The paper, *Ethics and Education: Curriculum Issues* confirms that there is an examination of codes existing at the Midwestern University to cover ethical considerations related to computing. This too I believe can be another way of presenting or forcing learning of software engineering ethics in higher institutions.

The establishment of "Ethics Centers" such as the Centre of Computing and Social Responsibility in some universities which address the social and ethical impacts of information and communication technologies through research, consultancy, and Education were encouraged in the paper. He says like in many other areas like general engineering, risk management, nursing etcetera, courses should be designed for the pure purpose of integrating ethical considerations into standard course content.

Who should regulate software engineering professionals as a body?

Effy Oz made a very bold attempt at answering the question on who should regulate software engineering professionals as a body in his paper, *Ethical Standards for Information Systems Professionals: a Case for a Unified Code*. He says it is better to unify the professional bodies and their individual code of ethics so that all the obligations concerning software engineering ethics (obligations to client, employer, fellow employees, profession and the society) can be fully considered. He believes this will achieve the five objectives of inspiration, sensitivity, discipline, advice and awareness that professional ethics are supposed to achieve.

A huge concern about these various business theories and professional organizations is their lack of priority among the list of obligations. While Effy Oz talks about a possible unification of these ethical codes from the various professional bodies, we suggest that a universal IT code of ethics that covers how situational ethical dilemmas should be tackled be made. This universal IT code of ethics should prioritize the stakeholders and remove the ambiguities that currently exist in these IT professional codes of ethics. There also been talk of ethical considerations in software engineering having project specific code of ethics; that also wouldn't be so wide off the mark.

Conclusion

We have learnt that the major frameworks for software engineering ethics currently in literature come from business. The normative framework which includes the stakeholder, stockholder and the social contract theories carefully translates to what we refer to as the software engineering code of ethics. Also, we have gathered that for software engineering ethics courses to be successfully taken within higher education institution, the student needs to be a little bit experienced in the field and be able to relate to the case studies which are highly recommended for such studies. Case studies should be used side by side the professional ethics code for teaching software engineering ethics course.

Conclusively, with all these frameworks and software engineering ethics course out to guide the software engineer, I still propose that a universal software engineering body that will draw positives from all the leading software engineering bodies and associations be formed. This will bring about a more unified and unambiguous software engineering code of ethics.

References

"Computer Society and ACM approved Software Engineering Code of Ethics" http://www.computer.org/portal/cms_docs_computer/computer/content/code-of-ethics.pdf. Viewed 1st April, 2009.

Dodig-Crnkovic, Gordana and Ivica Crnkovic (2005). "Professional Ethics in Software Engineering Curricula" http://www.it.uu.se/research/group/cetuss/Events/2005-12/program/Gordana.pdf. Viewed 1st April, 2009.

Gotterbarn, Donald (1995). "Three Levels of Professional Software Engineering: the moral responsibility of software developers". *The Journal of Information Ethics,* Pp 54-64. Spring 1995.

IEEE Code of Ethics, February, 2006. "IEEE-CS/ACM Software Engineering Code of Ethics and Professional Practice" http://hum.iit.edu:8080/aire/sea/1/Final.htm. Viewed 1st April, 2009.

Oz, Effy. "Ethical Standards for Information Systems Professionals: A Case for a Unified Code" *MIS Quarterly Vol. 16, No. 4*. Management Information Systems Research Center, University of Minnesota. March 1992.

Smith, Jeff H. and John Hasnas (1999). "Ethics and Information Systems: The Corporate Domain" *MIS Quarterly Vol. 23, No. 1*. Management Information Systems Research Center, University of Minnesota. March 1999.

"Software Engineering Code of Ethics and Professional Practice" *http://www.computer.org/portal/site/ieeecs/menuitem.c5efb9b8ade9096b8a9ca0108bcd45f3/index.jsp?&pName=ieeecs_level1&path=ieeecs/content&file=ethics.xml&xsl=generic.xsl&;jsessionid=J6DYS3TWX1j9q6MDqfvd1dTSnhcg0wgntW6ThJqq9rR1YRvgNsQ6!289239620*. Viewed 1st April, 2009.