Ethics in Software Engineering

Pankaj Kamthan
Concordia University, Canada

INTRODUCTION

As software becomes pervasive in our daily lives, its values from a purely human perspective are brought to light. Ethical conduct is one such human value.

There are various reasons for discussing the issue of ethics within a software engineering context. By participating in a software development process, software engineers can influence the final product, namely the software itself, in different ways including those that may be contrary to public interest. In other words, they could engage in an unethical behavior, inadvertently or deliberately. This could lead to personal harm, and potentially result in loss of confidence in software and loss of trust in organizations that own them. This can adversely affect the acceptance of software as a useful product, question the credibility of software engineering as a profession, lead to legal implications, and impact the bottom line of the software industry at-large.

This article is organized as follows. We first outline the background necessary for later discussion. This is followed by a proposal for a quality-based framework for addressing ethics, and software quality treatment of a software engineering code of ethics. Next, avenues and directions for future research are outlined, and finally, concluding remarks are given.

BACKGROUND

By viewing software engineering as a profession, we define ethics as a code of professional standards, containing aspects of fairness and duty to the profession and the general public.

Since a software can either be a benefit or a hazard to its potential users, the issue of ethics in its engineering arises. Software failures (Sipior & Ward, 1998) that have led to loss of human life, rendered computer systems unusable, led to financial collapse, or caused major inconveniences are grim reminders of that.

In this article, we discuss the issue of ethics from the viewpoint of software product quality considerations in practice. There is an apparent symbiosis between ethics and quality. For example, the causes of the aforementioned failures were attributed to violations of one or more quality attributes such as reliability, safety, and so forth, and/or to lack of proper validation/verification of these.

Indeed, in the Software Engineering Body of Knowledge (SWEBOK) (Abran, Moore, Bourque, & Dupuis, 2001), ethics has been placed within the software quality "knowledge area." The issue of information technology in general, and the role of quality in software development in particular, have been addressed in (Reynolds, 2003; Tavani, 2004). Moreover, software quality is viewed as an ethical issue from a philosophical perspective (Peslak, 2004). However, these efforts are limited by one or more of the following issues: quality and ethics are often viewed as a tautology, treatment of software quality is at a very high level and often as a single entity, and there is lack of specific guidance for improvement of software quality within the domain of software ethics.

One way to enforce ethical standards in a software project is by explicitly documenting the ethical expectations from stakeholders such as via a *code of ethics*. The Software Engineering Code of Ethics and Professional Practice (SECEPP) is a recommendation of the ACM/IEEE-CS Joint Task Force on Software Engineering Ethics and Professional Practices. SECEPP puts forth eight categories of principles decomposed further into clauses that software engineers should adhere to in teaching and practicing software engineering. However, these principles and associated clauses suffer from several issues (expounded in the next section): lack of separation (of concerns), recency, precision, completeness, reachability (to certain audience), and specificity, which makes their realization difficult. The relevance of SECEPP for practical purposes has been questioned (Qureshi, 2001), however the view is largely managerial rather than oriented towards the software product.

ETHICS IN SOFTWARE ENGINEERING AND SOFTWARE PRODUCT QUALITY

For the purpose of this article, our understanding of the discussion on ethics in software engineering is based on the following interrelated hypothesis:

Hypothesis 1. Ethical behavior is dynamic, rather than static. Specifically, by appropriate means (such as code of ethics), ethical actions of software engineers could be regulated and with education even be instilled.

Hypothesis 2. Ethics is a "meta-concern" (Qureshi, 2001) leading us to adoption of steps for software quality assurance and evaluation. Specifically, ethics and software quality are related by direct proportionality, and so overall improvement in the quality of a software product leads to an improvement in ethical considerations related to that product.

A Theoretical Framework for Addressing Ethics from a Software Product Quality Perspective

In order to address the practicality of introducing the ethical dimension in software engineering, we first need a theoretical foundation. To do that, we separate the concerns involved as follows:

1. View ethics as a qualitative aspect and attempt to address it via quantitative means so as to minimize the potential for heuristics and to make the evaluation repeatable.
2. Select a theoretical basis for communication of information, and place ethics within its setting.
3. Address software product quality in a systematic and practical manner by means of adopting a quality model. In particular, select the quality model that separates internal and external quality attributes.

Using this as a basis, we propose a framework for ethics from the perspective of software product quality (see Table 1).

We now describe each of the components of the framework in detail.

Semiotic Levels

The first column of Table 1 states the semiotic levels. Semiotics (Stamper, 1992) is concerned with the use of symbols to convey knowledge. From a semiotics perspective, a representation can be viewed on six interrelated levels: physical, empirical, syntactic, semantic, pragmatic, and social, each depending on the previous one in that order.

The physical level is concerned with the physical representation of signs in hardware and is not of direct

Table 1. A framework for ethics in a semiotic approach to software product quality

Ethical Concern	Software Product		
Semiotic Level	**Levels of Quality Attribute**	**Example(s) of Quality Attributes**	**Decision Support**
Social	External: Tier 1	Credibility, Trust	Feasibility
	External: Tier 2	Legality, Safety	
	External: Tier 3	Privacy, Security	
Pragmatic	External: Tier 1	Accessibility, Maintainability, Usability	
	External: Tier 2	Interoperability, Portability, Reliability	
Semantic	Internal	Completeness, Validity	
Syntactic	Internal	Correctness	
Empirical	Internal	Characters, Character Set	
Physical	Internal	Hardware Characteristics	