Code Reuse: a response Response to Parnas in SEN

Don Gotterbarn, Keith Miller, Simon Rogerson

.

We were a little surprised by the response of a highly respected software engineer, David Parnas, to the Software Engineering Code of Ethics and Professional Practice. In his comments he raises some very important issues for the professionalization of software engineering. We could not agree with him more that there is a need to define the core skills of software engineers. But we are concerned that his interest in this has led him to underestimate the role of a Code of Ethics in a profession. His letter to the editor contains two sets of issues; criticisms of the Code per se and the claim that any code is irrelevant- "a distraction"- to the real issues of software engineering.

THE CODE

He criticizes the Code as both vague and unnecessary. To include a detailed level of software engineering standards, such as requiring mutation testing for all mission critical software, in a code would be a mistake. Specific software engineering standards which have been proposed are controversial and in constant flux.

The Code, as written does encourage us to exploit standards --but it cannot name them. When the profession settles these issues, then we can incorporate them into a code. Codes have different purposes, among them guidance, inspiration and education. The code can help lead us to serious discussions about the details of standards; the code cannot prematurely set those standards.

He justifies the "vagueness" criticism in by various uses of the term"motherhood"

Motherhood-1. He claims that the Code contains statements that "no one would disagree with". Having led the development and final approval of the Code, we have a less optimistic view of the intuitively obvious nature ethical standards of software engineering. For most of the world software engineering is, at best, an emerging profession with conflicting views about the professional standards. The Code's review process, described in the original article (SEN January 1999), gives clear empirical evidence of this diversity. In 149 pages of comments developed during the IEEE-CS formal review, the following conflicting comments were made by practicing "professional" software developers: add the clause "avoid criticism of software engineers", remove 1.08 ...volunteer professional skills to good causes',... make 1.08 mandatory, remove references to maintenance, include all forms of maintenance, "...not all jobs require the highest professional standards", a plethora of opinions on the ethics of using bootlegged software, delete "consistent with the public interest", "...there is no specific reference to Y2K in the Code", "...remove the clause on testing. Testing is the clients responsibility." Comments like these are empirical evidence against his optimistic assertion that "no one can disagree with" any line of the code.

He claims that

> "... statements were made universally acceptable by the use of words like "adequate"... I am

told that I must do adequate testing but never told how much testing is adequate or how to tell if I have done enough testing. How would I know whether or not I am violating the code?" The answer is that if one does not have professional judgment sufficient to answer this question then he/she is violating the Code just by working on the project. Parnas' ad hominem is not justified. He knows that there are different techniques used for different systems and the word "adequate" is used to accurately reflect the breadth of software engineering. Reading the whole Code will provide a variety of ways to make the adequacy judgement, including talking with other competent software engineers who are familiar with the development domain. These methods will work today and they will work when new testing standards have been developed.

Motherhood-2. Even if we were to agree that the Code's statements are mostly descriptions of important generally accepted ethical principles learned from our parents, it does not indicate that these statements should not be removed from the Code. Being reminded of these "ethical principles" is considered important to many religions and nations. We have applauded software engineers who have publicly asserted such "motherhood" principles as "One's obligations as a professional precede other obligations. One must not enter into contracts which conflict with one's professional obligations." (David Parnas, " Professional Responsibility to Blow the Whistle on SDI.") This public commitment to an ethical principle was an important declaration, even though someone may have called it "motherhood". The removal of that statement would have significantly weakened the ethical significance of his letter.

These principles need repeating to inspire us to stand-up for our ethical responsibilities, to remind us of our ethical obligations, to help educate students and society about what can and should be expected of software engineers.

Motherhood-3. We have been challenged to show anything new or controversial in the Code. Unlike most engineering codes, the Preamble of the Code provides a method to make decisions when two of its imperatives such as "Honor Contracts" and "Work for the Enhancement of Human Welfare" are in conflict. The Code advocates pro-bono work. The Code requires whistle blowing and 6.12, 6.13 suggests a procedure to follow in whistle blowing. This Code advocates a stakeholder analysis which goes beyond the limited concerns of safety schedule and budget. The National Society of Professional Engineers' Code requires that an engineer report to a client, " if in the engineer's judgment, a project is likely to fail", whereas this Code requires that a software engineer report "...if in their judgment a project is likely to fail, to prove too expensive, to violate intellectual property law, or otherwise be problematic.". This code addresses "...maintenance with the same professionalism as new development." This is a partial list of some of the new and controversial elements in the Software Engineering Code.

It is difficult to respond to his charge that the Code is unnecessary. He first claims that the Code is vague and adds nothing to the vague platitudes of other codes, but he next asserts that these other codes are all we need; and we should REUSE these other vague -- platitude- ridden--

codes?

This code does not fit “under the vague platitudes of other codes”; it includes details that other codes don't, some specific to software engineering, others recast in the software engineering realm.

In the 1980's Parnas needed to educate and argue that testing needed to be done on a safety critical system, to argue that is was ethical to blow the whistle on this development. The motherhood of this Code now requires whistle blowing for a defective system. The weight of all the software engineers who reviewed the Code and the support of the professional societies that approved it make it easier to defend these kinds of actions as consistent with the moral standard of software engineers. A Code that has been reviewed , voted on, an adopted by professional societies provides ethical direction for new software engineers and gives support to the ethical decision of other software engineers.

A DISTRACTION

We are more concerned with the second criticism-- the Code of Ethics is just a distraction from the real issues of software engineering-- than with his claims about the details of the Code. He asserts that “ethical codes are just a distraction from the real problem”... “identifying what engineers call the “core body of knowledge” the knowledge that should be posed by all software engineers.”

This kind of emphasis is a terrible mistake. We heartily agree with the need for identifying a core body of knowledge. The code does NOT distract from this search for identity, it encourages us to search harder. Parnas thinks of the code and the search as competing for attention. We think of the code and the search as two complementary activities by people who care about the profession.

The IEEE-CS/ACM have been working on addressing the body of knowledge question. They have recently formed the Software Engineering Coordinating Committee to continue the work which supported the development of the Software Engineering Code of Ethics. Their plan calls for the Code of Ethics to inform both the body of knowledge and the curriculum design.

They understand that ethics and values are relevant to what we will define as our core body of knowledge. Without a broad concern for stakeholders in the software product and software development it would be reasonable to train and expect software developers to develop the shoddiest most harmful products possible. If the Code is not related to the body of knowledge then a “good” software engineer is someone who develops junk effectively.

Parnas realized this in his “Professional Responsibility to Blow the Whistle on SDI”. When

arguing for the technical problems with SDI with his fellow scientist, he “found none that disagreed with my technical conclusion.” Having the technical skill was not sufficient. Parnas brought ethical judgement to the technical task; ethical judgement needed to correctly respond to the technical task.

In addition to a body of knowledge, one element that separates professionals (in the philosophical sense) is that they pledge themselves to certain moral responsibilities and a higher order of care. The code leads to that kind of pledge. Is that sufficient to establish the profession? Of course not. Is it a necessary step? We think so. The body of knowledge is incomplete without values.

The code is not meaningless and it is not worthless. It is not perfect nor is it a panacea. But the fact that it has given David Parnas an opportunity to press for better standards in software engineering illustrates that the code can lead to more awareness of where we must go to improve. Don't fight the code, use the code in the good fight for improved professional standards in software engineering. The code is not a distraction, but trivializing the code is a distraction.