
From: Felipe Cabrera[SMTP:cabrera@MICROSOFT.com]
Sent: Sunday, October 06, 1996 5:31PM
To: Mechler, Edmund
Subject: Ethics Final Draft

Ed,

Could you address the comments made below?
Some are tough and general, and perhaps they cannot be addressed.
Yet the more pointed ones would be nice to clarify and fix.
Please let me know.

-- Felipe

>----------
>
>Here are my comments on the proposed code of ethics.
>
>In general, I think this expresses some good ideals, but is entirely
>impractical in a number of cases. The problem is that many of the things
>being proposed are vaguely defined, impossible to accomplish, or
>not under the control of the software engineer. A few others might
>not be generally accepted as the "right" thing to do.
>
First The SE Code of Ethics was built using other professional codes as models. Ethics doesn't have a practical/impractical side that I have ever experienced. Usually, ethics are vaguely defined and are not under any one groups control. I don't think that any of the items are impossible to do as a professional. Finally, I'll give 100 to 1 odds that under any professional code there are items that some will think are not the "right" thing to do.
>Specific comments and suggestions are identified below.
>
>> >CODE OF ETHICS FOR SOFTWARE ENGINEERS
>> >
>> >INTRODUCTION
>> >
>> >Computers now have a central and growing role in commerce,
>> >industry, government, medicine, entertainment, and ordinary life.
>> > Because the utility of computers depends in large part on the
>> >instructions written for them, those who design, develop, and
>> >test software have enormous opportunities both to do good and to
>> >cause harm. To assure, as much as possible, that this power will
>> >be used for good, software engineers commit themselves to making
>> >the design, development, and testing of software a distinct,
>> >beneficial, and respected profession. In accordance with that
>> >commitment, software engineers shall adhere to the following
>> >standards of conduct. The seven main paragraphs state general

>> >rules.
>
>The following sentence makes no sense to me:
>
>> >Each subsidiary clause is a specific application of its
>> >general rule, one experience has shown needs express statement;
>> >but no set of subsidiary clauses exhausts the general rule.
General disclaimer saying we cannot give all instances of a rule in ethics.

>
>
>> >
>> >Rule 1: PRODUCT. Software engineers shall, insofar as possible,
>> >assure that the software on which they work is useful to public,
>> >employer, customer, and user, completed on time and at reasonable
>> >cost, and free of significant error. In particular, software
>> >engineers shall, as appropriate:
>> >
>> > 1.01. Assure that specifications for software on
>> > which they work have been put in writing,
>> > satisfy the user's requirements, and have the
>> > customer's approval.
>
>I see no reason why the ETHICS would require that specifications be
>put in writing. And I do not believe a software engineer is in a position
>to assure that specifications have met the other criteria. Examples:
>- you are brought in to fix something where you have(no) know knowledge
> of the history of its requirements
>- you are given the requirements by your organization and have no
> interface with the customer and do not know if the customer has
> approved
Specs in writing could be put "record specs" but at present writing is the safest way of assuring transfer of data. The example given is one of the problems we are trying to decrease the occurrence of.
>
>> > 1.02. Assure that they understand fully the
>> > specifications for software on which they work.
>Would that this were true. It never is in practice. Things are gray here.
Again, as professionals, another area we are trying to decrease the occurrence of. Shouldn't we strive for better specs and be unethical if we didn't.
>
>> > 1.04. Assure proper goals and objectives for any
>> > project on which they work.
>How can this be done? What is "proper"? By whose standards?

There seems to be a contradiction here - if standards, companys/industry, exist then "proper" is defined. These are not difficult to do, although we don't, if we follow project manage techniques.
>
>> > 1.05. Assure proper development methodology on any
>> > project on which they work.
>Same question as above.
See 1.04
>
>> > 1.06. Assure proper management on any project on
>> > which they work, including proper procedures
>> > for control of quality and risk.
>This is patently impossible. A software engineer is not in a position to
>assure anything about management unless they are the management.
As any type of engineer I can assure proper mgmt of my responsible areas and shouldn't I report mismgmt.
>
>> > 1.07. Assure proper estimates of cost, schedule,
>> > personnel, and outcome on any project on
>> > which they work.
>Once again, what is "proper" and notwithstanding that, how can a software
>developer assure this in practice? Chances are they were added to the
>project long after estimates were done.
Another area we are trying to decrease the occurrence of.
>
>> > 1.08. Assure adequate documentation on any project
>> > on which they work, including a log of
>> > problems discovered and solutions adopted.
>This is determined by company processes and procedures and policies and
>specific customer requirmenets, not by software engineers desires.
What is the problem? Why shouldn't a professional add to these in areas of their expertise?
>
>> > 1.09. Assure proper testing, debugging, and review
>> > of software and related documents on which
>> > they work.
>What is "proper"?
See 1.04
>
>> > 1.10. Assure that software and related documents
>> > on which they work respect the privacy of
>> > those who will be subject to the software.
>This is vague and untestable. Who will be "subject to the software"?
>What does it mean to "respect the privacy"?
This is an ethics statement not a program being written. I don't think it is vague but an assurance that the SE thinks about the privacy issue.

>
>> > 1.11 Assure that raw information used in software
>> > is accurate, derives from a legitimate source,

>> > and is used only in ways properly authorized.
>Assurance may be difficult.
It may be difficult but we owe it to the public and who we are doing the job for.
>
>> > 1.12. Assure ethical, economic, cultural, legal,
>> > and environmental issues are properly
>> > identified, defined, and addressed.
>Few software engineers even know what these issues are.
Then maybe they are not SEs and this is a goal we should be aiming for.
>
>> > 1.13. Promote maximum productivity and minimum
>> > cost to employer, customer, user, and public.
>A nice ideal.
Is the commentator saying we are not doing this now? Are we a group of cheats?
>
>> > 1.14. Avoid fads, departing from standard
>> > practices only when justified.
>This is a judgement call. Highly inappropriate in a code of ethics.
Ethics is a set of principles or values that guide our judgments.
>
>> >
>> >Rule 2: PUBLIC. Software engineers shall, in their professional
>> >role, act only in ways consistent with the public safety, health
>> >and welfare. In particular, software engineers shall:
>> >
>> > 2.01. Disclose to appropriate persons any danger
>> > that the software or related documents on
>> > which they work may pose to the user, a third
>> > party, or the environment.
>> > 2.02. Approve software only if they have a well-documented
>> > belief that it is safe, meets specifications, and
>> > has passed all appropriate tests.
>This is too arbitrary. Why must their belief be well documented? Doesn't
>this depend on the intended use of the software? Doesn't "safety" also
>depend on the intended use?
Again 1.14 response; taken from PE Code.
>
>> > 2.03. Affix their signature only to documents
>> > prepared under their supervision and within
>> > their areas of competence.
>This would eliminate most documents one signs while seeking

>employment, such as non-disclosure agreements and
>various employment condition documents, such as intellectual property
>rights agreements.
I do not see what these examples have to do with Rule 2.
>
>> >
>> > 4.01. Provide service only in areas of competence.
>This would eliminate most practicing software engineers. I mean this.
>Who determines competence?
The commentator paints a very bleak picture of our industry.
>
>> > 4.02. Assure that any document upon which they
>> > rely has been approved by someone qualified
>> > to approve it.
>And how are they to determine the qualifications of the person who
>approved a document that was prepared by another organization or
>a long time ago?
See 1.14 response
>
>-- (snip) ---
>
>> > their opinion, a project is likely to fail,
>> > to prove too expensive, to violate copyright
>> > laws, or otherwise to turn out badly.
>I assume client means employer in most cases.
Yes
>
>-- snip --
>
>> > 7.01. Improve their knowledge of recent
>> > developments in the design, development, and
>> > testing of software and related documents.
>I don't see why this is required for them to be ethical.
>Ditto for the rest of this section. Sounds more like what one would
>expect for them to get a license, not to be considered ethical.
In the fastest growing field today, and perhaps in history, this is definitely an ethical issue. I have met too many people giving advice and guidance who I hope were not up to date and not just lying
>
>End of Comments
>