This article is part of the IEEE Reliability Society 2007 Annual Technology Report.

Information Integrity and I.T. Professionals' Integrity, Intertwined

Keith W. Miller and Jeffrey Voas
Email: j.voas@ieee.org

1. Introduction

"Information Integrity can be defined as the dependability and trustworthiness of information. More specifically, it is the accuracy, consistency and reliability of the information content, processes and systems." http://www.informationintegrity.org/index.php/c/What_Is_Information_Integity%3F

"PROFESSION - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest." [ACM/IEEE Software Engineering Code of Ethics and Professional Practice]

Information Assurance and Quality Assurance are intertwined with the idea of information integrity. When a consumer accesses information, the consumer wants to be able to trust the information. As computer professionals, our reputation with the public depends in large part on how the public perceives the information we deliver to them. If the public trusts the information we provide, the public trusts us. In so far as the public mistrusts that information, it mistrusts us.

Information integrity is attacked in myriad ways. In this paper, we focus on three threats to information integrity that map directly to characteristics of information-processing professionals: incompetence, conflicts-of-interest, and a lack of transparency. In each case, characteristics of individuals (micro-ethics concerns) and characteristics of the profession as a whole (macro-ethics concerns) interact to undermine information integrity. The paper concludes with strategies for improving information integrity by focusing on these three inter-related problems.

2. Professional Responsibility to Whom?

At least according to traditional codes of ethics, a professional has a primary responsibility to society at large. In addition to this primary responsibility to the public, the ACM/IEEE Software Engineering Code of Ethics and Professional Practice ("SE Code") also describes a computer professional's responsibilities to clients, to employers, to clients (or customers), to colleagues, to the profession as a whole, and to him- or herself. As we shall see, this entire set of responsibilities is tied to how well the professional protects informational integrity, because all of these stakeholders are affected if that integrity is compromised.[1]

3. Ethics-Related Threats to Information Integrity

The integrity of information is threatened on all sides: information systems can fail due to software defects and hardware failures; outsiders can attack security and, if successful, steal and/or degrade the information that was to be protected; information can be intentionally skewed or faked at its source; and external disasters can wipe out storage devices, including backups. Out of the wide range of important threats, we will select three threats that are traceable to issues that map directly to professional ethics, namely: incompetence, conflicts of interest, and lack of transparency. For each of these three threats we will list relevant excerpts from the SE Code.

Threat 1. Incompetence

"…software engineers shall, as appropriate: …

1.1 Accept full responsibility for their own work."

2.01. Provide service in their areas of competence, being honest and forthright about any limitations of their experience and education.

3.04. Ensure that they are qualified for any project on which they work or propose to work, by an appropriate combination of education, training, and experience.

No one wants to be known as incompetent, but is it really an ethical issue? According to the SECode, it certainly is. The issue of professional competence is, as the quotes above illustrate, a central issue in several of the code's sections.

The relevance of incompetence to information integrity is straightforward. No matter how fervently you want to do a good job generating, storing, protecting, or retrieving information, you are unlikely to preserve information integrity if you don't know what you are doing. As individual professionals, this means that we should only undertake projects in which we have an informed confidence that we can deliver the desired outcome within the time and budget constraints presented to us. This is not to suggest that each slipped schedule and each software failure arises from a distinct moral lapse; ethical software engineers make mistakes too, and information processing entails risk. However, individual software professionals are called upon for a good faith effort to undertake only those projects for which they are qualified, and to then apply due diligence on each of those projects. Without that kind of individual effort, information integrity will be undermined repeatedly as incompetence degrades data, opens security vulnerabilities, and delays access.

Unfortunately, as is pointed out elsewhere in this issue, there are systematic and pervasive problems with information integrity; certainly many of these problems can be traced to individuals who have acted incompetently. However, the scale and the universality of these problems (for example, see http://catless.ncl.ac.uk/risks) suggest that this is not merely a case of a small subset of individual lapses. As a profession, computer experts have not been able to convincingly demonstrate an ability to adequately control the quality, nor even precisely define the qualifications of, computer professionals. The long standing struggles to codify a software engineering body of knowledge attest to the difficulties of this task, unless and until professional standards are established for software engineers, security technicians, data processing programmers, and a whole range of IT workers, competence will be difficult to measure and difficult to attain.

Threat 2. Conflicts-of-Interest

4. JUDGMENT - Software engineers shall maintain integrity and independence in their professional judgment. (SE Code)

4.05. Disclose to all concerned parties those conflicts of interest that cannot reasonably be avoided or escaped.

In order to focus on one problem at a time, let us assume for the moment that a computer professional is working on a project for which she is qualified. A problem distinct from competence is the presence of conflicts of interest. In most professional situations, there are competing interests. A professional has responsibilities to the public, often to an employer, to customers, to co-workers, and to people who will be indirectly affected by the information produced. It is rare indeed for all these interests to be compatible, so part of the job of being a professional is to judiciously balance these interests, interests that are always in tension. Conflicts of interest become an ethical problem (and the term is generally reserved for situations in which there is a problem) when for a particular professional is unable to make an objective judgment on the proper balancing of these interests. [http://www.ethics.ubc.ca/people/mcdonald/conflict.htm]

For example, if a IT professional P is asked to make a judgment about which of three vendors should get a contract, and if P has an undisclosed financial interest in one of the vendors, then P is unlikely to be able to give an unbiased professional opinion on which vendor is best. For this example, the ethical problem is clear, as is the solution: P should reveal the interest in the vendor and either not render an opinion, or render the opinion along with a disclaimer about the financial interest. Unfortunately, not all conflict of interest problems are as obvious or as easy to resolve for IT professionals.

For example, a classic problem in information integrity is trying to determine when an information system is ready for deployment. (This system might be for generating, storing, protecting, or accessing information; the principle will be the same for all of these systems.) Any decision about releasing a system is a complex judgment requiring the balancing of many stakeholder interests. The developers of the system are likely to lose money if the release is delayed for long. Customers want the highest quality, but also want the software promptly.

Again, let’s call the IT professional making the release or not release decision P. Let’s assume that either internal policy or an explicit contract requirement dictates that P (and P alone) is to sign off on a project before it can ship. P has loyalty to an employer, feels responsibility to the customer and others who will be affected by the system, and also wants to maintain a personal reputation for quality work. As before, such conflicts are inevitable, and the IT professional is required to make a subjective, but well informed decision about the release, a decision that takes into account these competing interests. However, the IT professional’s objectivity may be compromised. Because it is a complex judgment, no matter what P decides (go or no go), P’s manager may object. Discussing the decision with the manager is fine; professionals can disagree. However, at some point P has to make this decision, and should be allowed to make a good faith professional judgment. (After all, that’s why the policy or contract designated P in the first place.) But what if the manager insists that P change that judgment? What if the manager threatens P with demotion or termination unless the decision is changed?

In such a situation, the P faces a difficult conflict of interest. On the one hand, P has determined a good faith judgment about system release; but the manager is trying to override that judgment. If this were a situation in which the manager and not P took responsibility for the release, the manager could simply override P and both could take responsibility for their separate actions. But if the manager requires P to sign off on the manager’s judgment despite P’s reservations, this means that P will either capitulate (and compromise his professional integrity) or refuse to go along (and suffer personally damaging consequences).

As MacFarland pointed out in a similar case (McFarland, M. C. 1991. Ethics and the Safety of Computer Systems. IEEE Computer 24, 2 (Feb. 1991), 72-75.), this difficult situation is not of P’s making. Instead, the manager (and by extension, P’s employer) have put him in this situation unfairly. The employers should not demand that P lie about a professional judgment. Even so, when such a situation occurs (and many IT professionals know first hand of such situations) there are two important issues besides the culpability of P’s employer. Once this situation arises, what should P do, and what, as a profession, should IT professionals collectively do to deal with these situations.

An entire literature exists on P’s dilemma, including much work on the pros and cons of whistleblowing. We will not review that literature here (see http://www.ethics.org.au/about-ethics/ethics-centre-articles/ethics-subjects/whistleblowing/index.html and http://www.onlineethics.org/ for resources on whistleblowing and engineers); suffice it to say that in our case P now will have to wrestle with his conscience in trying to choose from several desperately bad obvious options, or trying to creatively develop a better option that is not immediately obvious. How P makes that decision is a topic for micro-ethics. However, there are many macro-ethics issues connected to this case in addition to the clear ethics violation by P’s manager. One important issue is support for whistleblowers.

There is a history of courageous whistleblowers, including whistleblowers among IT professionals. However, the story about what happens to such whistleblowers after they alert the public to a problem is not always a happy story. For example, a group of engineers went public with their reservations about the automated Bay Area Rapid Transportation system that was going to be deployed despite their objections. (Friedlander, G. The case of the three engineers vs. BART. IEEE Spectrum, October 1974: 69-76.) Although the engineers’ concern were subsequently shown to be valid, they were fired.

One of the side effects of the BART case was that it helped motivate the IEEE to establish an ethics committee and, eventually, an IEEE ethics hotline. There was also a proposal to establish a fund, supported to voluntary contributions, to assist whistleblowers who were retaliated against by employers. However, soon after the hotline was established and before the fund was established, the IEEE stopped both projects, citing legal concerns. (The assault on IEEE ethics support Unger, S.H.; Technology and Society Magazine, IEEE Volume 18, Issue 1, Spring 1999 Page(s):36 - 40) This is an example of how larger institutions do (or do not) act to support engineers (and IT professionals) when they attempt to act ethically, especially when faced with difficult conflict of interest problems that involve their employers. The less institutional support that the profession gives them, the more difficult it is for IT professionals to act on their consciences.

Threat 3. Lack of transparency

“…software engineer should, when appropriate…

1.06. Be fair and avoid deception in all statements, particularly public ones, concerning software or related documents, methods and tools.

2.05. Keep private any confidential information gained in their professional work, where such confidentiality is consistent with the public interest and consistent with the law.

6.07. Be accurate in stating the characteristics of software on which they work, avoiding not only false claims but also claims that might reasonably be supposed to be speculative, vacuous, deceptive, misleading, or doubtful.

Perfect information integrity is a useful aspiration, but it is not a practical goal. In order to be useful to humans, information must be accessible to humans. Accessible information is always at risk of degraded integrity. Thus the generation, storage, protection, processing and retrieval of information always entails risk. Communicating (and not communicating) about that risk is the final ethical issue we will explore.

No one can honestly guarantee complete information integrity, but a conscientious IT professional seeks to maximize that integrity within practical constraints. How then should an IT professional communicate with a customer and with users responsibly about the risks to information integrity that are inherent in a particular system? The ethical choice for this communication requires openness. This openness has its limits. The IT professional has obligations to protect trade secrets and to maintain confidentiality, as pointed out in clause 2.05 above. But when it comes to information integrity, there is much the professional can be candid about. Unfortunately, this kind of transparency has become rare in commercial software.

An example of non-transparency is the treatment of known software bugs. Cem Kaner is a leading advocate for software consumers (and also is an experienced software developer). He suggests a simple rule about known software bugs: “The software company or service provider must disclose the defects that it knows about to potential customers, in a way that is likely to be understood by a typical member of the market for that product or service.” (Software Customer Bill of Rights, http://www.satisfice.com/kaner/?p=8) In today’s climate that might seem like a strange idea, but years ago it was not uncommon for commercial products to publish lists of known bugs. To this day, open source projects maintain such lists, and the web makes it relatively inexpensive to keep such lists up-to-date and accessible.

There are legal, marketing, and practical challenges associated with being open about known bugs. However, past history and the open source practice demonstrate that it is not impossible to meet these challenges. And it seems clear that openness about known bugs is consistent with the ethical obligations described in the SECode. We can again look at the issue from micro- and macro-ethical perspectives.

When an individual IT professional is in a position to make the decision, he/she can be candid with the customer about known bugs. The customer and the developer can then discuss the importance of these defects and the likely cost of repairing them. This can lead to a well informed negotiation about how next to proceed. If a customer decides that ignorance is bliss about known bugs, the customer can make that choice; but the IT professional has made the process transparent none-the-less.

Many IT professionals in a corporate setting will not have direct contact with customers and users, and will not be able to discuss or publish known bugs with outsiders. In this situation, the organization, not the individual professional, bears the ethical burden. From an even wider perspective, the case can be made (Cem Kaner makes this case) that governments should require this kind of openness. Such laws, conscientiously enforced, make for a level playing field that uniformly mandates ethical openness about software bugs.

Openness about known software defects is only one example of openness about information integrity. But the idea of openness generalizes: ethical professional practice requires that an IT professional should be an active partner with customers in helping the customer understand not only the strengths but also the weaknesses associated with our attempts to deliver information that has integrity.

4. Conclusions

There are subjective judgments by all three of the issues we have explored. It is non-trivial to determine adequate competence for a particular job. It requires diligence to recognize serious conflicts of interest. And being straightforward about the limits of our information integrity requires courage as well as care. But an IT professional who is serious about living up to ethical standards is obligated to make a good faith effort to meet these challenges. Furthermore, business and governments should be proactive in creating an environment that encourages IT professionals to follow their conscience in their professional lives.

Reference:

[1] Recent work by philosophers has also explored “information ethics.” According to this emerging theory, computer professionals also have a responsibility to the information itself. Although this idea seems particularly relevant to the issue of informational integrity, the literature on information ethics is controversial and evolving. To read more about the theory, see Floridi, L. 2006. Information ethics, its nature and scope. SIGCAS Comput. Soc. 36, 3 (Sep. 2006), 21-36. DOI= http://doi.acm.org/10.1145/1195716.1195719