

ICT GOVERNANCE AND WHAT TO DO ABOUT THE TOOTHLESS TIGER(S): PROFESSIONAL ORGANIZATIONS AND CODES OF ETHICS

Don Gotterbarn, Professor Emeritus

East Tennessee State University

ABSTRACT

ICT has had numerous unfortunate incidents in planning, development, and delivery. A typical response to these incidents is to both complain about the toothless tiger of unenforced and unenforceable technical and professional standards and to also advocate the development and implementation of strong government regulations – licensing and legislation. These regulations constitute one form of what has been called “ICT governance”. Unfortunately, there are significant limitations to both the constraining regulations approach and the vague toothless tiger approach to ICT governance. There is an approach to ICT governance which takes advantage of and strengthens some roles of professional organisations and avoids the dangers of giving government malformed regulatory teeth which limit the positive potentials of ICT and introduce overt harms. The purpose of this paper is to define strategies for professional organisations to meet their responsibilities to the ICT profession and ICT professional; a strategy which moves toward regulation without curtailing ICT potential with ineffective sanctions. Professional organisations need a strategy for reducing negative incidents and improving professional responsibility without simply introducing sanctions on a narrow range of practitioners who happen to be members of that organisation. There are ways in which the toothless tiger(s) can have a significant positive influence.

INTRODUCTION

Information and Communication Technology (ICT) has been with us for many years and in the past ten years there has been a growing interest in something called “ICT Governance” as a means of reducing information system disasters. There have been national organizations formed, professional organizations have organized sub-committees to address ICT Governance and it has even been called a “discipline”. The unwrapping of this concept, like many concepts, has several inconsistent interpretations and ineffective implementations in industry. In some cases the concept has been modified to meet a particular sectors needs. This broadening of concepts to fit individual needs is not new and sometimes is quite useful. There is however a fundamental mistake in the narrowness of most interpretations of ICT Governance which make it less likely that it will achieve its ultimate goals. I believe this mistake can and should be addressed by professional computing organizations.

In what follows I will examine the various approaches to ICT governance, the difficulty it tries to address and I will argue for what I consider its critical limitations. I will then show how professional organizations can address the weakness of ICT governance using tools they already have at hand.

THE PROBLEM FROM THE SOFTWARE PERSPECTIVE

In the early days of computing 1960's people worried about "the software crisis"; worried about the failure of software systems. Software workers addressed these problems by developing models for building well engineered software. The focus of the computing community was primarily internal; focusing on how to develop and test a program. This was the period of mathematical modeling of software development in books on Software Science (Halstead, 1977) and the Discipline of Programming (Dijkstra 1976). The focus was on making computing into a reliable engineering-like discipline. The impacts and concerns addressed were local to the particular system being developed.

In 1968 the term "Software Crisis" was coined to describe problems with developing correct and reliable software programs. The responses to these software problems generated and continues to generate many single mode solutions; do this one thing in software development and the problems will be solved. These approaches included an emulation of an engineering approach to development of software, then a structured approach to program design, then a formal proof of software system requirements, then object-oriented software development, a focus on individual programmers counting the number of errors they make (Personal Software Process) and now an agile or extreme programming approach to software development. These single mode methods were regularly interspersed with approaches which emphasized measuring the software's size, reliability, and space and time efficiency. The indication of the lack of success of these approaches is the rise of ICT Governance to address the negative impacts of the Software Crisis on industry.

I believe there were three major difficulties with these approaches by the software community. First the software crises is a complex problem and single mode approaches over emphasize one piece of the problem and tend to ignore other pieces of the problem. Second, computing technology and applications are constantly changing, and changing at a very rapid rate. This means the domain of the software crisis is also constantly changing; advances like computerized robotic surgery was not an issue in the 1970's. E-commerce had no meaning 15 years ago. The software crisis is still a problem with software development but the software being developed has expanded applications and the relevant stakeholder communities have correspondingly increased. The third problem is that the solution to a problem with software interaction with business and society was addressed only from the software side of the problem. These single mode approaches were focused on what software developers do. Since 1968 the answer has been the same in different clothes- and the mistake is the same. If I am an honest person and work hard following a good process then the problems will go away. Unfortunately this has consistently not worked. There are still significant systems failures which lead to major corporate failures. The effect of programmed trading on the stock market in October 1987 took years to recover from.

There are numerous standards organizations such as the American National Standards Organization (ANSI), the IEEE, The International Standards Organization (ISO), and Standards Australia which have worked on developing rules and standards to control and monitor software developments.

The diverse and developing nature of ICT makes the application of these rules to software development difficult.

- Since no common education is required many ICT workers do not know the Standards/Rules
- Since there are few sanctions, oversight, or threat to losing an ICT license obeying the rules is voluntary and it needs to be made clear what is gained by following them/
- Since the environment is changing so rapidly rules either no longer apply by the time they are finally written or they are inconsistent because they address different sides of an issue. Who enforces fraud legislation on international Internet sales?

The focus to solving the software crisis was internal on the technology and on how to be a good technician; the software developers were going to fix it. Another manifestation of these single mode internally focused approaches was a method of software development called 'over the wall development'. At its worst such development had the following scenario. The skilled software technician would gather requirements for a business software project and then develop a complete system without any further consultation with the client. The system developed (as the software person misunderstood it) would be presented to the customer (thrown over the wall). The assumption being that the technician had some how understood all of the important business issues and addressed them in the software. This systemic disconnection of developer and customer led to working systems being delivered which had nothing to do with the customers needs. Since corporate executives did not fully understand how IT worked, when a corporate manager complained about the style or function of the delivered system ITC personnel could simply declare that what they delivered was the only way the computer could do the requested task.

This problem is not unique to software development but occurs where there is discrete development elements and no communications between those working on different parts of a project. There is a disconnection between the world of the computer astute developer and the customer who is skilled in their own domain. The two parties speak different languages. The programmer who understands the syntax of a programming language is overwhelmed when requested to write a system to model genetic mutations which calculates the sequential effects of genetic drift, inbreeding, selection, gene flow, and mutation upon the proportion of a population's gene pool comprised of the second of two alleles. To think that the difficulty of developing this project could be resolved by focusing on software technology is a mistake. This programmer optimism is repeated I numerous projects.

History repeats itself

In the development of ICT Governance, I think history is repeating itself. I think a similar set of mistakes is being made in ICT Governance's attempt to address issues with ICT systems. Many areas in ICT Governance are taking a single mode approach which focuses primarily on its side of the problem, on the business element, minimizing the ICT elements and it does not take into consideration the rapid change of the domains of ICT and business.

ICT Governance

In general ICT Governance emerged as an attempt of business to deal with the impacts of major software system failures on business. Although it was primarily a part of corporate governance which focused on information technology, many ICT professional organizations contributed to the process of developing standards. These organizations advocated that their members adhere to these standards.

In 2003 Australia published several other standards for Corporate Governance including: Good Governance Principles (AS8000), Fraud and Corruption Control (AS8001), Organizational Codes of Conduct (AS8002), Corporate Social Responsibility (AS8003), Whistle Blower protection programs (AS8004) so the pattern was set for a document from the corporate side addressing ICT Governance. Standards were developed such as AS8015 – (2005). This Australian Standard for Corporate Governance of ICT to deal with in-house development issues and the fact that outsourcing to vendor's interests was not always consistent with those of a user organization. This is similar to one of the problems of over the wall software development.

The interest in ITC Governance is international and quality development from the Corporate side is defined in the Information Technology Infrastructure Library <http://www.itil.co.uk/>; a detailed framework with hands-on information on how to achieve a successful governance of IT. As in the attempt to deal with the software crisis described above, there is an ISO standard to deal with corporate governance. "Because inadequate information technology (IT) systems can hinder the performance and competitiveness of organizations or expose them to the risk of not complying with legislation, the new ISO/IEC 38500 standard provides broad guidance on the role of top management in relation to the corporate governance of IT." <http://www.iso.org/iso/pressrelease.htm?refid=Ref1135>.

The material on ICT Governance is not consistent in describing the level where ICT Governance comes into play. One should not be misled to think that there is a single standard of ICT Governance. There are compliance standards like Sarbanes –Oxley (section 404) that focus on configuration management. On the other hand, Weil (2005) in a study of 300 enterprises around the world "did not identify a single best formula for governing ICT." They say that "ICT governance specifies accountabilities for ICT- related business governance and helps companies align their ICT investments with their business priorities." ICT Governance performance for them involves the corporation's "Cost effective use of ICT, ICT for growth, ICT for asset utilization and, ICT for business flexibility." They argue that "ICT governance is the decision rights and accountability framework for encouraging desirable behavior in the use of ICT." This is very different from viewing ICT Governance as managing the interface between ICT development and corporate management.

Sense of 'Control'

But if we follow exactly the Australian definition of ICT Governance; it uses evaluation and control but it takes many forms; both in what is controlled – ICT systems development and the current and future corporate use of ICT. This same diversity exists in the standards being developed by the International Standards Organization. There is also diversity in what is meant by "control". Control is characterized both as a rigorous highly delineated control and a general directional guidance and principles to help the professional make judgments.

Single mode

As in software development, it looks like ICT Governance takes a single mode solution approach and that there are a variety of single modes. There are 'structuralists' who think everything is handled by structure and that the primary goals for information technology governance are to (1) assure that ICT generate business "VALUE" and reduce the risks that are associated with ICT by implementing a carefully defined organizational structure. This is sometimes connected with who is in charge or has ownership of the system; an over the wall problem.

Over the Wall- it is mine!

The definition of ICT Governance is tied to CORPORATE governance and relates the business focus of an organization and ICT management. It mandates that ICT decisions are owned by the corporate board rather than by ICT managers. This has the same problem as over the wall development. There are indeed limitations to what ICT can do and the ways it does things will have different effects. Balanced ICT Governance needs the ICT side in their systems guided by ICTers.

Limited view of stakeholders

Several of the problems associated with software development are recurring in ICT Governance. There is an additional problem which is common to the each of the issues of software development. There is a limited view of who constitutes the stakeholders in a project. The limited view of project stakeholders is constant- the developer and the customer. The effect of an ICT system on a business or on extended stakeholders is not considered. The current concept of ICT governance is modeled on the traditional business ethics concepts of who needs to be considered as the relevant stakeholders, namely those who have some financial interest in the business (Agle 1999) The current concept of ICT Governance stakeholders is "ICT governance implies a system in which all stakeholders, including the board, internal customers, and in particular departments such as finance, have the necessary input into the decision making process." This limited view of the system context and the stakeholders in that context is also supported by ISO standard which characterizes ICT governance as the management system used by directors. "ICT governance is about the stewardship of ICT resources on behalf of the stakeholders who expect a return from their investment." We shall return to how this problem can be addressed by professional organizations.

In a recent survey of ethical issues conducted by CAPPE (Lucas and Weckert, 2008) the limited understanding of stakeholders is clearly indicated when most of them say they do not consider their work to be related to larger segments of society.

The Concept of ICT Governance is closely tied to some concept of "Control." The use of words like 'control' and 'govern' imply some type of enforcement or sanctions for failure to follow the rules, but in ICT professional societies these rules seem like toothless tigers. There are no real sanctions for failing to follow the rules. Professional societies can have a significant roar without any associated bite.

We see the attitudes of many computer practitioners toward attempts to promulgate rules of behavior clearly in Lucas (2008). A member of the Australian Computer Society (ACS) complained about the rules because they wanted them to apply to everyone practicing ICT. "The ACS can discipline its members for breaches of its ethical code but that is no barrier to employment and it has no effect on the vast majority of workers in the industry who are not ACS members." (Lucas and Weckert, 2008)

This failure of universal application of principles and regulations is a common complaint of honest hard working computer practitioners. It should be noted that these are not complaints about the importance of following such rules. The complaints have other motivations. There is the problem of fairness that they are held accountable as members of the ACS. "Why should I follow the rules if those outside the ACS are not bound by them? Another motivation for the complaint is that the practitioners realize that following these standards will reduce some of the harms caused by software and improve the lot of humanity and as such everyone ought to follow the rules. The same need to 'enforce' compliance is held in ICT Governance. The primary motivator for following the

rules is the addition of teeth – sanctions- for not following the rules. As we have seen this is done with some compliance standards like Sabanes-Oxley and ISO Standards.

THE limitation of the ICT governance approach

It seems to me that ICT Governance is repeating some of the same moves made by software developers in their attempts to address the software crisis and I believe ICT Governance is heading for a similar set of problems as those faced by software developers. The development of systems software has to develop an interface between the technology of computing and the nature of the enterprise. The nature of this interface must be guided by a consideration of the impact of the system on a broad range of stakeholders. Software developers focus on the nature of the software and how to reduce errors in the programs. They tend to focus on a limited set of stakeholders in the system: developer, customer, sponsor, and vendor and tend to pay limited attention to those who will be impacted by the deployment of the software. ICT Governance, likewise, seems to have an internal focus on the business side of the software system and a narrow view of the stakeholders as those with a financial interest in the system (Agle 1999, Weil 2005). We can see some of the consequences of this narrow approach by looking at the information technology treatment of ‘software risk’

The narrow stakeholder focus in Software risk

Although the need for high quality software is obvious to all and despite efforts to achieve such quality, information systems are frequently plagued by problems (Ravichandran 2000). A narrow approach to risk analysis and understanding the scope of a software project and information systems has contributed to significant software failures. Informaticians have been evolving and refining techniques to mediate risks of developing software products that meet the needs of their clients. The risks focused on include: missed schedule, over budget, and failing to meet the system’s specified requirements. In spite of this attention to risks, a high percentage of software systems are being delivered late, over budget, and not meeting all requirements, leading to software development being characterized as a “software crisis” and a mistrust of software systems. Risk management generally consists of an iterative series of steps like the ones shown in Figure 1.

The context referred to in the top box—the context in which the project is being developed—includes the organizational structure, and its competitive and political position as well as its risk management structure. The risk identification process identifies potential negative impact on the project and its stakeholders. AS/NZS lists potential negative areas of impact such as

“Asset and resource base of the organization, Revenue and entitlements, Costs, Performance, Timing and schedule of activities, and Organisational behaviour” [AS/NZS, 1999, p. 39]

The risk analysis process divides the identified risks by their severity and the likelihood that they will occur, producing a given level of risk. This level of risk is generally determined using statistical analysis or calculations with fault trees and event trees. A typical calculation is ‘Risk Exposure’, a metric derived by multiplying the anticipated costs by the probability of the event occurring.

Two forms of exposure are commonly calculated. The first method using quantitative risk analysis provides quantitatively expressed assessment of the negative consequences of an event as the outcome of an event e.g., “A delay of one day will cost \$3000 in sales.” The second, qualitative risk analysis is often used to address risks which are not readily quantifiable, other than by

describing the degree of risk e.g. “The delay will upset our distributors causing significant loss of goodwill”. Generally “qualitative analysis is often used first to obtain a general indication of the level of risk... or where the level of risk does not justify the time and effort for a quantitative analysis...” [AS/NZS, 1999, p. 14]. Its role primarily is to characterize and identify the impact of a risk generally asserted in terms of dollars. Surprisingly, in standard risk methodologies the qualitative risk approach typically looks at quantifiable data which can be easily prioritized and facilitates analysis. These standard methods of risk identification and risk characterization have been used extensively in software projects but are incomplete.

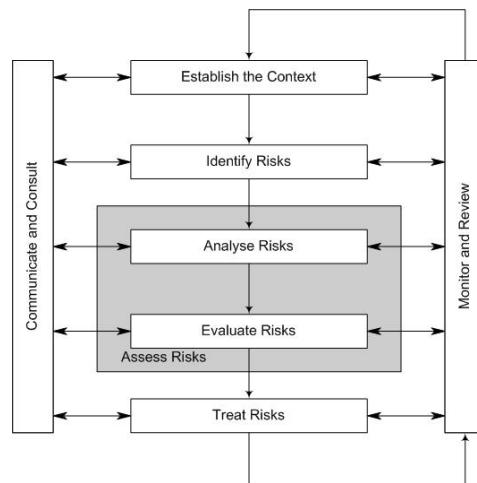


Figure 1. Risk Management [AS/NZS, 1999 p.16]

The Association for Information Systems defines “system quality” in terms of currency, response time, turnaround time, data accuracy, reliability, completeness, system flexibility and ease of use [AIS 2005]. Even after using these generic models of risk analysis, Information systems have been produced which have significant negative social and ethical impacts. The risks of these impacts are not traditionally included in the tripartite concept of software failure-over budget, late, or not meeting stated functions. The extended stakeholders in developed software are all those who are affected by it even though they are not directly related to the use or financing of a system. The political candidate who is not elected because of a difficult voting machine interface is a stakeholder in the development of that voting machine. The person who suffers identity theft because of a flaw in the security for an information system is a stakeholder in that information system. The developer’s obligations to these stakeholders are not included in the generic concept of software failure.

These systems may have been a success in terms of being developed on schedule, within budget, and delivered on schedule but were a failure because they failed to take into account the conditions in which they were used. The user interface which met specifications had a significant impact on the

lives of others. The system used to record dosages of pediatric medicine correctly handles negative interactions of dosages but was awkward to use in emergency situations resulting in 3 medication errors out of every 100. (Pediatrics 2006)

Contributing Factors

Two interrelated factors related to system stakeholders contribute to missing these professional and ethical failures. First, limiting the consideration of system stakeholders to just the customer/client, software developer and those who have a financial stake in the system ignores the needs of other relevant stakeholders. Some have realized that the focus on technical risks is too narrow; but unfortunately the risk focus only expands to other internal issues related to the development of the system. For example Ravichandran (2000) says “Research in software quality has focused largely on the technical aspects of quality improvement, while limited attention has been paid to the organizational and sociobehavioral aspects of quality management.” This is similar to the position maintained in the Australasian risk analysis model above.

A second factor is limiting the scope of software risk analysis just to technical and cost issues. A complete software development process requires 1) the identification of all relevant stakeholders and 2) enlarging risk analysis to include social, political, and ethical issues. A complete risk analysis requires a process to help identify the relevant stakeholders and broaden the scope of risks anticipated.

To meet the goal of quality software, developers focus on particular risks including: project and schedule slips, cost increases, technical and quality risks, the timeliness of the product, risks that the final product will not fit the business for which it was designed. Nevertheless, developers use Risk Exposure to help them focus on the most critical risks. The use of easy to read fonts or an easy to use back up system may be ignored in an effort to get a product out in time or at lower cost.

The risks that are addressed are those with the highest Risk Exposure. All consequences are given dollar values. Even qualitative risks are turned into a numerical hierarchy. The resulting risk of the September 11th disaster was calculated in terms of the number of deaths that occurred on that day or lifetime dollar earnings potential of those who died. The negative effects that need to be addressed in risk analysis include both overt harm and the denial or reduction of goods. An automated surgical system which randomly moved inches instead of centimetres hurting patients would have a negative effect; as a pay phone system which disabled all usage, including 911, without an approved credit card would also have a negative effect preventing the report of an accident. These stakeholders, patient and someone hurt in a fire, are not normally considered. The scope of a project needs to be identified in terms of its real stakeholders.

This extended domain of stakeholders includes: users of the system, families of the users, social institutions which may be radically altered by the introduction of the software, the natural environment, social communities, informatics professionals, employees of the development organization and the development organization itself. The design of many of the USA’s voting machines correctly counted votes but made it difficult for people to enter their votes.

Modifying the approach to stakeholders.

In the software developers response to the ICT crises the focus was internal on the technology and how to be a good technician. Analogously ICT governance- is going through a similar, though much shorter, life cycle. As initially developed, ICT Governance focuses on the governance within an

organization; “evaluating and directing the plans for the use of ICT to support the organization and monitoring this use to achieve plans. It includes the strategy and policies for using ICT within an organization.” (AS8015, 2005). The focus is clearly internal with statements like “to better understand their obligations and work more effectively to maximise the return and minimise the risks to the organisation from ICT” The internal focus suffers from the same problems as ‘over the wall’ software development.

THE ROLE OF PROFESSIONAL ORGANIZATIONS

Professional organizations have been involved in the development of these ICT Governance Standards. For example the ACS was involved in the development of AS8015 and its narrow approach. Both ICT and the business sector have contributed what they view as solutions from their own sectors – object-oriented software design and financial flexibility of corporate systems—which model single mode solutions. They have difficulty enforcing these solutions and worry about system ownership. They are each concerned that their side of the wall has control of the system, and of course they also differ as to which stakeholders primarily need to be considered when developing these systems.

The professional organizations have the ability to resolve this. One of the roles a professional organization needs to play in resolving these problems is to broaden the range of stakeholders considered in the current concept of ICT Governance. ICT professional organizations have Codes of Ethics and Codes of Practice which actually address each of the ICT governance problems discussed above. Their codes of ethics require the IT professional to consider a broader range of stakeholders, to consider a range of stakeholder which includes all those whose lives are affected by the IT project and the way it was implemented. The Code of Ethics is very useful as a model of governance but is ignored because these codes are viewed as less enforceable than other types of regulations. They are considered worthless because they are like toothless tigers.

Professional Computing Codes of Ethics can address the ICT Governance problem

To see how Codes can be useful we need to look at some of the general functions of Codes and how they are managed in professional organizations.

Functions of Codes

Codes of ethics are developed by professional organizations for a variety of reasons.

Codes of ethics have served a variety of functions and are frequently directed at several audiences. At the simplest level, a code of ethics is a statement of the obligations of individual computing professionals in the conduct of their profession. The code will generally embody a moral commitment of service to the public. Sometimes they are used to clarify expectations and appropriate behavior of professionals. More positive functions of codes of ethics include: making a statement to members or aspiring members of the profession about shared commitments and agreed upon rules, sensitizing members to new issues, and provide guidance to individual members when they are confronted with ethical situations.

Sometimes codes are used to win public confidence and stave off external regulation. Some codes or parts thereof are disciplinary in order to convey a sense of self-regulation by the profession. They also help set expectations for employers and clients about dealing with members of profession and socialize novices in field. Codes are sometimes used to express and strengthen the community orientation of the group. Some Codes are very general and are therefore hard for the individual practitioner to apply in concrete cases.

Some confusion about codes of ethics arises from a failure to distinguish between closely related concepts about codes which direct the behavior of practicing professions. Moving from less restrictive to more restrictive codes, we first mention codes of ethics which are primarily aspirational giving a mission statement for the profession. There are also codes of conduct which describe professional attitudes and some professional behavior. Codes of practice are very specific and tied closely to the practice of the profession. They are the easiest to use as a basis for legal action. Because practicing professionals deal with human affairs, the underlying ethical principles are the same across professions. Studies have shown that most codes are a hybrid of these three types of code (Berleur, 1994)

Several authors have discussed the various goals of a code of ethics (Martin). These goals of codes of ethics could be ranked from the most benefit to society to most benefit to the individual member of the profession. These goals include:

1. Inspiration and Guidance for ethical conduct.
2. Support those seeking to act ethically by appealing to the public system of ethics established in a code.
3. Education and Shared understanding (by the professional and the public) of standards of practice.
4. Deterrence and discipline for specific actions by sanctioning code violations.
5. Protect the Status Quo by stifling dissent and state minimal level of ethical conduct.
6. Promote business interests by forbidding competitive bidding.

One way to evaluate codes of ethics is to examine which of these functions assumes prominence in the code. Codes also indirectly educate the public at large about what professionals consider to be a minimally acceptable ethical practice in that field, even as practiced by nonprofessionals. The Swiss Computer Code suggests that the responsibility of a national and/or professional Society is to be in charge of making the public aware of the Society's guidelines. It also advocates regularly publishing information about Code violations as a means of sensitizing the public about what is expected of a computing professional. As we shall see later some Codes also include responsibilities of the ICT Profession itself as well as about the responsibilities of the individual practitioner.

Two common problems for codes in computing are that they need to be able to address a rapidly changing environment and there are difficulties in enforcing them. Turning a code into law makes it static and eliminates some of the other important functions of Codes of Ethics.

Recent codes have become more specific about the ICT workers responsibility to society and responsibility to a broad range of stakeholders (Gotterbarn 1996). The Canadian Information Processing Society (CIPS) Code of Ethics and Standards of Conduct addresses a diverse audience. The CIPS code has imperatives for six audiences: Colleagues, Clients, Students, the Public, Myself, and the Employer and Management. By separating the client and the employer it avoids the possibility that the interests of the client and the employer may not be identical. This code and

others are based on the belief that a set of ethical obligations--professional ethics— based on responsibility to the client is based on the societal impact of the occupation. Because of the nature and impact of computing, a higher level of care is required. Consistent with this, many codes advocate the avoidance of negative consequences of professional activities. In the Association of Computing Machinery (ACM) Code there are general statements dealing with responsibilities in the event of negative consequences. For example Section 1.2 dealing with the responsibility for negative consequences says that a person is obligated to undo or mitigate negative consequences as much as possible. This is clearly a shift from earlier codes. The earlier codes were designed to protect the computing professional. This and numerous other sections are designed to protect society, almost in the sense of strict liability. It does not say that the member is obliged to undo the negative consequence only if they were due to a failure or malpractice of that member. Even if the negative consequence were the fault of the customer's incorrect use of a product, the member is still responsible. The code, at this and numerous other places, first protects society, and then the professional. The development of a computer system requires a consideration of all stakeholder's rights. For example, section 3.5 states "*Articulate and support policies that protect the dignity of users and others affected by a computing system.* Designing or implementing systems that deliberately or inadvertently demean individuals or groups is ethically unacceptable. Computer professionals who are in decision making positions should verify that systems are designed and implemented to protect personal privacy and enhance human dignity." The ACM and the IEEE have adopted a Software Engineering Code of Ethics and Professional Practice (SECEPP). This code has been adopted by numerous ICT professional organizations and like the other Codes above it clearly points to the developer's responsibility to a range of stakeholders which goes beyond the software developer and the client.

"These obligations are founded in the software engineer's humanity, in special care owed to people affected by the work of software engineers, and in the unique elements of the practice of software engineering. ... In all these judgments concern for the health, safety and welfare of the public is primary; that is, the "Public Interest" is central to this Code."

But as we have seen simply mentioning a broad range of ethical obligations does not seem to satisfy those who want to see the codes enforced with severe sanctions. Lockheed Martin enforced its ethics regulations and fired 28 software engineers for violating the ethics policy. The enforcement of the policy struck many as toothless since the fired software engineers will easily get other jobs and Lockheed Martin and their staff suffered because of a staff shortage. It sometime looks as if professional organizations actually do not want to enforce the code. There is a problem that some codes seem vague and are neither easily enforceable nor useful in making decisions. There is the already mentioned problem of jurisdiction only over an organizations membership. How can you enforce a Code on people who are not members of your organization and have not agreed to be bound by the Code? Professional organizations can leave their Codes as hollow statements or do something to help them have an impact.

Two major difficulties with Codes for Professional Ethics Committees.

We will trace the development of the ACM's Code of Ethics, a development which matches the most Code's development and also maps a difficulty with ICT development. The two major themes running through the Code's development are the questions concerning enforcement- by whom and how- and currency-- how address changing technology. (Berleur, 2008)

The early years: Ethical standards in search of identity

On November 11, 1966 the Association of Computing Machinery (ACM) adopted an ethics standard. The ACM Council adopted a set of guidelines called “Professional Conduct in Information Processing” (Parker 1968). The pattern of concerns and development of this ethics standard is similar to the patterns of many such developments. The 1966 discussion revolved around questions of: whether Information Processing was really a discipline, whether it was a single discipline which could have a single standard; what types of effective enforcement it could have, was it meaningful to merely expel miscreants from membership in the ACM, who would determine when to enforce the standard, and a narrowly US-Centric concern that the enforcement of a professional standard by the ACM might alter the ACM’s tax status as a scientific society. At least two trends from this early approach continue through the ACM’s development of ethical standards: there is recognition that the rapid and unanticipated changes in the profession will require modification of the ethical standards at some level and there is a recognition that agreement on enforcement is difficult to manage.

The 1966 standard handled both the enforcement and the change issue in the same way. The recognition of change was addressed by calling the first ethics document a “guideline.” The label was also intended to address the enforcement issue – “the ACM Council has wisely adopted ethical rules as a guide to members rather than a code to be enforced”. As a result there is no enforcement function directly related to the Code/Guidelines. Approaches to the issue of change have been constant while in the ACM there has been a significant change in the approaches to enforcement. The society’s means of addressing these issues of enforcement and technological change dictate the role of the ACM’s committees related to ethics.

1970-1992 from Guidelines to Standards and the need for enforcement

Just 4 years later there was a change to article 3 section 4 of the ACM constitution that stated ‘demonstrating a lack of integrity’ was a reason for being ‘admonished, suspended, or expelled’ and gave the authority to ACM Council to impose these sanctions. The amendment also mandated the development of a Code of Professional Ethics. This led to the development of a code with detailed ethical statements which are more amenable to enforcement, and whose violation is easier to determine. There was an obvious flaw in the implementation of this approach. A fundamental concept was overlooked. In order to obey the rules you need to know what they are. This is education problem is not unique to the ACM. According to Lucas and Weckert (2008), more than 30% of those questioned are unaware of the ACS Code. This is an education problem. Perhaps education needs to be emphasized more than enforcement.

There is also a conceptual problem with the approach. There is a basic problem with a precise checklist approach to ethics especially when it applies to a technical field. The computing field changes and advances very rapidly. What was considered best practice and mandated 20 years ago may be dangerous now. Early medical practices are now considered very dangerous. A precise ethics checklist, which is easy to enforce, is out of date almost as soon as it is off the press and what it advocates may be inconsistent with current best practice. This code was adopted in 1974; nevertheless an enforcement procedure was not approved until 1978 (Smoot, 1981) reflecting the continuing uneasiness about enforcement (Perspective). During this time the ACM did have a committee on ‘Professional Standards and Practices’ whose role included services to individual ACM members who face ethical problems such as whistle-blowing, product reliability and safety issues, and employment problems.

The adoption of the 1974 Code and later adoption of a policy designating the enforcement method as the sole responsibility of ACM Council was a significant change from the ACM's original ethics guideline. Addressing and enforcing ethical issues related to ACM members was now the sole responsibility of the governing organization of the ACM - the ACM Council. According to the ACM constitution article 6 section 8 "a member may be admonished, suspended or expelled for demonstrating lack of integrity" by a three fourths vote of Council after a hearing". This still left open the nature of the hearing.

1990 - The Emphasis on Guidelines returns

In 1990, adopting the insights from 1966 that ethical guidelines for computing need to change to address unanticipated changes in the profession, Ron Anderson proposed that the current ACM Code which had a structure that listed possible ethical violations needs to be revised. He further argued that the "ACM needs a revised organizational structure for an ongoing review, reformulation, interpretation, and application of its Code of Ethics and Professional Conduct."(Anderson, 1990) ACM Council supported this request and two years later, on 16 October 1992, a Code of Ethics and a suggestion for a revised review standard was presented.

This 1992 Code was developed over a two-year period, which included multiple drafts and reviews by ACM members. The ACM approved a new Code of Ethics which de-emphasized enforcement and emphasized education of members, of prospective members and of the public. The Code's use as an aid to decision making was also emphasized. The Code, which is still in use today, has a two level structure. It consists of 24 ethical imperatives each of which has an associated guideline illustrating the imperatives application in computing. The imperatives are divided into four sections. The first section gives a set of general moral considerations, the second identifies additional ethical principles applies to computing professionals, the third section pertains to organizational leaders, and the final section deals with issues of general compliance with the code.

It was envisioned that the high level imperatives would be constant and that the lower level clauses would require updating when technology and practices changed. In addition to the approval of the Code, a Committee on Professional Ethics (COPE) was also established to meet the need for revision and nurturing of the Code. The charge for the COPE Committee was to 1) promote ethical conduct among computing professionals by publicizing the Code of Ethics and by offering suggested interpretations of the Code; 2) plan and review activities to educate the membership in ethical decision making on issues of professional conduct; and 3) review and recommend updates, as necessary, to the Code of Ethics and Professional Conduct and its Guidelines.

These changes should have addressed the concerns of 1966 about enforcement and Code revision. The nature of the Code, emphasizing voluntary compliance and consisting of aspirational and normative imperatives rather than disciplinary imperatives helped reduce the concern about sanctions and enforcement. The structure of the Code with fairly constant imperatives and flexible guidelines helped address the state of flux of the computing profession.

COPE was a committee with a charge but without a structure. Many of the items described above were not addressed in the establishment of COPE. The president of the ACM appointed the chair of COPE. The Chair of COPE determines its structure and guides its activities. The outreach functions of COPE are fairly straightforward and modelled to some extent on the work when the Code was first passed. COPE members present papers and participate in computer ethics workshops. They write articles that offer interpretations of the Code. (Miller, 2003) Other professional societies, the

ACS and German Gesellschaft für Informatik for example, have used the original case studies developed when the 1992 ACM Code was passed.(Anderson, 1993) COPE is currently working on a specific set of examples related to Internet issues because the World Wide Web only achieved prominence after the general Code was approved in 1992.

The role of COPE extended in 1999 when the ACM and the Institute of Electrical and Electronics Engineers, Inc. (IEEE)-Computer Society jointly developed and adopted the Software Engineering Code of Ethics and Professional Practice as a standard for a sub-specialization of computing. COPE's domain now includes both of these Codes. COPE has helped with translations of the ACM's Codes by professional organizations that want to adopt these Codes. The Software Engineering Code so far has been translated into 9 languages. (SECEPP). Other computing organizations have adopted SECEPP. For example, in September 2006, the Association of Software Testing resolved to adopt the ACM Code of Ethics as a series of principles to guide and govern practice among its membership.(Software Testing) The Australian Computer Society (ACS) adopted the Software Engineering Code of Ethics.

In meeting its charge COPE also gets involved in design of posters of the Code of Ethics distributed to member organizations, works on the design of web pages, and includes a commitment to the ACM as a separate item on membership renewal forms. COPE has primarily limited its education function to the membership of the ACM and has only reached out in terms of getting the Code included in appropriate textbooks and conducting workshops a computing conferences. The original charge to COPE is merely a starting point. The absence of a fixed structure, including a regular schedule of meetings, has led COPE to function in response to external requests. The committee is one of the ethical focal points within the ACM. COPE members are asked to review many of the ethics articles submitted to the Communications of the ACM. It also responds to ethics complaints forwarded to it by ACM headquarters. These complaints vary from the trivial to very significant such as the development of a plagiarism policy that is consistent with the ACM's Codes of Ethics. In many cases committee members are not knowledgeable in the domain of the ethics problem and need to bring in other committees who have a better understanding of the situation.

The absence of a charge which involves the ethics committee in all such issues means that on occasion some very significant ethical issues do not have their ethical component addressed adequately. For example there was a significant issue raised by one ACM member – regarding religious discrimination – that was never brought to the attention of COPE. (Complaint) One of the reasons for the absence of such a charge is that COPE only serves in an advisory role. Without a clearly defined structure it is sometimes difficult for COPE to achieve its goals. Another problem is the separation of primarily proactive and advisory functions given to COPE, and enforcement functions given to the ACM Council. Often this structure may contribute to a perception that the role of COPE is less important.

On one hand, the lack of a clearly defined structure makes it difficult at times to achieve its goals. On the other, the absence of a defined structure has the virtue that when an unanticipated issue arises, such as the relation of the Code to a Plagiarism Policy, COPE can be involved in those situations without having to wait for a formal meeting.

Every professional organization ought to have an Ethics committee which promotes the Code of ethics. There are several things required for such a committee to function effectively:

1. When an ethics committee is established both its charges and structure should be specified.
2. All ethics issues should be passed through the ethics committee.

3. The method of updating a Code of Ethics needs to be clearly defined by either the national/professional Society or by its ethics committee. This method should be as rigorous and cautiously entered into as the original creation of the Code.
4. The structure of the ethics committee should not limit the issues it can address.
5. Ethics committees should have a regular venue in the society's publications to help promote a proactive approach to ethics. This should be accompanied by an annual ethics award included in a National/Professional Societies repertoire of awards.

The Ethics Committee and ICT Governance and the Software Crisis

An ethics committee promoting the Code of Ethics amongst the ICT community is a help in addressing the three common difficulties identified in ICT Governance and in addressing the software crisis. A major role of the ethics committees is public education. The Codes of Ethics all focus on the professional's responsibility to a broad range of stakeholders. This responsibility to other stakeholders is paramount in the codes. This lessens the significance of the controlling influence of the discussion of 'who owns the system' and raises to the forefront the discussion of what are the consequences of the system for all concerned. The breadth of concern makes clear that single mode solutions are inadequate for systems with broad stakeholder responsibility. A Code advocates a quality of action and not a particular technology. A constant reminder of the social and ethical impacts of ICT systems makes clear that simply focusing on inline documentation in a software program is an irresponsible approach to building an electronic voting system.

BUT CODES HAVE NO TEETH

Even though the concerns about the constant flux of the computing profession and enforcement of a Code's imperatives have been addressed by modifications to the structures of the Codes of Ethics there remains the underlying concern of how to get everyone to follow the Code. Codes are not the teeth of an organization. They do not contain the due process and sanctions within an organization but they do describe the conscience of a profession.

Within organizations there is generally limited enforcement of a Code of Ethics. Codes get some teeth when they are used by a professional organization to make decisions. The ACM Code of Ethics was used in developing the ACM Policy and Procedures on Plagiarism based on "Respecting intellectual property rights is a foundational principle of the ACM's Codes of Ethics". The ACM publications board then clearly defined the procedure for reporting alleged plagiarism, the procedure for investigation and how confidentiality during the investigation will be managed. If the paper has already been published the "ACM will post a Notice of Plagiarism based on the investigation on the ACM Digital Library's citation page of the plagiarizing paper and will remove access to the full text." This give the plagiarism charge some teeth. Codes of Ethics do not have teeth. They do not define the discipline for violating the Code. Codes are not self-referential, Organizations have bylaws and the code is a bylaw. The due process and sanctions for violating the code is defined outside the Code. Codes do not have teeth, they are the mind and conscience of a profession. The profession is what gives the rules nurture and teeth.

The original concerns of the ACS member about those outside the ACS not being bound by the standards remain. In a recent case major sections of a student's master thesis were copied and submitted to a conference and happened to be passed to a reviewer who was the original author's thesis supervisor. The conference was not an ACM event, the plagiarizer was not an ACM members

so the processes defined by the ACM did not apply. But that does not prevent the Ethics Committee from informing the reviewer and the conference about how the ACM understands professional standards and the action they would take with members.

CODE OF ETHICS IMPOSE ACCOUNTABILITY ON THE PROFESSIONAL

ORGANIZATION

A code of ethics is like a Swiss Army knife, serving many important and useful functions. It is a statement to members about the ethical stand of an organization and profession, a conscience of the profession, an announcement to non-members what the profession standards for (although most often stated in terms of the actions of individuals), it imposes functions on an ethics committee to educate the membership, and it imposes responsibilities on the professional organization itself.

One of the functions of an ethics committee is to help its own professional organization understand the importance of the role played by ethical standards. Sometimes professional organizations lose sight of these responsibilities or get distracted. In 1972 the IEEE set up an ethics task force in response to the BART disaster and because the committee was not publicized, they had no cases until 1978 when it was publicized and they had 11 cases to address. In 1990 they set up hotlines and sent copies of the code out with membership renewals. With each renewal for the ACM members sign that they will adhere to the ACM Code. The IEEE hotline provided a direct channel for IEEE members to get help when they were faced with painful professional dilemmas. For a variety of reasons the hotline was discontinued.

In 2003 the ACS established the Committee on Computer Ethics (CCE) to promote the development of Computer Ethics policies (in Australia). They are charged to work with ACS SIGS, help develop policies for government, and promote the importance of computer ethics in the ICT community.

MICRO- MACRO ETHICS CONFUSION

The distinction between macro and micro ethics is important for these committees and for the use of Codes of Ethics in addressing the ICT Governance/Software Crisis problem. The lack of attention to this distinction is a problem that pervades many codes of ethics, ethics committees and professional organizations. Generally they all focus on the responsibility of the individual computing professional.

The general view is that Codes and regulations are about the behavior of the individual member and not about the organization nor of the profession as a whole. Micro focused Codes of ethics talk about 'You' and not specifically about the professional organization or about the profession.

The complaints about an absence of sanctions view the code as being primarily about the ICT person's behavior. 23% of public sector workers in Australia did not see their work as related to a larger whole (Lucas, 2008). It is the ICT individual who is asked to be ethical. This narrow understanding of the scope of a Code of ethics and professional standards is a problem. This view of the scope of a code of ethics affects their work. They "view themselves as NOT part of other systems but as separate ICT folk" (Lucas, 2008, p 23). If micro-ethics is the focus then enforcement

is the primary consideration and the professional loses sight of the contribution 'doing it right' makes to the quality of life.

Unfortunately few codes have sections dealing with macro ethics. Sometimes it is not clear in an ethics committee's terms of reference. The terms of reference for the ACS Committee on Computer Ethics has it right when it says they should promote "the value and importance of Computer Ethics within the wider Australian, as well as regional and international, ICT community". The standards are not just for ACS members but for the ICT community at large. They also recognize the responsibility to society as a whole in the charge "to advise the Society and the ICT community on 'best practice' in relation to Computer Ethics;"

This awareness and promoting of the professional organization's responsibility to the whole profession, not just to ACS members addresses the whole software development standard. It shows a major function of professional ethical standards beyond the punishment of miscreants. This Macro understanding justifies an approach to encourage adherence to the Code. If the standards become common knowledge, the public knows what ought to be done then those who do not follow the standards will receive less business, provided the failure to follow the standards is made known as in the Swiss model. For example, there was some concern about the lead content in the paint of some children's toys sold in the USA. Citizens were made aware a) of the danger of the levels of lead and b) that toys exceed these levels of lead. In spite of a lack of government testing on all toys, when the knowledge spread that X toy was below this safety standard the toys are no longer purchased.

The ethics committees need to clarify and formally state principles important to the profession as a whole. This need was supported by the comments in Lucas and Weckert (2008) which indicate the importance of training, but the commenters see ethics solely as an individual responsibility:

- any other comments- ethics- Punitive measures [when asked about them] are also useful but you can't take action unless you communicate your expectations. The message is: communicate your expectations, police them, and then maintain them. If there are no consequences people will not be motivated to behave ethically. So you need to put the sign post up (i.e. 80 km/h), communicate it and police it.
- Needs to be incentives, and the value of ethics for the business to be recognized.
- Organizations need to communicate their ethical expectations from staff and then ensure that they are aware of them. Management should encourage staff to take their ethics training so that they do not have an excuse when it comes to expectations.
- Needs to be more promotion of ethics. Frequently overlooked. Essentially it is up to individual, but training helps.

CONCLUSION

Codes of ethics and regulations can have teeth. If Professional organizations are committed to elevating professional practice and truly committed to the standards in their codes then they should publish a list of expelled miscreants who violated the standards. Maintaining silence in the light of such violations is inconsistent with the content of the Codes of Ethics. The ACS takes a different view. ACS reprobates are not named but "describe[ed] in general terms (ie, to protect the privacy of

those involved) the breaches that have occurred and the actions that have been taken by the ACS in respect of those breaches.” The important stakeholder is not the reprobate. The important stakeholders are everyone affected by that persons actions and the profession which maintains silence about unprofessional actions.

Professional committee need to be aware of:

1. A focus on micro ethics contributes to the single mode solutions and who owns the system problems. It also is used to justify ignoring a sense of responsibility for the whole project, for the whole profession.
2. The worst model of leadership is the whip. The committees must educate about the what and the why of regulations.
3. The rules are guidelines with a purpose. Codes and ICT Governance are not checklists which will be out of date shortly after they are written.
4. There needs to be as much if not more emphasis and financial support for education than on enforcement. Enforcement/encouragement can take a variety of forms.
5. Take the Codes seriously-publish the list of those ejected for Violation of Code of ethics. Be the Better Business Bureau for software quality and list if there are open complaints.

Codes of Ethics and regulations may be toothless tigers, but they can still be heard. Their message is important to address the problems at the interface of ICT Governance.

REFERENCES

ACM Policy and Procedures on Plagiarism

<http://www.acm.org/pubs/plagiarism%20policy.html>

ACS See <http://www.acs.org.au/news/060404.htm>

ACS Procedure <http://www.acs.org.au/index.cfm?action=show&conID=memberdisciplining>

Agle, B.R., & Mitchell, R.K., Sonnenfeld, J.A. (1999). “Academy of Who matters to CEOs? An investigation of stakeholder attributes and salience, corporate performance, and CEO values.” *Management Journal*: 42(5) October, 1999: 507-525.

AIS (2005) “Definition of System quality” – http://business.clemson.edu/ISE/html/system_quality.html 2005

Anderson, R. (1990) “A Rationale for the Proposed Revision of the Association for Computing Machinery’s Code of Professional Conduct,”http://www.southernct.edu/organizations/rcs/resources/research/comp_and_priv/anderson/background.html

Anderson, R. (1995) “The ACM Code of Ethics: History, Process, and Implications,” *Social Issues in Computing*, McGraw-Hill, New York, , pp. 48–72.

Anderson, R; Johnson, D.; Gotterbarn, D.; Perrolle, J; (1993) “Using the New ACM Code of Ethics in Decision Making”, *Communications of the ACM* 3, 2 February 1993

AS/NZS (1999) “Risk Management, Standard 4360-1999”, Standards Australia, Standards New Zealand

- AS8015-2005 "Corporate governance of information and communication technology.", Standards Australia Committee IT-030, ICT Governance and Management in 2005
- Berleur, J (2008) Ethics of Computing Committees: Suggestions for Functions, Form and Structure, To Promote Discussion Inside the IFIP National Societies, On behalf of IFIP-SIG9.2.2: Jacques Berleur, Oliver Burmeister, Penny Duquenoy, Don Gotterbarn, Philippe Goujon, Kari Kaipainen, Kai Kimppa, Benjamin Six, Debora Weber-Wulff, Diane Whitehouse, Eds., IFIP Press, Laxenburg-Austria, 2008, 32 p., ISBN 978 3 901882 24 3. The account of the ACM code of Ethics development is based on my report to this group.
- Berleur, J. & d'Udekem-Gevers, (1994) "Codes of Ethics, or of Conduct, Within IFIP and in Other Computer Societies", 13th World Computer Congress 1994 pages 340-348.
- Bowern, M. (2006) "Ethics" Information Age 14/02/2006 13:21:16 <http://www.infoage.idg.com.au/index.php/id;1260028245;fp;131072;fpid;0>
- The British Computer Society documentation on its Code of Conduct and Disciplinary Procedures can be found from a keyword search at : www.bcs.org
- Complaint, "My Complaint against the ACM - A leading technological society condones employment discrimination against some of its own members. Copyright © 2001-2003 by David E. Ross, <http://www.rossde.com/acm.html>
- Cockburn, A. (2004) Crystal Clear, Addison-Wesley,
- Dijkstra, Edsger W. (1976), A discipline of programming, Prentice-Hall, Englewood Cliffs, N.J
- Gotterbarn, D. (1996) "Software Engineering: The New Professionalism," The Professional Software Engineer, C. Myer, ed., Springer-Verlag, New York.
- Gotterbarn, D., K. Miller and S. Rogerson, (1998) "Software Engineering Code of Ethics", [online: available, <http://www.acm.org/serving/se/code.htm> accessed 1/27/2007] Communications of the ACM.
- Halstead, Maurice H. (1977) Elements of Software Science, Operating, and Programming Systems Series Volume 7. New York, NY: Elsevier.
- Highsmith, J. (2002) Agile Software Development Ecosystems Boston: Addison-Wesley.
- Lucas, R. and Weckert, J. (2008) "ETGOVICT 2008 Conference, Briefing Paper for Presenters" Centre for Applied Philosophy and Public Ethics, Australian National University ©2008
- Martin, M. W. et al., Ethics in Engineering, 2nd ed., McGraw-Hill, New York, 1989.
- Miller, K. Gotterbarn, D., "Computer Ethics in the Undergraduate Curriculum: Case Studies and the Joint Software Engineer's Code." Small College Computing Conference Journal 2003
- Parker, D. (1968) "Professional Conduct in Information Processing," Communications of the ACM, (11) 3 March
- Paulk, Mark C. (1995) The Capability Maturity Model: Guidelines for Improving the Software Process. Reading, MA: Addison-Wesley Publishing Company,
- Pediatrics (2006) accessed 1/25/2007 <http://pediatrics.aappublications.org/cgi/content/abstract/118/5/1872>,
- Perspectives on the Professions, Centre for the Study of Ethics in the professions at Illinois Institute of Technology, http://ethics.iit.edu/perspective/pers1_1mar81_3.html

- Ravichandran, T. (2000) "Total Quality Management In Information Systems Development: Key Constructs And Relationships" *Journal of Management Information Systems*, 1999/ 2000, 16 (3) ,p119,37p
- SECEPP <http://seeri.etsu.edu/Codes/TheSECode.html>
- Smoot, http://ethics.iit.edu/perspective/pers1_1mar81_3.html
- Software Testing, read Resolution on adoption of the ACM Code of Ethics <http://www.associationforsoftwaretesting.org/ethics.html>
- Weill, P & Ross, J.W. (2005) "How effective is your IT Governance," <http://mitslsoan.mit.edu/cisr/pdf/Weill-Ross%20-%20Effective%20IT%20Governance-final.pdf>