13. ICT governance and what to do about the toothless tiger(s): Professional organisations and codes of ethics[1]

Don Gotterbarn
East Tennessee State University

Introduction

Information and communications technology (ICT) is infamous for unfortunate incidents in planning, development, and delivery. A typical response to these incidents is to both complain about the toothless tiger of technical and professional standards that are not enforced, or enforceable, 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 approaches to ICT governance.

The purpose of this paper is to define strategies, which professional organisations can use to meet their responsibilities to the ICT profession and the ICT professional; strategies that move toward regulation without curtailing ICT's potential with ineffective sanctions. Professional organisations also need strategies for reducing negative incidents and for improving professional responsibility without the introduction of sanctions that apply only to practitioners who happen to be members of that organisation. There are ways in which the toothless tiger(s) can have a significant positive influence.

Outline

ICT has been with us for many years and, in the past 10 years, there has been a growing interest in ICT governance as a means of reducing information system disasters. National organisations have been formed, professional organisations have organised subcommittees to address ICT governance, and ICT has even been

1 A version of this chapter was published in *The Australasian Journal of Information Systems*, 2009, vol 16, pp 165–84. Published here with permission.

called a 'discipline'. The unwrapping of the concept of ICT governance results in 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 is sometimes 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 organisations. 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 organisations can address the weakness of ICT governance using tools that they already have at hand.

The problem from the software perspective

In the early days of computing, the 1960s, people worried about 'the software crisis' — a term coined in 1968 — or the failure of software systems. Software workers addressed this possibility 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 during which mathematical modelling of software development was published in books on The Elements of Software Science (Halstead, 1977) and The Discipline of Programming (Dijkstra, 1976). The focus was on making computing a reliable engineering-like discipline and the impacts and concerns addressed were local to the particular system being developed.

The response to the perceived 'software crisis' generated and continues to generate many single-mode solutions that suggest undertaking one particular process will solve all of the perceived problemshe particular single mode solutions trie shifted from emulating an engineering approach to the development of software, then to a structured approach to program design, to a formal proof of software system requirements, then to 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 have been regularly interspersed with approaches that emphasise measuring 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 the approaches adopted by the software community. First, the software crisis is a complex problem and single-

mode approaches overemphasise 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 rapid rate. This means the domain of the software crisis is also constantly changing; developments, such as computerised robotic surgery, were not issues of concern in the 1970s. 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 increased correspondingly. The third problem is that the solution to the problem of software interaction with business and society has been addressed only from the software side of the problem. These single-mode approaches are focused on what software developers do. Since 1968, the answer has been the same, worded differently — 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. It took many years, for example, for the world economy to recover from the negative effect of programmed trading on the stock market in October 1987.

There are numerous standards organisations, such as the American National Standards Institute (ANSI), the Institute of Electrical and Electronics Engineers Inc. (IEEE), the International Organization for Standardization (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, however, makes the application of these rules to software development difficult.

- Since there is no required common education program for ICT workers, many have not been taught the standards/rules.
- Since there are few sanctions and minimal oversight, and no threat of losing an ICT license, obeying the rules is voluntary and it is not always clear what is to be gained by following them.
- Since the ICT environment is changing so rapidly, rules either no longer apply by the time they are written, or they are inconsistent because they address different sides of an issue. Who, for example, enforces fraud legislation on international internet sales?

The solution to solving the software crisis has been seen as internal and focused 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, somehow, understood all of the important business issues and addressed them in the software. This systemic disconnection of developer and customer led to the delivery of working systems that had nothing to do with the needs of the customer. Since corporate executives did not fully understand how ICT worked, when a corporate manager complained about the style or function of the delivered system, ICT 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 anywhere there are discrete project elements, and no communication 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 that 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.

History repeats itself

In the development of ICT governance, I think history is repeating itself. A similar set of mistakes is being made in the way in which the industry is attempting to address the issues of ICT governance as those that it made in facing the problems that confronted ICT systems. Many areas in ICT governance are taking a single-mode approach, which focuses primarily on the industry's side of the problem, on the business element; thereby minimising the ICT elements and giving no consideration to the rapid change of the domains of ICT and business.

ICT governance

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

In 2003 Australia published several standards for corporate governance including: 'Good governance principles' (AS8000), 'Fraud and corruption control' (AS8001), 'Organisational codes of conduct' (AS8002), 'Corporate social responsibility' (AS8003), 'Whistle-blower protection programs' (AS8004), thus setting the pattern for a document from the corporate side addressing ICT governance. Standards, such as AS8015 (2005), were developed to deal with issues of in-house development and the fact that outsourcing to benefit the vendor's interests was not always consistent with those of a user organisation. This is outsourcing problem is similar to one of the problems of over-the-wall software development.

The interest in ICT 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 organisations 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.38500.org/>

As the multitude of standards illustrates, there is no single standard of ICT governance, and nor is there a consistent approach to the appropriate stage at which to implement ICT governance principles. 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. On the other hand, Peter Weill and Jeanne Ross p.14 (2004), in a study of 300 enterprises around the world, 'did not identify a single best formula for governing IT'. They say that 'IT governance specifies accountabilities for IT-related business governance and helps companies align their IT investments with their business priorities.' IT governance performance for them involves the corporation's 'Cost effective use of IT, IT for growth, IT for asset utilisation and, IT for business flexibility.' They argue that 'IT governance is the decision rights and accountability framework for encouraging desirable behavior in the use of IT.' This is very different from viewing IT governance as managing the interface between ICT development and corporate management.

Sense of 'control'

But, if we follow 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 in the current and future corporate use of ICT. This same

diversity exists in the standards being developed by the ISO. There is also a difference in what is meant by 'control'. Control is characterised both as rigorous highly delineated control, and general directional guidance principles to help the professional make judgements.

Single mode

As in software development, ICT governance takes a single-mode-solution approach, and there are a variety of single modes. There are also 'structuralists' who think everything is handled by structure and that the primary goals for ICT governance are to ensure that ICT generates business 'value' and reduces the risks that are associated with ICT by implementing a carefully defined organisational structure. This is sometimes connected with who is in charge or has ownership of the system; again, 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 organisation to ICT management. It mandates that ICT decisions are owned by the corporate board, rather than by ICT managers. This results in the same problems as over-the-wall development. There are indeed limitations to what ICT can do and the ways in which it does things will have different effects. Balanced ICT governance needs the ICT side in their systems guided by ICT workers.

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, namely, there is a limited view of who constitutes the stakeholders in a project. Because the view of who constitutes the stakeholders in a project is limited to the developer and the customer, the effect of an IT system on a business or on extended stakeholders is not considered.

The current concept of ICT governance is modelled on the traditional concepts of business ethics regarding who needs to be considered as the relevant stakeholders, namely those who have some financial interest in the business (Agle et al, 1999). The current concept of ICT governance stakeholders is 'IT 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 view of the system

context and the stakeholders in that context is also supported by ISO standard 38500 ''ICT Governance which characterises ICT governance as the management system used by directors. 'IT governance is about the stewardship of IT resources on behalf of the stakeholders who expect a return from their investment.'

The briefing paper on a recent survey of ethical issues conducted by the Centre for Applied Philosophy and Public Ethics (CAPPE) (Lucas, 2008) the limited understanding of ICT workers about 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 the concept of control. The use of words like 'control' and 'govern' imply enforcement of rules or sanctions for failure to follow them, 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 behaviour 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, 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. Rather, they are complaints about the fairness of their being 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 realise 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 perceived 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

ICT governance is repeating some of the same mistakes 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 focus on a limited set of stakeholders in the system: developer, customer, sponsor, and vendor, and pay limited attention to those who will be impacted by the deployment of the software. ICT governance, likewise, has an internal focus on the business side of the software system and a narrow a view of the stakeholders as those with a financial interest in the system (Agle et al, 1999; Weill & Ross, 2005). We can see some of the consequences of this narrow approach by looking at the ICT treatment of 'software risk'.

The narrow stakeholder focus in software risk

Although the need for high quality software is obvious to all, 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 moderate the risk of developing software products that do not meet the needs of clients. The risks include: missed schedule, going 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 delivered late, over budget, and do not meet requirements, leading to software development still being characterized as a ''software crisis'' and leading to a general mistrust of software systems.

Risk management generally consists of an iterative series of steps, similar to 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 organisational 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 4360-1999 lists potential negative areas of impact such as

> Asset and resource base of the organisation, Revenue and entitlements, Costs, Performance, Timing and schedule of activities, and Organisational behaviour.' (AS/NZS, 1999: 39)

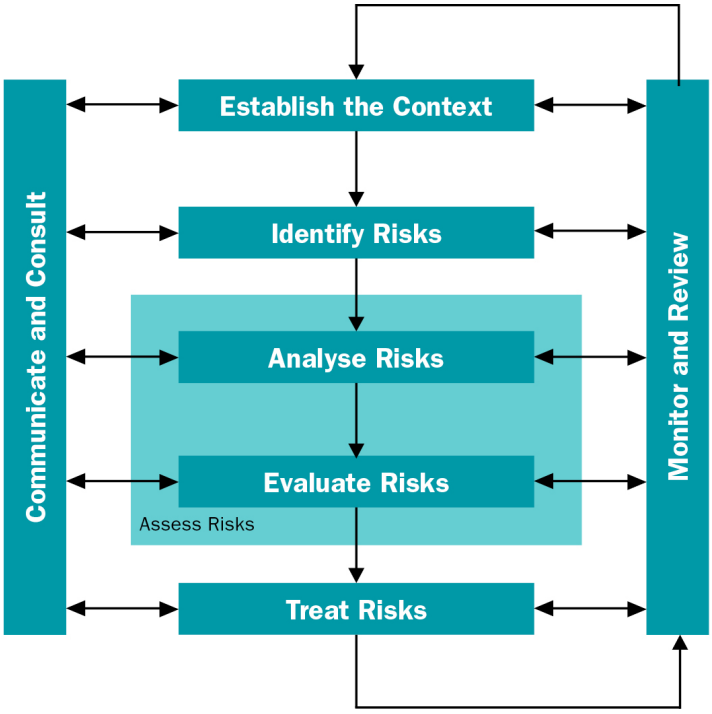

Figure 1: Risk management

Source: AS/NZS, 1999, 16

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; for example, 'A delay of one day will cost $3000 in sales'. The second method, qualitative risk analysis, is often used to address risks which are not readily quantifiable, other than by describing the broad degree of risk; for example, '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: 14). Its primary role is to characterise 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 prioritised and facilitates analysis. These standard methods of risk identification and risk characterisation have been used extensively in software projects.

The Association for Information Systems (AIS) 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 and 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 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. For example, the system that was used to record dosages of paediatric medicine correctly handles negative interactions of dosages, but was awkward to use in emergency situations, resulting in three medication errors out of every 100. (Walsh, 2006)

Contributing factors

Two interrelated factors related to system stakeholders contribute to these professional and ethical failures being overlooked. 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 realised that the focus on technical risks is too narrow, but, unfortunately, the risk focus only expands to other internal issues that are related to the development of the system. For example, Thiagarajan Ravichandran writes 'Research in software quality has focused largely on the technical aspects of quality improvement, while limited attention has been paid to the organisational and socio-behavioural aspects of quality management' (p. 119, 2000).

A second factor arises from 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 the particular risks including that they perceive as a threat to a project, such as budgets, timelines and suitability of the product. This focus may mean that other critical aspects of the product, such as the use of easy-to-read fonts or back-up systems, are not given adequate consideration.. 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 produced at a 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, thus hurting patients, would have a negative effect; just as a pay-phone system which disabled all usage, including emergency numbers, without an approved credit card, would also have a negative effect. These stakeholders, patient and someone hurt in a fire, are not normally considered.

The scope of a project must identify all stakeholders to eliminate the possibility of negative effects.

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 organisation and the development organisation 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

The response of software developers to the ICT crises was internal and focused 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 organisation; '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 organisations

Professional organisations 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 organisations have the ability to resolve this. One of the roles a professional organisation needs to play in resolving these problems is to broaden the range of stakeholders considered in the current concept of ICT governance. ICT professional organisations have codes of ethics and codes of practice that address each of the ICT governance problems discussed above. Their codes of ethics require the ICT professional to consider a broad range of stakeholders, including all those whose lives are affected by the ICT project and the way in which 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 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 organisations.

Functions of codes

Codes of ethics are developed by professional organisations for a variety of reasons. They serve 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 behaviour 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, sensitising members to new issues, and providing guidance to individual members when they are confronted with ethical situations.

Codes are also used to win public confidence and stave off external regulation. Some codes 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 the profession and socialise novices in field. Codes can express and strengthen the community orientation of the group. The general nature of some codes makes it difficult for the general practitioner to apply their directives in concrete cases.

Some confusion about codes of ethics arises from a failure to distinguish between closely related concepts about codes which direct the behaviour of practicing professions. The less restrictive codes can be primarily aspirational, in that they provide a mission statement for the profession. There are also codes of conduct which describe professional attitudes and some professional behaviour. Codes of practice are specific and closely tied 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).

The goals of a code 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 for 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. Protection of the status quo by stifling dissent and state only minimal levels of ethical conduct.
6. Promotion of 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 Information Society Code of EthicsComputer Code (SIS Code 2005) 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 informing the public about what is to be expected of a computing professional. Some codes also include the responsibilities of the ICT profession itself.

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 ICT workers' responsibility to society and 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 starts from the belief that a set of ethical obligations — professional ethics — is in part based on the high social impact of the profession; because of the broad and significant impacts of computing the computing professional owes a higher order of care to their clients. 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, which deals with the responsibility for negative consequences, states that a person is obligated to undo or mitigate negative consequences as much as possible. This is clearly a shift from earlier codes, which were designed to protect the computing professional. These, and other sections of the ACM code, are designed to protect society. Some codes limit corrective responsibility to merely fixing one's own mistakes. In the ACM code, however, even if the negative consequence were the fault of the customer's incorrect use of a product, the member is still responsible. The code first protects society and then the professional on the basis that 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 organisations and it clearly points to the developer's responsibility to a range of stakeholders. The Preamble states

> 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 satisfy those who want to see codes enforced by severe sanctions. A Code such a SECEPP can still be enforced even if the Code does not include mention of specifice sanctions. A large aerospace firm 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 the firm and their staff suffered because of a staff shortage. It sometime looks as if professional organisations 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 organisations membership. How can you

enforce a code on people who are not members of your organisation and have not agreed to be bound by the code? Professional organisations 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

It is useful to trace the development of the ACM's code of ethics because it is similar to the pattern occurring in the development of most codes and it maps a difficulty with ICT governance. The two major themes running through the code's development are the questions concerning enforcement — by whom and how — and currency — how to address changing technology (Berleur, 2008).

The early years: Ethical standards in search of identity

On 11 November 1966, the ACM adopted an ethics standard. They 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 that 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 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 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.

1970–92: From guidelines to standards and the need for enforcement

Just four years later, a change was made 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 authority to the 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, the violation of which is easier to determine and which are easier to enforce.

There was an obvious flaw in the implementation of this approach: in order to obey the rules, you need to know what they are. This education problem is not unique to the ACM. According to Lucas (2008), more than 30 per cent of those questioned are unaware of the ACS code. Perhaps it is education that needs to be emphasised, more than enforcement.

There is also a conceptual problem with the approach. A basic problem with a precise checklist-approach to ethics arises particularly when it is applied to a technical field. The computing field changes and advances very rapidly. What was considered best practice 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.

The ACM code was adopted in 1974, but an enforcement procedure was not approved until 1978 (Smoot, 1981), reflecting the continuing uneasiness about enforcement (Perspective, 1981). During this time, the ACM's committee on 'Professional standards and practices' was responsible for services to individual ACM members who faced ethical problems such as whistleblowing, 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 organisation 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 a code like 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 organisational structure for an ongoing review, reformulation, interpretation, and application of its Code of Ethics and Professional Conduct'(1990). The 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, during which there were multiple drafts and reviews by ACM members. The ACM approved a new code of ethics which de-emphasised enforcement and emphasised education of members, of prospective members and of the public. The code's use as an aid to decision-making was also emphasised. 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 application of the imperative 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 which apply to computing professionals, the third section pertains to organisational 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 publicising 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 1966 concerns about enforcement and code revision. The nature of the code, emphasising 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 within 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, who determines its structure and guides its activities. The outreach functions of COPE are straightforward and modelled to some extent on the work outlined 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 IEEE–Computer Society jointly developed and adopted the SECEPP as a standard for a sub-specialisation of computing. COPE's domain now includes both of these codes.

COPE has helped with the translations of the ACM's codes by professional organisations that want to adopt them. The SECEPP has so far has been translated into nine languages.

Other computing organisations 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. The ACS has also adopted the SECEPP (ACS, 2004).

In meeting its charge, COPE is also involved in the design of posters of the code of ethics, which are distributed to member organisations, 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 conducing 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 that are 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 COPE in all such issues means that on occasion some very significant complaints and ethical issues do not have their ethical component addressed adequately. For example, there was a

significant issue raised by one ACM member on his website — regarding religious discrimination — that was never brought to the attention of COPE (Complaint, 2003). One of the reasons for the omission 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 the primarily proactive and advisory functions given to COPE, and the enforcement functions that are given to the ACM council. Often, this structure contributes 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 hand, 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 organisation ought to have an ethics committee for the promotion of 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 that is included in a national/professional society's 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 helpful in addressing the three common difficulties identified in ICT governance, and in addressing the software crisis. A major role of the ethics committee is

public education. The codes of ethics focus on the professional's responsibility to a broad range of stakeholders. This broad focus lessens the significance of the controlling influence of the discussion of 'who owns the system' and emphasises the question of the system's consequences, 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 something with the impact of 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 organisation. They do not contain the due process and sanctions within an organisation, but they do describe the conscience of a profession.

Within organisations generally there is limited enforcement of a code of ethics. Codes get some teeth when they are used by a professional organisation to make decisions. The ACM code of ethics, for example, was used in developing the ACM policy and procedures on plagiarism based on the imperative that 'Respecting intellectual property rights is a foundational principle of the ACM's Codes of Ethics'. The ACM publications board defined the procedure for reporting alleged plagiarism, for investigating the allegation, and managing confidentiality during the investigation. If the offending 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 plagiarising paper and will remove access to the full text.' This response gives teeth to the charge of plagiarism.

Codes of ethics do not have teeth and they do not define the disciplinary action for a code violation. Codes are not self-referential: organisations have bylaws and the code is a bylaw. The due process and sanctions for violating the code is defined outside the code. Codes are the mind and conscience of a profession. The profession is what nutures the code and gives the code 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 masters thesis were copied and submitted to a conference where the original author's supervisor was present. The conference was not an ACM-sponsored event and the plagiariser was not an ACM member, so ACM processes for dealing with plagiarism 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.

Codes of ethics impose accountability on the professional organisation

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 organisation and profession, a conscience of the profession, an announcement to non-members of 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 organisation itself.

One of the functions of an ethics committee is to help its own professional organisation understand the importance of the role played by ethical standards. Sometimes, professional organisations lose sight of these responsibilities or get distracted. In 1972 the IEEE set up an ethics task force in response to the fireing of engineers reporting ethical problems in the development of the Bay Area Rapid Tansit system. Because the existence of the committee was not publicised, no case was referred to it until 1978 when, following advice of its existence being sent to the membership, it received notice of 11 cases to address. In 1990 the IEEE set up hotlines and sent copies of the code out with membership renewals. With each renewal, ACM the members agree to 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. Unfortunately, and 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). The committee is charged with working with ACS Special Interest Groups, to help develop policies for government, and to 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 organisations. Generally, they all focus on the responsibility of the individual computing professional.

The general view is that codes and regulations are about the behaviour of the individual member and not about the organisation, or the profession as a whole. Micro-focused codes of ethics talk about 'You' and not specifically about the professional organisation or about the profession.

Complainants about an absence of sanctions tend to view codes as being primarily about the ICT person's behaviour. Of public sector workers in Australia, 23 per cent 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 and affects their work. Individuals who 'view themselves as NOT part of other systems but as separate ICT folk' (Lucas, 2008: 23) are taking a an exteme form of micro-ethics approach where enforcement of a narrow set of rules is primary and they lose sight of the positive contributions of ethical development, lose sight of the contribution 'doing it right' makes to the quality of life.

Unfortunately, few codes have sections dealing with macro ethics. Sometimes the need for a macro consideration is not clear in an ethics committee's terms of reference. The terms of reference for the ACS Committee on Compuiter Ethics is correct in asserting that it 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 recognise 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 promotion of the responsibility to the whole profession, not just to ACS members, addresses the software development standard. It shows a major function of professional ethical standards being 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 and 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 those principles that are important to the profession as a whole. This need was supported by the comments in Lucas (P.57) which indicate the importance of training, but the commentators see ethics solely as an individual responsibility

> 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 recognised.
>
> Organisations 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 organisations are committed to elevating professional practice, and to the standards in their codes, then they should publish a list of expelled miscreants who violate those 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 (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 person's actions, and the profession which maintains silence about unprofessional actions.

Committees need to be aware of:

1. A focus on micro ethics contributes to single-mode solutions and mistakenly ignoring obligations to a broad range of stakeholders. 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 on and financial support for education thenis given to methods of enforcement. Enforcement/ encouragement can take a variety of forms.
5. Take the codes seriously — publish the list of those ejected for violation of codes 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

ACS , 2004 *Image* See <http://www.acs.org.au/news/060404.htm> accessed 2008.

Agle, BR, Mitchell, RK & Sonnenfeld, JA, 1999,', Academy of *Management Journal*, vol 42, no 5, October, pp 507–25.

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/rccs/oldsite/resources/research/comp_and_priv/anderson/references.html>

——, 1995, 'The ACM code of ethics: history, process, and implications,' *Social issues in computing*, McGraw-Hill, New York, pp 48–72.

——, Johnson, D, Gotterbarn, D, & Perrolle, J, 1993, 'Using the new ACM code of ethics in decision making', *Communications of the ACM* vol 3, 2 February.

Association for Computing Machinery (ACM), 2006, 'Policy and procedures on plagiarism', revised 2010, <http://www.acm.org/publications/policies/plagiarism_policy?searchterm=Policy+and+Procedures+on+Plagiarism>

Berleur, J, 2008, *Ethics of computing committees: suggestions for functions, form and structure: To Promote Discussion Inside the IFIP National Societies*, Berleur, J., Burmeister, O., Duquenoy, P; Gotterbarn, D. ; Goujon, P, Kaipainen, K; Kimppa, K, Six, B.; Weber-Wulff, D.; Whitehouse, D. (Eds)., IFIP Press, Laxenburg-Austria, 2008, ISBN 978 3 901882 24 3. The account of the ACM Code of Ethics development is based on my report to this group.

—— & d'Udekem-Gevers, M 1994, 'Codes of ethics, or of conduct, within IFIP and in other computer societies', *13th World Computer Congress*, pp 340–48.

Bowern, M, 2006, 'Ethics: part of being a professional', *Information Age*, June/July p 51-52.

Cockburn, A, 2004, *Crystal clear*, Addison-Wesley.

Dijkstra, EW, 1976, *A discipline of programming*, Prentice-Hall, Englewood Cliffs, NJ.

Gotterbarn, D, 1996, 'Software engineering: the new professionalism', in C Myer (ed), *The professional software engineer*, Springer-Verlag, New York.

——, Miller, K & Rogerson, S, 1998, 'Software engineering code of ethics', viewed 27 January 2007, <http://www.acm.org/serving/se/code.htm>

Halstead, MH, 1977, *Elements of software science, operating, and programming systems series*, vol 7, Elsevier, New York, NY.

Highsmith, J, 2002, *Agile software development ecosystems*, Addison-Wesley, Boston.

Lucas, R 2008, 'ETGovICT briefing paper for presenters', Centre for Applied Philosophy and Public Ethics, Australian National University, Canberra.

Martin, MW & schinzinger, Roland 1989, *Ethics in engineering*, 2nd ed, McGraw-Hill, New York.

Miller, K & Gotterbarn, D, 2003, 'Computer ethics in the undergraduate curriculum: case studies and the joint software engineer's code', *Small College Computing Conference Journal*.

Parker, D, 1968, 'Professional conduct in information processing', *Communications of the ACM*, vol 11, 3 March.

Paulk, MC, 1995, *The capability maturity model: guidelines for improving the software process*, Addison-Wesley Publishing Company, Reading, MA.

Perspectives on the Professions, Centre for the Study of Ethics in the professions at Illinois Institute of Technology, <http://ethics.iit.edu/perspective/v1n1%20perspective.pdf>

Ravichandran, T, 2000, 'Total quality management in information systems development: key constructs and relationships', *Journal of Management Information Systems*, vol 16, no 3, pp 119–37.

Ross, DE, 2003, 'My complaint against the ACM — a leading technological society condones employment discrimination against some of its own members', <http://www.rossde.com/acm.html>

SIS Code, 2005 Swiss Information Society Code of Ethics. <http://www.s-i.ch/fileadmin/daten/si/SI_Code_of_Ethik_V1.pdf>

Software Engineering Ethics Research Institute, 1999, 'Software engineering code of ethics and professional practice (5.2)', <http://seeri.etsu.edu/Codes/TheSECode.htm>

Smoot, 1981, http://ethics.iit.edu/perspective/v1n1%20perspective.pdf

Software Testing, <http://www.associationforsoftwaretesting.org/about/governance/>, Resolution on adoption of the ACM code of Ethics.

Standards Australia, 2005, *Corporate governance of information and communication technology*, Australian standard AS8015.

Standards Australia & Standards New Zealand, 1999, *Risk management*, Australian and New Zealand standard 4360.

Walsh, K,, Adms, W., Bauchner,H., Vinci,R. 2006, 'Meidcation Errors Related to Computerized order Entry for Children', *Pediatrics,* vol 118, n 5 pp 1872-1879.

Weill, P & Ross, JW, 2005, 'IT governance on One Page', MIT Sloan Working paper nimber 4517-04 <http://papers.ssrn.com/sol3/papers.cfm?abstract_id=664612>