THE ROLE AND VALUE OF ETHICAL FRAMEWORKS IN SOFTWARE DEVELOPMENT

A thesis submitted for the degree of Doctor of Philosophy

by

Sean Dodd

Department of Information Systems and Computing, Brunel University

May 2003

Abstract

Software development is notorious for failure, typically defined as over budget, late delivery and/or poor quality of new information systems (IS) on project completion. The consequences of such failure can be enormous, particularly financially. As such, there is consensus by practitioners and academics alike that this practice is unacceptable. Yet with a variety of accepted development methods and tools available for use by software developers and project managers, there is still no significant reduction in the size or frequency of failure reported. In an attempt to understand the conflicts which arise in the development environment in which developers and project managers must operate, the research area is the role and value of ethics in the development of managed software projects. A definition of ethics in this context was provided by the IEEE/ACM Code of Ethics. Research was additionally conducted to understand how other professions and business areas define and enforce ethics in their respective working environments. These were (UK) Law, Finance, Retail and, law practice in the European Union. Interpretive research was then conducted to enable software development practices to be understood from the view of developers and project managers in industry. Unethical practices were then identified in a large IT company based in west London via a single, six month in-depth case study, with the data collected analysed via a series of repertory grids. Analysis and triangulation of the data collected via interviews, document analysis and observations led to an improved understanding of the causes of the unethical practices found. Conclusions and recommendations are then provided relating to implications for (a) the company participating in the research, (b) the application of the IEEE/ACM Code in industry (c) theory for ethicists.

Contents

List of Tables

List of Figures

Acknowledgement

My thanks and gratitude goes to my supervisors, Dr Mark Lycett and Professor Robert Macredie, for their advice and support throughout the PhD program. The help received from many others is also very much appreciated, including friends and family, other PhD students, lecturers in the department and the DISC administration staff. My thanks also goes to the project teams and business units at Telco which participated in the research, particularly Gordon Anderson and the VTC project team. Finally, my thanks goes to Dawn, for her continuous support and encouragement.

Dedicated to my sister, Tangie

Publications

- Dodd, S. and Lycett, M. 2002: Ethical Frameworks and Software Project Management: A case study to identify the role and importance to practitioners of the IEEE/ACM Code of Ethics and Professional Practice for Software Developers. United Kingdom Academy of Information Systems (UKAIS) Conference *Co-ordination and Co-opetition: the IS Role*, Warwick University, Warwick, UK
- Dodd, S. and Lycett, M. 2001: Ethical Frameworks and Software Project Development: A case study to identify constraints on their implementation. United Kingdom Academy of Information Systems (UKAIS) Conference *Information Systems Research, Teaching and Practice*, Leeds University, Leeds, UK
- Dodd, S. and Lycett, M. 2000: Ethical Frameworks and Software Project Development: A review. United Kingdom Academy of Information Systems (UKAIS) Conference *Information Systems in the Digital World*, Plymouth University, Plymouth, UK

Chapter 1. Current Information Systems Development & Problems

1.1 Introduction

Most businesses in the current market place now depend on computerised information systems (IS) in order to both function effectively and compete successfully to enable their survival (ICSE 2000, SWECC 2000). With such a high level of dependency, it is surprising to find software development known notoriously for failure (Benyon-Davies *et al* 2000). With various methodologies and tools available for software developers and project managers to utilize which were created and are accepted by professionals in the industry, with consensus among many professionals in IS that such frequent and costly failure is unacceptable the cause(s) of such failure warrants investigation. With the current climate of blame pointing at software developers and their project managers, research into the existence, or non-existence, of ethical practices in managed software development is necessary. The awareness of ethics in project teams and the scope of its practice/non-practice is, therefore, determined.

Section 1.2 provides a background into software development and examples of project failure which have led to the notoriety associated with the discipline. Section 1.3 describes the nature of software development today, followed in Section 1.4 by a description of the palliatives available to address some of the problems identified. Section 1.5 identifies pertinent soft issues in software development, leading to Section 1.6 which describes a need for ethical awareness in software development. Section 1.7 then presents the aims and objectives for the research, followed in Section 1.8 by an overview of method and, in Section 1.9, an outline of the dissertation.

1.2 Background

When compared to other disciplines, software development is a nascent discipline falling under the remit of Information Systems (IS). Software development is typically a managed process involving customers, analysts, software developers, project managers, technical consultants and technical authors. Albeit a relatively new discipline, software development and software project management are taught at universities, with methodologies and tools used to encourage both consistency in approach and a high level of quality in managed IS development (ISD).

The failure of ISD comes in many forms, with symptoms existing at the end of software projects invariably consisting of (grossly) exceeded budgets, far longer development times than agreed at the outset and/or, inferior quality. This quality also comes in many forms and, in this context, is defined as conformance to requirements (Yeates and Cadle 1996). Millions of pounds are lost each year due to poor quality ISD, resulting in new IS requiring significant change when implemented or, being scrapped without ever becoming operational. Other examples of software failure include:

- The air traffic control centre at Swanwick went into live operation over seven years later than was expected and was many millions of pounds over budget
- In July 2000 both the main power and backup power for the Sydney air traffic control system failed
- The bank records of over 17,000 suppliers were accessed by a hacker breaking into the Australian Treasury departments computer records
- Norton 'Antivirus' 2000 was found to crash computer systems by confusing device drivers with script files
- The $125m Mars Climate Observer probe approached Mars on target but got 100km too close for the probe to survive - blamed on erroneous commands sent to it
- A UK rail disaster was blamed, in part, on the Automatic Train Protection (ATP) system being switched off because the train driver had not been trained in its use
- A second UK rail disaster was blamed, in part, on the train's automatic warning system (ATP) being switched off due to malfunctioning earlier in the day

For safety critical systems, achieving and maintaining high quality should be paramount in the development process. One computer company was found guilty, however, of contributing to a plane crash which killed 159 passengers. Risks with the software were known about by the software company eleven months earlier than the date of the crash - as shown by a memo used as evidence against the company in court - but not corrected. In a separate context, NATO scientists created a computer virus 'by mistake' leading to military secrets being made available on the internet. When testing the software for self protection against external virus attacks, the experiment went wrong and the developers unleashed the virus on themselves. On another occasion, the names of CIA agents were made available through a newspaper's website, despite having a layer of black boxes over the names. Anyone viewing the site could disable the black boxes by freezing the page, resulting in the names of the secret agents being made fully visible. The list of examples is by no means definitive, but shows a need for research into why ISD failure with large scope and impact occurs with unacceptable frequency.

1.3 Software Development Today

The impact, cost and frequency of ISD failure has led to much active research into how and why it occurs. One cause found is that new IS are large and complex (CLIST 2000, Pouloudi 1999), while another cause found that some methodologies used to develop and implement new IS take little or no account of the social, organisational, political, economic or cultural issues (Walsham 1993). Other difficulties in the development process include what Paul (1994) describes as the delivery of dead systems. These are new IS built to meet the requirements of a specification created at some point in the past - which may be in terms of years for larger systems - known as the project approach. The dynamic and complex market-place in which (*living*) companies operate and compete in today, however, requires systems which can evolve with the company in order for it to survive. Despite this necessity, new systems are typically implemented which were designed sometimes years earlier and are static in their operation - hence a dead system is delivered into a living organisation. The need for living systems to be delivered was first advocated by Burns and Stalker (1961) and later, Lawrence and Lorsch (1967), yet still some four decades later, the IS industry still invariably depends on the project approach for the development of new software - resulting in frequent high impact failure as defined above.

Software size is known to have grown by over 80% a year, but a study on bug density found little progress in its reduction over the last 20 years (Dakin 1996, Halang 1998). Yet still software is used at the highest levels of safety criticality, often with fatal consequences - often without fatal consequences as well, but just one unnecessary fatality is surely one too many. In addition, software developers often have many more opportunities to do good or cause harm, or to influence others to act in the same way than other professionals (Gotterbarn 1999a, Sedlet 1999). Postmodernity is described as fragmented, invariably associated with change and uncertainty and, consists of multiplicities of experience and differentiation. Consequently, hyper-individualism has been created, fuelled by the new contemporary 'me' generation (Johnson and Smith, 99, Walsham 1993). As such, these opportunities to do harm or good are achieved not only through developers' behaviour, but also through their programming skills. Program code is typically mentally owned by the developer and as such, elegance in structure and use of impressive algorithms are commonly given a higher priority over producing code which is dependable (Dakin 1996, Halang 1998, Oliver 1998). This practice is thought to support or boost the ego of developers who believe they know better than their colleagues. This esoteric knowledge - resulting in a guru status - is also rewarded by financial bonuses to individuals (particularly in the West) when achieving specific targets, reflecting typical pay structures focused toward individuals rather than project teams as a whole. The resulting code is thus unnecessarily and artificially complex, which goes against the *raison d'être* of most software development methodologies: KISS (Keep It Simple, Stupid) (Salt 1998). The sharing of information is necessary, in conjunction with team rewards, to facilitate a reduction in ISD failure. It is acknowledged, however, that code which looks complex may actually be simple in operation but structural simplicity is traded for performance. The presence of elegant code compounds the problem of determining the impact of new technology which, even on its own, is difficult to calculate and additionally, errors and violations in the code may lie dormant beyond the end of a contract (Timpka 1999, Irani 2000). This may be deliberate of course, for contractors hoping to have their contracts extended and thus keep themselves employed (Harris 2000). The consequential result - in this case complex code - and initial justification for it, highlights a need to understand the reasoning behind actions to enable a correct attitude towards it.

Schnettler (2000) and Sommerville *et al* (1998) argue that it is not the development methodology used that determines the success of a project, but ensuring that the project has the correct technical specifications, a complete project plan, set milestones, realistic deadlines and provisions for changes in scope and user requirements. Problems that arise are frequently caused by bad planning, poor communication and/or unrealistic expectations by either the customer or the project manager. The gathering of requirements from a customer for a new IS, however, is known to be problematic. Individual interpretations and subsequent decision-making are all based on subjectives, conjectures, prejudice, bias, skills and experience. These qualities are all unique to each individual, the corollary being our interpretations and decision-making abilities will never exactly match those of anyone else (Introna *et al* 1999, Oliver 1998, Walsham 1993). It is not possible, therefore, for new IS requirements to be understood correctly and completely by any two people involved in ISD. Paul (1994) further argues that even if we could have the same correct and complete understanding as everyone else, customers do not know what they require of a new IS anyway. We additionally need to find a way of eliciting what customers want, therefore, so that we can develop new IS which meets their exact requirements and is not seen to fail when implemented.

Patrick, F. (2004) found the practical reality of making and keeping to agreements necessary for successful projects is hindered by common problems: people on projects are reluctant to promise the unknown, plans are disrupted by re-work, schedules are stretched – at best – by contention for resources that are involved in more than one project. Resources are often used in mixed and multiple project environments. A limited pool of people and resources are assigned to mixed responsibilities, such as development and maintenance – and/or are shared across multiple concurrent projects. The resulting compromise on ethical development practices are a major source of difficulty in promising and delivering software projects. Bouncing back and forth between unfinished tasks in an effort to show progress merely delays all the milestone deliverables and wastes valuable time in unnecessary set down, set up and 'Where was I?' questions at every restart. Furthermore, development projects are undertaken in an uncertain environment: management is, therefore, about prediction. There is never enough time to develop and manage a project correctly but, time is always found for re-work. Estimates are requested long before many of the details of the project are really understood and, are often accompanied by a stream of caveats or disclaimers as a consequence. An environment is needed that allows people to do their best work on the task at hand. Estimates should, therefore, not be seen as fixed commitments. Where a rational planning process is in place, it invariably becomes a negotiation process where no-one is really satisfied with the outcome. The fear of promising something with many unknowns leads to many unethical behaviours, such as high initial estimates offered with the expectation that they will be reduced. Making project promises that everyone is satisfied with is difficult. Keeping promises that are the result of negotiation is even more difficult. Uncertainty must be acknowledged by all involved and managed appropriately. Projects are about dependent and interdependent efforts, uncertainty all the way and, the allocation of resources to tasks in pursuit of a goal. As these are important, they deserve focused attention and common-sense management. Planning to plan as the project progresses is thus required, knowing that factors will change as time progresses.

Software development moves from gathering requirements to specification of the modules of the code required to deliver those requirements. Thompson N. (2003) supports Patrick above by acknowledging that developers may discuss the necessary requirements with customers but also, may additionally/instead imagine what they actually require and, then write the code directly. A separate specification of the necessary modules of code then exists in the mind of each of the developers involved in the project, with the number of differences between the perceived requirements increasing with the number of developers in the team. Thompson further argues the formalised approach to development advocating 'best practice' typically results in a development environment containing the following unethical attributes:

- Excessive volumes of documentation
- Blind adherence to bureaucratic rules
- Excessive attention to quality and process maturity concepts of unproven value
- Advocation of standards over skill, initiative and judgment
- Disregard of team members who may be excellent at what they do but do not have official qualifications

Kaner *et al.*, (2002), advocate teaching the use of 'tips and tricks' of reading code with the goal of avoiding future pitfalls rather than finding errors as the code is developed. The authors offer over 200 lessons constructed from years of experience in software development, with explanations and examples of each problem provided to help illustrate each lesson's objectives. The practical value, however, of each lesson still needs to be established against industry standards. With a nascent profession seeking consistency and uniformity through the implementation of de jure standards, the practices offered by these authors may not correlate with those of a governing body. With the acknowledgement and acceptance of the public interest prevailing at all times (advocated by both Kaner *et al.* and the ACM/IEEE bodies) when implementing the tips taught, however, the practical value of the lessons proposed may prove more valuable than an official code known to contain weaknesses.

One of the early stages of software development is referred to by Miller, S (2004) as 'analysis paralysis.' This is when the gathering of requirements takes far longer than originally anticipated and analysis of the requirements delays the start of the project further still. Practices which aim to facilitate the end, or prevent, the paralysis stage altogether can be found in the literature, including Joint Application Development

(JAD). This approach requires the stakeholders to meet and have brain storming sessions early on in the project to facilitate the gathering of desired requirements and ensure a uniform understanding of them. The aim of JAD is to prevent the unethical practices described above. When changes are required, however, it is essential to know what to change, how to change it, how to cause the change to happen and, knowing who the right people are to implement it. Process improvement models available may not fit the team culture. What appears to be progress to some may appear as obvious to others, whereas some may not comprehend the progress at all. Miller identifies the following issues as important when managing an software project: (a) team size and criticality of the project, (b) critical projects require publicly visible correctness and, (c) rigid and highly disciplined methodologies do not work in reality as well as intended by their respective authors as team members are individuals and may not conform to the authors' assumptions with regard to their skills, experience, commitment, etc.

A myriad of IS models exist with the purpose of enhancing the development and management of ethical software projects. Maturity models are just one example available to practitioners, and include: (a) Capability maturity model for software (SW-CMM), (b) Software engineering Capability maturity model (SE-CMM), (c) Project management maturity model (PMMM), (d) Automated software testing maturity model and, (e) Software reliability engineering maturity model. What these models have in common, however - as well as another thirty-something maturity models identified - is that they focus on the maturity level of the software development processes in place and not on the developers or managers implementing those processes. Copeland (2003) argues another maturity model is required to focus on those actually implementing the development processes to measure their own personal maturity. He identified five levels in his new Maturity Maturity Model (M3), as defined below:

- Level 1: People are not influenced by thoughts of duty, responsibility, peer approval, accepted forms of behaviour as defined by society, ethics or benefits. They behave autonomously and speak freely
- Level 2: Maturity is acknowledged but not practiced. There is commitment to non-commitment. This level is for those who grow older but do not grow up
- Level 3: People at this level are fussy about detail, but do not know what for or why. They have their head buried in the sand
- Level 4: People seek corporate success here, with the top of the ladder remaining out of reach regardless of progress upwards. They seek and hold

tightly to arbitrary rules as a coping mechanism in their hectic little worlds. They are guided by reward and punishment. They are often anxious, hypersensitive, hypocritical and often bad tempered

- Level 5: People reflect at this level about the ethical and moral values attached to their actions. They willingly amend their ways to fit within these values

The model is intended to guide practitioners in understanding their own maturity which may then lead to ethical working practices. The process of application of individual evaluation and consequent reflection and desired course of action are not stated, although the findings of developers for project managers may result in tailored management to meet individual needs.

At the ETHICOMP Conference 2002, 75% of survey respondents carried out by Rogerson and Prior (2003) stated they would not work on a project which they considered to be unethical, although this contrasted with a 78% response in favour of working on a challenging project irrespective of its purpose or overall objectives. In 2000 the survey found 91% of respondents in favour of this (latter) statement. What the survey also found was there appeared to be a link between answers obtained and ages of the respondents. 75% of those who were either indifferent or in favour of working on a project regardless of its purpose or overall objectives were under 40 years of age. In deed, some respondents stated they were prepared to use another employee's password - with their permission - to gain access to data they were not authorised to see. Other respondents employed in the IS sector stated they preferred a non-transparent relationship with their customers. Where new software projects are running late or over budget, a reduction in the testing of software was found to be acceptable to some respondents, as found in earlier surveys. Furthermore, 69% of respondents from developed countries were in favour of developing software more quickly. Only 7% were opposed to the idea. Without the context of the answers provided, however, the percentages are not meaningful. The survey authors subsequently made recommendations to companies which include:

- Establish an ethical code of conduct for all employees if they do not currently have one in place
- Establish whistle blowing procedures
- Provide a working environment which encourages ethical working practices which promote commercial advantage

- Support developers and managers to reject temptations to bow to commercial pressures which may lead to unethical working practices

For ethical codes to be easy to remember they also need to be short and clear. For the many situations where ethical considerations arise, however, such codes lack the detail required. The IEEE/ACM Code attempts to address this dilemma with an aspirational preamble and subsequent detail for more specific circumstances. Even then, behaviour delimited by the requirements of a code does not necessarily warrant it as ethical *per se*. For example, argues Fairweather (2004), a requirement to only access other people's computer files with their permission prevents surveillance from investigating possibly corrupt behaviour on the part of the owner of the file. Another requirement of the IEEE/ACM code is the prohibited use of a company's computing resources without proper authorisation. In the event of an emergency, use of such a computer may be necessary for a specific purpose. It is important to consider the context of a situation requiring ethical decision-making, therefore, in order for an acceptable choice to be made. Furthermore, consideration of ethical issues when decision-making - although encouraged - is pointless without subsequent and appropriate action by the decision maker.

Rogerson (2004) argues UK companies are lacking in recognising their obligations to ensure ethical behaviour is conducted in the workplace. Even in society at large, parents are turning a blind eye to the conduct of their children, oblivious to it, or even learning from them. This can take the form, for example, of copying music files illegally from the internet. A public inquiry has begun, therefore, in an attempt to close loopholes in the Computer Misuse Act to further ensure only ethical behaviour is exercised. The Swedish government recently exercised control by closing down a website which it deemed to breach acceptable use of the internet. China also closed down two websites whose content was found to be unacceptable. In a time where freedom of expression is advocated, censorship by governments still prevails with the aim of securing the public interest. In a modern, complex and global society, the ability, tools and confidence to address unethical behaviour are still necessary to prevent a legacy of the consequences of such from being passed to future generations. If unethical behaviour cannot be prevented, however, Rogerson suggests the taking out of adequate insurance cover against such consequences.

1.4 Characteristics of Palliatives in Use

With so many problems it is clear to that a solution, if one exists, will be difficult to find. As stated previously, much research has been conducted to reduce or omit at least some of the known unethical practices. Possible remedies proposed to date include new, or amended, development approaches and respective models. For example, for each of the three problems identified above (i.e. over budget, over time and inferior quality), the following have been proposed in an attempt to remedy them: Rapid Application Development (RAD), Component Based Design (CBD) and the Capability Maturity Model (CMM), respectively. These approaches have the following primary aims: (a) RAD aims to have new systems finished within the budget constraints by building the systems quickly, (b) CBD aims to reduce development time by assembling a new system made up of components which are written, tested and documented prior to assembly and, (c) CMM aims to improve quality by improving software development processes.

Although each of the above approaches aims to facilitate successful completion of software projects with the use of ethical development practices, they are only facilitators and offer no guarantees for project success. In reality, all three approaches only address individual weaknesses in software development and, therefore, do not provide the panacea ideally sought (Macredie 1998). The reasons for this are as follows. RAD fails as it requires quality to be redefined, as systems built quickly are not able to meet fully the original requirements or, be completely tested in the time available. CBD fails as dependencies accrue over time between the components in the system, resulting in potentially very expensive maintenance as the ripple effect of adding new components cannot be determined before insertion. Moreover, testing is problematic, again with the ripple effect unknown and the possible requirement of the system to be switched off for the test duration. CMM fails as it assumes all problems are managerial, without regard for either technological or social issues (Macredie 1998). Social issues were also ignored, for example, by SSADM (Structured Systems Analysis and Design Methodology), which is one reason for its failure to successfully address the problems outlined above (Nicholls 1987).

Furthermore, the above development approaches only provide *prima facie* solutions as they are focused on the development processes *per se*, and do not accommodate the process as part of a much bigger picture, which, of course, it is. This mechanistic approach to software development invariably produces systems which are rigid and inflexible - i.e. dead systems, as described above (Walsham 1993). The bigger picture comprises of not only the business environment within which the new system must function, but also soft issues, i.e. the people involved. Each of these groups of people has their own set of needs which should be met for a new IS to have a chance of succeeding. Any attempt to create a complete and correct set of user-requirements without regard for the social context, multiple perspectives and ambiguities of organisational life is unlikely to succeed (Kyng 1991). From a managerial perspective, such mechanistic approaches to software development are further likely to result in developers' poor productivity and lack of motivation (Macredie 1998, Sommerville 1998, Yeates and Cadle 1996, Walsham 1993). As such, RAD, CBD and CMM are only palliatives, and do not provide the remedy sought to remove the notoriety from the IS profession.

1.5 Soft Issues in Software Development

It is now apparent that approaches to system development that encapsulate soft issues - for example, developers' needs - are necessary, due to the inherent social process of communication and learning (Walsham 1993). A new approach needs to accommodate the large part people have to play at all stages in a systems life cycle, from inception through to decommissioning. Sommerville *et al* (1998) argue ethical systems depend on ethical people to build them - supported by Lembke *et al* (1998). Soft issues are now regarded, therefore, just as important as technological and managerial issues (Themistocleous, 2000, Walsham 1993).

System development in the workplace today is invariably team-oriented, and has been for some time. Research has found, however, that the optimum size of a project team to develop new IS is 6 to 8 members. Project teams with a larger number of members were not found to have a relative increase in production, due to the increased communication overhead (Elzer 1988). Regardless of team size though, rapport between team members needs to be good in order for the team to be productive, i.e. *esprit de corps* needs to be present throughout the duration of development. It is the project manager's responsibility to identify, track and resolve any conflicts that arise, both intra and inter team, that team members are not able to resolve themselves (Kawano 1997, Yeates and Cadle 1996, Freeman 2000). Imposed decisions, although necessary in a tie-break situation to prevent inaction, should be avoided wherever possible, argues Walsham

(1993) as success of any kind usually requires consensus from all pertinent stakeholders. Ultimately, the best solutions result in a win-win situation for all parties involved (Introna and Pouloudi 1999).

The delegation of responsibility for decision-making from a project manager to the team members identifies a management style encouraging empowerment and autonomy - such as Laissez Faire, where developers only consult with their project manager when they choose to do so. This could lead us to trying to define what a good manager is, such as someone who is forceful yet sensitive, intelligent yet not too clever and so forth (Sommerville *et al* 1998). A list of desirable attributes of a good project manager could be compiled, but when complete it would explicitly identify the impossibility of finding managers who possess all the attributes sought and, not having any (negative) others. But ultimately, how much responsibility is delegated will be different for every project manager with demarcation contextually defined. To compound the problem yet further, more companies than ever before now have international offices and, with the advent of the World Wide Web, access to the global marketplace has been enabled for all companies - regardless of size or available resources. As a result, software development can now be conducted across multiple continents with geographically dispersed project teams - which exacerbates the task of finding a solution to the unethical practices identified above.

Whatever approach is taken to forming a project team, ideally the team should ultimately consist of a manager and developers who are ethical (Armstrong 1993, Elzer 1988). Not only should this facilitate good relationships among the developers leading to higher productivity, it should also accommodate the current lack of *de jure* standards needed to facilitate ethical software development (Yeates and Cadle 1996).

1.6 Ethical Awareness in IS

The literature shows a consensus on the need for a greater focus on ethical issues, not only to address the problems outlined above in system development, but also to facilitate the growth of the software engineering profession as a whole (Stephens 2000). Furthermore, it is argued that ethical issues need to be addressed as they cannot be safely ignored (Hasnas and Smith 1999), with early attempts to produce ethical guidelines in ISD concentrating on confidentiality, accuracy, property rights and accessibility (Timpka 1999). Ethics is proposed as being inherent in ISD and, certainly lives longer than any hardware used. But at all stages of a system's life are unethical actions commonplace, with software development developers specifically taking the full force of the criticism in the media (Sedlet 1999, Abi-Raad 1999, Dakin 1996). This dissertation, therefore, is concerned with the ethics affecting project teams during the development of new IS with the symptoms of unethical practices found to include (Hirschheim and Smithson 1988, Currie 1989, Sedlet 1999, Grodzinsky 1999, Timpka 1999, Dakin 1996, Walsham 1993, Gobold 1999, JSB 2000):

- Supply of inaccurate/untimely information
- Software released with known bugs
- Inappropriate political influences
- Strategy formation outside the control of local management
- Complacency resulting from success
- Game-playing with budgets

Other unethical practices which may be 'hidden' include user dissatisfaction, absenteeism and, stress felt by any of the many people involved - whether they be, for example, the software developers or end-users (Walsham 1993). The causes of these symptoms could be summarized as a lack/omission of quality, reliability and/or safety - or simply unethical behaviour (Halang 1998). It is postulated here that their cause, at least partly, lies in ethical tensions between the constraints of quality, cost and time and, partly because the state of the world today can be described as disordered (Cooper and Fox 1989). The list above is certainly not exhaustive and shows a need for an ethical contribution to ISD. This contribution is needed not only during system development (by developers and project managers alike), but also during a system's operation and maintenance and by senior management, as frequently highlighted in the media. As the benefits of a newly implemented information system are often difficult to quantify in a meaningful way, intangible and uncertain, this could also hinder or prevent actions being taken from an ethical basis as the benefits are not clear (Symons and Walsham 1988).

Pouloudi (1999) and Sedlet (1999) argue that awareness of ethics is growing and that it should be inherent in all under-graduate modules used for teaching ISD. This is due in part to the problems identified above and business issues focusing predominantly on power, politics and profits. Total power and control is not possible anyway and should be realised by management so as to prevent poor planning, reduced organisational effectiveness and consequent bad feeling in project teams (Abi-Raad 1999, Gobold 1999, Walsham 1993). The toleration of unethical behaviour by managers can subject them *per se* to personal and organisational liability (Dakin 1996). The status quo is not acceptable to stakeholders either, who include shareholders, end users, maintainers, senior management and other employees (Gobold 1999, Walsham 1993). Pouloudi (1999) notes, however, that these stakeholders are neutral and consideration of their ethical views is contingent on the manager of the IS. Walsham (1993) argues the role of an IS strategist includes acting as an ethical agent, concerned with the creation and recreation of ethical values for individuals, the organisation and society at large, which could address some of the concerns highlighted above. ISD is always reacting to crises, however, with information available highlighting past problems and not predicting possible future problems.

The computerized worldwide interconnections in place today have created a knowledge society in a global village, which enables decisions taken by developers locally to have a global impact. Never before has a business environment without boundaries been available or accessible to all (Abi-Raad 1999, Computing 2000a, CLIST 2000, Svensen 1998). It is unfortunate then, that this scale of magnitude is equally reflected in the negative impact of unethical practice. Power of this size must then necessitate responsibility and accountability. This is widely supported by the great philosophers, including Hobbes, Smith, Will and Schmitt. There is a small group of philosophers who contend the issue of course, including Godwin, although the former stance is taken by the author. There are certainly many more opportunities at this juncture for developers to conduct themselves ethically or unethically, with the consequence often hard to calculate (Abi-Raad 1999, Grodzinsky 1999, Timpka 1999). Software development must be conducted with ethical ISD as its goal, therefore, but as a discipline is still relatively nascent when compared to other more established disciplines such as law or medicine. Unfortunately, the manifestation of software development is founded on buzzwords with decisions frequently based on what is fashionable at the time - as against the technical merits at hand. It is argued that decisions taken in software development should not just be based on technological issues, but should place (at least) equal consideration on social and ethical issues (CLIST 2000, Gobold 1999, Halang 1998). Existence of the unethical development practices identified above provides additional support for a wider consideration when decision making, in an attempt to reduce or remove the problems completely - although the software development environment is known to be socially complex - which hampers any corrective effort (Dakin 1996).

The scope of impact clearly determines the scale of consequence. If customers of new IS feel the negative impact of unethical practice, a loss of (possibly repeat) business can be expected in the worst case, a damaged relationship at best. A generic product may be boycotted and internally, developers may leave (voluntarily or coerced) or, they may be considered untrustworthy by their colleague(s) and/or themselves (Gobold 1999). Mill argued the implementation of ethical practices additionally engenders positive self-worth, resulting in an increased capability of higher value to others (Oliver 1998).

A continuum of unethical behaviour will lead us from examining ethics from a philosophical viewpoint to a legal discussion, as has happened in the US which introduced new laws in an attempt to combat current problems (Dakin 1996, Grodzinsky 1999). This could be seen as an extreme measure, as some Americans are known, for example, to believe the introduction of privacy laws is only wanted by those with something to hide. Conversely, the Europeans are less suspicious and more welcoming of new legislation to establish and maintain ethical control (Halang 1998, Dakin 1996, Pouloudi 2000). Legislation has also been passed in the UK in an attempt to address some the problems outlined above, for example, the Computer Misuse Act (1990), the Copyright, Designs and Patents Act (1990), the Data Protection Act (1984) and more recently, an act giving employers the right to monitor employees' email (Bott *et al*, 1996). Bad development practice needs to be overcome, therefore, as unethical developers not only destroy their own credibility but also destroy the credibility of other - ethical - developers and that of the companies which employ them. This could be achieved formally through legislative requirements or more informally through frameworks and codes of ethical practice.

Clearly no panacea exists in the global environment in which most companies now operate and compete due to the complexity and interaction of business and societies (Abi-Raad 1999, Pouloudi 2000). Even Lyotard and Montaigne argued it may not be possible to find a solution to unethical practices encountered (Oliver 1998). The chances of this undesired result occurring is increased if Oliver is to believed, who argues ethical values are weakening significantly in the modern world (also supported by Hurst 2000). Lyotard, a contemporary philosopher, argues society has now willfully abandoned ethical standards (Oliver 1998). The proposition here is that:

> With the failure of software projects still prevalent after the advent of accepted software development methodologies and tools, we should try to determine the cause(s) of the bad practices identified. We need a framework, therefore, to firstly define what ethics actually is in the context of ISD and, then determine the cause of the unethical practices that lead to the invariable software project failure which give the profession notoriety.

Ethical awareness in IS is therefore growing. To enable faster growth, however, it useful to consider the role of ethics in other areas. This will enable lessons to be learned from established professions and facilitate the successful implementation of an IS ethical code. Other professions are considered below.

1.7 Ethical Awareness in Other Professions

To understand the role of ethics and how it is implemented in other contexts, with the objective of providing suggestions for improvement to an IS ethical code, research was conducted into the following: UK Law, European Law, the Retail sector (Marks and Spencer) and the Financial Services sector (Financial Services Authority). The findings are presented below. Additional information can be found in Appendix C for the interested reader.

1.7.1 UK Law

The rules and principles which govern the professional conduct of solicitors are created and implemented by the Law Society which is a voluntary body established under Royal Charter (1845). The purpose of the Society is to promote professional improvement and to facilitate the acquisition of legal knowledge. Much of the work below is taken from Taylor, N. (1999).

In a society founded on respect for the rule of law a solicitor fulfils a special role. He must serve the interests of justice in addition to meeting the needs of his client. He therefore also needs to be his client's advisor as well as his defendant of rights. The role creates both legal and ethical obligations on solicitors then, towards his client, the courts, the legal profession and his colleagues. These obligations additionally extend to the public, binding the two together by respect for rules which the legal profession creates itself in order to protect human rights.

The Law Society's code evolves to reflect the changes in society and, the role of solicitors in society. A working party exists which examines the principles which underlie its rule-making function and, to facilitate consistency in-rule making in line with those principles. Crucial to this process is the input from the society's members. The motivation for these processes is the goal of a rule book which reflects the reality of running a practice in modern times and, of equal importance, which protects the public interest. The rule book, known as The Guide, is intended to maintain a proper standard of conduct as a hallmark of the profession and vital to the administration of justice. The Professional Ethics department of the Law Society is responsible for producing The Guide - which consists of approximately 900 pages. Topics covered include (a) solicitors in practice, (b) client relationships and care, (c) conflict of interest, (d) obligations to others, (e) financial regulations and, (f) disciplinary processes.

The Law Society keeps a roll of all practicing solicitors as required – by the Solicitors Act 1974 and the Solicitors (Keeping of the Roll) Regulations 1999. In all cases the Law Society has the power to waive in writing any of the provisions of the regulations and to revoke such waivers. Practicing certificates are renewed annually for all solicitors. New certificates specify its date of commencement, its date of replacement and, any conditions imposed by the Law Society. Application for new certificates must be completed correctly and in full, including two signatures from other practicing solicitors who are not related or employed or partners of the applicant. Where forms are completed and signed by a solicitor other than the applicant, he must take reasonable steps in ensuring the details are correct. Solicitors without practising certificates can remain on the roll if they notify the Society annually in writing. The Society can remove their names if they have not requested to be included in the Roll by firstly issuing a notice of intention. Solicitors cannot practice unless they have been admitted as a solicitor, have a practicing certificate and, are on the Roll. Exceptions exist, including someone '…who is supervised by a qualified person'. Applications for admittance to become a solicitor should be submitted simultaneously with an application for a practicing certificate. A solicitor who has been suspended can still obtain permission from the Law Society to be employed but only to do work carried out normally by a solicitor's clerk. For example, only a solicitor with a current practicing certificate can authorise withdrawals of money from a client account – Solicitors Accounts Rules

(1998). In normal circumstances, a practicing certificate must be held and is renewed annually, in accordance with the Solicitors Act 1974 and the Practicing Certificate Regulations 1995. A fee must be paid to renew the certificate each year by the applicant. Applications for certificates can be refused, for example, where a solicitor has failed to give a satisfactory explanation of his conduct, or where a solicitor has entered into difficulties with his creditors or where a solicitor has delivered late an accountants report. An appeal against a refusal or against the imposition of a condition must be made to the Master of the Rolls (any judge of the Supreme Court) or the High Court within 4 weeks – as required by the Solicitors Act 1974 and the Master of the Rolls (Appeals and Applications) Regulations 1991. Solicitors cannot practice as a sole partner unless they have been practicing for a minimum of 3 years. He cannot, therefore, supervise an office as described below. (The Master of the Rolls is empowered to 'make any order as he thinks fit' (p56).)

An annual contribution is also required to be made by all certified solicitors to the Solicitors' Compensation Fund. The Compensation Fund provides compensation to those who suffer loss resulting from a solicitor's dishonesty. The fund is a fund of last resort, with no grants made available where recovery is possible from another source, such as an insurance policy or another person.

The rules and principles of professional conduct which fall under the remit of this research are found in the first chapter of The Guide. These are set out as follows:

1.01 Practice rule 1 - basic principles

1.02 Basic principles - additional guidance

1.03 Sources – statutory and non-statutory

1.04 Practice rules – as made by the Law Society, covering professional conduct, practice and discipline

1.05 Accounts rules – business accounts and monies of clients

1.06 Indemnity rules – fund to cover professional indemnity in private practice

1.07 Waivers and other dispensations – contained within most rules regarding conduct

1.08 Behaviour outside legal practice – must not bring the profession into disrepute

1.09 Advice and help – offered confidentially regarding conduct expected

1A Solicitors' Practice Rules 1990

1B Solicitors' Anti-Discrimination Rule 1995

In more tangible terms, Practice Rule 1 (basic principles) is described as: 'A solicitor shall not do anything in the course of practising as a solicitor or permit another person to do anything on his behalf, which compromises or impairs or is likely to compromise or impair any of the following: (a) the solicitor's independence or integrity, (b) a person's freedom to instruct a solicitor of his choice, (c) the solicitor's duty to act in the best interests of the client, (d) the good repute of the solicitor or of the solicitor's profession, (e) the solicitor's proper standard of work and, (f) the solicitor's duty to the Court.

The basic principles of all conduct governing the professional practice of solicitors is summed up in Practice Rule 1. When a conflict arises between two or more of the principles, the determining factor over which should take precedence must be public interest. The requirements of professional conduct arise from many statutory and non-statutory sources. A list of these sources can be found in Appendix D.

The Law Society provides guidance on conduct, for example, including (1) the principles and other guidance contained in the chapters of The Guide, (2) the Solicitors' Anti-Discrimination Code and, (3) the International Code of Ethics of the International Bar Association. The latter non-statutory sources provide input on matters of complaint, regulation and discipline. The bases of conduct of both sources are closely intertwined and do not function independently of each other. A statutory rule, for example, may be based on a common law ethical requirement. Non-statutory guidance may be based on an interpretation of statutory rules. Some changes to non-statutory guidance regarding a solicitor's conduct requires - by the Courts and Legal Services Act 1990 - approval by the Lord Chancellor and four judges. The Guide points out that the professional conduct required of solicitors should not be confused with the general requirements of the law. The expected professional conduct of solicitors is described in The Guide and, also in 'Cordery on Solicitors' (Butterworth). Some information exists in The Guide under the heading of Practice Information, which is for information purposes only and not to be considered requirements of professional conduct.

Many codes of conduct have been created by the Law Society to ensure professional conduct is exercised by solicitors and include:

- The Solicitors' Publicity Code 1990
- The Solicitors' Introduction and Referral Code 1990
- The Employed Solicitors Code 1990
- The Solicitors' Separate Business Code 1994
- The Law Society's Code for Advocacy

Rules have also been created and implemented to facilitate achievement of the same goal:

- The Solicitors' Anti-Discrimination Rule 1995
- The Solicitors' Investment Business Rules 1995
- The Solicitors' Overseas Practice Rules 1990
- The Solicitors' Incorporated Practice Rules 1988

The monitoring of solicitors is the responsibility of the Monitoring and Investigation Unit of the Office for the Supervision of Solicitors. Anyone acting as a solicitor without qualification will be guilty of an offence and liable on conviction of indictment to imprisonment of not more than 2 years or to a fine or both. He will also be guilty of contempt of court in any trial in which he may have been participating and punished accordingly. The Office for the Supervision of Solicitors has powers which, where required, include:

- Reprimanding a solicitor
- Disallowing part or all of the solicitor's costs
- Directing a solicitor to rectify an error at his expense
- Directing a solicitor top pay compensation up to £5,000.00
- Directing a solicitor to take any other action at his own expense as specified
- Directing a solicitor to pay interest
- Refusing to issue a practicing certificate or a conditional certificate
- Imposing conditions on a current practicing certificate
- Recovering money and papers for a client's new solicitor
- Ordering an inspection of accounts
- Intervening in a solicitor's practice
- Instituting disciplinary proceedings before the Solicitors Disciplinary Tribunal

Clearly these powers are lacking in the IEEE/ACM bodies and their introduction would facilitate the administrators' quest for acceptable and uniform behaviour of its members. In addition, when a solicitor goes bankrupt his certificate is automatically suspended.

The regulation for management of a solicitor's office is of particular importance to this research as efficacious project management is key to ensuring the delivery of new IS within the constraints of cost, quality and time as required by customers. Rules exist which require solicitors ensure their offices are and can reasonably be seen to be properly supervised in accordance with the following minimum standards: (a) he shall spend sufficient time there to ensure adequate control, (b) he shall hold a current practicing certificate of not less than 3 years and, (c) offices shall be attended when telephone calls from clients are possible for all the hours in that duration by a solicitor.

Also within the requirements of The Guide is an obligation for solicitors not to behave in their professional or private lives in a way which is fraudulent, deceitful or otherwise harmful to the profession. A Tribunal within the Law Society exists to investigate cases of possible misconduct. The most fundamental purpose of the Tribunal is:

> *'to maintain the reputation of the solicitors profession as one in which every member, of whatever standing, may be trusted to the ends of the earth. To maintain this reputation and sustain public confidence in the integrity of the profession it is often necessary that those guilty of serious lapses are not only expelled but denied readmission. A profession's most valuable asset is its collective reputation and the confidence which it inspires. The reputation of the profession is more important than the fortunes of an individual' (p860).*

Lifelong learning is seen to be vital to the competitiveness of solicitors. As a result, they are subject to a scheme of compulsory continuing professional development as described in the Training Regulations 1990. Failure to keep up to date could compromise their standard of work contrary to the requirements of the regulations. The requirements apply to practicing solicitors who are self-employed or employed and, irrespective of whether they hold a practicing certificate. Continuing professional development means a course, lecture, seminar or other programme or method of study – requiring attendance or not – that is relevant to the needs and professional standards of solicitors and complies with guidance issued from time to time by the Society. In the first year after admission to a solicitor he must undertake one hour of study for each whole month in legal practice or employment between admission and the next 1st day of November. In the first 3 years of admission, he must attend any courses the Law Society may prescribe, in addition to 16 hours of study in each of the 3 years. In each subsequent 3 year period thereafter, a solicitor must undertake 48 hours of study. He must keep a record of all study completed and submit the record to the Law Society when requested. For solicitors who work part time, one hour of study per year is required for each two hours worked in a week.

1.7.2 Law in the European Union

The continued integration of the European Union (EU) – formerly the European Community (EC) and the increasing frequency of cross-border activities of solicitors within the community have made it necessary for some common rules which apply to all solicitors operating in the Community regardless of the Society they belong to at a local level. The purpose of the common rules is to eliminate any potentially inconsistent practices and national rules between legal societies and nations, i.e. double deontology. The Council of the Bars and Law Societies of the European Union (CCBE) created a code whose rules are accepted by all legal bodies in the EU and are adopted as enforceable. The code is called the CCBE Code of Conduct for Solicitors in the European Union and the rules contained therein are considered by the individual member states when dealing with national issues. This facilitates harmonisation of the European Code with national codes as time progresses. Each are applied whenever possible in line with the other and, application of the rules when consistency cannot be established is recognised as needing to be based on interpretation. The requirements of the code are to be complied with in conjunction with the requirements of the Declaration of Vienna. This declaration was adopted by Heads of European Bars and Law Societies in 1975, to provide guidelines relating to professional assistance between solicitors of different European States. Much of the work below is taken from Taylor, N. 1999: The Guide to the Professional Conduct of Solicitors, 8th ed., Law Society Publishing, London.

Rules of conduct are designed to ensure proper performance by solicitors and are binding on them. Failure to comply must result, as a last resort, in disciplinary action. The particular rules of each bar or association arise from its own traditions. They are adapted to the organisation of the profession in the member state concerned and to its judicial and administrative procedures and to its national legislation. It is neither possible nor desirable that they should be taken out of context nor that an attempt be made to give general application to rules which are inherently incapable of such application. This is made explicit in the following statement:

> *'The Code shall apply to the cross-border activities of a solicitor without prejudice to the pursuit of a progressive harmonisation of rules or professional practice which apply only internally within a member state.' (p207)*

The particular rules of each bar and association, however, are based on the same values and therefore demonstrate a common foundation. This can be seen in the attributes required of a solicitor as defined by the EU code when compared with those described by the Law Society. The EU code requires a solicitor to have absolute independence, particularly from personal interests or external pressure. Such independence is as necessary to trust in the process of justice as it the impartiality of the judge. Advice given to a client by a solicitor has no value if given only to serve his own interests or in response to external pressure. Relationships of trust can only exist if a solicitor's personal honour, honesty and integrity are beyond doubt. These are seen as both traditional virtues and are professional obligations. Confidentiality - without time limits - between a solicitor and his client is fundamental in securing the trust required for justice to succeed. This confidentiality extends to colleagues and office staff and, continues after a solicitor ceases to act for his client. To further secure and maintain his independence, some occupations carried out in conjunction with his legal activities which could compromise this, are prohibited.

The corporate spirit of the legal profession requires a relationship of trust and cooperation between solicitors for the benefit of their clients and in order to avoid unnecessary litigation. It can never justify setting the interests of the profession against those of justice or those who seek it. A solicitor should recognise that all other solicitors of member states are professional colleagues and act fairly and courteously towards them. It is the duty of a solicitor not to accept a case in an area where he is not competent and, in this instance, should help his client locate a solicitor with competence in that area. Where solicitors cooperate with solicitors from other member states, both have a duty to take into account the differences which may exist between their legal systems and professional organisations, competences and obligations. Solicitors have a further duty to inform themselves of the rules of a host member of the EU which may affect them in their work. The Solicitors' Services Directive (1977) contained the same requirement:

> 'A solicitor pursuing legal activities shall observe the rules of professional conduct of the host member state, without prejudice to his obligations in the member state from which he comes.' (p209)

The International Bar Association is a federation of the National Bar Associations and Law Societies with membership throughout the European Union. Most companies have Codes of Legal Ethics as models for or governing the conduct of its members. In some jurisdictions these codes are imposed on all the practitioners by their respective Associations or Societies. Except where the context requires otherwise, it is this Code which applies to a solicitor having contact with a solicitor from another jurisdiction or where his activities are conducted in another jurisdiction other than his own. This code does not, however, permit a solicitor to ignore his obligations to the law or rules of conduct in his own jurisdiction. This code acts as a guide as to what is considered acceptable conduct by all solicitors engaged in international law. The code consists of 21 rules and the rules pertinent to this research are described below.

Solicitors are required at all times maintain the honour and dignity of their profession. They shall abstain from any behaviour which may discredit the profession, both in their private and professional live. They shall not engage in any other business if by doing so they cease to be independent. Solicitors are required to treat their professional colleagues with the utmost courtesy and fairness. Solicitors who undertake to work with foreign colleagues shall always keep in mind that the foreign colleague has to depend on them to a much larger extent than in the case of another solicitor from the same country. Their responsibility is much greater therefore, both when giving advice and handling a case. For this reason it is improper for solicitors to accept a case unless they can handle it promptly and with due competence, without undue interference by the pressure of other work. Any oral or written communication between solicitors shall be kept confidential. Solicitors shall always maintain due respect for a court and never knowingly give to a court incorrect information or advice. They shall at all times give clients a candid opinion on any case. They shall offer assistance with scrupulous care and diligence and never encourage undue litigation. They should not acquire a financial interest in a case they are conducting. Furthermore, solicitors should never disclose what has been communicated to them in their capacity of counsel, even after a case concludes, unless ordered to do so by a court. They should be most diligent and prioritise the interest of their clients and the administering of justice over seeking compensation for their services. Fees charged should reflect the amount of time and labour involved and all other personal and factual circumstances. A contract for a contingency fee should be reasonable under all circumstances of the case, including the risk and uncertainty of the compensation. Solicitors should not permit their professional services or name to be used in a way which would enable others to practice law who are not qualified to do so. In addition, solicitors should not delegate to a legally unqualified person not in their employ or control which are normally performed by a qualified solicitor.

Conflicting requirements exist between the EU Code and the Law Society with regard to which values should prevail at all times. The Law Society requires that its members support each other over their client's interests so the profession never suffers. The EU Code requires its members to protect their client's interests over their own or those of the other members of the legal profession. Solicitors are required to advise and represent their clients with diligence, promptly and conscientiously. He is not permitted to accept a case which he knows he does not have the competence to conduct professionally. A solicitor is further prevented from withdrawing from a case where the client is unable to find other legal assistance and will suffer as a result. This also conflicts with the Law Society's Code where a solicitor is preferred not to withdraw in such circumstances. Once accepted, a solicitor cannot generally withdraw without safeguarding the client's interests. It is also the responsibility of the solicitor to ensure he understands correctly the wishes of the client. Delegation of work to others does not remove the responsibility that came when accepting a case.

An interesting point in the EU Code is that solicitors may not accept a case where knowledge obtained in confidence from a previous client may give an undue advantage to the new client if that confidence was breached. In the context of IS development, this would prevent one company's competitors gaining unfair advantage by employing the same IS employee at a later date by becoming aware of their business from information supplied – regardless of any possible motivation – by the IS employees.

In an attempt to reduce misunderstandings between EU members, solicitors are encouraged to offer training to new solicitors from other states.

If a solicitor believes a colleague in another member state has breached the code of conduct, he should inform him of this first. Where a dispute arises between solicitors of different member states, they should 'try to settle it in a friendly way' (p221). Where this is not possible, the law societies of each should be informed so they may assist in reaching a settlement before any formal proceedings are started.

1.7.3 The Retail Sector

Companies operating in other industries also have ethical codes which they create and implement and, consideration of one such code may prove beneficial to this research in terms of implementation and adherence to requirements. One such company in the retail sector is Marks and Spencer (M & S) which joined the Ethical Trading Initiative in 2000 and then produced a guide in 2003 entitled Corporate Social Responsibility (CSR) Review. In 2003 M & S joined the new FTSE4Good Ethical Company index. Much of the information discussed below came from www2.marksandspencer.com/thecompany/corporatesocialresponsibility/index.shtml (05/04/04).

Marks and Spencer is a multi-national retail outlet which started in 1884 and now has over 330 stores in 26 countries. Its objectives are described as: Vision – the standard against which all others are measured; Mission – making aspirational quality available to all and; Values – quality, value, service, innovation and trust. The company has survived tough conditions in recent years with one factor responsible for this survival: a concentration on three key principles. These are shown below, with descriptions - pertinent to this research - provided.

1. Take care and act responsibly in delivering high quality products and services
    - Recognizing that when a profit is made a responsibility has to be accepted
    - Strive to achieve the best balance of quality, value for money, social well being, environmental protection and animal welfare
    - Listening and responding to the needs of stakeholders openly and honestly
2. Create a great place to work
    - Employees and suppliers feel engaged in helping to grow a good and responsible business

- Valuing the quality of the relationships with each other and the stakeholders
- Encouraging the growth of diversity and responding to better work/life balances
- Working with business partners who share our aspirations beyond our principles

3. Help make the communities good places in which to live and work

The chairman, Luc Vandevelde, summaries these three key principles as ‘Everything we do is influenced by the notion of good corporate behaviour’. Anticipating customer needs and acting in their interests is how M & S aim to meet their expectations. Ethical conduct in this context includes, for example, from 2002 the sole use of free-range chickens to produce the egg products sold. The life cycle of all the products sold are viewed as shown in Figure 1 below and includes much more than just the distribution and selling processes.

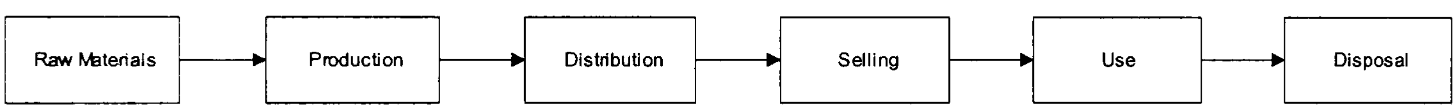


Figure 1. M&S Product Life Cycle

The ethical conduct of the company is over-seen by a board level committee which meets at least three times a year, supported by a network of managers. The standard of behaviour expected by the stakeholders sets the remit for the CSR committee, with stakeholders consisting of all the people and interest the company might affect. These include shareholders, customers, employees, suppliers, campaign groups and researchers. Understanding their needs is seen as fundamental in the development of policies. This is achieved through listening and learning – noted as the highest priorities. The ways in which the stakeholders are listened to is shown below (where appropriate - and/or different - the relative group in the context of IS is shown in brackets):

Group	How the views are obtained
Employees (developers and project managers)	• Business involvement groups – representing employees locally, regionally, and nationally throughout the company • Confidential helpline • A regular employee survey entitled 'How are we doing?' • Internal communications including an employee magazine, weekly business updates and daily team briefings • Listening Groups that senior management hold during regular store visits • Three month CSR consultation programs
Customers	• Sales information • Weekly and monthly monitoring of views • Surveying customers at new and refurbished stores • Customer panels • 350,000 communications to head office • Annual independent CSR survey
Shareholders	• Annual general meeting • Regular meetings and presentations with institutional investors • Regular surveys of institutional investor satisfaction • Participation in CSR surveys and benchmarking (e.g. FTSE4Good)
Suppliers including franchisees	• Regular visits and meetings • Regular surveys of satisfaction • Attending major UK shows and running listening groups • Direct relationships with important raw material suppliers • Taking part in the Ethical Trading Initiative
Communities	• Working on a wide range of urban initiatives, e.g. crime prevention • Regular meetings with key charity partners • Partnerships initiatives to address issues of health, community safety, education and employability • One Community' pilot, working through the employees, customers and neighbours
Environment and animal welfare groups	• Regular meetings to discuss key issues • Active participation in benchmarking and surveys • Partnership initiatives across a wide range of issues such as free range eggs and food pesticides
Government and regulators (ACM, IEEE)	• Regular meetings with bodies such as the food standards agency, department for the environment, foods and rural affairs, health and safety executive, environmental regulators, financial services authority and key government departments • Local authority partnerships on environmental health and safety, fire and trading standards

Table 1. M&S Stakeholder Communication

Only the timescale for review of these issues appears to be inconsistent with the initial enthusiasm expressed throughout – the reviews are described as occurring in terms of 'Over the next few years'. In this time scale, any amendments or additions will take a long time to be subsequently distributed, educated and implemented in practice, although some aspects of CSR performance are already reported. The CRS committee sees its role as helping all the stakeholders to get as much out of M & S products as possible. This is achieved by working with them to agree on what is and is not acceptable and how improvements can be made. Ethical trading is accomplished by active membership of the Ethical Trading Initiative (ETI) which includes an annual submission of a performance report. Ensuring good working conditions is one requirement of the ETI, as is sourcing local produce where appropriate - 70% of the produce sold comes from the UK and Ireland. Systems for ensuring ethical trading are reviewed by independent specialists. These consist of M & S audits and independent audits and, supplier self-audits. Suppliers are further encouraged to adopt this approach with their own suppliers. This is formally documented as Global Sourcing Principles which sets out the minimum working standards required. The areas covered are: working hours, health and safety, conditions, rates of pay, terms of employment and, minimum age of employment. Suppliers are expected to raise their standards in addition to meeting the - minimum -requirements. M & S have produced Self Help Guides in 9 languages to facilitate suppliers' compliance and progress. M & S actively participates in 9 pilot ETI programs. Specialists not employed by the company are commissioned to advise on improvements and, comprehensive training is provided for suppliers to create local best practice programs. In addition, there is a business ethics policy in place regarding gifts and insider trading and, financial donations to political parties ceased in 1997.

Employees of M & S were surveyed to establish what they considered important about working for the company and the results found it was considered important to be a valued member of a close-knit team, the team is able to get results and the team members enjoy each others company. For M & S to be considered a 'great place to work' (as identified above) the survey found it needed to offer (1) consultation and communication of the company's goals and for the employees to be actively involved, (2) reward and recognition for superior results and, (3) the chance to develop skills so that employees felt equipped to perform current and future roles. Consequently, employees are regularly asked which medium they prefer to receive company information. It can be provided via the staff magazine, meetings, email, team events, telephone conferences, videos and road shows. A regular survey entitled 'How are we doing?' gauges morale and pinpoints areas for improvement. It achieves a very high response rate. There is a business involvement group in every office and shop created and run by the employees, with the leader of each allocated 2 hours a week to organise and communicate the findings to head office. The Chief Executive chairs meetings to discuss the findings. Employees receive a fixed salary with a bonus paid for individual performance and another bonus for business performance. A package of benefits is also offered which includes 20% off store products, pension scheme and preferential loan rates. Most M & S employees are also customers and shareholders of the company. The company acknowledges a challenge is balancing financial rewards for promoting teamwork and individual achievement. Career plans and objectives are discussed annually with line managers for all employees and used as a basis for setting individual pay bonuses. In a survey of its own employees, M & S found 60% of workplace illness related to muscular strains and the second highest cause was anxiety and depression. As a result, in 2002 all employees were issued with a booklet on personal safety and a personal security alarm.

Priorities for the CSR committee are to continue to gather evidence of compliance to the principles outlined above. An improvement in performance is sought regarding the ETI, in addition to helping suppliers accomplish the same goal. A reduction is also sought in the workload and costs of supplying ethical trading information to retailers. Participation in local authority best practice initiatives is to continue. The issues identified as important to the CSR committee arose from the CSR members' own understanding of the issues raised by stakeholders in the surveys conducted. This leads to possible difficulties with language and ambiguity as previously discussed. Furthermore, every area of M & S is developing its own CSR action plan, prohibiting consistency across the company. The advantage of course is the guidelines that will result from the action plan should meet local needs. Again, a balance needs to be reached between local and company wide issues.

1.7.4 The Financial Services Sector

The Financial Services Authority (FSA) - formerly known as the Securities and Investments Board - has created a framework of principles founded on ethical values and developed a risk-based approach to the regulation of companies in its remit. The

FSA views mechanical compliance as unhelpful in preventing problems from arising and, consequently, compliance to the principles created simply for the sake of complying to something, is not sought. Much of the work below is taken from Davies (2002).

The financial services sector believes it suffers from a negative public image of being interested in making money for those who work in it, at the expense of everything else. This perception continues, despite immediate condemnation of large financial scandals in the press such as Enron and the Allied Irish Bank. The financial sector feels the pressure of such notoriety and the FSA has produced a long and detailed handbook on conduct expect of professionals working in the financial industry in an attempt to address the issue. The principles contained therein have been found to be misunderstood an applied inconsistently by some of those employed in this sector. Some of those are asking 'Show me where it says I can't,' rather than 'how can we improve our standards and conduct our business with integrity?'

The response of the FSA to recent scandals has been seen to be initiated by the bad actions of a few which will affect the good actions of many. Those employed in the industry with consistently good practices should be regulated less than those with consistently bad practices. The values displayed by companies influences the regulatory relationship the FSA has with them. This interplay is illustrated in Table 2 below, in a model for the development of values and the nature of relationships between companies, customers and the FSA per se. The model works in terms of individual firms, a sector, or the financial industry as a whole. Companies are not expected to move in one direction only – movement in a backwards direction and development at different rates is also expected and accommodated. The model does, however, give a direction for companies, sectors and the industry as a whole with an ethical dimension.

Company Values	Regulatory Relationship
Minimum Standards • Unthinking, mechanical compliance • Does as little as can get away with • Culture of dependency • Tries to abdicate decisions and responsibilities	**Policing** • Monitoring boundaries • Detecting and responding to crises • Enforcement 'lessons' • Basic training

Compliance Culture	Supervising/Educating
• Reliant on guidance • By the book • Unaware of some risks • Bureaucratic	• Developing ethics and competence • Looking for early warning signs • Early action to bounce firms back on track • Themed/focused visits
Beyond Compliance • Risk focused, self policing • 'Buy in' at senior level • Ethos integrated into most business processes • Ethos seen as assisting business	**Educating/Consulting** • Facilitating the development of competence and culture • Values scorecard • Lighter touch
Values-led Business • Internalise ethos of core values • Spirit not just the letter • Values focused, goes beyond rules, not just compliance • Well developed individual responsibility and a sense of involvement by (all) staff • Focus on prevention • Continued reassessment and improvement on approach • Awareness and discussion of ethical considerations at all levels • Open relationships • Strong learning culture	**Mature Relationship/Benchmarking** • Reinforce good practice • Lead by example • Reallocate resources to problem firms • Sustainable regulation

Table 2. Development of Values

Adherence to the model may not produce tangible business benefits. In deed, unsubstantiated, high ground claims about how a moral case complements a business case every time are not sought. A gap is perceived to exist, however, even when it does not, between a business case and doing the right thing. This does not mean that an ethical approach to decision making is incompatible with increasing shareholder value.

Where issues of consideration hold equal merit it can be difficult to decide on which course of action to take. An additional problem is that it may be impractical to seek advice at the time of having to make a decision. The ethical framework developed by the FSA attempts to address these difficulties by encouraging the recognition, application and balance of values when making decisions and taking subsequent action. The FSA principles – in conjunction with the Financial Services and Markets Act – embody a framework of core values:

- Open, honest, responsive and accountable
- Relating to colleagues and customers fairly and with respect
- Committed to acting competently, responsibly and reliably

These values can be seen to correlate with those found above in the IT/IS profession, the legal profession and the retail sector. To facilitate the application of the spirit of the values identified in the principles above, the FSA designed the following questions to help recognise, apply and balance values in everyday decisions and actions, grouped under the following headings:

Open, honest, responsive and accountable

- Who is left out or kept in the dark? Why?
- How happy are we to be associated with our decision/actions?
- Are we listening or just hearing?
- What can we learn? How do we help others to understand us?
- How do we recognise and deal with conflicts of interest?

Relating to colleagues and customers fairly and with respect

- Do we treat everyone as we would like to be treated?
- Do we deal with people with respect and without prejudice?
- How do we keep rights and obligations in balance and proportionate?
- How do we hold to our commitments and resist 'fudging'?
- Who benefits and who loses out? Should they?

Committed to acting competently, responsibly and reliably

- Do we do what we say we will do?
- Under pressure do we swap co-operation for co-ercion?
- Do we dither or delay? How is error treated?
- Do people trust us? If not, why not?
- Can we meet our commitments and plans?

To embed these values in the thought processes and consequent decision-making of professionals working in the financial sector, the FSA considers additional questions need to be considered:

Developing vision and a values-led approach

- What needs changing? What prevents change?
- What is the long term outcome? What is sustainable?
- Do we sufficiently recognise and act on our stakeholder responsibilities?
- How do we develop share purpose, loyalty and fulfilment?
- Do we apply ethical criteria simply to gain an advantage or because we believe we should?

The value of discussion of practical day-to-day examples is recognised. These can tease out the kind of issues that people have to face and how they might deal with them and the value judgements they consider in doing so. A variety of hypothetical situations were developed by the FSA to illustrate where it might it be more difficult to decide on the right course of action. The situations are intended to initiate thought and encourage a deeper understanding about the ethical behaviour expected of its members. One situation is described below. Others can be found in Appendix C.

Scenario 1: You are a project manager for an internet company. You are selecting a project team for a major piece of work that could last up to 6 months. Your boss has recommended that you make use of a particular member of staff who has many of the technical skills you need. However, from past experience, you know that this person has difficulty working in a team, tends to turn up late to important meetings or get the date wrong and, you had difficulty communicating this to him last time you managed to get him on a project. Questions: (a) Do you raise the issue with your boss? (b) What are your considerations?

The goal of creating the ethical model of value development and hypothetical situations above is to facilitate a change in the public perception of the financial services industry. This would enable the provision of expertise in conjunction with an integrity that engenders mutual trust.

1.8 Aims and Objectives

To meet the needs of the proposition above, the following objectives have been defined:

- Evaluate the IEEE/ACM Code of Ethics for Software Developers as an ethical framework for developers, project teams and organisations
- Identify how working practices, policies and procedures of organisations are influenced by the Code
- Identify how the application of the Code is influenced by an organisation's working practices, policies and procedures
- Identify how the enforcement of ethical practices in other professions can enhance the enforcement of ethics in the IEEE/ACM Code

1.9 Overview of Method

An interpretive and in-depth single case study was carried out over a six month period. The case study explored the suitability of the IEEE/ACM Code as an ethical framework after using it to firstly provide a definition of ethical practices in the context of managed ISD. This definition enabled the identification of unethical practices implemented by the software project managers and their respective development team members, in addition to other business units participating in the research. The objectives were achieved with the use of interviews, document analysis and, observations - recorded in the author's log book. A version of repertory grids were then used to codify the predominantly qualitative data collected, enabling largely quantitative analysis of the findings and a discussion. A summary and consequent conclusion are then presented, including recommended changes to the IEEE/ACM code in an attempt to reduce or omit the unethical behaviour as outlined above - thereby contributing to a reduction in IS failure. Finally, areas of further work are provided for the interested reader.

1.10 Dissertation Outline

Section 2 starts the discussion by defining the research area. Current problems in ISD which are described in the literature as unethical are identified. The role of ethics in the managed development of information systems is thus ascertained, in an attempt to minimise or omit unethical behaviour or actions. Ethical practices defined and sought by philosophers and religious groups are then examined to ascertain their strengths and weaknesses. The IEEE/ACM Code of Ethics and Professional Conduct for Software Developers is described to identify its objectives and enable a definition of ethics in the context of managed software development projects to be determined. Implications of the Code in practice are also ascertained. The chapter concludes by establishing the focus of the thesis as research into (a) the IEEE/ACM Code to determine its suitability to address the unethical practices found in software development implemented by developers, project teams and organisations (b) the influence of ethical codes on organisations' working practices, procedures and policies and, (c) the influence of organisations' working practices, procedures and policies on the application of ethical codes.

Section 3 begins with a description of the four main 'soft' paradigms for consideration as the research approach to be adopted: interpretivism, radical structuralism, radical humanism and functionalism. Each paradigm is described with its respective ontological and epistemological stance and then the justification of adopting the interpretivist paradigm for the research is given. The techniques permitted within this paradigm identify a single in-depth case study as being the most appropriate for meeting the aims and objectives of the research. A critique of the case study technique is provided, followed by a protocol containing details of the case study design, the data collection techniques used and, details of the company chosen and justification for its selection. Data collection methods utilised in the research are described, namely documentation analysis, semi-structured interviews and observation.

Section 4 describes the evidence collected from the case study relating to all eight principles of the IEEE/ACM Code of Ethics for Software Developers. Each principle requires the compliance of project managers and software developers in order for ethical practices to be exercised. The evidence collected in the case study and considered important is presented, enabling the research questions of 'What' unethical practices exist and 'How' they occur, to be answered. The evidence of unethical practices found and considered important were related to the research objectives stipulated in Chapter 2 and, either occurred more than five times during the case study or were allocated a '3' by the author for being considered most important - or both (with '1' allocated for least important).

Section 5 discusses the unethical practices found by the researcher conducted by all the software project managers, software developers and other business units participating in the research. Only the unethical practices which directly relate to the issues drawn out in Chapter 2 are included at this juncture, relating to moderating the interests of all parties involved ISD, high quality in ISD, the professional conduct of maintenance and, project planning. This discussion enables the more important research questions of 'What' and 'Why' the unethical practices identified occurred to be answered, by identifying their cause. Areas which were found to be addressed by multiple principles - due to repetition in the Code - are noted as such and discussed only once to avoid duplicate work. A description of each pertinent principle requirement to which the unethical practices identified relate is provided, prior to the evidence and subsequent explanation for non-compliance.

Section 6 assesses the IEEE/ACM Code of Ethics for Software Developers for suitability as an ethical framework under the categories identified in the literature of: design considerations, inclusion of extant framework strengths, exclusion of extant framework weaknesses and its effect on unethical working practices. The strenghs of codes identified include: advocation of quality, education, identification of stakeholders, consideration of context and significance, deterrence, identification of value categories and, reasoning and attitude. The weaknesses of codes identified include: ambiguity, arbitrary increase in workloads, unclear or conflicting work practices, job insecurity, use of alien language, over-simplification of instructions, designed solely by managers and, discipline. The literature additionally identified resistance to codes when being implemented, in relation to the protection of job status, a requirement to work with new people, perceived arbitrary increases in workloads and unclear new working practices. How the IEEE/ACM Code addresses such resistance, or otherwise, is discussed, also under these headings. Other professions/business areas were researched to identify how they enforce ethical working practices on their members in the workplace. The professions researched were (UK) law, law practised across the European Union, Finace and Retail. How these professions/business areas implement ethical practices is discussed.

Section 7 provides a summary and conclusions for the dissertation. The suitability of the IEEE/ACM Code as an appropriate ethical framework is determined and, its affect on reducing or omitting the current problems inherent in software development as identified in Chapter 1 is ascertained. A summary of the unethical practices - as experienced by the four software project managers and their development teams simultaneously with the four other business units taking part in the research (i.e. Recruitment, Human Resources, Quality Control and Telco as a whole) is provided in conjunction with a description of how ethical practices might be implemented or facilitated. The affect of working practices on the application of ethical codes is identified and what businesses can do to facilitate their successful implementation. A summary of the contribution made is then provided, followed by recommendations of further work.

Chapter 2. The Role of Ethics & the IEEE/ACM Code Introduced

2.1 Introduction

With ethics in IS now established as the research area, a contextual definition of ethics and its implications for use by software developers and project managers is now needed. The IEEE and ACM organisations jointly created a Code of Ethics to address the problems identified in software development. The Code provides a definition of ethical practices in the context of managed software development, leading to a clear scope for the thesis: the role and value of ethics in the development of managed software projects. The objectives of the research are then determined.

Section 2.2 provides a background into ethics, by describing types of behaviour sought by different religious and philosophical groups. This behaviour is addressed by what can be generically called codes of practice. Their alignment with personal attributes considered ethical by academics is determined, to acquire a broader understanding of how ethical practices might be facilitated in the software development process. Section 2.3 discusses the literature search findings which include the identification of management influences on software development practices and, are discussed in conjunction with a project team's legislative obligations. After the scope of unethical practice is defined, the affects of multi-cultural project teams typically used at this juncture are presented, followed by an investigation into whether ethical behaviour can be learned. Section 2.4 provides a critique of existing codes and frameworks, consisting of the use of language, common weaknesses identified and the requirements of a successful code. This is followed by Section 2.5 which describes the IEEE/ACM Code, its objectives and the eight principles which constitute its structure, justifying its use to provide a definition of ethics in the context of this research. Implications for application of the Code in managed software development are also identified.

2.2 An Introduction to Ethics

Ethics has an integral part to play in all areas of IS development and, for the symptoms outlined above to be reduced or even omitted, practices need to be judged as ethical or not to enable better decision-making about whether to proceed with the development practice at hand. It is argued, though, that consensus exists regarding a sliding scale of ethical correctness being more appropriate than a simple binary decision (Abi-Raad 1999, Baldwin 2000). This sliding scale complicates the question further of whether an action can be ethical, moral and legal, or any of the possible permutations. The legality of an action is easily determined by (timely) legislation - something is either legal or not, but determining the ethical stance of something based on a sliding scale is not so clear. For the - data analysis - purposes of this research, development practices are evaluated as either ethical or unethical. But before we can investigate the role of ethics further, in an attempt to reduce or omit unethical practices and their respective impacts, a definition is needed of what ethics actually is. The Oxford Concise Dictionary (p463, 1996) defines ethics as 'Morally correct concerning human conduct; honourable'. The dictionary definition provided is quite general, however, and a contextual definition regarding software development is sought for the purpose of this research.

2.2.1 Definitions from Religion and Philosophy

The contribution from religion and philosophy is important as they are both significantly connected to human behaviour - religion frequently tries to influence it and philosophy tries to understand it. Both disciplines have a potentially large part to play in the quest, therefore, in identifying, measuring and encouraging ethical practices in software development in the attempt to reduce or omit unethical practices (Walsham 1993, Kling and Iacono 1989). The personal ethical attributes identified in the literature by religions and philosophers and which are pertinent here are shown in Figure 2 below (Oliver 1998)*.

- Content
- Supportive of the community
- Harmonious
- Passionate
- Hard working
- Decent
- Peaceful
- Caring
- Supportive of equality
- Not frivolous
- Fair
- Self-Reflective
- Safe
- Honest
- Diligent
- Single
- Courteous

- Charitable
- Congenial
- Happy
- Reduce suffering
- Sincere
- Humble
- Seek peaceful resolutions
- Disciplined
- Determined
- Selfless
- Minimises harm

Figure 2. Ethical Attributes by Philosophers and Religions

*To aid clarity, attributes identified and considered synonymous with those already listed have been excluded from the table.

The ethical attributes in Figure 1 above are not supported by all philosophers and religions, with one such contention being the attribute 'Happy'. Although advocated by many, Erasmus thought the pursuit of happiness was folly as he considered it synonymous with contentment - which led to numbness of the senses. Happiness is subjectively defined, of course, with software developers possibly finding happiness in the use of the latest technology or methodology or, reaching or surpassing some measurement of achievement. Other developers may find happiness when finishing work early or receiving a financial bonus. It is worth distinguishing here that happiness is an emotion from the senses - a state of being, rather than an action which is a state of doing. The latter relates to the unethical practices being researched.

A common difficulty with codes, argues Baudrillard and Habermas (two modern philosophers), especially codes with an international following, is that codes grow weaker in value as the number of followers increase (Oliver 1998). At this juncture, known as the Information Age, the value of codes is also weakened as followers become aware of alternative codes around them. Not only are similarities identified between different codes, but also the strengths and weaknesses of each. As a consequence, the codes being followed become liberal in order to keep their appeal (Buerk 2000). This begs the question of who is determining the values to be upheld: the authors of the code or the followers? The stance taken here is that ideally a code should be created with collaboration by both the code authors and the follows to facilitate its success when implemented.

2.2.2 Definitions from Academia

Academia additionally describes ethics as a code which promotes positive activity on several levels: individually, locally, nationally and globally (Gobold 1999, Pouloudi 2000). The values of such a code require more than compliance to the legal details of a contract, argues Abi-Raad (1999), although this argument is rejected by Calvin who advocated only strict obedience is necessary for a code to work. Compliance may only be fully achieved when ethical values are within a person and satisfied automatically, which is thought to be accomplished when, according to academics, the attributes shown in Figure 3 below are present in a person's character (Dakin 1996, Grodzinsky 1999, Halang 1998, Kelley 1999, Timpka 1999).

- Strives to do better
- Respects others and their community
- Avoids causing significant harm
- Proactive self-management
- Produces high quality work
- Aims for company goals
- Shows appropriate behaviour
- Has self-discipline
- Bases work on earlier generations
- Resists fashions/market forces
- Doesn't abandon good practices for worse
- Takes on additional responsibility
- Works overtime
- Finishes all work started
- Volunteers help
- Uses initiative
- Concerned
- Courageous
- Considerate
- Integrity
- Selfless
- Committed
- Honourable
- Honest

Figure 3. Ethical Attributes by Academics

Although the list appears to be in general accordance with the ethical attributes identified by philosophers and religions in Figure 1 above, it is worth noting, however, that definitions are invariably timely and contextual in practice.

It is interesting to find the spirit reflected in the taxonomy of attributes identified in Figure 2 by academics largely reflects that created by the attributes identified above by religions and philosophers. The additional attributes can be seen to be 'Base work on earlier generations' and 'Resist fashions/market forces'. These were excluded from philosophers' findings as the philosophers were attempting to answer slightly different questions. These include: 'What makes humans happy?' etc., as against an academic's more specific questions of: 'What makes software developers happy, how and why?'

Ethics is a very active research area at present, with over 300 categories proposed in the academic arena. Examples from these categories are provided in Table 3 below (Ambrose *et al* 97, Hasnas and Smith 99, Johnson and Smith 99, Van der Ven 98).

Ethic Categories	Author
Action-based	Grodzinsky (1999)
Business	Timpka (1999)
Computer	Grodzinsky (1999)/Online Ethics 2000
Consequential	Gobold (1999)
Cultural	Gobold (1999)
Engineering	NSPE (1996)/Online Ethics 2000
Feminist	Pouloudi 1999
General	Sedlet (1999)
Method	Walsham (1993)
Personal	Grodzinsky (1999)
Professional	Grodzinsky (1999)/Sedlet (1999)
Research	Singer *et al* (1998)/Online Ethics 2000
Rights	Gotterbarn (1999b)
Rule-based	Gobold (1999)
Virtue	Grodzinsky (1999)/Abi-Raad (1999)

Table 3. Ethic Categories Found in Academia

Academics in IS struggle to differentiate between ethics and morals and, believe it is for others, such as linguists, to accurately define (Baldwin 2000, Pouloudi 2000). Lewis (2000), a psychologist, argues the two terms are interchangeable and are used as such by academics. She further states for something to be ethical it must be good or right, but she does not make it clear if these values are absolute - which would conflict with Abi-Raad and Baldwin above. For the purposes of the this research, (a) software development is the development of any new IS which has a project manager and is therefore a managed software development project and as such, (b) enables ethics in ISD to be defined by the IEEE/ACM Code described below. The Code uses the term ethics and further states that ethical practices are binary values although are only generically defined in the Code and should be contextually defined.

2.3 Use and Implications of Ethics Within ISD

2.3.1 Managerial Influences on ISD

Ethical guidance is expected from, and of, IS professionals who are mature, in responsible positions or are valuable to a company. Even Socrates, regarded by some as the founder of philosophy, argued that ethics was an essential quality for managers to have (Oliver 1998). Project manager's values, however, are commonly determined in relation to performance and without consideration for ethical consequences. This is due to the main, if not entire, focus of management education on corporate strategy and profit - with the additional covert use of controlling behaviour (Woodall 1996, Walsham 1993). This needs to change for several reasons. Firstly, not only are ethical considerations an integral part of IS management decision-making, (even if they are not explicitly recognised at the time), but a project manager's claim on ethical authority in the workplace is questionable as a result (Hasnas and Smith 1999, Johnson and Smith 1999). Secondly, change is now also necessary in project management due to the constant presence of dispute and conflict in the business environment, as well as the failing of traditional compliance-based management control (Johnson and Smith 1999, Woodall 1996). Thirdly, ethical awareness and adherence is essential for project managers if software developers are to take ethical practices seriously (Abi-Raad 1999, Halang 1998, Singer and Vinson 1998). Walsham (1993) found however, that managers take actions without the deliberate use of theories and are consequently sceptical towards any theories proposed. He advises, however, that strict adherence to a framework is not ideal and that a framework is best used as a valuable tool and not a rigid structure. In today's world of networked communications, managers now need to let go of old and familiar inapplicable domestic paradigms and adopt a new global perspective (Ford *et al,* 1996).

Change in not easy, however, which explains why project managers and developers alike have traditionally not been proactive in this area. Kling and Iacono (1989) and

Pettigrew (1990) found this was due to complexity, difficulty and risk - caused by the institutionalisation of beliefs and practices over time. Any changes to be implemented need to be considered initially in terms of context, process and content, to facilitate their chances of success. Outcomes of change are shaped not only by design, negotiated agreements and master plans, but also by chance, power and opportunism. Plans of change, therefore, cannot readily reflect a straight forward and rational process. As such, project managers need to be enthusiastic and persistent, with explicit support from senior management, i.e. a high level sponsor. Giddens (1984) and Orlikowski (1992) suggest that all action results in the initialisation of change, which indicates its level of importance for research. Walsham (1993) argues an IS strategist could be responsible for forming strategy content, understanding its context and for facilitating any necessary strategic change. He could additionally be responsible for reviewing any vision statements to identify the values which act as its foundation and, the organisation's involvement in creating the mission statement.

It is worth noting, however, that project managers do not always have the authority and/or responsibility to determine the strategies needed to address the above concerns. The literature identified many different approaches to strategy, for example, Pettigrew (1990), Quinn (1980) and Mintzberg and Waters (1985). The latter are frequently cited for identifying eight approaches, five of which are relevant here:

1. The planned strategy - detailed planning, normally from a vision
2. The entrepreneurial strategy - based on an individual's control
3. The umbrella strategy - setting guidelines without detailed plans
4. The process strategy - controlling the process but not the content of forming strategy
5. The imposed strategy - determined by pressures outside the control of local management

With the generic requirements of codes needing to be specifically defined at a local level, the umbrella strategy is necessary for that to be implemented (Dodd and Lycett 2001).

Project managers are implicitly assumed to know everything necessary to be able to make the best decisions for their project and, implicitly in the longer term, the developers they manage (Walsham 1993, Woodall 1996). Project managers expect to be neither questioned nor doubted, and, as a consequence, software developers are found to be indifferent, ambivalent, oblivious and even hostile to any new code or framework introduced (Woodall 1996). Woodall further argues a form of communication between a project manager and his team which is free from all distorting influences is what is needed. This, however, is hampered by the power and domination associated with management. Although power has capacity to achieve goals, it is not restricted solely to management, argues Giddens (1984). The influences destroy any trust previously held in a manager and can only be re-established with a quality relationship between the software developers and their project manager. Woodall (1996) defines this quality relationship as:

> 'Where both parties are committed to and see as legitimate, the reciprocal rights and obligations as realised through their interaction.'

As such, codes and frameworks should induce mutuality between software developers and project managers to encourage co-operation and reduce any bad faith. It is no longer possible for authority to be effectively forced upon others or held by an individual. Walsham (1993) argues it is essential for managers to be aware of the subtle difference between autonomy and control if they wish to manage successfully. It can be argued that project teams determine the success of a project and consequently, their involvement should be identified and agreed when designing an ethical framework which they are then expected to follow. Caution should be taken though by software developers against reference to a project manager's own ethical code, as it may be misleading (Timpka 1999). This conflicts with manager/developer relationships where developers do as instructed by their managers, but the stance is supported by many philosophers, such as Bentham, Godwin, Locke and Plato. They argue the greater good of the community takes priority over any local management needs. An authoritarian approach that transcends local management towards meeting that objective, however, is not the best way to go about it. This is due to it entailing regulation and control which would hinder the very practices it was trying to encourage.

A role of a developer which includes analysing a proposed new IS, for example, requires ethical consideration. He may take on one of four roles, as described by Hirschheim and Klein (1989), each with a different focus on end-users. The roles are described as (a) the system expert, (b) the emancipator or social therapist, (c) the labour partisan and, (d) the facilitator. The system expert role (most common) only considers the system design from a technical perspective, whereas the emancipator role focuses on free and open discussion by all parties involved - regardless of any power hierarchies in place. As a labour partisan, the developer tries to balance power equally between the end-users and management, although in practice management impose a decision on the end-users when a consensus cannot be reached. A developer working as a facilitator focuses on developing a system with approval from all parties involved, else the system is not considered to be legitimate. Walsham (1993) argues this latter role is unworkable due to our natural human instinct to be critical after an action has been monitored and its consequence identified. With current IS seen to be large and complex, the role of a system expert is necessary, but coupled with the emancipator role to ensure the mutuality relationship described above. The labour partisan role is insufficient as decisions are not continent on democracy - i.e. a majority vote - when a consensus cannot be reached.

2.3.2 Acknowledgement of Cultures Present

Much progress in software development methods has been made since the 1970s, but problems are still large, expensive, sometimes fatal and all too frequent, as described above. Many problems are blamed on poor management, which adds further support to the inclusion of cultural and sociological considerations in the design process (Dakin 1996, PC Week 1999). These are known to help (and hinder) the success of a software project and in response, diverse employee populations are now encouraged in the workplace. This diversity includes gender, age, race, social class, ethnicity, religion and specialist groups. A project team so disparately constructed is also expected to be more sensitive to identifying and meeting the needs of the users of a new IS (Abi-Raad 1999, Grodzinsky 1999).

Inter and intra cultural differences in project teams may be enlarged by technological advances - instead of erasing them (Dozier *et al* 1996, Johnson and Smith 1999). One reason for this is that project teams become insensitive to signals from their environment (Woodall 1996). She argues that subcultures must be allowed to exist for IT companies to succeed. This is due to the fact that successful companies now consist of a series of partners arranged in such a way that all partners benefit over time. Companies are no longer the monolithic structures that seek to maximise profit from each and every project. Cultures and subcultures will continue to exist anyway, proposes Woodall (1996) and Walsham (1993), regardless of any ethical codes introduced by project managers, as no team exists with a homogeneous culture present. Managers of cultures assume those cultures to be identifiable, static and manipulable, but they may in fact be resilient, distinctive and resistant to change. Cultures are an active, living and changing phenomenon. Their total control (probably infeasible anyway due to language anomalies (Symons 1990)) by management is not encouraged, argues Walsham (1993), as the overall benefit would be decreased. Subcultures which are less strong and less coherent facilitate project success as they accommodate diversity and prohibit cultural imperialism (Johnson and Smith 1999). This contrasts with the 'excellence' movement, argues Woodall (1996), which advocates strong cultures are needed to increase *esprit de corps*: project team ownership, commitment and strength of purpose.

Ethical diversity prevents a consensus from being formed and therefore a universal framework is not possible, argue Klein *et al* (1990) and Woodall (1996). This is contended by Johnson and Smith (1999) who argue the identification of a possible consensus could be turned into the foundation of a company-wide framework. As such, project managers should encourage and support existing subcultures, resulting in two-way communication highlighting consensus. Cultural plurality should then enable the ethical judgment of something from several bases and the transition can be made for rational choices to move from the realms of ideology to reality (Smircich and Stubbart 1985, Walsham 1993). Kling and Iacono (1989) argue that cultures can hinder or facilitate ethical change, described as the intangible dimensions of social context. A metaphor of culture is offered by Young (1989) as simply *tension* - between the need for co-operative action necessitated by unity and, fragmentation, implied by distinctive subcultures. Walsham (1993) differentiates the cultural metaphor from the political metaphor described above, with the key elements shown in Table 4 below. With contemporary project teams now typically consisting of developers located internationally, the accommodation of unavoidable diversity is necessary to facilitate project success. *Esprit de corps* still exists, but redefined to reflect the inter-continental displacement of developers.

	Cultural	Political
View of Organisations	Organisations as patterns of symbolic discourse and action.	Organisations as loose networks of people with differing interests.
Some Key Ideas	Culture is an active living phenomenon through which people create the world in which they live. Subcultures maintain distinctive character and ascribe different meanings to the same events.	Power is intrinsic to all human activity. Exercise of power is continuous with subtle, local properties. Morality is involved in the exercise of power. Can include domination, but this is never total.
Management	Cannot control culture, but can influence its evolution. Need to manage for multi -perspectives.	Need to actively manage the precarious balance between autonomy and control at multiple levels.

Table 4. Elements of Cultural and Political Metaphors

Politics within an organisation plays a key role alongside cultural recognition, therefore, with much research conducted in this area by Giddens. He introduced Structuration Theory which joined the previously separate entities of humans (and their actions) and the structure of social systems - invariably synonymous with process and context, respectively. He found structures draw on human interactions and in doing so, produces and continuously reproduces the social structures. This can be illustrated schematically as shown in Table 5 below, whereby human interaction and social structure are broken down into three dimensions and then interlinked via respective methods or manners. Firstly, human communication interprets knowledge known about the actions of self and others, resulting in structures of meaning regarded significant. Secondly, power is utilized in interaction by drawing on facilities such as human or material resources, leading to the creation and reinforcement of structures of domination. Finally, actions are sanctioned by drawing on norms or standards of morality, thereby maintaining or amending social structures of legitimation.

Interaction	Communication	Power	Sanction
(Method/Manner)	Interpretation	Facility	Norm
Structure	Signification	Domination	Legitimation

Table 5. Gidden's Structuration Theory

Walsham (1993) argues that IS in organisations incorporate interpretive schemes, provide control and co-ordination facilities and, harbour norms and standards of morality. Information systems are therefore deeply implicated in the methods and manners that link social action and structure, which are drawn on in interaction, resulting in the confirmation or amendment of the social structures identified: Signification, Domination and Legitimation.

Although the cultural factors of family, class and race have an impact early on in IS professionals' lives, competent and professional project managers are what is ultimately needed to address the problems in IS identified above, argues Dozier *et al* (1996). A thorough cultural understanding is needed if ethical decisions are to be made, due to the complex differences found in the global marketplace in which we all now work (Ford *et al* 1996, Walsham 1993). The cultural factors for consideration when designing a new ethical code or framework, found Ford *et al* (1996) include: language, level of education, law, politics, concept of time, social organisation, technology, values and attitudes. The needs, wants and motivations of each should be determined, which would reduce the conflict that exists in all types of cultures (Johnson and Smith 1999). A developer's best intentions may be judged by others to be unethical, but whether the developer was unethical or simply a victim of circumstance should be clarified (Dakin 1996, Gobold 1999, Introna and Pouloudi 1999). Additionally, Montaigne argues that what one developer considers to be unethical may actually be judged by others to be ethical (Oliver 1998). An example of this is the hiring by project managers of - cheaper - software developers in India by UK companies. This is happening for two reasons. Firstly, because salaries are significantly cheaper in Asia and secondly, because there is a shortage of skilled developers in the UK (Dodd and Lycett 2001).

The scope of unethical practices crosses not only geographical boundaries, therefore, but also social contexts (Walsham 1993). As such, cultural factors clearly need to be considered, but whether ethical values are held autonomously by developers needs to be determined and is now investigated, followed with an assessment of the suitability of ethics as something which can, or cannot, be taught.

2.3.3 Nature Versus Nurture

Grodzinsky (1999) argues we have three views of ourselves: the perceived self, the real self and the ideal self. These perceptions describe how we see ourselves, how we really are and how we would like to become, respectively. (The philosopher Proudhon believed it man's destiny to strive to create his ideal self (Oliver 1998).) The latter perception suggests a desire for positive growth and the development of practical wisdom. This view coincides with Kant's view on how ethics should be taught - as a list of principles describing how to behave in specific situations (Grodzinsky 1999). This argument is rejected by Aristotle who attested to behaviour being autonomously controlled by each individual and thus ethical behaviour can be practiced in any and all situations. This contrasts greatly with the predetermined behaviour for a finite list of scenarios as proposed by Kant. Most notable philosophers fall into one of these two categories; for example, Locke is supportive of Aristotle with Proudhon and Socrates supportive of Kant. Before judgment can be made as to which is possibly right, a look at autonomous human behaviour is necessary. The question 'What are we?' is one of the questions which has kept philosophers busy for almost three thousand years, although Smith, Locke and Proudhon, in conjunction with Muslims and Hindus, argue man is naturally good, considerate, non-violent and sociable - with many more philosophers arguing against. These include: Erasmus, Socrates, Hobbes, Kant, Descartes and Foucault, in conjunction with Christians and Buddhists. They found man's natural behaviour to include being: selfish, immoral and irrational. Philosophers have been active in their quest to additionally determine if rationality and reason (the intellect) are separate from our senses. Philosophers in the period known as Enlightenment argued they were separate from our senses, whereas the succeeding philosophers, the Romantics, strongly argued they were not. The quest for an answer to this quandary continues to this day. If a separation is found to exist, that would support the argument that we can learn without experience. What this means in the context of this research is that it would be possible for developers to learn to become ethical as they are not under the control of their individual senses. This supports the former group of philosophers from the Enlightenment era, which a modern-day philosopher, Habermas, argues the benefits of which still need to be fully realised (Oliver 1998). The overall stance taken here by the author is that man is frequently content with his real self but, is also able to perceive - and does strive for - his ideal self. Man is also naturally good, considerate, non-violent and sociable, yet the pressures and environment of everyday modern life make other, less desirable attributes, surface - i.e. he is a victim of circumstance, as argued by (Dakin 1996, Gobold 1999, Introna and Pouloudi 1999).

2.3.4 The Teaching of Ethics

Most decision-making needs to take account of strategy, finance, ethics and, invariably deals with uncertainty. IS introduces new factors into decision-making and ethics in software development is certainly non-trivial and non-simplistic (Abi-Raad 1999, Grodzinsky 1999). When faced with an ethical conflict, being told to change your ethical hat to a managerial one is not the best solution (Gobold 1999). Nor is the easier option of ignoring such issues completely (Introna and Pouloudi 1999). Collaboration is needed to include values across personal, professional and organisational boundaries, whereas priority in the past was predominantly given to the latter (Pouloudi 1997, Praxis 1999, Timpka 1999).

The teaching of ethics is best done by experts - ethicists, who study ethics and only ethics - even argued Socrates, almost 2,500 years ago (Oliver 1998). Even then, argues Montaigne, Sartre and others, including the teachings of Taoism, it is not possible to understand human behaviour completely as it is too complex. The corollary being the impossibility of creating a taxonomy for use in developing a code of ethical behaviour, whether it be for a finite list of scenarios or guidance for general behaviour. Walsham (1993) contests this, however, having found similarities in people's behaviour and interactions over large periods of time. The following professional organisations also refute this finding by creating their own Code of Ethics (for software developers): the Institute of Electronic and Electrical Engineers (IEEE) and Association of Computing Machinery (ACM).

If professional teaching has its outset at university, then the responsibility of teaching ethics lies with doctors and professors, although inclusion in a student's first-ever lesson on a computer at junior school might be preferred (Dakin 1996, Grodzinsky 1999). But being novices in the study of ethics, it is easily conceivable that ethicists should be employed as a better alternative to academics (Abi-Raad 1999, Baldwin 2000, Grodzinsky 1999, Pouloudi 2000). Socrates certainly thought so, the reason being that students may take on the stance adopted by their - untrained - professors (Sommerville and Dalziel 1998). The employment of ethicists would ultimately enable software developers to make an informed decision when considering the abandonment of an ethical stance in favour of a business, technological or logical stance, as is common practice (Gobold 1999). Additionally, a classroom environment would provide a safe environment to explore one's own ethical values (Grodzinsky 1999). An awareness of ethics should have a positive influence on a developer's professional conduct (Pouloudi 1997) and ideally ethics should be a component of every module taught (Botton 2000, Pouloudi 2000). This view of teaching ethics assumes its grasp, however, by students when included in their syllabus, although computer science students may not have the interest or the aptitude for such issues (Sedlet 1999). Dewey, (supported by Macredie 1999), argues the solution is interactive learning to get the student's mind engaged in the material, as against traditional rote learning (Oliver 1998). Interactive learning, then, by trained ethicists, is the preferred method of teaching ethics (Dodd and Lycett 2002).

Developmental psychologists argue, however, that an individual's values are determined and established by the time he reaches the age of seven and it is this immature set of values that are taken into the workplace many years later (Hebel 2000). Ethical judgments initially made by managers and developers as children are influenced by those around them: parents, other adults, printed media and, to a lesser extent, the television. Reinforcement or punishment of a particular behaviour also influences the development of future behaviour. This is known as social cognitive theory in psychology. Vicarious learning, also from social cognitive theory, advocates modeling - the supply of information regarding certain behaviours and their respective consequences. Reasoning also develops with age, as experience and education increases (Dozier *et al* 1996). These authors also found other influences on behaviour, known as ethical carriers, to be foreign visitors, employees of multi-national companies, the media and environmental groups, for example. Ethical awareness and education also comes from discipline, socialisation, development, clarification and emotional and character formation (Van der Ven 1998). Ambrose *et al* (1997) and Woodall (1996) argue that there are no comprehensive theories, however, to account for the development and change of any ethical values that a person holds. They argue this is true for influences both within a person, and externally. These external influences include inducements, coercion, persuasion or threats. Punishment for unethical practices, therefore, may not result in corrective action.

Although ethics is a relatively new dimension in IS, a broader and more integrated approach is necessary to enable the encapsulation of the many contributory disciplines. These include not only technology and philosophy, but also psychology, anthropology and sociology (Grodzinsky 1999, Sedlet 1999). (A list of other associated disciplines can be found in Appendix G.) This might culminate in what could be called 'best ethical practices'. As ethical principles covering every conflict of interests that could ever arise cannot be feasibly known or taught, Aristotle's advocation of ethical behaviour being driven from within each of us becomes appealing. If true, the need for an undergraduate course would be unnecessary as developers would naturally know how to behave in an ethical manner. This suggests, again, the unethical practices conducted by a developer make him a victim of circumstance and not an unethical person *per se*. This raises the issue of whether we manage our ethical values or whether they manage us - a corollary of the former being self-regulation for each developer and therefore, eliminate any possible consistency of ethical practice across the profession (Svensen 1998).

Managers and developers use ethical frameworks to try to understand the world they live in and then to judge decisions made by themselves, colleagues and management. Frameworks also influence their reaction to an ethical decision - they might discuss it with the aim of finding a resolution in the case of conflict, report it, ignore it or even resign (Ambrose *et al* 1997, Hasnas and Smith 1999, Johnson and Smith 1999). Ethical reasoning and attitudes, therefore, need to be acknowledged as separate (Dozier *et al* 1996, Johnson and Smith 1999). Reactions to a process or outcome differ according to whether the focus was on the process, the outcome, or both. A project manager may focus on a process being ethical, for example, whereas a developer may focus on the outcome, with each unaware of the other's (differing) focus. An unfavourable outcome is more welcome by a recipient when the process leading up to it is considered fair (Ambrose *et al* 1997). The distinction between reasoning and attitude is necessary for project managers to understand the cause of any conflict, argue Dozier *et al* (1996). It is also possible for a number of forms of ethical reasoning to conclude with the same ethical attitude. This is achieved when there is input from both the project manager and his team of software developers in a decision-making process.

2.4 A Critique of Existing Codes

2.4.1 Use of Language

Habermas, Heidegger and Lyotard argue communication is the key to increasing awareness. As English is spoken by the majority of the world's population it has, therefore, has been largely adopted as the language for business - to facilitate wide understanding. But languages are difficult to learn and use and English is no exception, also argues Oliver (1998). Language anomalies also hinder understanding, increasingly so for those whose mother tongue is not English (Symons 1990). Walsham (1993) highlights the content of vision statements as being particularly ambiguous, for example, but due to their rather general nature. Native speakers can exacerbate the problem by deliberately making mistakes. For example, at a presentation given by one project manager to his 20-strong development team, he spoke of delegation as the transfer of responsibility and not authority, which is incorrect (Anderson, 2001). Metaphors contribute to the problem, with an example of a negative metaphor being 'professional'. This is considered negative as it suggests professional knowledge is above and inaccessible to end-users themselves, enlarging the divide between end-users and IS professionals (Walsham 1993).

Montaigne argues each reader of a code will have his own unique interpretation of it (Oliver 1998). This argument is also supported by Macredie (1998) and Paul (1994), with each interpretation determined based on personal subjectives, experience, skills and conjectures, as found earlier. Much care is therefore needed to minimise, if not omit, ambiguity and confusion. Occam and Erasmus both argue 'Less is more' (Oliver 1998). Although a language as rich and colourful as English will never be without problems, that should not deter a concerted effort from being made to produce a meaningful and useful ethical code which is consistent, complete and as unambiguous as possible. The answer, argues Zuboff (1988), resides in the introduction of a new vocabulary. With a new vocabulary requiring definition and consistent and accurate understanding of it, the author feels the problems of ambiguity and confusion would still be prevalent. A new vocabulary, therefore, is not the way forward (Dodd and Lycett 2002).

2.4.2 Common Weaknesses Found

Although frameworks and legislation aim to enforce ethical practices beyond ordinary morality, high profile cases in the media show the lack of success in the regulation of satisfactory IS practice (Davis 1998, Johnson and Smith 1999). An attributable factor for this, found Hasnas and Smith (1999), is that the application of ethical frameworks in IS are more complex than those applicable to the business community. Conversely, the oversimplified instructions contained within a framework may substitute the ethical issues, but be of little practical use in resolving the issues at hand (Johnson and Smith

1999). The problem of knowing how much detail is the right level of detail - without being too vague or complicated - then needs to be determined. General guidance on ethical practice, with specifics explicitly left to local practitioners to define, therefore appears to be the solution (Dodd and Lycett 2002).

Hasnas and Smith (1999) attest the failure of ethical codes and frameworks to several factors. Firstly, the use of philosophical language in frameworks makes them difficult to understand. Secondly, frameworks commonly conflict with the duties and responsibilities of both project managers and software developers. Johnson and Smith (1999) additionally argue knowledge of the different moral philosophies on which frameworks are based is needed by project managers for an informed choice of framework to be made. These authors also note the creation of a hybrid solution by project managers should be avoided as finding the optimum solution would then not be possible. It is the preferences of the code designers, however, that determine its content, argue Johnson and Smith (1999) and Walsham (1993). Even if the correct choice of framework can be made, argue Hasnas and Smith (1999), managers and software developers do not make ethical decisions acting as free individuals, but as agents for their company. Drawing on the evidence above relating to management training now requiring a change of focus from corporate strategy and profit, open consultation across all levels is now advocated to facilitate the successful implementation of an ethical code.

The limitations of current frameworks are additionally due to the complexities of society in which we live, and their affect upon social life consequently being problematic (Johnson and Smith, 1999). One such consequence is the common perception of a company appearing unfair to its software developers - and possibly management - in comparison to other companies. Frameworks provide a vehicle for the comparison and confirm that perception, or not (Ambrose *et al,* 1997). Another consequence is that of liberated and empowered software developers challenging any frameworks introduced by their respective project managers, though this is questionable in practice (Buchanan and Hucyznski, 1991). The point, however, is that ethical standards are individually and autonomously ascertained as a result (Hasnas and Smith 1999, Johnson and Smith 1999, Woodall 1996). This individual setting of ethical standards, however, argues Ford *et al* (1996) is the only way for company goals to be achieved. While project managers may try to impose their own frameworks, software developers must be allowed to achieve company goals their own way. In this vein, Woodall (1996) found that resistance to frameworks included:

- Protection of job status
- Need to work with new people
- Opposition to arbitrarily increased workloads
- Clarification of new working practices

It could be argued that other company processes could be used to institutionalise ethics, such as recruitment, selection and training, but each of the processes need the ethical standards to be defined and agreed first in a framework for reference - implicitly introducing the need for good documentation (Kling and Iacono 1989, Johnson and Smith 1999). Institutionalisation here is defined as a sequence of stages, influenced and conditioned by previous events and actions - leading to progress, of any description, being slow (Walsham 1993).

Frameworks should not be created by project managers based on their own preferences, argue Johnson and Smith (1999). This is due to the subsequent ambivalent and hostile feelings felt by their project teams and in the worst case, the failure of a framework as a consequence of excluding the team in the design process (Woodall 1996, Walsham 1993, Johnson and Smith 1999). In addition, the lifeblood of any project, i.e. enterprise, initiative and creativity, are likely to be lost when compliance to a framework is based on reprisals - or just excessive managerial control (Walsham 1993, Woodall 1996). Woodall also argues that software developers cannot be expected to own a culture change in the workplace enabled by an ethical framework which was initiated, managed and implemented solely by a project manager or other outsider. The goal of implementing an ethical framework successfully is hampered further when a project manager is perceived not to adhere to the framework himself. Markus (1983) additionally warns of team involvement during the development of a code when management has previously decided on its content. This situation would result in increased resentment as the developers would be aware of their *prima facie* contribution and the evident reality of none. Walsham (1993) points out, however, that in case studies carried out where issues were presented to stakeholders involved with options available regarding the way forward, commitment could not be obtained towards some necessary action. It is not known whether a majority vote decided the way forward or whether management imposed a decision. Open discussion across all levels was highlighted above as the best way forward at this juncture.

For a framework to be successful some changes are required of the developers. Change is seldom welcome, but must be accommodated to prevent the introduction of a new framework from becoming futile. The stages of change people pass through when confronted with change typically consist of: denial, anger, abandonment and adjustment (Woodall 1996). This is no surprise, argue Johnson and Smith (1999), as feelings and emotions are two essential attributes of a person. People progress at different speeds through these stages, known collectively as a Personal Transition Cycle, and performance and/or productivity may fall during this time. The change is traumatic, she argues, because previously acceptable behaviour (sanctioned by management) is no longer acceptable. Symons (1990) argues this trauma is possibly also due to the perception of enforced change, without clear and justifiable reasoning, from a way of working which they were reasonably happy with. Consequently, feelings of apprehension, anxiety, cynicism and stress are also common. Other feelings include: uncertainty and insecurity. There may also be a feeling of infringement on personal privacy, autonomy, self-esteem and equal treatment, which all need to be addressed. The time needed to adjust is usually not respected, and the trauma is dealt with by constant criticism of the old ways. The right to hold views other to those now required is also not respected. The correct way of handling these difficulties is by counseling or to create a process which enables the expressions of anger and frustration to be heard (Woodall 1996). Both take time and money, scarce commodities in any competitive company. One solution advocated by Ford *et al* (1996), to reduce the stress suffered, is to involve the developer's spouse and children in the training given. As we have seen from issues raised above, many of these problems can be overcome with open discussion, resulting in general agreement and commitment from all involved.

2.4.3 Requirements of a Code

As a result of the problems associated with adopting codes with strong theoretical foundations, project managers are now creating their own codes, but with no guarantee of alignment with codes previously put forward (Hasnas and Smith 1999). These authors also argue the success of a new framework is also dependent not only on the software developers but also their company and societal contexts. Furthermore, IFIP (1995) found enforcement of frameworks to be frequently weak, with one cause found to be the often lacking explicit or consensus-based rules describing what is, and is not, appropriate behaviour supporting the framework (Hasnas and Smith 1999). This is caused by the general nature of codes, the consequential ambiguity of their requirements and (necessary) lack of detail for implementation at a local level.

Business ethicists created their own codes in an attempt to address the problems outlined above. The codes are called NTBEs (Normative Theories of Business Ethics). As normatives they describe what managers and their employees should and ought to do, and have been written in everyday English. There are three main NTBE theories: stockholder, stakeholder and social contract theories. No more than one NTBE can be used at a time and ideally should not be altered from its original state. If the NTBE selected by a manager must be altered in any way, it should be done before its application and only then. NTBEs are not designed for use in the public sector or for non-profit making companies (Hasnas and Smith 1999). They are all designed for use in the same domain: by profit-making companies in a competitive arena and as such, offer ethical guidance in that environment. At a practical level, Hasnas and Smith (1999) also argue, more information is needed before they can be used effectively. Even then, they only act a guide to the individual conscience, independent of any ethical framework being instilled in addition by a manager on his team. This independence enables NTBEs to be used as a standard from which employees can judge themselves, their manager's decisions and their company's policy/culture/code of conduct. As such, NTBEs offer helpful guidance that exceeds that offered even by traditional philosophical theories (Hasnas and Smith 1999). Due to the weaknesses identified earlier, however, the values advocated in current NTBEs need further clarification.

It is sometimes necessary for a disastrous failure to ensure the views of stakeholders are considered and accommodated in the creation of a code - or a revised attempt, argues Symons (1990), although constant failure in IS has not led to any workable solutions accepted across the profession. Walsham (1993) found the successful creation and implementation of a new code are not disparate activities, but in fact intertwined. It is necessary, however, to decide who will determine what is ethical and what is not, what information is needed, where it will be obtained and how. Dozier *et al* (1996), proposed the practices which need to be determined as ethical in the workplace in the context of managed software development can be broken down into six categories:

1. Project manager's or software developer's behaviour affecting the

company

2. Company or project management practices affecting software developers
3. Software developer's behaviour affecting other software developers
4. Company practices affecting customers
5. Company practices affecting shareholders
6. Company practices affecting the general public and the community

Hasnas and Smith (1999) recommend the creation of a new framework or, the amendment of an existing code. The role of language on a person's behaviour should also be accommodated, argue Ford *et al* (1996). Codes previously created by professionals and laws should also be considered as they are as important as any moral theory (Davis 1998). The setting of new goals, milestones and benchmarks, however, serve to legitimise good working practices, and remind developers that old standards were unsatisfactory (Woodall 1996). An example of an appropriate amendment could be the inclusion of an ethical checkpoint in the development life cycle model being used (Hasnas and Smith 1999). Care should be taken though, not invade anyone's privacy by measuring behaviour and attitudes in addition to their tasks (Woodall 1996). Habermas argues the increased understanding of the world we are continually acquiring also weakens the value of codes (Oliver 1998). Yet completion of a framework's acceptance follows a period of integration and consolidation ending any uncertainty and insecurity an employee may have felt toward his work task or relationships (Woodall 1996, Walsham 1993).

In an attempt to address the concerns highlighted above, two professional organisations - the IEEE and ACM - jointly commissioned and subsequently adopted a Code of Ethics specifically for software developers. A description of the Code, its structure and implications for use, are provided below.

2.5 The IEEE/ACM Code of Ethics for Software Developers

2.5.1 Background

An ethical code specifically for software developers was created by the Institute of Electronic and Electrical Developers (IEEE) and the Association for Computing Machinery (ACM) in an attempt to address the weaknesses of other codes and ultimately reduce the problems in ISD. The Code was chosen for this research as the IEEE is the world's leading organisation of computer professionals and has the vision of being the leading provider of information and services to the world's computing professionals - achieved by fostering communication, co-operation and information exchange among its members. Further information about the IEEE/ACM and the Code can be found in Appendix B.

Having a code in place helps establish software development not only as being a profession, but also identifies it as being different from other IT areas. (After all, if software development is not regarded as a profession, then any kind of behaviour is surely acceptable (Dakin 1996)). With so many categories of codes and attributes of ethics proposed, the appropriate category(ies)/attribute(s) specifically for software developers needs to be determined. A study was carried out by a joint steering committee called SEEPP - Software Engineer's Ethics and Professional Practice - commissioned by the IEEE and ACM, with their conclusions being adopted in the Code of Ethics and Professional Practice for Software Developers which was published in December 1998. A new committee called SEPEP - the Software Engineering Professional Ethics Project - was then created to nurture the impact of the new code - by introducing educational material for distribution to practitioners (Gotterbarn 1999a/b). This should facilitate the recognition, understanding and ownership of the Code by software developers worldwide, necessary for its success. The material to be distributed need not only be paper-based, however, argues Svensen (1998), but could also be in the form of an intranet/extranet, the internet, interactive TV, multi-media kiosks, freebies, compact disks, 3½" floppy disks, video conferencing, workshops and/or seminars.

2.5.2 Description

Software developers from all continents were consulted during the Code's creation, culminating in a code which has global consensus for representing the standards of behaviour expected of professional software developers, by professional software developers (Gotterbarn 1999a/b). The core element of the Code is the focus on public interest prevailing at all times, with regard to health, welfare and safety. Mandatory compliance to the Code is not sought, however, only co-operation. Although this may initially appear ineffective in practice, this approach is necessary, argue Hebel (2000) and Pouloudi (1999), as it is not possible to change a person's values - only a change in their priorities is possibly achievable. Demanding adherence would be ineffective, therefore, even if only seeking co-operation lays the Code open to abuse (Introna and Pouloudi 1999). The preamble to the Code stipulates it should not be read as a finite guide to ethical behaviour and, is different from other codes proposed in four ways, in that it:

- Provides guidance on ethical principles
- Provides a decision-making strategy
- Addresses conflict in ethical values
- Addresses three levels: humanity, professionalism and professions

The guidance on ethical principles includes, for example, a requirement for developers to be fair and avoid deception. The words 'avoid deception' were chosen over the initial choice of 'be honest' as the former was later considered to be open to exploitation. The decision-making strategy also acts as a framework by which a decision made can be judged by others to be ethical or not. Conflicting ethical interests are accommodated in a set of principles itemising pertinent areas in a prioritized order so that the most important areas of interest can be identified. This prioritizing approach is also supported by Baldwin (2000), Halang (1998) and Pouloudi (2000). Dilemmas in ISD are not easily settled and this is acknowledged by the Code which seeks a satisfactory solution by identifying and addressing the three levels identified above: humanity, professionalism and profession. This is achieved by detailing, respectively, how (a) to aspire to be human (e.g. strive for integrity and justice), (b) to expect to be a professional (e.g. identify general professional obligations) and lastly, (c) how developers should expect to use good tools (e.g. explicit testing and documentation procedures).

An additional function of the Code is to educate software developers into promoting and protecting positive values, without encouraging any whistle-blowing activities. It also promotes education, training, support, guidance and inspiration (Timpka 1999). The Code is produced in two forms - one is high level and aspirational, the other is low level and detailed. Ethical standards are described, as well as managerial and technical, although the latter category has since been disputed. For a developer to be ethical he must be able to argue his viewpoint from an ethical stance. The Code provides him with ammunition to argue his case and to be able to say, 'No' when a situation requiring a compromise on ethical standards arises (Gobold 1999, Singer and Vinson 1998). Stakeholders in the development of new IS are identified in the Code as managers, clients, suppliers, communities, other professionals, other employees and society (Gotterbarn 1999b).

It is worth noting that although the Code was created after lengthy consultation with software developers in the international arena, the IEEE/ACM do not now hold a monopoly on ethical behaviour. This joint body is still a 'player' amongst players - who may take the form of other discipline professionals or cultures worldwide, probably with their own codes of ethics (Abi-Raad 1999). The existence of other codes is not the only problem faced by anyone attempting to create a new one. Other problems include differences in language and even strong negative feelings between nations (Hebel 2000). The approach taken by IEEE/ACM in consulting the software developers is supported by Singer and Vinson (1998), arguing that without this, the Code might attempt to impose practices on the - global - profession which are inappropriate at a local level.

2.5.3 The Eight Principles

For the Code to succeed in achieving its goal of making the software development profession beneficial and respected, it sets out eight key principles to which developers and project managers are requested to adhere (Gotterbarn 1999b). A brief summary of each is thus provided and sets the context for the thesis:

1. Public. Software developers shall always behave appropriately to public interest - defined as health, safety and welfare
2. Client and Employer. Developers will always behave in the best interest of their employer and client - as long as it is consistent with public interest
3. Product. The highest professional standards possible will be sought by developers for their products and their respective modifications
4. Judgment. Developers will exercise integrity and independence when making professional judgments
5. Management. An ethical approach will be promoted and exercised by project managers and team leaders in the management of software development and maintenance
6. Profession. The integrity and reputation of the profession will be increased, consistent with public interest
7. Colleagues. Developers will be fair and supportive toward their colleagues

8. Self. Developers will take part in life-long learning of the profession's practices and promote an ethical approach to the practices used

2.5.4 Practical Implications

A credible and used code should inspire confidence in both practitioners and users alike, culminating in an enhanced image of ISD (Gotterbarn 1999a, Timpka 1999). The Code provides a measurement tool by which a developer's behaviour can be judged - by himself, his peers, management, end-users or the public, although the eight principles of the Code can be amended as necessary when found to be inappropriate. Whether ethical standards provide a minimum or maximum level of acceptable development practices still needs to be ascertained, although an ambivalence of opinions does provide an arena for the right answer to surface - assuming such a thing exists (Abi-Raad 1999, Dakin 1996, Gotterbarn 1999a/b).

The English language creates ambiguity which adds confusion to the difficulties of implementing the Code, as noted previously. For example, in the eight principles above, the description of Principle 3: 'Product' requires the highest possible standards to be sought by developers. The Code attempts to clarify these standards as the striving for high quality, acceptable cost and a reasonable schedule. The definitions provided to reduce the ambiguity present actually augment it as the definitions introduce additional wording which also needs further clarification: striving, high, quality, acceptable and reasonable. How these will be defined on a practical level will invariably be different for every person who attempts to define them, with their definitions individually, timely and contextually determined. Principle 5: 'Management' requires an ethical approach to be promoted. Again, a definition is provided but this also needs clarification in order for the requirement to be correctly adhered to. Definitions provided requiring further definitions dominate the Code. Principle 6.08 requests that only significant errors be corrected and reported. Again, what a significant error is, how it should be corrected and to whom it should be reported, will be interpreted differently by every reader and is, as noted above, invariably individual, contextual and timely in definition (Abi-Raad 1999, Baldwin 2000, Dodd and Lycett 2002). Furthermore, Pouloudi (1997) argues reporting an error, or conflict of any kind, might not always be beneficial. Although an ambiguous code, refinements can and will be made, but only by experimentation and subsequent learning (Dakin 1996). Repeating an action the same way and hoping for an improvement in the results is highly unlikely - partially effective standards should at least create a demand for better. In this way, the Code will move towards becoming a *de facto* standard and ultimately a *de jure* standard.

The adoption and inclusion of the Code described above in ISD is welcome, but on its own the Code has little value. Control also needs to be exercised so that sanctions can be introduced when a project manager or developer is non-compliant, i.e. unethical in his work practices (Dakin 1996, Gotterbarn 1999a). The IEEE/ACM Code cannot possibly cover everything and does not address the issues of discipline, deterrence or cowboys (Gotterbarn 1999a). Insufficient authority is present to ensure compliance, with only co-operation sought from the developers as a consequence. Gotterbarn (1999b) argues that as the profession acquires knowledge enabling the Code to be modified, standards will improve - even though the revision process takes a minimum of a year for each amendment. Refinements cannot be made to the Code without the help of software developers - who are believed to be bright and therefore willing and able to learn, develop, and share their knowledge and experiences (Svensen 1998). To encourage adherence to the Code in Texas, USA, a local register was established to list compliant developers working in the area. Developers are not encouraged to become isolated ethical heroes however, and ethical boundaries still need to be determined to allow for mistakes to be made - developers are still only human after all (Timpka 1999).

In support of Svensen's findings above, Kelley (1999) found that software developers considered most successful had a common ambition of aiming to meet organisational goals first and foremost - in alignment with management focus on corporate strategy and profit identified above - as against working for the public interest at large which the Code stipulates must take the highest priority and prevail at all times. But for software developers typically employed in the western Europe, remuneration packages are individually tailored and, knowledge is power (Knights and Morgan 1991, Svensen 1998). Unique knowledge is frequently rewarded with bonuses, salary increases and promotion. The consequence is an anti-sharing environment, with the paradoxical situation of knowledgeable developers not wanting to share any of that knowledge - resulting in their 'knowledge guru' status (Cooper 1986, Computing 2000b). The situation might exist therefore, that if a developer only appears to know the same as everyone else, a project manager might then believe he has employed the wrong 'specialised' person for the job. The information could be being withheld, however, due to a fear of it being used unethically by other parties with alternative motives. Information sharing in ISD therefore needs to become a component of a developer's performance so that it can be expected as part of the norm, as against rewarded for its *ad hoc* appearance (PC Week 1999). This will facilitate the identification of weaknesses in the Code and corrective measures being taken where appropriate. Feldman and March (1981) support this tactic as they argue the act of requesting information is actually symbolic of competence - refuting Cooper above. Furthermore, the ramifications of key developers leaving a project prior to sharing their knowledge is the possible significant and long term loss of revenue to the company, damaged client relationships and the loss of best practice(s). Key developers should be retained long enough then for their esoteric knowledge to be shared with the remaining develops in the project, as being productive yet unethical is no solution to the problems outlined above either (Kelley 1999, Hebel 2000, PC Week 1999).

2.6 Summary

Ethics was introduced through its contribution from philosophy and academia and, the unethical practices found in software development were summarised. Management training was found to contribute to the lack of awareness/concern of ethics in ISD, in conjunction with the reward structure of IS professionals in western Europe. The failure of current codes and frameworks was researched, together with their cause. This led to what is necessary for an ethical framework to be successful in ISD. The IEEE/ACM Code of Ethics for Software Developers was examined to determine its structure and objectives. The eight key principles were described which scoped the definition of ethics in ISD.

The focus of management education was found to be primarily on corporate strategy and profit, with a need for change identified by (a) ethical considerations forming an integral part of IS management decision-making (b) a project manager's claim on ethical authority in the workplace is currently questionable, (c) the presence of dispute and conflict in the business environment is constant, as well as the failing of traditional compliance-based management control and, (d) ethical awareness and adherence to a code is essential for project managers if software developers are also to take ethical practices seriously. The values found to constitute ethical behaviour found consensus amoung academics, philosophers and religious groups. In the context of managed software development, the following six categories were identified (a) project manager's or software developer's behaviour affecting the company, (b) company or project management practices affecting software developers, (c) software engineer's behaviour affecting other software developers, (d) company practices affecting customers, (e) company practices affecting shareholders and, (f) company practices affecting the general public and the community.

The issue of whether developers manage their own ethical values or whether the values manage them was discussed - a corollary of the former being self-regulation for each developer which would eliminate any chance of consistent ethical practice across the profession. The latter would result in developers becoming a victim of circumstance whenever an unethical practice occurred. If developers act as agents for their employer and not independently, we have a third possibility of the role of a developer. The stance taken by the author is the former - developers manage their own ethical values. No comprehensive theories were found to account for the development and change of any ethical values that a person holds. These external influences include inducements, coercion, persuasion or threats. Punishment for unethical practices - and non-adherence to a code, therefore, may not result in corrective action.

Frameworks were found to be the most important and most common way of institutionalising ethics. Limitations and weaknesses of frameworks were identified as (a) being highly ambiguous, (b) arbitrarily increasing workloads, (c) introducing unclear/conflicting new working practices, (d) creating a need to work with new people (e) threatening jobs, (f) using alien language, (g) over-simplifying instructions, (h) designed solely by managers or outsiders and, (i) do not address discipline, deterrence or cowboy presence. The creation and implementation of codes were found to be intertwined and, not separate activities, with open communication necessary for codes to gain acceptance. Change is not easy, however, due to complexity, difficulty and risk - caused by the institutionalisation of beliefs and practices.

The umbrella strategy - setting guidelines without detailed plans - was identified as being the most appropriate method to institutionalise ethics in other departments in organisations, such as recruitment, selection and training. Each department would need the ethical standards to firstly be defined and agreed in a framework for reference. Designers typically determine the contents of new codes, which should be avoided via open communication across all levels prior to its construction. Weak cultures - defined by diversity - were found to facilitate project success. Cultures within development teams were ideally found to be diverse and should consist of differences in gender, age, race, social class, ethnicity, religion and specialist groups. Such a development team is expected to be more sensitive to identifying and meeting the needs of the end-users. Four roles of an analyst were identified and with current IS seen to be large and complex, the role known as a system expert was found to be necessary, coupled with the emancipator role to ensure the mutuality relationship described above.

The preamble to the IEEE/ACM Code of Ethics stipulates it should not be read as a finite guide to ethical behaviour and that it: (a) provides guidance on ethical principles, (b) provides a decision-making strategy, (c) addresses conflict in ethical values, (d) addresses three levels: humanity, professionalism and professions, (e) educates software developers into promoting and protecting positive values, without encouraging any whistle-blowing activities and, (f) promotes education, training, support, guidance and inspiration. Language problems typically exist in codes and the IEEE/ACM Code is ambiguous in its requirements.

The following chapter determines which research approach is most appropriate for the objectives identified to be met: to form a critique of the IEEE/ACM Code's suitability as an ethical framework; identify and describe unethical practices conducted by project managers, their project teams and other business units; and determine the cause of the unethical practices identified.

Chapter 3. Research Methodology & the Case Introduced

3.1 Introduction

For the research objectives to be met, an appropriate research methodology needed to be selected. The philosophical stance adopted by the author had to be determined, enabling the research approach used and associated methods adopted - for data collection and analysis - to then be selected, justified and described.

Section 3.2 describes the philosophical foundation taken typically in terms of ontology and epistemology, together with the stance taken by the author. The research approaches advocated by Burrell and Morgan (1979) are then described - Interpretive, Functionalist, Radical Humanist and Radical Structuralist - culminating with the interpretive approach being selected and justified as the most suitable. Section 3.3 provides the identification and critique of the data collection methods associated with this approach, with a six month in-depth case study chosen to be conducted to enable the collection of pertinent data through observation, document analysis and semi-structured interviews. Section 3.4 describes the company selected to participate in the research, relating to its structure, the development environment and the role of the author whilst conducting the research. A discussion on data analysis is provided in Section 3.5, identifying the use of repertory grids as suitable.

3.2 Determining the Research Approach

3.2.1 Philosophical Foundations

The philosophical foundations of research are typically argued for, or against, on the two concepts of ontology and epistemology. Ontology is the science of understanding reality - in context with the study at hand. Doctrines here describe two opposing viewpoints: from the belief that all social and physical entities exist independently of how they are perceived by people, to the belief that all social and physical entities are mental constructions - and everything in between these two values. Epistemology is the science of understanding knowledge and how it might best be learned. Doctrines here also describe two opposing viewpoints: from the belief that all knowledge can be defined by a set of laws and/or facts which can be justified with measurement, to the belief that knowledge is down to subjective interpretation and understanding on an individual level. Both the ontological and epistemological concepts are included below in the description of each of the four paradigms that were considered. The vast and complex 'real' world in which we live leaves the author mid-way between the opposing doctrines of both ontology and epistemology. For the purposes of this research, however, the ontological stance taken was that all social and physical entities are mental constructions, with the epistemological stance taken of knowledge is down to subjective interpretation and understanding on an individual level.

3.2.2 Recent Approaches

At the highest level, research methods can be divided into two camps: scientific and non-scientific - with the former taking a functionalist/positivist stance and the latter taking an interpretive stance. Each (respectively) concerns itself with the collection and subsequent analysis of quantitative and qualitative data, but the clear distinctions determining their area of use are no longer exclusive. As the heart of this thesis is about people (developers and project managers, teams and organisations), a non-scientific approach was taken. This approach consists of four (sociological) paradigms: Interpretive, Functionalist, Radical Humanist and Radical Structuralist, as described by Burrell and Morgan (1979) and Hirschheim *et al* (1995) – although other descriptions exist. The approach enabled an in-depth case study to be used and, a critique of the case study method identified the need for a single embedded descriptive case study. To ensure the validity of the data collected - via respective data collection methods associated with the case study method - and the consequent reliability of the case study - the case study protocol as advocated by Yin (1984) was adopted and is described below. The protocol consists of several components which ensure the case study 'stands up' under close examination.

As academia has been actively searching for answers to numerous disparate questions over a very long period, it is not surprising to find a plethora of categories and approaches already in existence. Although a taxonomy has been created to differentiate between the research categories and methods, there is strong support of a hybrid approach, taking the best from each category as appropriate (Avison 2001, Tashakorri and Teddlie 1998). A hybrid approach has at least three benefits. It can: (a) help to reduce bias, (b) reveal errors in measurement and, (c) verify and cross-check the data collected (Housden 1992, Remenyi 2000, Nissen *et al* 1990, Swetnam 1998). A common problem with research method choices is that local institutions can 'dictate' which category and/or method should be used (Nissen *et al* 1990, Swetnam 1998). This may be more prevalent when research students are employed as research assistants and have their research activities more closely guided. Leedy (1997) argues, however, that it is better for the data requiring examination to determine the approach and method to be used by the researcher and, this view was shared by the author. Two general classifications of many of these categories are possible and, are shown in Table 6 below, with pertinent authors.

Non-Scientific	Scientific
Qualitative (Walsham, Myers, Cavaye, Strauss and Corbin)	Quantitative (Yin)
Subjective	Objective
Post-Positivist	Positivist (Benbasat)
Interpretivist (Zuboff, Orlikowski and Gash, Gibson and Burrell, Cavaye, Walsham)	Functionalist
Phenomenology (Zuboff, Mingers)	Empirical (Layder, Mumford)
Ethnography (Smircich, Van Maanen, Mingers, Suchman, Myers)	Nomothetic
Hermeneutics (Boland and Day, Lee, Mingers, Giddens)	Traditionalist
Idiographic	Interventionist
	Experimental
Radical Humanist	Radical Structuralist
Naturalistic	Constructivist

Table 6. Taxonomy of Research Categories

There are many methods available for use within each of these two generic categories; the most popular are shown below in Table 7.

Qualitative	Quantitative	Others
Case/Field Study	Theorem Proof	Historical
Ethnography	Lab/Field Experiment	Normative
Phenomenology	Case Study	Longitudinal
Experimental	Surveys	
Grounded Theory	Questionnaires	
Subjective	Interviews	
Game/Role Playing	Observation	
Action Research		
Participative		
Observation		

Table 7. Taxonomy of Research Methods

Despite the introduction of NTBEs (as described previously), an attempt to address unethical practices with the application of normative theories is often neglected - in this context normative theories are based on the idea that applicable universal principles can be created and understood by all project managers and software developers (Hasnas and Smith 1999). Although total standardization would result in cultural insensitivities, argue Ford *et al* (1996), it is possible through a prescriptive approach. Such an approach would consist of normative theory being applied from a rationalist standpoint (Johnson and Smith 1999). Though total localisation may be an aim of the framework designer, it is prevented by cost (Ford *et al* 1996). The quest for universally applied principles should be abandoned, therefore, in favour of relativity and subjectivity (Campbell and Marshall 1999). A relativistic approach accepts that any claim to moral authority is highly questionable, and consequently rejects the notion of cultural imperialism. This would enable a focus on one's own interests rather than the common good. This is supported by Johnson and Smith (1999) who agree that ethical obligations to others can only be considered after priority has been given to one's own desires and needs. Though these and other sociological approaches exist, they can be categorised into just four paradigms which are used very frequently (Burrell and Morgan 1979, Hirschheim *et al* 1995, Walsham 1993). These paradigms are: Functionalism, Radical Structuralism, Radical Humanism and Interpretivism. The latter is described below as it was chosen for the purpose of this research, including a brief description of the philosophical stance taken, with the other three described in Appendix A.

3.2.3 Interpretivism

Interpretivism aims to understand the world how it is and not how it could be - at the level of subjective experience. Reality is seen as complex and with a different interpretation of it for everyone. It provides explanations on consensus and social order by referring to interpretations given by participants individually, as against the use of mechanical or biological analogies. The vocabulary used refers to values, issues and concerns, referring to factors which are human, organisational and political. (Note the deliberate use of 'factors' and not 'criteria' - due to its hard scientific aura, emphasizing quantitative and technical data.) Consequently, no objective reality is identifiable for others, for example researchers, to replicate - quite the converse to the Functionalist approach described above. Every experience is individually interpreted with even holistic approaches just providing *prima facie* snapshots of reality for any one individual and, therefore, is subjective (Oliver 1998). This paradigm sees the world as an emergent social process where relevant issues are related to the status quo, consensus, social order, solidarity, social cohesion and integration. Everyday life consists of each individual's understanding, informal evaluations and subsequent perspectives are taken seriously by interested parties (Hirschheim and Smithson 1988). The ontological status is of nominalism (constructivism), where the social world is regarded as questionable and problematic - reality is the product of the mind. The epistemological stance of anti-positivism is taken, with the will and need to make sense of oneself and the situation at hand. An organisation's reality evolves through changing traditions of social laws, conventions, cultural norms and attitudes, with an emphasis on negotiation and clarification to reduce uncertainty. Interpretivism questions whether organisations exist in anything other than the conceptual sense and, shared meanings are the result of inter-subjectivity rather than objectivity. Multiple realities are accepted, as against challenged for resulting from communication problems. Global consensus is not sought if individuals' interests are considered subordinate and everything is relative - only acceptance of the issue at hand is what matters. The weaknesses of Functionalism are addressed with Interpretivism and it is accepted that universal laws and principles cannot be successfully applied. Heidegger (1962) summarised Interpretivism as,

'Existence is interpretation and interpretation is existence'.

Improving understanding by describing the dynamic interactions in the real world as interpreted by the individual stakeholders is the work of the Interpretivist researcher (Walsham 1993). The corollary of this is that qualitative data is invariably collected for analysis, as pages of numbers are unable to convey the rich descriptions and explanations afforded by language and, the subjective experience of individual participants was key to addressing the research objectives identified above.

3.3. The Research Approach Taken

It could be viewed as a paradox that the multiple approaches and respective techniques for research as described above not only enable the most suitable to be adopted by a researcher, but this plethora also creates uncertainty. The corollary being that a reduced number of options would reduce the uncertainty present when a researcher is selecting an appropriate research methodology - of course a reduction would also hinder or prevent the hybrid/pluralistic approach advocated above. Interpretivism was selected now been described, which ultimately meet the need of a PhD to either add something of value to the body of or, to show independent and critical thought:

- Identify the causes of unethical practices in managed software development
- Evaluate the IEEE/ACM Code of Ethics and Professional Conduct for Software Developers as an ethical framework able to address the ethical conflict identified
- Determine how organisational practices, policies and procedures are influenced by the framework
- Determine how the application of the framework is influenced by organisational practices, policies and procedures
- Identify the stance taken by other professions/business areas and the methods implemented to enforce their requirements on their members
- Provide recommendations for improvements to the IEEE/ACM Code

The research design is now able to be shown diagrammatically, and is provided in Figure 4 below.

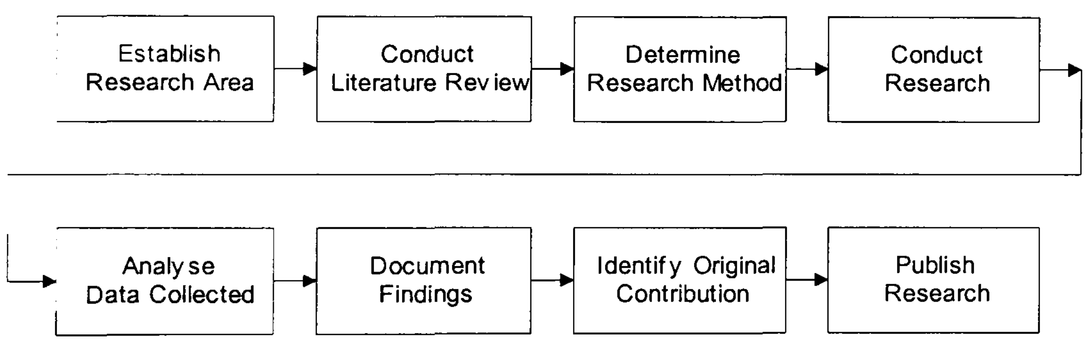


Figure 4. Research Design

3.3.3 A Critique of the Case Study Method

Case studies report on real-life situations in their natural environments, by enabling the observation and subsequent description of facts and relationships - the holistic and meaningful characteristics of real-life events, in the real world, leading to a greater understanding (Klein *et al* 1990, Yin 1984, Yin 1994, Cavaye 1996, Sauer 1993). These might include organisational and/or managerial processes, individual behaviours and international relations. Consequently, case studies come under the remit of both qualitative and quantitative research, which made it an ideal research method in this context (Walsham 1993, Orlikowski and Baroudi 1989, Benbasat *et al* 1987, Cavaye

1996). For this reason, case studies are the most commonly used method in the IS arena (Orlikowski and Baroudi 1991, Yin 1984). In addition, case studies are flexible to a degree with regard to their design and consequently make their use by academics to be both desirable and appropriate. Furthermore, the focus of the research was on a contemporary phenomenon, i.e. adherence to the IEEE/ACM Code as an ethical framework in ISD and, situated in a real life context. Case studies are also the preferred choice when *how* and *why* questions need to be asked - as we know to be necessary in this research. For example, *how* did an unethical practice happen and *why* did it happen? This compares with *who, where* and *when* questions (Housden 1992, Yin 1984). A summary of the relevant situations for different research strategies is shown in Table 8 below, adapted from Yin (1984).

Strategy	Research Question	Requires Control Over Events?	Focus on Contemporary Events?
Case Study	How/why	No	Yes
Experiment	How/why	Yes	Yes
Survey	Who/what/where/ how many/how much	No	Yes
Archaic Documents	Who/what/where/ how many/how much	No	Yes/No
History	How/why	No	No

Table 8. Different Research Strategies

The use of interviews in a case study strengthens the research by supporting or challenging those answers provided by other data collection techniques used. This then addresses the concern of Swetnam (1998) who argues only certain attributes can be measured with the use of questionnaires. As such, a case study was considered to be the best method available for this purpose, as it can be applied to both qualitative and quantitative data, as noted above (Cavaye 1996, Stake 1994). It is this large collection of data that is the strength of the case study method that others, such as laboratory experiments, are unable to match. Sauer (1993) identified the value of a systematic and rigorous case study may then - usually to the researcher conducting the case study - take the form (for the purpose of this research) of:

- Raising problems about the phenomenon of ethics
- Stimulating theories of the causes of non-adherence to ethical codes
- Stimulating the development of ethical adherence mechanisms

Sauer (1993) additionally identifies other advantages of conducting case studies:

- The processes by which non-adherence occurs are shown
- Case studies are a surrogate for costly experience
- The complex social, political and economic context in which new systems are developed becomes more identifiable
- Knowledge gained can be shared through published reports, providing an additional route to understanding through secondary data without having to experience unethical practices first hand

In reality of course, nothing is perfect and in-depth case studies are no exception. Problems can be encountered when the researcher is not able to recall his observations accurately, or does not disclose any important feelings he may have about something pertinent to the research. Additionally, participants in the company may be wary of revealing information which may reflect badly on themselves, their colleagues and/or manager(s) at a later point. The main points of concern raised in academia - which are addressed below - are that case studies (Yin 1984):

- Are less rigorous than other research strategies
- Provide little basis for scientific generalisation
- Take too long
- Produce lengthy documentation

Case studies can be less rigorous when a researcher has been careless or allowed personal views to influence the data analysis and/or final conclusions (Remenyi 2000). This weakness can also apply to other strategies. Single case studies, however, enable in-depth research to be carried out - usually over a lengthy period - which increases a case study's rigour by enabling the examination of continuing processes in context. The significance of various inter-connected levels of analysis can also be drawn. Consequently there is an opportunity to identify the sources of causation and connectivity which are fundamental to the identification and explanation of patterns present in a process. Rigour is also achievable with good organisation and planning. The data in this instance relates to the unethical practices identified. It is paramount, therefore, that the data collected is accurate. Only then can the two strategic uses of a case study be achieved with confidence: to collect evidence and create knowledge. The use of a Case Study Protocol, as advocated by Yin (1984), ensures the integrity of a researcher conducting a case study and the rigour required - further information regarding the Case Study Protocol implemented can be found in Appendix E.

Generalisation from a single case study is often brought into question. External validity addresses this commonly asked question of single case studies: how can they justifiably be generalisable to a much larger, or even universal group, from just one company? Two points address this question. Firstly, the examiner is (implicitly) contrasting the case study method with survey research where the sample used readily generalises to a larger group due to the formal methods of sampling. The analogy to samples and universes is not a scientifically acceptable comparison as survey research relies on statistical generalisation whereas case studies (as with experiments) rely on analytical generalisation. This analytical generalisation aims to generalise a particular set of results drawn from a case study to some broader theory. The statistical representation of case studies for generalising to a universal population, therefore, is not at the core of interpretivist research. Secondly, such a generalisation also needs to be 'scientifically' acceptable (possibly a paradox as the research approach is categorised as non-scientific) and is so when the concluding theory is tested through a 'replication' of the original company environment. When the conclusions of the replicated test show consistency with the initial conclusions, then the theory is accepted for a larger group of similar companies. Again, this argument could also apply to the generalisation offered by other research strategies, such as experiments. The findings and conclusions drawn are then offered for debate and further research.

The latter two weaknesses, that case studies take too long and produce lengthy documents, are more justified arguments than the former two. Case studies have historically taken a long time and consequently produced unreadable documents, but it is no longer necessary to always conduct case studies this way (Yin 1984, Murray 1989, Remenyi 2000, Walsham 1993). A final point worth including here is that the use of the formalities described above in ensuring high quality case studies are more important when a multiple case study is being carried out or, when multiple researchers are involved, or both (Yin 1994). Although rigour was achieved in this research with the use of the case study protocol, neither of these latter concerns was applicable in this instance.

3.3.4 Justification for a Single Case Study

There is wide academic support that any study of people is both complex and problematic - to paraphrase Satre cited earlier, 'Each person is a whole new world,' i.e. vast and complex. In addition, there were only a limited number of resources and a finite amount of time available in which the research needed to be completed for the author - who worked alone. A single case study, however, was expected to reduce both the complexity and size of the problem to a level which would enable quality research to be carried out. An in-depth case study is additionally able to collect large amounts of information from which generalisations derived from empirical results are then possible (Klein *et al* 1990, Walsham 1993, Woodall 1996). Many single case studies have been carried out in the past which have, therefore, made significant contributions to knowledge due to the insights obtained by the researcher(s) (Walsham 1993). This was important as the role of ethics in software development was relatively new and consequently there was not a plethora of research readily available in this - nascent - area. The research conducted can also be continued in the future by other interested parties (Yin 1984).

A case study, then, is a reliable way of investigating an empirical topic, with reliability ensured with the use of the case study protocol. The protocol is a detailed statement setting out what is aimed to be achieved and includes a plan of how those objectives will be met. The protocol, therefore, forms the research design and as such, is a model of logical proof. This is achieved by offering guidance in the areas of: (a) the questions to be studied, (b) what data is pertinent to the study, (c) the selection of data to collect and, (d) how the results should be analysed (Yin 1984, Yin 1994).

An in-depth case study then, to be conducted over a period of six months was chosen for three main reasons.

- Because *how* and *why* questions needed to be answered
- There was little or no control over the unethical practices to be identified
- The emphasis was on a contemporary phenomenon within a real-life context

The design of the case study can take one of four forms. Only two of those forms were applicable here as the use of a single case study had been identified as most appropriate. The four initial choices are shown in Table 9 below.

	Single	Multiple
Holistic	1	3
Embedded	2	4

Table 9. Case Study Design Choices

As the use of a single case study had been selected, the choice of whether a holistic or an embedded approach only needed to be determined. A holistic approach looks at the area under investigation as a whole, i.e. the big picture, whereas an embedded approach looks at the area as a collection of sub-units - possibly forming an initial hierarchical structure. As the research aims and objectives identified earlier referred to investigating the role of ethics in software development teams, associated business units and the suitability of a framework to address ethical concerns, the embedded approach was found to be the most suitable. This enabled multiple units of analysis, known as logical subunits. The identification of subunits can add significant opportunities for extensive analysis - the corollary being enhanced insights into the case study. The only weakness of this approach is the possible lack of focus on the unit higher up in the hierarchy: the company. As such, the company selected was also studied as one of the business units - as explicitly included in the aims and objectives of the research, ensuring that it was not overlooked. This led to the units of analysis to be identified as the developers, the project teams, the project managers, other business units and the organisation as a whole.

It could be argued here that the research was moving towards positivism with such 'controlled' design and analysis. This argument is accepted as the chapter initially identified that a pluralistic approach would be taken as there are strengths to both positivist and interpretivist camps that can be utilised in the context of the research to compliment each other. It is also worth noting that the needs of the research leading to a description and justification of a single case study might encourage thoughts of an enthnographic study. As this research aimed to obtain the interpretations of project managers and software developers through the collection of predominantly qualitative data, it was not an ethnographic study as hard scientific descriptions were not sought. Furthermore, at this juncture in academia, these terms are used almost synonymously and are no longer as clearly defined as they might previously have been (Miles and Huberman 1994, Avison 2001).

3.3.5 Testing the Case Study

On completion, a case study needs to pass several tests to be acceptable - the case study protocol is a major tactic in ensuring this, as described earlier. Identifying the tests and the tactics for dealing with them ensures the success of a case study. Yin (1984) identifies the tests as: Construct Validity, External Validity and Reliability. (Internal Validity is not appropriate here as it is a test for case studies which are not descriptive or exploratory). Each test is shown with its respective tactic in Table 10 below, adapted from Yin and then described in more detail.

Test	Tactic	Phase of Research in Which Tactic Occurs
Construct Validity	Use multiple sources of evidence, Have individuals study review draft case study report	Data Collection Composition
External Validity	Establish domain to which the findings can be generalised	Research Design
Reliability	Use Case Study Protocol	Data Collection

Table 10. Case Study Tests and Tactics

Concerns raised over the subjective nature in data collection are addressed by construct validity, by providing a set of measures which are sufficiently operational. The selected measures must demonstrate a reflection of the types of change also selected. For example, as the views of software developers were sought with the use of interviews, then the choice of using an interview has to be justified. Several methods exist which increase construct validity, such as collecting multiple sources of evidence, asking participants in the case study to review draft copies of the final report and lastly, by establishing a chain of evidence. The latter method allows any reader of the final report to work his way forwards or backwards through the report following a logical and satisfactory path. Finally, the identification and grouping of commonly felt feelings and/or meanings experienced by those participating in a case study can provide a powerful critique of a shared situation (Zuboff 1988).

Generalisations have been found to be inappropriate with this research approach as the company selected was not chosen at random from a sample population using a mathematical tool as discussed above - the choice was very much context dependent. The domain to which the findings could be generalised, however, consists of large companies developing new IS, have a structure similar to that of the company taking part in the case study and, be set in a similar economic climate with an active American parent company.

The aim of the reliability test is to reduce (and ideally minimise) the number of errors and/or bias in a case study. The objective is to obtain the same conclusions by repeating the same case study. Note the same 'case' study i.e. the case is the individual(s)/ company(ies)/etc. studied originally. This repetition is considered possible when the first study of the case was properly documented. Good documentation has the added benefit of reducing any third party suspicions on the reliability of a case study and is achieved by adopting the case study protocol.

3.4 The Company Selected to Participate

3.4.1 Introduction

The company required to take part in the case study needed to promote ethical working practices, regardless of whether they were explicitly identified or referred to in this way or not. This then should facilitate some rich insights, enabling an evaluation of the framework selected which could, if found appropriate, be applicable across the industry for all project managers and software developers to benefit. After the consideration of several companies, a company was selected to take part in the case study and will be referred to as Telco - due to confidentiality agreements. Telco was based in Stockley Park, west London and, was additionally geographically located convenient to Brunel University which facilitated communication, triangulation of data, the seeking of assistance, etc.

Founded in 1984 by computer scientists from Stanford University in the US, Telco first delivered new IS to customers in 1986 and, is now a world leader in networking for the internet and creating end-to-end networking solutions - its Ethernet business alone is worth over $3 billion a quarter. The parent company is still based in the US, with its main offices located in Herndon and San Jose. Telco did not include the use of ethical practices in its mission statement or description of the culture present, although an internal code of ethics was in place and described on the company's intranet. Telco's mission statement was 'Changing the way we work, live, play and learn'. The culture present within the company was formally described as:

'Quality team, no technology religion, stretch goals, teamwork, empowerment, trust/fair/integrity, drive change, frugality, market transitions and open communication'.

3.4.2 The Working Environment

The American parent company exercised a 'hands on' approach to its offices in other continents and, therefore, set corporate policy to be implemented globally. The Chief Executive/President was in regular contact with the entire workforce, typically via group email. An atmosphere of honesty, openness and competition between employees at all levels was actively encouraged, which follows in spirit with the ethical attributes identified earlier. This was confirmed in initial telephone conversations with one project manager at Telco who was interested from the outset in participating in the case study and, who later went on to note that the company encouraged ethical working practices from the bottom up (Anderson 2001). In addition, the company had a universal reputation for excellence which should encourage academia and practitioners alike to study the research findings carefully. Telco's advertisement campaign in January 2001 reflected its attitude towards its customers, which stated the desire not only to provide customer support but, customer delight (Telco 2001).

The company was split into various business units, including, Recruitment, Human Resources, Sales, Development and Quality Control. The offices were well equipped with training and personal development actively encouraged at all levels. All software developed by the company was created using an in-house methodology called GEM: the Great Engineering Model. The model consisted of the following stages: project planning, system definition, design, implementation, unit testing, internal verification, external validation, release and sustaining (maintenance). Test cases were contained in test plans and test specifications. The development teams were located across several continents, with the development of some software components outsourced to other companies. The software for new IS was developed and tested by almost disparate project teams implementing different methodologies.

Shortly after the case study began, however, the US economy plummeted and huge cutbacks by the company were planned, i.e. 8,500 of the workforce globally were made redundant (including 14% of the company's UK workforce) and, the share price crashed from over $80 to approximately $12 over a 12 month period. An 'expansion to

Reading' became considerable 'down-sizing to Reading' - a cost cutting exercise - as the number of offices leased in west London was reduced significantly with the majority of employees relocated to the new (cheaper) offices.

3.4.3 The Role of the Author

Initially the author's role entailed assisting one software project manager with both project management tasks and preparation for ISO 9001:2000 certification. The ISO preparation began with an internal audit - led by a quality manager, which was conducted shortly after a questionnaire was administered by the author to the project team members. The questionnaire results were later not used but the exercise did facilitate the direction of the research. Due to the mass redundancies, the manager did not present the results of the internal audit to the (reduced) project team. The overall plan, therefore, became to:

- Establish research direction
- Assist in project management tasks
- Assist in preparation for the (indefinitely postponed) ISO assessment
- Collect relevant data to meet the research aims and objectives

3.5 Data Collection and Analysis

A case study involving a company was selected for use in the research as against a case study conducted in a library and/or via a telephone as the latter would not have provided enough detail of a situation for the case study to be considered rigorous research - as is required. Important sources of evidence that can be used in case studies to provide this detail are documents, interviews and discussions - with individuals and/or groups, observations (direct or participant), physical artifacts and archival records - each is described in Appendix F (Yin 1984, Walsham 1993). These sources of evidence could clearly not have been collected via a series of telephone calls made from a researcher's desk or a visit to a library. To ensure that the sources chosen were used correctly, however, the author needed to become familiar with them and also utilize them independently. This list of data sources is noted by Yin not be finite, however, with additions possibly including photographs, for example. The sources should be viewed as complementing each other - with as many as possible used where appropriate to facilitate a high quality case study.

A summary of the techniques to be used for each known research activity is shown in Table 11 below.

Activity	Techniques Used
Identify role and importance of ethics in software development teams	Questionnaire, Interviews, Document Analysis, Observation
Evaluate the IEEE/ACM Code as an ethical framework	Questionnaire, Interviews, Document Analysis, Observation
Identify the influence of the Code on working practices	Questionnaire, Interviews, Document Analysis, Observation
Identify influence of working practices on the Code	Questionnaire, Interviews, Document Analysis, Observation

Table 11. Case Study Techniques

3.5.1 Ensuring the Validity of Data Collected

Although each of the above sources of data collection has been used solely for complete case studies, the triangulation of sources ensures the verification and validation of data collected and subsequent findings. (In deed, it is this opportunity to use a wide selection of data sources that forms a major strength of adopting the case study approach.) Four types of triangulation are possible (Patton 1987):

- Of data sources
- Of methods
- Of perspectives on the same set of data
- Among different researchers

The maximum benefits of any source selected can be obtained by following three accepted principles: (a) using multiple sources of evidence, (b) creating a case study database and, (c) maintaining a chain of evidence. The principles aim to increase both case study reliability and construct validity, as described above. They do not aim, however, to restrict the researcher's ability to be creative, innovative or insightful and as such, should be adhered to as guidelines only and, not as legalities. A description of each of the three principles follows.

Multiple sources of evidence cannot only be used simultaneously but also merged to form a new hybrid source if found to be necessary. This is only possible with case studies - unlike surveys or experiments. The development of converging lines of inquiry is perhaps the most important advantage of this principle as any finding or conclusion put forward by the researcher is likely to be more convincing - and thus acceptable - to an examiner. In addition, potential problems with construct validity are also addressed with multiple sources of evidence, with case studies adopting this approach being rated more highly than those which rely on single sources of data collection. Naturally this approach does not offer itself as a panacea to a researcher. To be able to collect data from different sources, more than one technique needs to be learned. This attaches a resource cost on the researcher regarding time and materials. If the techniques to be used are not familiar to the researcher at the outset of a case study, then the beneficial opportunities described above may be lost or not exploited to their full potential. Furthermore, techniques to collect data may be used and then not included in the final report. For example, a questionnaire was administered by the author on a project team manager at Telco and his team of developers and data collected from the questionnaire was ultimately unused - although the exercise did facilitate the direction of the research. As such, issues regarding the design of questionnaires are not included.

A Case Study Database increases the reliability of a case study by explicitly separating the data collected from the case study report. This enables the data to be analysed independently by others as required. This is made clearer, for example, when quantitative data needs to be examined but is encapsulated within pages of text. Typically this explicit separation has not been done with previous case studies, although it is now on the increase. Two activities are thus necessary for a researcher to carry out: the separate storage of data and the construction of the report containing pertinent data as required - from the database. The problems associated with creating such a database are categorised in terms of: notes, tabular materials, narratives and documents - each of which is described in Appendix F for the interested reader.

Maintaining a chain of evidence increases the reliability of the case study by enabling any reader of the report to navigate forwards or backwards in a logical and guided sequence of steps, as described above. He may choose to begin with the conclusion and then trace the necessary steps back to the initial research questions, supported with references, evidence and other supporting material as indicated in the report at any particular point. He may choose to navigate his way through the report in the opposite direction. Whichever way he chooses, the chain of evidence will enable him to do so in both a logical and convincing manner.

3.5.2 Allocation of Unethical Practices to the Eight Principles

The three generic areas of unethical practices in the workplace identified previously were: (a) six categories of questionable practices, (b) common worst practices and, (c) current problems in ISD. These three areas were aligned with the eight principles of the IEEE/ACM Code (shown in Appendix E). This enabled each area covered by the eight principles of the Code to be analysed where pertinent data was available from the case study - showing the existence of unethical practices and leading to the identification and understanding of their respective cause.

3.5.3 The Use of Statistics

In order to make an informed decision about the appropriateness of the use, or non-use, of statistical methods in this research, it was necessary for the author to know a little about them first. Much of the work described below was taken from Galliers (1992).

The predominantly qualitative data collected by the author via interview transcripts, documentation analysis and participation-observation (recorded as log book entries) - were entered into tailored repertory grids and, therefore, some statistical knowledge was required. This was not only necessary for the analysis of the data once collected, but also for the design of the repertory grids and, for the construction and administration of the initial questionnaire. Many questions asked in the questionnaire required an answer in the form of a tick against a number or word forming part of a measurement scale such as, for example, 1 to 5. As described earlier in the chapter, the questionnaire was later discarded after refining the direction of the research and, therefore, additional information about the use and construction of the questionnaire is not included.

3.5.4 Techniques for Data Analysis

Whichever analytical approach is adopted, Yin (1994) identified four key principles for consideration by a researcher to ensure high quality data analysis:

1. All relevant data should be analysed
2. All major rival interpretations should be analysed

3. The analysis should concentrate on the most important aspect of the case study
4. The researcher should include his own experience(s) and knowledge where appropriate

In addition, Yin recommended that analysis should be seen to be carried out with expertise. For example, a researcher should use care in the organisation and presentation of the case and, facilitate readability of the narrative given. This helps the researcher in addition to any methodology and technique(s) adopted, which Yin argued are difficult to use, making data analysis the hardest stage of conducting a case study.

3.5.5 Selection and Use of Repertory Grids

The development of a new coding system (scheme) for data analysis is both difficult and time consuming. As there is a plethora of available coding systems developed, tried and tested, it made sense to use one already proven: in this instance repertory grids were found to be most appropriate. A brief introduction into the description, justification and use of repertory grids for the analysis of the data collected is, therefore, presented below. The data collected was then able to be categorised according to the eight principles of the IEEE/ACM Code and, additional information regarding the choice and, use of repertory grids is provided in Appendix E for the interested reader. Much of the work below is taken from Norris (date unknown) and Robson (1996).

Although repertory grids were considered to be most appropriate tool for data analysis for this research, many other grids exist, such as rated grids, implication grids, interactional grids, resistance to change grids and, dependency grids. Repertory grids are most suited to measuring and/or comparing attitudes of individuals and/or groups of individuals over a brief time period or at intervals or over any time period, though they have other uses. These factors particularly suited this research as data to be compared and analysed was collected from four software project teams and teams from other departments (Quality Control, Human Resources, Recruitment and, the company as a whole). The data was collected during a six month case study. Used in this way, repertory grids can provide a large amount of data in a structured format. This was achieved by transforming the qualitative data predominantly collected during the case study into quantitative data - facilitating comparisons to be made and conclusions to be drawn. Furthermore, repertory grids can be used alone or in conjunction with other methods and, be used safely by relative newcomers to the technique. This quantitative approach to data analysis addresses two weaknesses of qualitative case studies: they generate a lot of documentation which is difficult to read and valuable information is buried within pages of text. The use of a few repertory grids is not only manageable by a reader but also highlights important information and ignores unimportant information or that which is irrelevant, duplicated or otherwise undesirable to the researcher.

3.5.6 Reliability and Validity

For repertory grids to be used, the researcher needs to be satisfied about the reliability and validity of the technique. Narrative accounts can be developed whilst on a case study, but it could be difficult for a researcher to be objective - a better approach might be the use of a coding scheme to reduce bias (Robson 1996). Reliability and validity are still essential issues with coding schemes of course, with reliability achieved when, for example, several researchers are able to develop the same, or similar, coding scheme when presented with an identical situation. As the author worked alone, no other researchers were available to ensure reliability this way. For the purposes of this research, reliability was ensured with the consistent use of a detailed coding scheme provided by the principles of the IEEE/ACM code and their respective area of focus. These same eight principles provided the definition of IS ethics above, used to set the context for this research. This was also highly desirable, as the nature of the case study almost dictated unstructured observations, compared to the structured observations carried out on relative certainties and simplified complexities presented in, for example, laboratory experiments. The implementation of unethical practices by the software developers, project managers and participants in other departments - and the subsequent possible identification of their cause(s) of their existence - could not have been scheduled to fit into the availability of the (and any additional) researcher(s). The unethical practices defined by the code are invariably not pre-determined, justifying further the selection of participation-observation as a data collection method to be implemented.

3.5.7 Practical Considerations

Grids are usually completed by the respondents on a one-to-one basis with a researcher, but this can also be done in groups and, also by a researcher himself using data collected, for example, from an interview transcript. This latter application was conducted by the author, with data also collected from document analysis and log book entries made possible by participation-observation during a case study. The design of an actual grid varies according to its application, with a bespoke grid typically necessary for each research project. No special abilities are required of a researcher to create and/or use repertory grids.

The practical benefits of using grids include flexibility and the ability to produce more subjective material than a questionnaire, for example - when used to collect data. Interviews can obtain nuances of meaning which can be shown in a grid and, coupled with the multi-dimensional nature of a grid, it can gather and present a huge amount of information in a relatively short amount of time - again, compared to a questionnaire, for example. A questionnaire may only enable a participant to tick a series of boxes relating to 'Yes' or 'No' answers, with all additional information pertinent to the context of the answer lost. The data collected in a grid reflects that obtained, or elicited, from meanings which are personal to the respondent(s) and, consequently a researcher's task of completing the grid is generally an enjoyable task as, at least in part, the data is of greater valuable. Grids have the additional benefit of identifying redundant information, which can be quickly ignored and/or omitted from analysis. As such, the design of a grid and the planning of analysis on the data it provides are carried out simultaneously. These characteristics helped determine the suitability of repertory grids for this research.

The author completed the grids from data collected via interview transcripts and the other data collecting techniques noted above, as it was considered obtrusive for the respondent to complete them after each question put to them by the researcher and therefore, would have impeded the rapport with the respondent - resulting in 'poorer' data being collected (Yin 1984).

3.5.8 Grid Design

For the purposes of this research, data collected in the case study was analysed in a series of repertory grids reflecting the multiple project teams and business units participating. Project teams were represented in the grids to enable the identification of individual developers, project managers and the teams as a whole. This was necessary as the units of analysis were previously identified as the developers, the managers and the teams, in addition to other business units and the company as a whole – which were also represented by separate grids. As each grid or series of grids is/are bespoke - created specifically for each research project - there is no standard grid form or template available. Grids are flexible in their design and this flexibility is considered highly valuable - and as such is one of their great strengths (Ryle 1975). The design of a repertory grid is based on a sorting procedure - in this case, the eight principles of the IEEE/ACM Code. Data is recorded in a matrix, with the number of rows and columns needed reflecting the sorting method and the number of items to be sorted: eight principles consisting of approximately ten sub areas - each with an associated compliance/non-compliance value - with approximately 6 team members per project, multiplied by eight projects/business units, identifying eight grids were necessary. The grids have a horizontal axis consisting of 'elements' and a vertical axis consisting of 'constructs'. The elements are the individual team members *per se* and the constructs are the eight IEEE/ACM code principles in detail.

3.5.9 Grid Analysis

As the purpose of every grid is unique, guidelines on analysis are vague, just as they are on design. A major strength in the use of grids is their flexibility and adaptability, as noted above, but this is also their weakness. This versatility prevents the creation of a single, all-purpose grid for all to use or, a single set method for analysis. Clearly, the nature and extent of any analysis undertaken varies according to the area of interest of the researcher. Construction of the repertory grids used was tailored to suit this research as follows:

- The constructs used were defined by the eight principles of the IEEE/ACM Code and, not defined by the project managers, software developers or other participants (who made up the elements), as would normally be the case. This variant was necessary as one objective of the research was to identify non-compliance to the IEEE/ACM Code *per se*
- The elements consisted of the project managers, the software developers and participants from other business units. Managers could not, therefore, select the people to be elements as is normally the case - here they were determined by each manager's team members
- 'Construct' 5 covers Principle 5 of the IEEE/ACM Code which was created only for project managers and, not all the elements on the horizontal axis of the grid, i.e. the software developers and other participants from other business

units. These elements, the software developers and other participants, are known in this context as 'non-applicable' elements

Although the constructs were predetermined in this instance, it is not uncommon and, is known to facilitate analysis. A series of grids were needed as four other business units were included in the research, in addition to the software team to which the author was assigned for the six month period and three other project teams, making a total of eight units of analysis in all. A series of eight grids was therefore used. In each of the cells, codes were entered identifying the type and location of data collected as evidence of compliance or non-compliance to the principles of the IEE/ACM code, as appropriate. An example of the grid structure used is shown in Table 12 below. The structure used might appear simple when compared to other repertory grid structures, but with approximately eighty constructs used (8 principles each covering approximately 10 sub-areas) as against the recommended maximum of 25, the ultimate analysis needed to be manageable (Ryle 1975). The code shown in the first data cell (I-JF-20-4-1) relates to evidence supporting compliance to Principle 1.1, is from an interview (I) with an identifiable manager whose initials are JF, and which can be found on page 20 of his interview transcript at line 4. (The initials entered into the grids are coded to preserve the confidentiality agreement with the participants.) The final digit of the code represents the level of importance of the evidence in supporting or not, compliance to the principle, with 1 being the least important to 3 being the most important - as further defined below.

	Project Manager	Developer 1	Developer 2	Developer 3	Grand Total
Principle 1.1 Compliance	I-JF-20-4-1		D-23-14-1		2
Principle 1.1 Non-Compliance					
Principle 1.2 Compliance					
Principle 1.2 Non-Compliance					

Table 12. Grid Structure for Data Analysis

There is no general validation available for repertory grid analysis. Due to its individual and adaptable nature, the extensive range of data available is open to interpretation in a number of ways, in addition to the technique being used for a variety of purposes. Due to the volume of data provided by the interview transcripts and other data collection methods used by the researcher, the incidents of unethical practices were analysed and recorded by both weighting (degree of importance) and frequency of occurrence. The weight scale used was a three-point Likert scale: 3 for the greatest importance/compliance and 1 for least importance/compliance, or unimportance/non-compliance respectively. The grids were then able to be analysed for each project team and department by identifying those unethical practices which occurred most often and/or were considered most serious – level 3 on the Likert scale used and described above.

The contents of the grids consisted of the locations of the evidence collected and the type of data collection methods used to collect a particular item of data. This enables any reader to locate the evidence and confirm that what is recorded is an accurate reflection of the situation found leading to the results produced. Two rows were allocated to each principle requirement to enable additional data to be recorded regarding compliance to the IEEE/Code code. The first row recorded the locations of evidence relating to non-compliant practices found - in line with the research objectives - and the second row recorded practices which were compliant to the code. This is shown in the row header as, for example, '8.15 N' and '8.15 Y' where the 'N' represents 'No' to compliance and the 'Y' represents 'Yes' to compliance. Although not a requirement under the objectives of this research, the additional data is recorded for interested readers to examine or just to inform a reader that not all practices at the participating company are unethical. (After all, the company was kind enough to allow the case study to be undertaken fully aware of the data required by the researcher.)

3.6 Summary

Having identified a non-scientific methodology as being most appropriate to enabling the research objectives, Burrell and Morgan's four sociological paradigms were described. A justification of why an interpretive approach was taken was provided, followed by a critique of the descriptive case study strategy used - which the interpretive approach permits.

A six month in-depth single case study was conducted as this was necessary for the ethical environment within which project managers and software developers must function to be able to be identified, understood and described. This was in conjunction with determining the suitability of the IEEE/ACM Code of Ethics as an ethical framework for project managers and software developers. The case study strategy was described further to facilitate a subsequent research design and, a single embedded case study was identified to be the most appropriate way of meeting the aims and objectives of the research. The activities and deliverables considered essential for a case study were then described and put into context. These included the case study protocol which is a major tactic in proving the reliability of a case study. The reliability and construct validity of the case study was found to be increased by adhering to three principles: the use of multiple sources of evidence, the creation of a case study database and, by maintaining a chain of evidence. Participation-observation, interviews and document analysis were techniques described and subsequently selected to collect both quantitative and qualitative data for analysis. A critique of repertory grids was then provided, with a major advantage identified as their 'tailorability' - and with no prior expert knowledge in their use required. A version of repertory grids was thereby chosen as most appropriate for categorising and analysing the data collected.

Now that a focused area of research and an appropriate research method have been identified, the next chapter presents the case study findings by describing the unethical practices found at Telco, either encountered or initiated by the project managers, development teams and other business units. The unethical practices are categorised by the eight principles of the IEEE/ACM Code, enabling the research questions of 'What' and 'How' to be addressed.

Chapter 4. Ethics in Practice - Evidence from the Case Study

4.1 Introduction

The need for a descriptive and single case study to be carried out over a period of six months to enable both in-depth research and the subsequent collection of rich real world data was established in the previous chapter, which enabled the difficulties about the issue of ethics to be raised, in the context of managed software development at Telco. The working practices of other business units with which the project teams needed to interact were also identified for - relative - assessment against the principles of the IEEE/ACM Code to facilitate the generalisation of analytical findings at the end of the research.

Section 4.2 provides an overview of all the evidence of the unethical practices encountered at Telco, categorised by each of the eight principles of the Code, enabling the research questions of 'What' and 'How' to be answered. The following sections provide the evidence collected in the case study of unethical practices which relate to the issues drawn out in Chapter 2. These issues are then discussed in the next chapter, categorised by the units of analysis: individuals, teams, managers and organisations. Section 4.3 provides a summary of Principle 3 relating to new IS - referred to in the Code as Product, followed by the evidence collected for each sub-area. Section 4.4 provides a summary of Principle 5 relating to management, followed by the evidence collected for each sub-area. Section 4.5 provides a summary of Principle 6 relating to the profession of ISD, followed by the evidence collected for each sub-area. Section 4.6 provides a summary of Principle 8 relating to developers themselves, followed by the evidence collected for each sub-area. Where multiple evidence collected refers to the same unethical practice identified, to aid brevity it has been presented only once. In addition, due to ambiguity and obvious repetition found in the Code, some points presented are briefly repeated, and noted for being so.

4.2 Non-Compliance to the Principles Found

Identifying the project managers, team members and other units of analysis as compliant or otherwise to the principles of the IEEE/ACM Code of Ethics was originally anticipated as a complex process due to the involvement of numerous individuals, decisions, organisational rules, social norms and, a mixture of both good and bad intentions. To compound the problem further, an interesting point made by two interviewees participating in the case study was that asking someone if they are unethical is like asking someone if they beat their wife - no-one will admit to it. Incidents of non-compliance were observed and recorded, however and, Table 13 below shows the total number of instances identified during the case study for each of the development teams and business units that participated in the research.

	DT 1	DT 2	DT 3	DT 4	Quality	Recruit-ment	HR	Telco	Total
1: Public	6	3	1	1	2	1	2	6	22
2: Client & Employer	8	1	0	1	0	0	2	10	22
3: Product	63	14	12	17	5	9	5	41	166
4: Judgment	12	0	0	2	3	5	1	9	32
5: Management	74	3	2	8	3	11	15	21	137
6: Profession	19	7	4	7	0	4	3	5	49
7: Colleagues	11	2	0	4	3	0	6	7	33
8: Self	14	3	3	2	0	4	2	3	31
Total	207	33	22	42	16	34	36	102	492

Table 13. Summary of Non-Compliance Identified

The table above shows the total number of incidents of non-compliance identified in the case study. With the Code created specifically for software developers, however, some of the issues covered by the eight principles were not directly relevant to the other business units. They were considered, however, in context where appropriate. It is also worth noting at this juncture that Development Team 1 (DT1) was the principle team participating in the research (to which the author was assigned), resulting in a larger amount of data collected for that development team than the other development teams or business units. This is reflected in the number of non-compliant incidents identified, rather than a development team which was found to be possibly more unethical than the other development teams/business units which took part. The unethical practices found were identified with either tangible evidence or with inference. Due to page limitations, only incidents of non-compliance considered to be important are described below, in conjunction with their respective evidence and explanations of their cause. Importance is defined here as those unethical practices identified which have a direct bearing in relation to the objectives drawn out in Chapter 2. The incidents considered to be less important than those described below are not included in the research.

The evidence collected of non-compliance to the principles is now presented, followed by explanations of how they occurred. The charts below show the number of incidents per project team/business unit for each sub-area of each principle. The number of occurrences is shown on the Y (vertical) axis, with the four development teams (1, 2, 3 and 4) and Quality (R), Recruitment (R), Human resources (HR) and Telco (T) shown along the X (horizontal) axis. Multiple colours are used for each principle in the tables to separate the sub-areas, for example, red for Principle 1.01, green for 1.02, blue for 1.03, etc. (The 'N' indicates Non-compliance to the principle). As tables can only present a maximum of six principles areas in any one table - restricted by the software used - multiple tables are presented for principles containing more than six.

4.5 Non-Compliance to Principle 3: Product

Principle 3: 'Product' is summarised as 'The highest professional standards possible will be sought by developers for their products and their respective modifications'. A breakdown of the non-compliance found across all the project teams and business units is shown in Tables 14, 15 and 16 below.

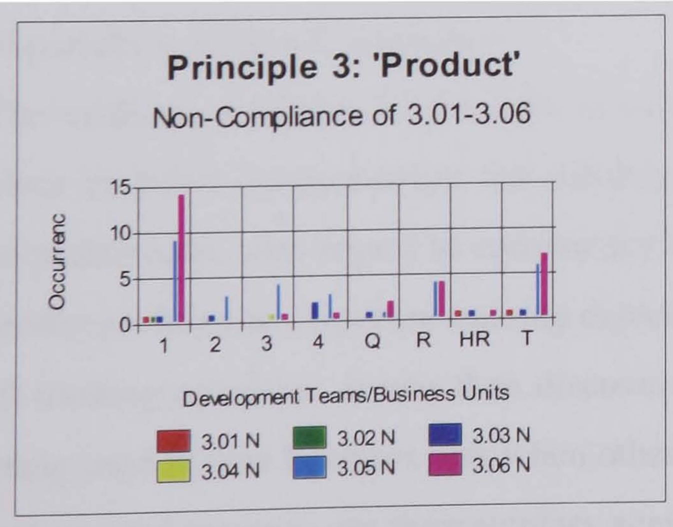


Table 14. Principle 3 Non-Compliance of 3.01-3.06

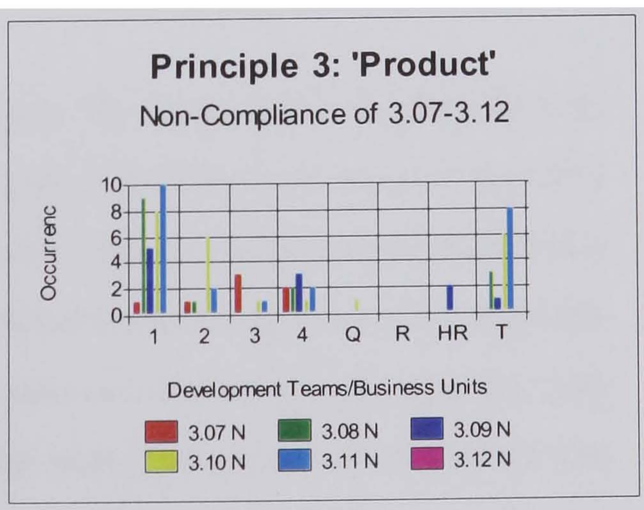


Table 15. Principle 3 Non-Compliance of 3.07-3.12

with respective data-collecting techniques and analytical methods, with justifications provided below.

3.3.1 Justification of Interpretivism

The Interpretivist paradigm was adopted, with understanding and meaning sought and viewed as subjective, with knowledge seen as a social construction. Furthermore, the identification of cultures and sub-cultures existing in organisations were identified in the previous chapter and, as viewed as a system of shared meaning and knowledge, required interpretation. Having characteristics which can be described as subtle, complex and indistinctive, research of cultures often attracts soft science interpretivists. This approach is necessary as the Functionalist approach, for example, would require identification, categorisation and measurement of distinctive variables which, in this context consisting of cultural and social issues, was not readily possible (Symons and Walsham 1988, Floder and Weiner 1983, Kumar 1990). The ultimate goal of this research was not to emancipate software developers but to understand their individual perception of reality, leading to an understanding and description of the role and importance of ethics in software development. This was in conjunction with ascertaining the suitability (or otherwise) of the IEEE/ACM Code of Ethics as an ethical framework - as a step forward towards finding a solution to the unethical practices conducted in ISD. This included the identification of possibly covert motives and actions, and determining their cause - albeit intentional by the respective software developers, project managers, project teams or the organisation as a whole, or otherwise.

The approach is best suited to enabling the aims and objectives of the research to be met, i.e. understanding and describing the role and importance of ethics in ISD and how the effects of an ethical framework influence working practices, in addition to identifying how these same working practices influence the application of the framework itself (Dodd and Lycett 2002). This is possible as interpretivism can be applied at any stage of an IS life (Walsham 1993, Orlikowski and Baroudi 1991, Benbasat *et al* 1987). The context of the selected framework was analysed and the various organisational systems and structures within which the framework was evaluated were determined. The social structures forming more subtle contexts were also analysed. The participants' individual interpretations of reality resulting in shared/contentious meanings resulted in a complex dynamic context within which the framework was evaluated. Individuals are known to feed on contextual elements such as resources or perceptions of authority to carry out their specific tasks, which then support these systems of resource distribution or power, or create new ones. Thus the human perceptions of the context in which they are formed are continuously changing, fuelled in addition by the IS in operation. (The latter can be thought of more specifically with regard to hardware, software, systems or data.) This hybrid approach - interpretive/qualitative/quantitative - has much support as there is thought to be no single 'best' approach (Campbell and Marshall 1999, IFIP 1995, Johnson and Smith 1999, Klein *et al* 1990).

In an interpretive context, the conclusions drawn are neither right nor wrong, or correct or incorrect - just simply interesting to some readers and not so interesting to others. This is the corollary of adopting the interpretivist research approach. The conclusions are still of value to a wider community than just the author, as they enable both written and verbal debate leading to broader judgments of value to be ascertained. Conclusions can then be compared, evaluated and improved in this manner. Additionally, they provide a vehicle for communicating this knowledge to others - facilitating learning and understanding (Walsham 1993, Yin 1994).

Each research methodology has its weaknesses of course, with heavy involvement - and therefore, influence - of a wide range of stakeholders being a weakness of the interpretivist approach. In addition, the time and resources available to a researcher working alone for a specified duration naturally result in these issues providing physical deterrents to some researchers considering this approach (Legge 1984). They were not, however, considered a major concern of the author.

3.3.2 Aims of the Research Approach

As the research aimed to primarily identify and understand the role and importance of ethics in ISD, this pointed to a qualitative approach, enabling the use of a case study. A company participating in the case study would enable the identification of situations where unethical conflicts are managed (or mismanaged), but this alone would serve no purpose. The management of these situations needed to be examined and their causes identified - from both the IEEE/ACM framework on the organisation's practices and vice versa. The stance on ethics taken by other professions/business areas also needs to be understood to enable lessons to be learned where appropriate and additional recommendations to the IEEE/ACM code may be made. Thus several activities have

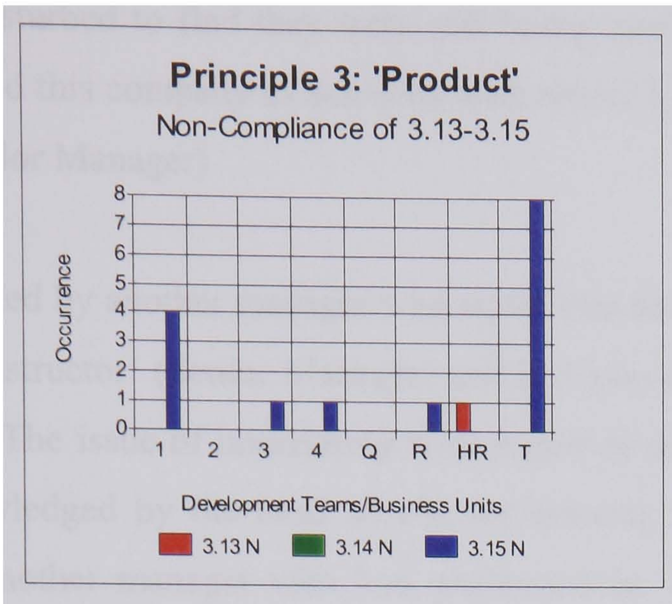


Table 16. Principle 3 Non-Compliance of 3.13-3.15

Principle 3.01 requires software developers to strive for high quality, acceptable cost and a reasonable schedule, ensuring significant tradeoffs were clear to and accepted by the employer and the client, and were available for consideration by the user and the public. From the data collected, three examples of non-compliance were recorded. The evidence found of incidents considered to be important included:

- 'There is a problem of inconsistency and quality' (Senior Manager)

This is further supported by observations recorded in the author's log book, which included:

- 'There is a culture of deliver it now and get it working later - which needs to change as customers expect their systems to work from the outset' (Developer)

Explanation of Non-Compliance

The evidence available in the data to ascertain how the above non-compliances took place included (respectively): the suitability of external training vendors was found to be problematic with regard to consistency and quality. Inconsistency was identified as a problem when the corporate training department based in the US attempted to centralise all training activities. It was then discovered that one training vendor in particular was being used by one business unit when other business units had previously used them and had decided never to use their services again:

> 'I was very disturbed to find they were still being used as I had long ago disqualified this company as someone with whom I would ever do business' (Senior Manager)

This view was supported by another manager who stated that he was highly annoyed by their supposed 'star instructor' (Senior Manager) and had also decided he would never teach at Telco again. The issue of uncertainty with regard to training consistency was identified and acknowledged by the head of UK recruitment. This point was further supported by a yet another manager who had exclaimed he was personally tired of always being different from the rest of Telco and had queried whether every software development department in Telco had their own processes.

The culture of 'Deliver it now and get it working later' with regard to ISD did not concern one project manager interviewed. His attitude to (imposed) completion dates was:

> 'Something will be delivered *on time* - but its reliability won't be known' (Senior Manager)

This could be considered surprising as most Telco employees - including project managers and software developers - had stock and stock options and, therefore, a vested interest. Project managers were found to be at ease with the release of software for shipment to customers with known (and sometimes severe) bugs still present in the code, comforted in the knowledge that the bugs present would be corrected at some point in the future via a support agreement.

Principle 3.10 requires software developers to ensure adequate testing, debugging, and review of software and related documents on which they work. From the data collected, twenty three examples of non-compliance were recorded. The evidence found of incidents considered to be important included:

- Code was released containing known severe bugs

Explanation of Non-Compliance

The evidence available in the data to try and ascertain how the above non-compliance took place included an acknowledgement that when a large deal depended on having a system delivered by a certain date, then that system would be delivered on that date regardless of the condition of the code. This was commonly achieved by a reduction in the number of tests conducted.

Project managers had discretion in requesting code reviews, with one developer stating that his project team did not conduct any formal code reviews. In project teams where code reviews were conducted, two team leaders noted that test cases should preferably - albeit informally - be written by the test developer as he goes along, although in practice test cases were usually written by developers prior to testing. The documenting of test cases was additionally viewed as problematic. A distinction worth noting here is that software released without sufficient testing was considered different from the release of software which was known to contain bugs but still found to perform satisfactorily.

Principle 3.11 requires software developers to ensure adequate documentation, including significant problems discovered and solutions adopted, for any project on which they work. From the data collected, twenty three examples of non-compliance were recorded. The evidence found of incidents considered to be important included:

- 'Documentation is not normally up to scratch - it's a matter of luck really' (Developer)
- 'We try to avoid documentation' (Developer)

This is further supported by observations recorded in the author's log book, which included:

- Source code written by one developer was found to contain a lot of 'one liners' which was unhelpful to other team members

Explanation of Non-Compliance

The evidence available in the data to ascertain how the above non-compliances took place included (respectively): the Product Requirements Document (PRD) template was invariably not used. Furthermore, an internal assessment conducted of working practices found that one developer had created his own templates for new documentation. Many randomly selected project documents were reviewed and all were found to contain anomalies, summarised as ambiguous, inconsistencies, containing typing errors and other negative observations. One project document was found to have been revised eleven times. Documentation was avoided by some software developers, with one project manager explaining that his team members did not document everything they worked on. As little time was spent writing documentation, one developer explained it would never be perfect. One design document was even described as 'crap' by the technical consultant in one project team. One anomaly discovered by an internal audit was the identification of incorrect revision numbers in document headers. In another instance, creation dates of some documents had been back-dated deliberately to satisfy a quality assessment. Furthermore, meeting minutes were officially required to be recorded for all project meetings, but one project team was found not to record their meeting minutes and, was consequently warned by the Quality Department to write up their minutes or be awarded an ETBP (Exception to Best Practice).

Many 'one-liners' of code were written by one senior developer who was knowledgeable about the structure and functionality of one particular section of code for which he was responsible. Whilst away on long sick leave, however, neither a team leader nor another senior developer in the team were able to make satisfactory progress on developing the section of code further due to their lack of his esoteric knowledge.

Principle 3.15 requires software developers to treat all forms of software maintenance with the same professionalism as new development. From the data collected, fifteen examples of non-compliance were recorded. The evidence found of incidents considered to be important included:

- Defects were not resolved in a timely manner
- Only 40% of upgrades reported fewer bugs

This is further supported by observations recorded in the author's log book, which included:

- 'Code is messed up by developers when maintained' (Developer)

Explanation of Non-Compliance

The evidence available in the data to ascertain how the above non-compliances took place included (respectively): a request to the technical support department to fix a bug concluded with an email from them stating it had been corrected - although no change was detected. Bugs were found not to be corrected in a timely manner and, to compound the problem still further, code was believed to be messed up by developers when it was maintained. Additionally, for some project teams, extra features were added to the main code unsupported, with one project manager stating that on occasion he had not been able to provide support for shipments of new IS within six months of release. Another project manager did not expect to provide any maintenance support until a future release of the software was made available. Even then, visits to a customer site were not made if they were located too far away from their local Telco office.

4.7 Non-Compliance to Principle 5: Management

Principle 5: 'Management' is summarised as: 'An ethical approach will be promoted and exercised by developers in the management of software development and maintenance'. A breakdown of the non-compliance found across all the project teams and business units is shown in Tables 17 and 18 below.

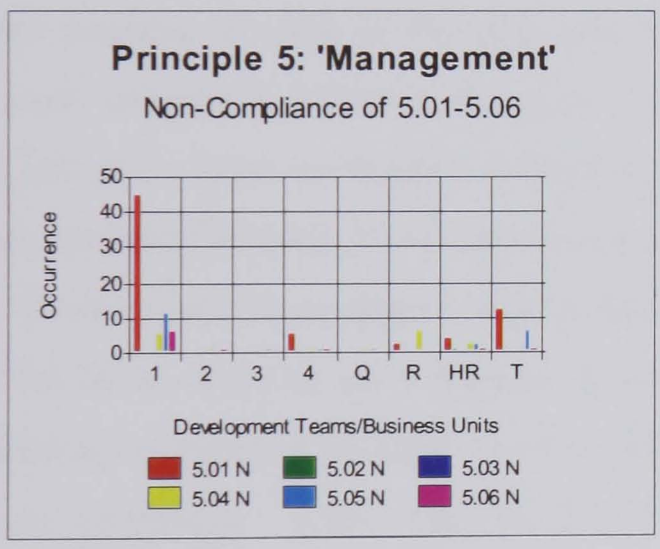


Table 17. Principle 5 Non-Compliance of 5.01-5.06

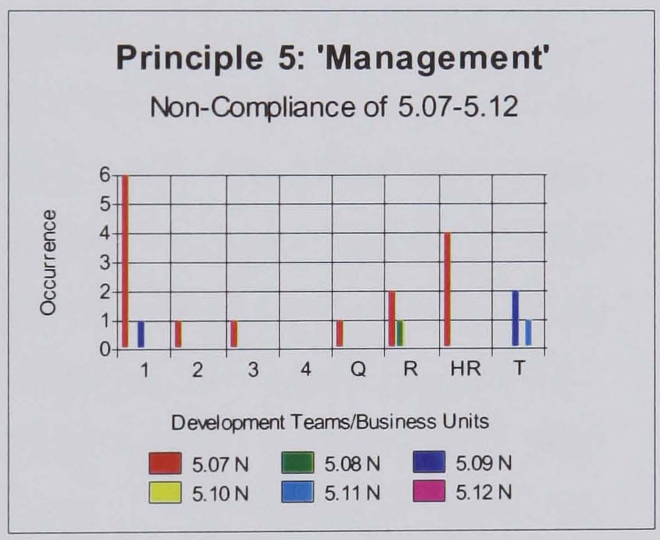


Table 18. Principle 5 Non-Compliance of 5.07-5.12

Principle 5.01 requires that project managers and team leaders ensure good management for any project on which they work, including effective procedures for promotion of quality and reduction of risk. From the data collected, sixty eight examples of non-compliance were recorded. The evidence found of incidents considered to be important included:

- One manager asked his team to check the project Gantt chart
- 'The development model is not exactly followed anymore' (Developer)

This is further supported by observations recorded in the author's log book, which included:

- Safety measures were compromised
- The quality manager identified a lot of small errors which he said would normally be reported together as a major non-conformity

Explanations of Non-Compliance

The evidence available in the data identifying how the above non-compliances took place included (respectively): one project manager asked his development team to check their project Gantt chart as he found he had been assigned to a task by a senior team leader on the chart without his knowledge. Another team leader looked over the Gantt chart and found he (a) did not understand some of the tasks, (b) was unaware he had been assigned to some of the tasks and, (c) found some tasks he was expecting to do himself assigned to others in the project team. A manager of another project stated that project plans were not usually created or followed, as was evident when one manager was reminded of tasks to do (and then entered them into his diary) when he saw them listed on a Gantt chart used by other project managers. On one occasion a developer was found to be working on a module of code of which his team leader was unaware. Information received by team members from senior management was sometimes found to be confusing - when received at all. As a result of the difficulties above, one developer confessed:

'Sometimes there is a lack of understanding of what we were going to do' (Developer)

Developers individually determined which processes to follow after the redundancy program had been implemented as the use of processes stipulated in the GEM model was then viewed as almost optional. New IS were then able to be developed quickly and to a satisfactory level of execution, but were known to be imperfect and not comply in their entirety to any official requirements or standards. In the context of HR, other examples of non-compliance to relative models or procedures included one project manager who was criticised as being insensitive to the feelings of those in his development team - in the period leading up to the redundancy notifications when morale was known to be low - by continuing to work from home. In addition, the decision as to who to keep and who to be made redundant was, on occasion, based on personality - albeit illegal. One employee was asked for the keys to his company car although he had not been issued with one and, another employee was made redundant at 9am and then, incorrectly, again by another manager at midday when he was still clearing his desk. These incidents were highly embarrassing and perceived to 'probably not be in the best taste', conceded the head of UK Recruitment. Furthermore, a newly hired developer was telephoned the day before his employment was due to commence to be told that he was no longer required and, after a lengthy and - heated - discussion, the project manager agreed to pay him three months salary and relocation expenses.

Safety measures were compromised when safety mats for the laboratories were not bought, although their use was strongly recommended by the manufacturers of the hardware equipment housed. In a separate context, one project manager had several live electrical cables trailing across his office floor which had to be stepped over to reach his desk, with no desire to make the area safe identified.

Many minor non-conformities in ISD at Telco were identified by a quality manager who stated they could be reported together as a major non-conformity, with a few examples described here. The incidents recorded were numerous and varied, including (a) public/bank holidays scheduled as working days in Gantt charts, '?' entered for task durations and, resources allocated in excess of their availability - for example '152%', (b) a spreadsheet for developers' holidays included developers who had been made redundant and were no longer employed, (c) management software used for budgetary control included one application called 'Expense Tracker' which was assumed by one project manager to show all figures in US dollars and not GB pounds. This assumption was not clarified during the case study, (d) one manager did not remember agreeing to holiday dates requested and then wanted to prevent the employee from taking the time off when the time arose, (e) electronic project records were not backed up where folders had been created by the developers themselves and not by system administrators, (f) one developer was emailed for several weeks by a project manager with tasks to be carried out after joining the company, which were not actioned, (g) tasks assigned to a senior developer and notified by email were neither carried out by the developer nor chased up by the project manager two weeks after the initial email had been sent, (h) one manager stated he had received nothing from HR regarding a login ID and password for a new software developer - although he later found he had received them two weeks earlier but had not read the relevant email on its arrival and, (i) data collected additionally highlighted the postponement of deadlines as - and this was a common complaint from those participating in the research:

'There is too much to do, to do it well' (Developer)

It was acknowledged by many of those interviewed that there was never sufficient time to carry out all of their assigned tasks to a satisfactory level. Developers were therefore found to work weekends prior to their holidays so as to help keep their projects on schedule whilst away. On one occasion, code reviews were found to be excessively delayed and other developers did not assist those conducting the reviews. One developer noted that projects were not discussed as much as they should have been. Scheduled staff appraisals were also frequently postponed by project managers due to insufficient time available. Furthermore, no evidence was found - albeit formally required - of review meetings having taken place. Quality goals were not defined and, one project manager interviewed only relied on trust with regard to progress and quality of software written by his developers, confessing that when developers told him the code was well written that he did not know for certain that it was. All he could do to check on their progress and the level of quality obtained was to look over their shoulder and try to read the screen of code they were working on at that time. The manager further conceded he was unable to review technical documents either - although they required his formal approval on completion. Other examples include the principle project team in the case study being initially called NSSU-UK (Networked Software Services Unit - UK) but was expected to change its name to VTC (Video Technology Centre). When the intranet was updated, however, the team's new name was shown as 'Call/Feature Control - Tandem'. The project manager was not aware of this new name. Furthermore, the intranet page showing project members' details was not updated to show the team's relocation from west London to Reading for several weeks after the move had taken place.

Principle 5.04 requires that project managers and team leaders assign work only after taking into account appropriate contributions of education and experience tempered with a desire to further that education and experience. From the data collected, sixteen examples of non-compliance were recorded. The evidence found of incidents considered to be important included:

- 'If we hire ten developers, it may be true that only three of them are going to be any good' (Senior Manager)

This is further supported by observations recorded in the author's log book, which included:

- Many training courses were either cancelled or inappropriate

Explanation of Non-Compliance

The evidence available in the data to ascertain how the above non-compliances took place included (respectively): one consensual view found of graduate developers on their first day in industry was that they knew nothing. One software developer, however, was initially rejected from the recruitment process but was later serendipitously hired by a project manager who recognised some of the attributes present in him that he wanted for his team. One project manager interviewed did not believe his team had 100% integrity, stating that they might be unethical in their approach to ISD. The redundancy process might had been useful in removing those developers considered by their managers to be unethical, but after the process was completed one manager exasperated,

> 'We need those people who have gone, we need skills like that but, we just fired these guys' (Senior Manager)

Training was typically found to be inappropriate or cancelled, or both. Two separate training courses booked for three software developers in one project team were cancelled due to the freeze on the training budget during the period of heightened frugality. One developer was in receipt of an aeroplane ticket for three weeks boot camp training in the US, which the technical director cancelled the day before the flight was due to leave the UK. Consequently, one developer acknowledged that there was a problem with the lack of knowledge to use some of the software tools available for development.

Training invariably took the form of being either on the job or *ad hoc*. Internal courses about methodologies were found by developers to be difficult with regard to transferring the examples provided in the course to their day-to-day project work. The examples provided appeared to match perfectly with simple scenarios, but when the developers considered what they were working on and how they had multiple independent systems all working together, knowing that problems would spread out through a network, they could not visualise how these problems would scale up. Some courses were compulsory but considered a waste of time when they had been completed previously, even if with a previous employer. Two internal trainers who came to west London from Australia for two weeks admitted they only knew some parts of the code they were supposed to teach. In another incident, a senior developer was unaware of the meaning and use of the 'Wait' state which reflected the status of a reported bug. Courses that were attended were found to be very time consuming and an alternative way for the developers to learn was preferred. One presentation given by the quality manager on peer reviews lasted one and a half hours and contained eighty five slides - with insufficient time available to show the last twenty. Software used in training sessions was itself described as buggy by the trainers, with requests for new releases without the known bugs rejected by senior management. Even a project manager in a presentation to his team incorrectly spoke about delegating responsibility and not authority. Moreover, a six month course in Switzerland for new developers was considered - by the head of UK Recruitment and others - to be:

> 'So unbelievably intensive it is amazing they don't come out of there with a nervous break down' (Senior Manager)

Principle 5.05 requires that project managers and team leaders ensure realistic quantitative estimates of cost, scheduling, personnel, quality and outcomes on any project on which they work or propose to work, and provide an uncertainty assessment of these estimates. From the data collected, twenty two examples of non-compliance were recorded. The evidence found of incidents considered to be important included:

- 'Time plans, product quality targets and time for testing were all unrealistic' (Developer)

Explanation of Non-Compliance

The evidence available in the data to try and ascertain how the above non-compliance took place included a common lack of resources. (Only a quality manager reported having the resources needed to carry out his role effectively.) Although the lacking resources consisted of time, money and staff, the dominantly lacking resource across all development teams and business units was time. This was further supported by a request from project managers to their developers to modify their tasks - to something less time consuming - in order for project deadlines imposed by senior management in the US to become achievable. This was in conjunction with a perceived high pressure to finish earlier than officially necessary, felt generally. Project plans were thus described by one project manager as being tough on everyone as they were considered to be extremely aggressive. Consequently, the US parent was generally seen to have a negative effect on the UK workforce. In practice, projects were delayed, with a request made on one occasion by a technical consultant to send someone to a customer 'begging for more time'. One completion date was set cautiously as 'September, approximately'. Telco was thought to be more willing to let deadlines slip than her competitors, noted one developer, with one project overrunning by 14 months and with the development still not complete. Completion dates were agreed by all parties concerned initially, but the level of quality to be expected on those completion dates was unknown. A summary of the situation typically experienced by one project manager was:

> 'We have done fag packet designs - although that sometimes results in the right decision not being made' (Senior Manager)

This situation was exacerbated further when, as the data collected shows, weekly project plans were not consistently created by project managers. Code reviews were thus delayed or shortened, with software shipped to customers with known bugs. Requirements were, on occasion, even found to be invalid at the end of the - long - development stage. When limited time was available and decisions had to made in deciding which functionality in the requirements was to be developed first, customers were not consulted. Functionality which was expected to create the earliest revenues were prioritized and developed first. Moreover, estimation tools provided by Telco for managers to aid their planning and project management were invariably found to be unused, with estimates typically based on individual experience and ability.

4.8 Non-Compliance to Principle 6: Profession

Principle 6: 'Profession' is summarised as: 'The integrity and reputation of the profession will be increased - consistent with public interest'. A breakdown of the non-compliance found across all the project teams and business units is shown in Tables 19, 20 and 21 below.

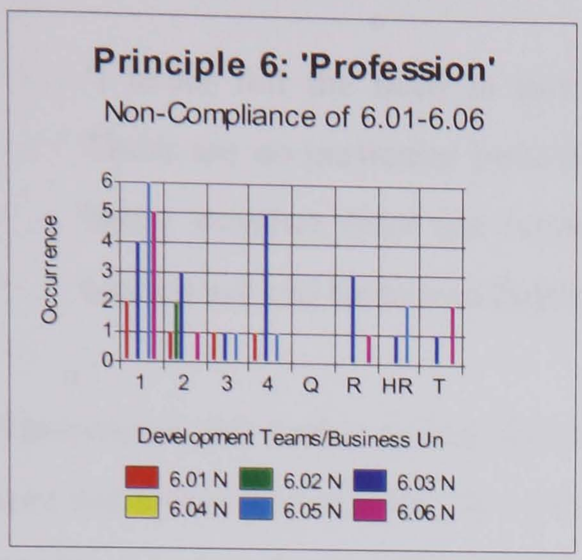


Table 19. Principle 6 Non-Compliance of 6.01-6.06

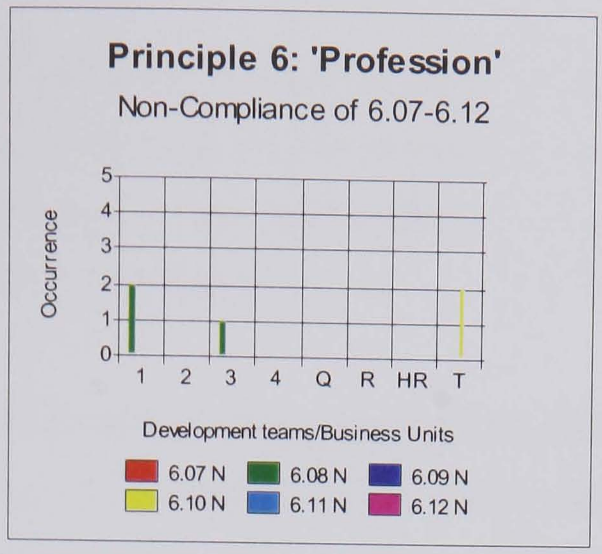


Table 20. Principle 6 Non-Compliance of 6.07-6.012

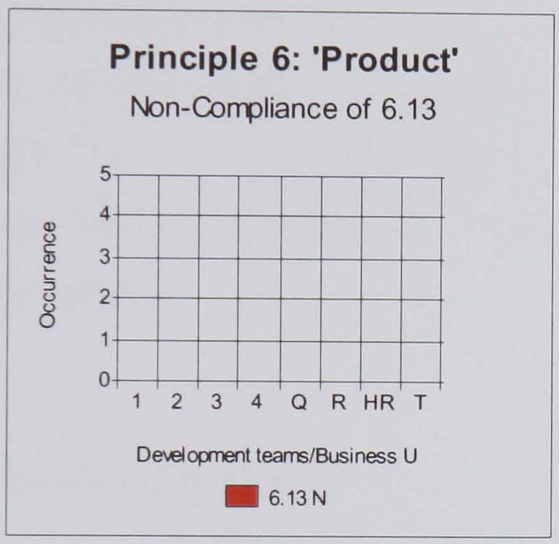


Table 21. Principle 6 Non-Compliance of 6.13

Principle 6.03 requires that software developers extend software development knowledge by appropriate participation in professional organisations, meetings and publications. From the data collected, eighteen examples of non-compliance were recorded. The evidence found considered to be important included:

- 'I am not a member of any professional body' (Senior Manager, Developer)

Explanation of Non-Compliance

The evidence available in the data identifying how the above non-compliance took place included: some developers spent so many hours working - in the office and at home - that when they did eventually finish they felt like doing something entirely different. This did not include attending any meetings hosted by professional organisations. The few reasons for not joining a professional organisation such as the IEEE were commonly held by those participating in the research:

'I never felt the need to join and I don't find them terribly relevant. There are no particular benefits. I can never be bothered to join and not being member does not actually hinder my work (Developer). Their leaders are old fashioned fuddy duddies' (Senior Manager)

Frameworks and codes created by professional organisations advocating best practice were not seen to be of value by senior management and, as such, Telco did not pay the membership fees for developers and project managers to join such organisations. Other departments within the company were perceived not to adhere to frameworks or codes of conduct which had a negative affect on the development teams. Entry criteria to the professional organisations for membership did not need to be met in order to join, as one project manager found - he then questioned whether he wanted to be a member of such an organisation which lets members of the public join without any proof of suitability. Furthermore, the leaders of professional bodies were found not to be respected by practitioners in industry.

4.10 Non-Compliance to Principle 8: Self

Principle 8: 'Self' is summarised as 'Developers will take part in life-long learning of the profession's practices and promote an ethical approach to the practices used'. A breakdown of the non-compliance found across all the project teams and business units is shown in Tables 22 and 23 below.

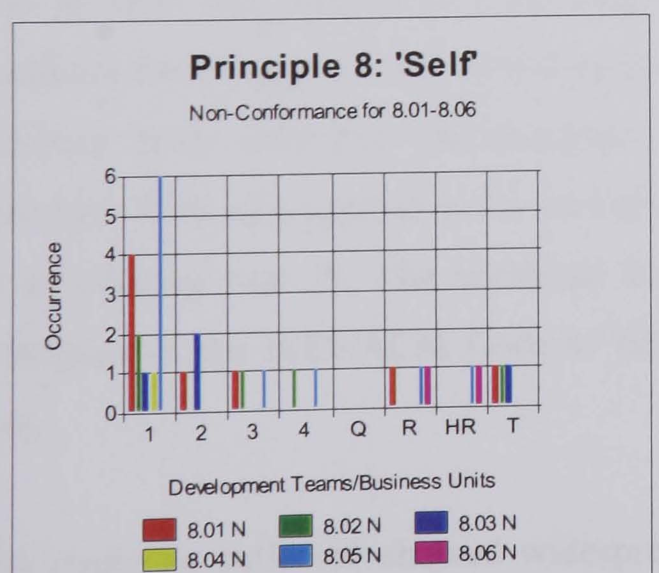


Table 22. Principle 8 Non-Compliance of 8.01-8.06

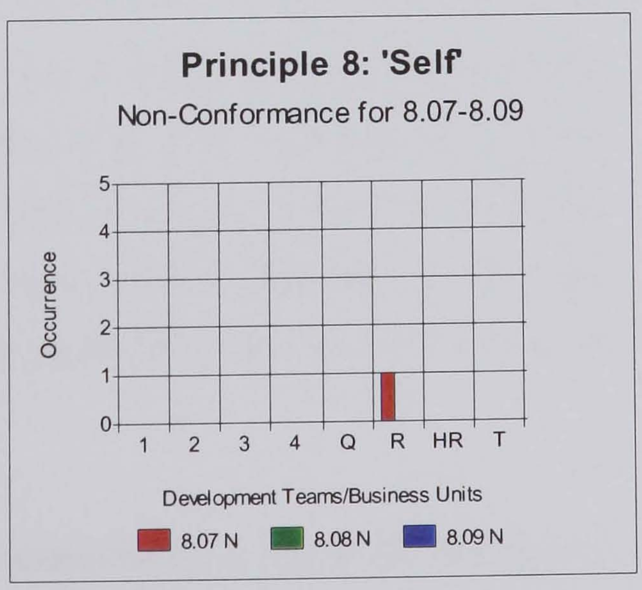


Table 23. Principle 8 Non-Compliance of 8.07-8.09

Principle 8.01 requires that software developers further their knowledge of developments in the analysis, specification, design, development, maintenance and testing of software and related documents, together with the management of the development process. From the data collected, eight examples of non-compliance were recorded. The evidence found of incidents considered to be important included:

- 'I don't attend courses of any kind' (Developer)
- 'I've got a lot of things I need to learn and it would probably do me some good to go on some courses' (Senior Manager)

Explanation of Non-Compliance

The evidence available in the data identifying how the above non-compliances took place included courses not being attended as they were considered to be unappealing. Developers admitted they had a lot of things that they needed to learn but it was thought unnecessary to attend courses to learn about them. Courses were acknowledged by some developers as beneficial but they were not mandatory. This was despite, on one occasion, one team leader and senior developer not understanding their own project Gantt chart. In a departmental meeting where several project teams were in attendance, only one developer present had been formally trained in testing procedures - which was conducted at a previous company and, therefore, unrelated to Telco's GEM model.

4.11 Summary

The in-depth and descriptive case study conducted enabled an understanding of the unethical environment within which project managers and software developers have to function to be identified and described, including their own unethical development practices. This also applied to the business units with which they interact in the course of developing new IS. The unethical practices found were categorised by the eight principles of the IEEE/ACM Code of Ethics - to enable the research objectives to be met.

The evidence collected showed widespread non-compliance to the Code principles - from software developers, project managers and the other business units - relatively where appropriate. As an objective of the case study was to identify and understand the unethical practices specifically in the software development process, it was to be expected that a higher number of incidents of non-compliance to Principles 3 'Product' - the development of new IS - and 5 'Management' - project management - were recorded than the other six principles which make up the Code. The evidence identified the following pertinent unethical practices as having taken place at Telco during the case study:

- There was a lack of understanding by project managers as to what exactly was required to be developed. This was facilitated by the infrequent use of Gantt charts, which were themselves complex and/or poorly maintained and, contained unrealistic estimates of the time required for ISD
- Staff development was typically unplanned, unmonitored and invariably inappropriate regarding content or teaching method. This led to general unawareness of the GEM model created internally by Telco for ISD
- Membership of professional organisations was scarce among project managers and software developers, contributing to general unawareness of timely legislation, except in HR where regular meetings were held with external consultants
- The membership of professional organisations was not perceived to be beneficial

A discussion of these findings is provided in the following chapter, enabling the more important research question of 'Why' to be addressed. The chapter describes the findings relating to the main principle areas which have greatest bearing on the issues drawn out in Chapter 2, contributing towards the aims and objects of the research. The discussion is structured around the units of analysis as described previously - at the levels of individual developers, project teams, project managers and organisations as a whole. A critical evaluation of the IEEE/ACM Code itself then follows in Chapter 6.

Chapter 5. Discussion of Results

5.1 Introduction

The data collected and presented as evidence in Chapter 4 showed not only widespread non-compliance to the IEEE/ACM Code by all project teams and business units participating in the research, but also across all eight principles. The evidence identified what unethical practices had occurred and how they were implemented. The causes are now discussed, which led directly, or indirectly, to these unethical practices, to enable a better understanding of why they occurred. The unethical practices will be discussed in relation to the units of analysis previously identified and described: individual developers, project teams, project managers and the organisation as a whole.

Section 5.2 discusses non-compliance to the Code by individual software developers. The non-compliance discussed relates to the process of development, maintenance issues, the value of documentation, the provision of training and approach to professional standards. Section 5.3 discusses non-compliance to the Code by project teams. The non-compliance discussed relates to the process of development, testing and code reviews and maintenance issues. Section 5.4 discusses non-compliance to the Code by project managers. The non-compliance discussed relates to the gathering of requirements, project planning and, recruitment and training. Section 5.5 discusses non-compliance to the Code by the organisation as a whole. The non-compliance discussed relates to project planning, the process of development, maintenance issues, the value of documentation, the provision of training and, approach to professional standards.

5.2 Evidence Collected Relating to Individual Developers

The evidence collected shows that the developers did not comply with at least 5 principles of the IEEE/ACM Code. The principles are: (1) Principle 3.01 Strive for high quality in the development process, (2) Principle 3.11 Create appropriate documentation, (3) Principle 3.15 Conduct maintenance professionally, (4) Principle 6.03 Participate in professional organisations, and (5) Principle 8.01 Increase knowledge in software development. The evidence from each is discussed below.

5.2.1 Process of Development - Principle 3.01

Principle 3.01 requires developers to strive for high quality, acceptable cost and a reasonable schedule, ensuring significant tradeoffs are clear to and accepted by the client and employer and are available for consideration by the public.

The development culture identified of 'deliver it now and get it working later' was found to have multiple causes leading to its existence. Customers invariably did not know what they required in the new IS they were requesting and, therefore, as one developer stated, there was no right or wrong when it came to determining the requirements. It was thus difficult for a developer to know what particular features were required as it was not possible to anticipate a customer's every requirement. When requirements were accepted by project managers, they still needed to be technically feasible and, sometimes there was a necessary trade off between what customers wanted and what was actually possible. On occasion the requirements acquired from customers were deliberately ambiguous so that when a developer considered the technical details he would then be able to determine the best way forward. This conflicted with a statement by the head of UK recruitment who explained that the developers were not permitted to change any requirements even when a better way was known - despite being initially employed due to their ability to think objectively. The process of determining complete and accurate customer requirements should therefore be improved, with a consensual level of abstraction in detail sought determined. Unambiguous communication should be sought between a customer and project manager, technical consultant and software developers, to facilitate pertinent and correct understanding before development commences.

The delivery of poor quality IS was caused, in part, by development and testing not being conducted thoroughly. Many developers admitted to not having read Telco's own manual on software development, called the GEM (Great Engineering Model) handbook. According to the technical director, this was due to the dynamic and changing methodology needed as development progressed. He further admitted it was difficult to define, or get right, the necessary development processes for the 1000 plus software developers in his division. This was supported by the findings of an independent quality audit, which concluded that almost no software development processes had been approved, or did not apply or, were too detailed and complex to be followed. When investigated through the interview transcripts, statements from project managers confirmed the findings of the audit. Two project managers explained that their software developers did not strictly adhere to anything, but that it was also necessary to forget all about the formal and detailed processes and procedures in place. This was due to the difficult application of development models in practice, with project managers describing them as extremely difficult and an absolute nightmare - as found by the quality audit described above. In addition, there was a commonly perception of insufficient time available and, new IS were considered too complex to achieve the level of quality desired. Furthermore, no universal Work Breakdown Structure (WBS) model, for example, was found to be used, although strongly recommended in text books on project management, such as those by Yates and Cadle (1996). The only contingency identified by the project managers and developers was that as software is more flexible than hardware, bugs could be fixed after shipment to customers (Senior Manager). This was also found to be relied upon in the case where a developer who knew nothing about a module to be tested would test all of it and, that was perceived to be a waste of resources. Awareness should, therefore, be raised - in conjunction with any training available - of both development and project management models and tools available, with the tailoring of formal and generic models to individual projects permitted. The responsibility of adherence by developers and project managers to pertinent models should also be actively encouraged by senior management. The development approach adopted should enable amendments or additions to the initial requirements with relative ease. (Any financial penalties incurred by the customer for amending or increasing the functionality required should be determined in the original contract.)

Further evidence of unethical development practices included the taking of short cuts which were expected to expedite the development process. These were not isolated incidents as team leaders also encouraged graduate developers to make conventions for themselves, clearly showing no formal processes were being followed. As noted previously, the literature search found only 5% of developers followed a methodology - the rest just 'write code'. Adhering to a methodology or standard was seen to improve software development in theory, but in practice such adherence was expected to result in development time being unacceptably delayed. Customers were understood to want their bespoke new IS earlier than agreed, in addition to generic projects needing to be completed before Telco's competitors finished theirs. Additional new features found to be required after the initial gathering of requirements were created as bugs as less paperwork was needed - but the development time needed still increased as a result. Furthermore, development and testing were not thoroughly conducted due to the complexity of the software required and the limited resources available. These limited resources, as noted above, included time - especially when even tighter completion dates were imposed by the US than those originally set.

5.2.2 Maintenance Issues - Principle 3.15

Principle 3.15 requires developers to treat all forms of software maintenance with the same professionalism as new development.

Software code was shipped to customers with known bugs caused by insufficient time for development and, with development being a sequential process, testing was invariably shortened when necessary. The redundancy program resulted in fewer developers available to conduct testing - and other development work also requiring completion – in addition to the maintenance of existing IS located throughout Europe.

Irrespective of the cause of bugs, the process for their subsequent resolution was found to be open to criticism by the developers themselves. The following instances occurred which were identified as hampering the maintenance effort:

- Debug output was sometimes edited
- Some debug output was missed due to the rate-limit restrictions on consoles
- Information recorded about bugs was not read but assumptions made upfront
- Information about bugs was apparently requested just to buy time
- Resolved bugs did not have evaluations conducted

The narrative so far has assumed bugs reported were accurately, completely and honestly described by customers. The evidence from the data collected, however, reveals this was not always the case. Customers occasionally used their IS in certain ways but then did not accurately report what they had done. Developers became knowledgeable over time as to the cause of some bugs reported and could identify the cause upon reading the initial bug report. An ongoing bug report - describing a bug which was not resolved initially - was often necessary due to customers not implementing the advice given initially. When bug reports were received, they were evaluated and then either nulled or fixed. They might be nulled due to a misunderstanding regarding the functionality actually provided. The customer then needed to be told it was not a bug but actually how the system was meant to work - which was quite common (Senior Manager). Unambiguous communication should, therefore, be sought between a customer and project manager, technical consultant and software developers, to facilitate a correct and complete understanding before even development commences. On one occasion, a request for a bug fix to the technical support department resulted in an email stating the bug had been resolved, although the system showed no sign of correction. The compilation of weekly bug reports for project meetings may have contributed to the reported fixing of bugs so as not to show untimely resolutions which would reflect negatively on the developer(s) assigned.

5.2.3 Value of Documentation - Principle 3.11

Principle 3.11 requires developers to ensure adequate documentation, including significant problems discovered and solutions adopted, for any project on which they work.

Project documentation was described by one developer as normally being unsatisfactory. Even when documentation was considered satisfactory, it was not expected to make much difference to the quality of the development process as there was wide consensus on developers not reading project documentation. Documentation templates should, therefore, be structured so that only pertinent sections are included and navigation is easy for the reader. The value of quality documentation should be given a high profile to encourage developers, project managers and others as necessary to take its creation and timely updates - with appropriate re-distribution, seriously.

An additional cause of the untimely resolution of bugs found was the initial design documents. These could consist of 60 pages or more with as much detail as possible entered into them. The situation frequently arose, therefore, where nobody could be absolutely sure of what was being delivered due to the sheer volume of information provided. It was also expected that customers would use software in unknown ways which would lead to additional bugs being reported which also incurred a time overhead. Here the sheer volume of information provided actually had a negative effect on ISD, whereas earlier insufficient information was identified as problematic. The problem exists therefore of knowing how much information is the right amount: too little leads to not enough being known to develop a new IS and, too much prevents everything from being known due to the time required to read the documentation, further compounded by navigational issues. The answer is subjectively decided by every potential reader of a document - dependent on their own knowledge, motivation and abilities.

5.2.4 Provision of Training - Principle 8.01

Principle 8.01 requires developers to further their knowledge in analysis, specification, design, development, maintenance and testing of software and related documents, together with the management of the development process.

Some software development tools available on the intranet were either not known about or not used by the developers. This was due to a common lethargy felt towards courses. The causes for this were found to be (a) insufficient time available to attend, (b) insufficient budget available to request any training, (c) course content irrelevance to day to day tasks, (d) the nature of development work was considered unsuitable for classroom-based learning, (e) course material was inconsistent and/or, (f) quality of material provided was inconsistent. Internal, yet formal, training using appropriate test methods should be provided for developers, which would address these weaknesses and facilitate a reduction in the skills shortage in the UK. The training also provided for developers to be able to carry out credible and sufficient testing was found to be limited. Little information was available for them on how to do testing, including via the intranet. It was assumed by managers that developers knew how to adequately test from their other training at Telco or previous employers.

Even with adequate training, testing was still seen to be flawed due to it being carried out to some specification, but that in itself was subject to the criteria of the test. If the test criteria were not exactly the same as what the customer was going to simulate in his environment, the results, therefore, were not necessarily going to be the same. Testing could be done to what were believed to be the customer's requirements, but the customer's environment may not necessarily have been what was expected by the developers conducting the testing. Relevant and timely training to developers and project managers should then be identified and included in project schedules and delivered via appropriate or preferred training mediums. Communication between project managers and customers should also be increased to facilitate correct and uniform understanding of the IS requirements and, the environment in which it is to be used, to enable a duplicate environment to be replicated at Telco with only pertinent tests conducted.

5.2.5 Approach to Professional Standards - Principle 6.03

Principle 6.03 requires developers to extend software development knowledge by appropriate participation in professional organisations, meetings and publications.

Developers and project managers typically found little or no relevance or connection to professional codes or standards. Furthermore, so many hours were spent at work in addition to those officially required, that when they did eventually leave the office they felt like doing something entirely different. This did not include attending meetings hosted by professional organisations. Few reasons were given for not joining professional organisations such as the IEEE but, were commonly held:

- There was no need to join
- The organisations were not terribly relevant
- There were no obvious benefits
- Developers could not being bothered to join
- Non-membership was perceived not to hinder one's work

The hours spent at work therefore need to be reduced to enable other activities - such as attending meetings hosted by professional organisations - to occur. Working in the evening was commonplace and found to be due to (a) flexible working time, (b) unrealistic timescales allocated to development as discussed above and, (c) meetings were arranged by US managers requiring UK attendance without consideration of the time difference. Flexible working time was offered as a benefit, but this was clearly only at a local level for the developers and managers who exercised this option. With the non-attendance of meetings hosted by professional organisations, the benefit was counter-productive for the profession as a whole. Where normal working hours can be achieved, the abolition of flexi-time would enable attendance of these meetings.

The evidence collected further showed that the leaders of professional bodies were not respected by those in industry, as:

> 'They are old fashioned fuddy duddies who are not in touch with what people need on a day-to-day basis' (Senior Manager)

The widespread non-membership to professional organisations contributed to companies as a whole not adhering to standards set out in frameworks or codes, such as the IEEE/ACM Code. The frameworks and codes created by professional organisations were generally not seen to be of value by either project managers or senior managers, hence their lack of advocacy and subsequent general non-compliance. Meeting the entry criteria for membership of one professional body was found not to be consistent as one project manager found - he then questioned whether he actually wanted to be a member of such an organisation. Furthermore, Telco did not pay the membership fees for employees to join professional organisations, as this had been abused in the past where some employees had joined executive type (golf) organisations. For membership of professional organisations - leading to compliance to their codes - to be seen as valuable by developers, it is necessary for management to see them as valuable too. The criteria for membership needs to be consistently required and met to facilitate the identification of value and, payment of fees by companies would encourage membership, with the organisations and levels of membership previously consulted and agreed between senior management and the organisations.

A further cause of non-adherence to professional standards was, as described by one developer, that being professional took the fun away. The developer might be better described as a maverick than a free spirit - with the latter sought in the interview process and, the former not. For those developers who had read the GEM handbook which described the standards and processes for Telco developers to follow, their ability to recall its requirements was found to be problematic. One cause of this was the handbook was found to be unclear or difficult to understand. Guidelines from professional bodies should, however, be taken seriously when creating company-wide processes and standards - which should then be tailorable as appropriate by project managers at a local level. These should then be available in multiple languages to reflect the diversity found in the workforce of large international IS companies. One senior developer noted, however, that standards would be dropped if a developer had to work with someone he did not enjoy working with.

A defined plan should be included in the preparation stage for any process improvement about to be implemented, noted a quality manager, which should be executed with commitment from the respective development team(s). Conflict which may arise at this stage might include time constraints on training and, the incompatibilities of time-to-market pressures versus the need to document the design details and test plans, as discussed above. The quality manager continued to explain that as a lack of resources was problematic, project managers needed to be shown the value-add associated with standards being implemented. As found in the initial literature search, when the context of a situation is known the opinions of developers concerned regarding any consequential action or benefit can differ significantly from those who are unaware of the context. This was supported by the quality manager above who believed consultation with developers and managers was necessary to facilitate the successful implementation of any process improvement plan. With legislation controlling many of the processes implemented in HR, the adherence here was the opposite found to the IEEE/ACM Code. The law demands adherence or face prosecution through the courts. The law was seen to change too quickly, however, for those employed in HR to be able to keep up with it. In an attempt to ensure laws were not broken, a consultancy firm was used on a periodic basis to brief the HR department in any recent developments pertinent to their work. With consultancy firms providing timely information to the HR department regarding their legal obligations, the project teams should also benefit from consultancy firms visiting the company to update them on both legal and professional software development/project management requirements - due to their explicit lack of time to explore these areas themselves.

With regard to Recruitment, a professional body did not exist for employees to be able to join, which would - theoretically - have led to their awareness of its code of conduct and their subsequent adherence. Achieving royal chartered status did not improve the public image of the CIPD, however, noted one employee in HR. He further noted that he had not felt the need to visit the company's code of conduct or know it verbatim since becoming a member of the CIPD. Time was unavailable to attend meetings and learn about new standards or better practices, stated one manager, as discussed above. Standard interview techniques laid down by its professional body - the Chartered Institute of Personnel Development (CIPD) - were not used by the Recruitment staff. Instead, the tests were Telco specific, i.e. the interview techniques used were generated in-house. This was due to the desire of Telco's President/CEO to concentrate on behaviour, as he considered it more valid than asking an applicant how he would approach something and then be provided with a past experience of how he did something. Here then, we see evidence of a (Chartered) professional body describing how a process should be conducted, which was then rejected - by senior management - in favour of a process designed in-house - and, created in the US but for use across the company worldwide. A simple acknowledgement of the cultural diversity present in the world would find global implementation of localised processes to be problematic - hence the evidence identified.

5.3 Evidence Collected Relating To Project Teams

The principles of the IEEE/ACM Code do not seek compliance from project teams explicitly. The Code is aimed at developers and project managers, which tacitly implicates teams. The evidence collected shows that the teams did not comply with at least 3 principles of the IEEE/ACM Code. The principles are: (1) Principle 3.01 Strive for high quality in the development process, (2) Principle 3.10 Ensure adequate testing and, (3) Principle 3.15 Conduct maintenance professionally. The evidence from each is discussed below.

5.3.1 Process of Development - Principle 3.01

Principle 3.01 requires project teams to strive for high quality, acceptable cost and a reasonable schedule, ensuring significant tradeoffs are clear to and accepted by the client and employer and are available for consideration by the public.

Following the global cutbacks and widespread redundancies, additional problems arose. One of the problems found to exist was that project teams were still trying to pursue processes that were written for projects with project managers, progress chasers and program managers. In other words, processes created and implemented for a large staff were no longer applicable after the cutbacks, but were being followed despite their newly found unsuitability. Relevant and timely training to project teams should, therefore, be identified and included in project schedules - via appropriate or preferred training mediums - to reflect organisational changes. A balance needs to be determined regarding the autonomy and empowerment given to departments and project teams and, the centralisation necessary to ensure company-wide quality and consistency in both project teams and other business units, as found above.

Due to the importance of a uniform understanding of new requirements, team-working skills were identified as being more important for those working from home than for those based in the office. Working from home was not found to be beneficial, therefore, albeit common among the teams. Graduate developers invariably worked in the office but found it difficult to seek help when other - more experienced - developers worked from home. In addition, office dividers found throughout Telco offices did not aid communication which further contributed to the lack of understanding of what was required to be developed. One developer actually preferred the dividers, stating conversation with his colleagues hampered his progress. Office-based working and the removal or reduction of barriers would therefore facilitate the correct understanding of new IS requirements.

The problem of understanding what was required for new IS, as identified above, was further compounded by the physical locality of the project teams involved and, the respective languages used. These physical boundaries were identified to be national, international and global, with teams located, for example, in Uxbridge, west London and San Jose in the US. The national origins identified as causing problems in communication were Italian, French, Chinese and Indian. Time zones, language difficulties and physical localities should, therefore, be identified and considered in all conduits of communication.

Feature Based Development implemented by the project teams taking part in the research did not reduce development time as expected as, in part at least, it could not be predicted how multiple independent components would perform under testing. Furthermore, modules from other project teams were found to be unstable themselves and, it was difficult to identify who was most suited to do the combined testing, noted one senior developer. Feature Based Testing was preferred by project managers to be an optional phase as it was considered not to be appropriate for small project teams with limited requirements scope. A further definition is highlighted here then to distinguish testing prior to release known as Feature Based Testing whereas customers exercised Solution Testing - i.e. post-delivery. This was to enable all functionality affected by bugs triggered to be identified by customers and, additionally circumvented the need to ascertain who might be most suitable in the project teams to conduct the intra/inter multi-component testing, noted as difficult above.

The documenting of test cases was seen to be problematic due to areas in templates not thought wholly relevant by the project teams because they were seen to be more related to hardware and other areas. This was supported by a quality manager who stated that project teams had stated that the templates were too hardware centric and conversely, the hardware teams had stated they were too software centric. Again, abstraction for all through centralisation did not work at a local level. One team leader even suggested creating a new generic test plan by using an old plan and then taking out the specifics. Templates should therefore be generic and tailorable, thereby enabling the completion of only necessary sections as appropriate for the task at hand.

Although Telco had a business code of ethics in place and which was freely accessible via the intranet, project teams generally unaware of its existence. Furthermore, the areas covered by the code were general business issues and not specific to software development. In addition, many developers in the project teams participating in the research had not read the GEM handbook. This was due to a commonly held perception that adherence to a methodology or standard was theoretically seen to improve ISD, but in practice increased the development time which could not be permitted. This was due to customers wanting their bespoke systems sooner and generic projects needing to be finished before competitors in the marketplace finished theirs, as discussed above. The technical director further explained it was difficult to define and get right the processes necessary for the large number of project teams in his division.

5.3.2 Testing & Code Reviews - Principle 3.10

Principle 3.10 requires project teams to ensure adequate testing, debugging and review of software and related documents on which they work.

The review process was considered to be inappropriate due to the possibility of new code requiring a formal test possibly consisting of two lines or a thousand. Due to the tight deadlines set by senior US management, some code was not reviewed which should have been - with the last 10% of test cases thought to be the most difficult and known to take up to 30% of the total time allocated for testing. Where reviews were conducted, test cases which were described as one-liners were not documented as there was no process in place for them. Consequently it was not possible to be totally honest at weekly project status meetings with the US regarding how many test cases had been completed or, how many tests had passed or failed.

One US manager wanted to impose the use of a generic test plan on the UK project teams but a UK based project manager rejected the request arguing it was not a documented requirement and, therefore, he was not obliged to do so. On one occasion, an additional month of testing was required after the formal testing period had finished. Testing was only conducted on small sections of code, with several tests carried out on the same section. The tailoring of formal and generic models to individual projects should, therefore, be permitted with the responsibility of adherence by developers and project managers to pertinent models highlighted. Furthermore, sufficient and realistic time for testing should be agreed at the outset without external influence and scheduled into the Gantt chart - for both development and testing, enabling testing to be both formally conducted and documented. A shortage of skilled staff further prevented the delivery of high quality IS. It took one project manager eighteen months to find a developer good at testing to employ. Only then was it was possible for the project manager to take on other developers to be trained in testing. The project manager believed he must have developers who could also test to a high standard or the code would suffer. The situation was exacerbated by the technical director who believed code reviews took too long and should even be shortened. The motivation for this statement was made evident in a project meeting when he clearly told the development team that at the end of the day they had got to get the stock price up again, having fallen from over $80 to under $12 in the previous twelve months.

Although there was a goal set by US management of a 95% pass rate for testing, code was released to customers containing known - and sometimes severe - bugs, otherwise the product was expected never be delivered due to the time required to test it fully. Although (some) test cases were written prior to testing and were included in the Test Plan and Test Specification documents, two team leaders stated that as it was infeasible to test for everything, test cases should be created and written by a developer as he progresses with testing and, not only solely beforehand. Additional reasons for this provided in the data include the difficulty or even inability to manage a large number of test case documents which themselves could become out of date very quickly. A project manager could decide not to have code reviewed if he felt it to be inappropriate, but that then led to untested code being shipped to customers. One developer confessed to his project team to being quite bad at testing, but stated that it could not be helped. This was found to be caused by insufficient time available (caused by tight deadlines and later, reduced manpower), inappropriate or non-provision of training, poor documentation and, a personal preference by the developers to concentrate on development. A distinction regarding code containing bugs found worth noting is that software released without sufficient testing was considered different from software released which was known to contain bugs but still worked to a satisfactory level. Unknown reliability was acceptable to some customers, however, when they were given software prior to the agreed delivery date to enable the tailoring of generic software to individual requirements by Telco's Professional Services Department. Due to the nature of development, customers should be educated, therefore, to understand that it is not possible to test new IS 100% and, consequently bugs will undoubtedly be present. Test cases could be written completely prior to the start of any testing activity, but dependent on the original design documents being complete, consistent, accurate and with consensual understanding of what was required. As none of these was provided at the outset of any new ISD, test cases should be created during the testing period to accommodate those missing from the test cases prepared previously from the design documents.

5.3.3 Maintenance Issues - Principle 3.15

Principle 3.15 requires developers to treat all forms of maintenance with the same professionalism as new development.

Although in practice this was found, the Code tacitly expects ethical professionalism to be implemented in development, which was not found at Telco. Even statistics collected internally by Telco found the most common reason for a system upgrade was to fix bugs. Unprofessional maintenance was found to be caused by activities both prior to, and post, shipment of new IS. The activities within each stage are now discussed.

Development time for new IS was reduced further by time needed for maintenance of existing IS. Investigations into some bugs reported by customers in software already installed were found not to be bugs at all. The code for the functionality reported to contain a bug was frequently found to not even exist in the software. The cause was found to lie in (a) a misunderstanding of requirements from either Telco's or (b) the customer's viewpoint, or both, or (c) the functionality was expected to be present due to promises made by salesmen - regardless of its technical feasibility - and was ultimately not included in the IS released. Customers should be educated to understand (a) it is not possible to test software 100% due to the inability to predict how it will be used and what future needs will be placed on it, (b) quality software costs time and, therefore, money. 'You get what you pay for' is just as true for new IS and, that (ideally) cost should not be a factor. In addition, a high profile should be given to the importance of appropriate and adequate testing and formal reviews. Salesmen should sell only feasible IS, as discussed above.

Bugs reported and corrected under maintenance contracts were caused, in part, by tight project development deadlines set by senior US management, preventing some code from being formally reviewed. In addition, the technical director believed that not all code needed to be reviewed and, code that should be reviewed should be completed in less time than was actually being taken. In addition, one senior developer explained that the generic review process was inappropriate due to the unknown length of code, as it could consist of 2 lines or 1000. Even then, testing was usually carried out on only a small part of the entire project code, with several tests carried out on the same section. The last 10% of test cases were generally found to be the most difficult to conduct and could take up to 30% of the total test time allocated. The number of bugs found post delivery, therefore, was greater than necessary due to the relatively small amount of testing conducted prior to shipment which ultimately increased the bug fixing process.

The causes of bugs reported post-delivery and found to be fixed in an untimely manner were numerous and, also included developers being unfamiliar with some areas of code where the bugs found were believed to be located. In addition, the quality of the code itself was reduced when maintained due to the relatively short period of time developers were employed and, consequently, no developers knew any sizeable amount of code well.

The process of bug fixing which contributed to untimely resolution was criticised by developers themselves as when bug reports where received, the following were often not provided:

- What triggered the bug(s) to occur
- What had happened that should not - and vice versa
- Whether the customer had recently upgraded from a version which was working
- Whether the bug was repeatable
- What network topology, i.e. manufacturer/model of equipment was involved
- If there were any other bugs reported which might be related - by this customer or any other

The above took place despite the internet-based form - requesting information from customers when they log bugs - requiring answers to many questions, although specific details about the actual problem were not sought until the fourth page. In this case, the process of eliciting as much information as possible was actually counter productive.

Moreover, other issues identified which delayed the bug fixing process included (a) information recorded about bugs was not read but assumptions were made instead, (b) information about bugs was requested to gain more time and, (c) resolved bugs did not have evaluations carried out for lessons to be identified and shared - both the strengths to be repeated and the weaknesses to be avoided. A lack of training, communication, time and documentation can all be seen to contribute to the delays experienced.

5.4 Evidence Collected Relating to Project Managers

The IEEE/ACM Code seeks compliance from project managers with Principle 5 addressed explicitly to the managers. The evidence collected shows that the project managers did not comply with at least 3 principles of the IEEE/ACM Code. The principles are: (1) Principle 5.01 Gathering requirements, (2) Principle 5.05 Project planning and, (3) Principle 5.04 Recruitment and training. The evidence from each is discussed below.

5.4.1 Gathering Requirements - Principle 5.01

Principle 5.01 requires managers to ensure good management for any project on which they work, including effective procedures for promotion of quality and reduction of risk. The evidence collected falls under this remit but also - and more explicitly - under the requirements of Principle 1.02 which requires the interests of software developers, the employer, the client and the users be moderated - with the public good.

Project managers was found not to moderate the interests of all parties involved in new ISD with a boxed solutions approach adopted for development to enable the creation of generic products for all its customers. This was due to the boxed solutions approach expected to result in facilitated project management, reduced development and maintenance costs and, reduced training for both developers and customers. Understanding the requirements of new IS was known to be difficult, but expected to be worse for bespoke ISD than the boxed solutions approach advocated.

As described above, the situation was compounded further by salesmen selling new IS with functionality which did not exist, resulting in the consequential extended development time-scales exceeding the expected period for revenue generation. Salesmen additionally sold new IS to customers too quickly, with little or no regard for the information required by a project manager. This information was essential to both try to understand the requirements and, create the necessary project plans with realistic estimates of time and other pertinent resources. This issue relates to the point made above where managers developed boxed (generic) IS rather than bespoke IS largely due to economic factors. On occasion, however, it was necessary to create designs 'On the back of fag packets', noted one manager, who exasperated,

> 'We already have fifty other priorities that we are looking at and we have got to make a quick guesstimate which might sometimes result in the right decision not being made' (Senior Manager)

In practice, salesmen were selling pseudo bespoke systems to close deals quickly - thereby securing their commission. Although there was evidence of a 'providing solutions' approach being exercised by salesmen - which was strongly advocated by one senior project manager - the project manager's own working practices did not enable him to follow this through and develop what was ideally required from a customer's viewpoint - i.e. affordable and bespoke new IS, as originally offered by the salesmen.

5.4.2 Project Planning - Principle 5.05

Principle 5.05 requires project managers to ensure realistic quantitative estimates of cost, scheduling, personnel, quality and outcomes on any project on which they work or propose to work and, provide an uncertainty of these estimates.

Gantt charts were used by project managers to facilitate the planning process and were known to be problematic - both in their structure and their use, with one team leader very reluctantly contributing to the creation and timely amendments of his project's chart. A boxed solutions approach was expected to result in fairly consistent use and structure of charts to reduce the problems associated with creating and using new charts for each new project. Delays experienced in projects were blamed, however, at least in part on 'messy' Gantt charts, with the team leader described above believing the title of his project's chart should be 'Lies!' One manager did not update his chart on a weekly basis as was expected, which resulted in one developer asking for a copy of the chart used by the author so that he knew what was expected of him and when. Increased awareness of Gantt charts regarding their use and value should therefore be sought, facilitated by the inclusion of accurate and timely known tasks to all team members.

Project completion dates were invariably set by US management and, although project managers accepted them, they were unable to confirm the level of quality expected on delivery. This process did not concern one project manager as he was not personally responsible for setting the milestones and completion dates. This was despite most project managers and software developers holding stock and stock options and, therefore, having a vested interest in the IS they were developing. On one occasion developers were asked to modify tasks to something smaller and less time consuming by their project manager in order for the deadlines set to become more feasible. One cause of the setting of tight deadlines was to prevent new bugs being discovered in an otherwise long development period - which could be expected to create additional delays. From the data collected, evidence found both customers' and Telco's competition contributing to the pressure to impose tight deadlines on project durations. This was in addition to the problems associated with salesmen motivation by commission rather than the sale of appropriate new IS with feasible deadlines and so forth, as described above. Consequently, a high pressure to finish early existed in the project teams. One developer explained that at a previous employer shipment dates for new IS were calculated by finding out from the customers when their competitors were offering to complete and, then reducing that time scale by six months. When determining anticipated duration times of new projects, therefore, a consensus should be achieved by customers and project managers on what is feasible so that realistic deadlines can be set.

Estimation tools provided by the company for project managers to create initial formal proposals were not used due to the perceived nature of estimation. Estimation was generally an art more than a science, noted one manager, and was therefore based on experience and ability. This approach - called the Analogy Method - to estimation is supported by Yeates and Cadle (1996). Gantt charts in the planning process of new ISD were created with a project manager's pseudo knowledge of what was required for the known tasks to be actioned by the team. Charts were typically created using 80% of each developer's available time with the remaining 20% allocated to distractions - including the sending and receiving of email. (Time available for project management was reduced for one manager by the receipt - and frequent response - to over 200 emails daily.) The 80% use of the total time available for each developer in a working week supports the work of Yeates and Cadle (1996), but these authors advocate the allocation of the remaining 20% of a developer's time to sickness and holidays - which was not done. As described above, one project manager received over 200 emails a day, thereby confirming the 20% allocation of time to email to be justified, or even insufficient. Better estimates of duration times for particular tasks were expected to be provided by developers. In one meeting a graduate developer was asked by his project manager if three weeks would be long enough to carry out a particular task. He said it would be, but then an experienced developer said it could be done in two weeks, which was then accepted - even though the graduate was going to be doing the work. On a separate occasion, two developers worked over a weekend to meet an imminent deadline, yet one of the developers stated that the testing would still not be finished for another month as he still had to execute another 180 test cases.

For potential new projects which were later confirmed, these preliminary designs caused at least in part, project plans which were then seen to be tough on everyone. There was also no standard WBS (Work Breakdown Structure) in use at Telco. The redundancies resulted in the reduction in size of development teams, which increased the delays further still. At one juncture, a request was made by a technical consultant to send someone to a customer 'begging' for more time. When the redundancy program had been implemented in full, the reduced project team to which the author had been assigned became the sole provider of maintenance support to all Telco's European customers and as a consequence, the developers expected all existing development to be delayed. In a pseudo attempt to prevent project delays, the completion date for one project was given only as an approximation although one developer found Telco to be quite willing to let projects slip. One manager conceded to having once overrun by 14 months and still had not completed the project. Where conflicts arose in deciding which requirements were to be developed first, the potential profit and its timeliness from each requirement was the determinant and, not any preferences a customer might have expressed. A balance needs to be determined, therefore, regarding the autonomy and empowerment given to project managers and the centralisation necessary to ensure company-wide quality and consistency in project teams and business units.

Developers were asked by one project manager to check the accuracy of his Gantt chart as they were 'messy' - made worse by the use of unclear definitions. Errors found in a Gantt chart included public holidays scheduled as working days, task durations were entered as '?' and, resources were allocated in excess of their availability, for example '152%'. Gantt charts were found to be difficult to understand without gridlines displayed to enable tasks to be lined up with their respective descriptions and start/finish dates, but could easily be over-cluttered with their use. On one occasion when a project Gantt chart was shown to team leaders containing all the project-related tasks, the team leaders requested all tasks be removed that were not directly relevant to them. The charts were also found not to include all the symbols shown in the key which caused confusion. Estimates of resources needed were thought to be more reliable when provided by developers. This supports the use of the Delphi estimation method - seeking the advice of (experienced) others - and, the Analogy method - using one's own experience and skills - as advocated by Yeates and Cadle (1996). Developers should be formally required to contribute to the estimation process, therefore, with customers educated to understand that resources are difficult to plan ahead as identification and quantification is ideally required by a project manager of both known and unknown tasks and bugs that can be expected. Furthermore, resources could not be readily identified and assigned to projects offering maintenance support only due to the nature of bug fixing: how many bugs could be expected, what severity could they be, how frequently could they occur, how many developers would be needed to investigate and remedy them, etc. As one project manager exclaimed:

> 'In today's market I am selling a machine that has to work with about fifty other products and you can't test every other companies' products inter-working capability with your own' (Senior Manager)

Some tasks found to be delayed were caused by a project manager who emailed one developer to notify him of his assigned tasks at his previous email address - for some weeks after joining the company - which were not actioned as consequence. Even then, tasks assigned to a senior developer and emailed to his correct email address were neither carried out by him nor chased up by the manager two weeks after the initial email was sent. On a separate occasion, one manager stated he had received nothing from HR regarding a login ID and password for a new developer, but on checking his email found he had received the information two weeks earlier but had not read it initially and then forgotten. Increased awareness of Gantt charts regarding their use and value - achieved by the inclusion of accurate and timely known tasks - would have enabled available time for the reading of email and monitoring of tasks completed. Impractical deadlines imposed by the US as described above also hampered the ability of project managers to be thorough.

The data collected identified a need for project deadlines to be delayed, as several developers and managers had noted there was too much to do in the time available to do it well. Time management was further tested by the receipt of occasional 20 plus page documents by project managers which needed feedback/approval within 24 hours. It was further acknowledged by many of those interviewed that there was never sufficient time to complete anything properly as there were always unforeseen problems, which although initially anticipated, could not be justified before a project began and therefore were not included in the project plans. Developers were, therefore, found to work (voluntarily and unpaid) at the weekends prior to their own holidays to help keep their projects on schedule whilst away. Furthermore, subsequent reductions in task durations were necessary due to customers demanding their new IS earlier than originally agreed and, the need to beat competitors who reduced testing to curtail both development time and cost.

One team leader only wanted the Gantt chart for his project to show activities on a monthly basis and, not day to day, due to the perceived excessive managerial control implied. Another team leader wanted two separate Gantt charts created for his team: one showing tasks assigned to his team and the other showing everything else associated with the project. Both team leaders were unfamiliar with some of the symbols shown in the Gantt charts created using Microsoft (MS) Project, with only one assuming MS Project would automatically update the predecessors and successors of tasks which were altered as a project progresses. In one project meeting, copies of a Gantt chart were distributed for the first time to the team leaders present. Other leader present had no Gantt chart at the prior to the outset of the meeting and, did not want to keep a copy at its conclusion. On one occasion when the Gantt chart was being updated with a team leader, he stated that as he would be on holiday the following week he did not really care about its content. Increased awareness of Gantt charts, therefore, regarding their use and particularly their value - facilitated by training and the inclusion of accurate and timely known tasks - to all team members should be achieved.

A spreadsheet which was used to record developers' holidays still included those who had been made redundant in previous months, justified by the project manager as being too much trouble to remove their details. This was due to software bought for use company-wide being adapted to enable a faster roll out to all departments internationally, but with a consequential decrease in user-friendliness. Software used for budgetary control was assumed by one UK project manager to show all figures in US dollars and not English pounds. This assumption was not clarified during the case study. One manager did not remember agreeing to holiday dates requested and then wanted to prevent the employee from taking the time off. In this instance it was apparently too much trouble to put into a spreadsheet - as opposed to the above. This is unless, of course, some project records were lost that had not been backed up, as was the case on another occasion. Training in the use of the new software should have been provided, therefore, to both facilitate use and clarify any ambiguities, for example, with the type of currency used.

One corollary to the multiple factors identified above as having a negative affect on project plans was that weekly project plans were either not always created by project managers and/or updated on a timely basis. To exacerbate the problem yet further, project files were known to have been deleted accidentally - by a *safety* feature in Telco's own IS.

One team leader held a meeting to brief developers in his team and used an out of date copy of the project's Gantt chart. No-one present in the meeting had a diary to determine what days certain dates fell on and there was no calendar on the wall, thereby preventing goals identified to be allocated to specific dates. On a separate occasion a project manager could not update his chart due to one of the developers having gone home early. The update then had to be postponed until the following day, with a hope that the developer would not be working from home. Increased awareness of Gantt charts regarding their use and value - facilitated in part by the inclusion of accurate and timely known tasks - to all team members should therefore be achieved. The value of quality documentation generally should be given a high profile to encourage project managers and others to take its creation - and timely updates with appropriate distribution - seriously.

Problems created by other departments (and external suppliers) were also given to development teams to resolve, reducing their available time for development yet further. For example, the late delivery of equipment ordered, a change in circumstances during development and, requirements being found to be invalid at the end of development. The latter can be seen to be caused in part by the high number and frequency of proposed new projects requiring estimates of time and cost from project managers.

Even without the impractical deadlines imposed by senior US management, under-estimation always occured, contributing to the late shipment of new IS due to the numerous factors and uncertainties which can affect a project. The reliability of hardware used in development was uncertain, as was its inter-operability with modules of code. Other pieces of hardware were also needed in addition to those initially identified. For example, difficulties arose between new software code written and the hardware used for its execution. Other problems arose between code written by outsourcing companies and the software used by the project teams. Uncertainties in the workplace compounded the problem further still, an explicit example being RiF - the mass redundancy program implemented shortly after the case study began which reduced the number of project managers and developers available.

Resources could not be accurately identified and assigned for projects where only maintenance support was required. This was due to the inability to foresee the future and predict which bugs would appear - in conjunction with their scope, impact, complexity and location within the code, with associated time and costs for correction, as discussed above (supporting the work of Paul 1994). This maintenance activity - bug fixing - was the only activity identified in the data collected, refuting the work of Yeates and Cadle (1996) who identified multiple maintenance tasks. They categorised bug fixing under the heading of Correcting, but they additionally identified the categories of Adapting, Performing and Perfecting. These latter categories were not found to be implemented by the developers responsible for maintenance.

One project manager was open to game playing with his budgets as he was prevented by senior management in the US from spending what was required on his project for it to complete satisfactorily. For example, the technical director had spent some of the project manager's budget, preventing him from buying a server which he had previously approved. The game playing included employing some permanent software developers as contractors as the head count for the project had already been reached and contractors were not included in this figure. Practices such as these extended to the US where one finance manager - as a personal favour to the project manager described above - deliberately left off pre-selected developers from a list the technical director had requested, identifying those who could be made redundant in the RiF process with minimum impact to the company. The project manager in question made a point of building bridges with other departments to facilitate his own work. The finance manager in the US had been described to the project manager as a battle axe when he first joined Telco, but he found that to be untrue. After meeting her in person and explaining how he worked and why and, ensuring that she had the information she required in a timely manner, she reciprocated by doing him favours. Communication between departments should therefore increase to facilitate a correct understanding of what each requires and when. As a result of the improved relationships, favours can be done and later reciprocated when difficulties arise - but unlike the situation described above, the increased understanding (at a corporate level) should result in honest practices being conducted which aim to facilitate the achievement of corporate goals and, not just local goals.

Some project expenses had been accommodated in one quarterly budget, but were covered by additional product revenues, freeing up some of the budget for other purchases to be made. A senior manager based in the US and responsible for this particular project was not told of this - initially unexpected - source of revenues. One project manager who had a request to purchase a new server for £200,000 denied by the technical director then went to his vice president for approval, which was granted, over-ruling the technical director's decision. The project manager above even stated people were 'Getting into bed' with him to get money. Another manager exclaimed that he had seen what people would do for extra cash and, it was not what he would call best business practice. As found above, improved communication between management levels should increase a correct understanding and justification of what each requires and when, resulting in open practices being conducted which aim to facilitate the achievement of corporate goals.

5.4.4 Recruitment and Training - Principle 5.04

Principle 5.04 requires managers to assign work only after taking into account appropriate contributions of education and experience tempered with a desire to further that education and experience.

When one senior developer was absent on long term sick leave, two other developers in the team took over the development of his module of code. The esoteric knowledge the senior developer had about the code's structure and functionality, in conjunction with his own advanced knowledge of programming, enabled him to write a lot of 'one-liners'. Although efficient in the number of lines of code written and subsequent increase in the execution speed of the module, the two developers who took over from him were unable to make satisfactory progress as their understanding of the code was limited. Delays to project plans were further caused by a lack of knowledge in the use of development software itself. Relevant and timely training to developers should be identified and included in project schedules, therefore, to enable adequate cover in times of absence.

A consensual view at Telco of graduate developers on their first day of employment was that they knew nothing. Additional problems began, however, at the outset of a developer's employment at Telco. One developer had no written job description and had spent the first three to four months of his employment with Telco by himself, searching the intranet for courses to attend and pertinent documentation to read. He had to wait three weeks for a computer after joining a project team and no mentor was appointed to answer his queries. The infrequency of staff reviews - if they occurred at all - resulted in many anomalies not being identified which was necessary for their possible correction to be determined and implemented as appropriate. Insufficient time available was identified as the cause by all those affected, with reviews postponed or cancelled at short notice almost systematically. This contributed to one developer wanting a copy of the author's project Gantt chart to identify what tasks he was responsible for, as he was uncertain. One senior developer was also unaware of the meaning and use of the 'Wait' state which had been created to reflect the status of a reported bug which had been identified and assessed as fixable - but with no-one assigned to carry out the work. Without the staff reviews to express or identify gaps in necessary skills to a project manager, the desired formal processes in place were not fully implemented, if at all. One manager explained they had not given developers a lot of information about how to do code reviews, for example, and there was no link in the intranet that informed developers about how to conduct them. One developer had received little in the way of supervision until informal weekly meetings were started by his project manager - but then only one was held. The manager's attitude towards the value of the weekly meetings became evident at the end of the first meeting when he told those present to bring beers the following week. This blasé attitude towards formality was found to be almost typical among project managers. Thus poor attitude needs to be addressed in conjunction with allocated time for reviews to correct the current infrequent/non-existent review process.

The existence of several definitions of terms used could have been identified in staff reviews and facilitated a reduction in development durations. The definitions of pass and fail with regard to testing new code, for example, were found to be the opposite at Telco to those taught on undergraduate modules at university. At Telco, a test passed if no bugs were found and failed if bugs were found. This was contrary to the definitions taught on undergraduate Computer Science modules. Subsequently, one manager explained that if he hired ten developers, it may be true that only three of them were actually going to be useful. Reviews would additionally have identified other areas relating to universities, possibly culminating in an improved alignment with the real needs of industry. The suitability of graduates was raised on more than one occasion, with development code for one project, for example, written in MDL, C and C++, whereas universities currently teach Computer Science undergraduates to program in Java. The quality of programming by graduates in MDL, therefore, was poor during the learning period. It took one project manager 18 months to find a good developer. Furthermore, the Gantt charts taught at university on Software Project Management modules were found to be small, simple and unrealistic when compared to those used by the project managers. University was seen, however, to be a good place for developers to mature and acquire some theoretical background.

With regard to testing, it was simply assumed developers knew how to test from training completed elsewhere, or from mentors or other business units. When a project manager based in the UK needed help to use a new application, he telephoned a colleague in the US for assistance. He justified this by saying people working in this field generally did not read manuals or attend courses, they just asked questions. An investigation into the data to ascertain why this was the case found:

- Too much literature available - a search in the company's database found 79,706 documents with the abbreviation 'EFT' (Early Field Trial) in them
- Inappropriate definitions or misuse of terms - 54 recorded incidents were identified in a cursory short search
- Some documents referred to obsolete processes
- One project manager explained that sometimes so much information is put into a document that navigation was time consuming and/or difficult

Having documentation containing everything was therefore not helpful, just as documentation containing insufficient information was not either. The corollary to these anomalies was that documentation was unread, skill gaps were unrecognised by project managers and, an acknowledged preference to learn by *ad hoc* experimentation on the job.

Training was down to individuals to initiate, whereby developers identified what training they needed and then booked an appropriate course on the intranet. In practice, however, training was drastically reduced due to the company-wide cutbacks. As a consequence, a technical director achieved this primarily by cutting back on training and travel expenses. Two separate training courses booked for software developers in one project team were thus cancelled. One developer was actually in receipt of an aeroplane ticket to the US for three weeks Boot Camp training, which the technical director cancelled just one day prior to the outbound flight. Inadequate training subsequently contributed to a lack of knowledge in the use of some of the development tools available. Another course that was cancelled would have led to an IEEE qualification for three other developers. The training cutbacks described were implemented at the time of heightened frugality in conjunction with the redundancy program, but some project teams were apparently oblivious to this drastic cost-cutting exercise. In one department a placement student still at university was booked on a 6 week training course in Brussels, for the third time, with no mention of a cancellation. Two other placement students in his team had already attended. One manager had to cut back on training and yet another manager was permitted to continue - even for placement students - due to the empowerment of directors to decide how to implement the budget restrictions imposed. In the latter case, training was not compromised, even when the reciprocated benefits would only be realised in the long term - assuming the students returned to Telco after their graduation. The severe reduction in training budgets therefore hampered the ability of developers to train in areas of need.

After the redundancy process and subsequent cutbacks were implemented, however, one project manager exasperated that they then needed the skills of some of the developers who have been let go. Regular staff reviews could have identified the timely skill-sets held by developers which would have resulted in a more accurate selection process implemented in the redundancy program. One project manager, however, additionally expected to take on a more technical role as his role was significantly reduced with only 5 developers remaining in his team. The manager had not written software since the early 1970s, however, and was additionally not able to review technical documents requiring his formal approval - both before and after the cut backs. With the turbulent economic environment experienced by Telco during the case study, financial constraints prevented the manager from receiving the training needed to contribute towards the cover necessary for the developers made redundant. In a steady and healthy economic climate, this situation may not have arisen.

The need for periodic reviews was strengthened by an examination of recruitment practices which found that the level of ethical awareness in prospective developers not addressed in a codified manner. Application forms for vacant positions did not ask any specific questions to identify ethical attributes either, other than questions relating to a possible criminal record. One might argue this is a pointless line of questioning as a developer with a criminal record would not confess to it anyway in fear of jeopardising his chances of employment. The situation was compounded yet further by the hire of a recruitment manager responsible for the UK as a whole who admitted to having no experience in recruitment. He explained that although he did not know who to recruit for particular positions initially, he soon got a feel for it when managers came back and said the new recruits he had hired for them were totally unsuitable. One graduate was rejected from the recruitment process but was later serendipitously hired by a project manager who recognised some of the attributes he possessed as those he wanted in his development team. Staff reviews, therefore, would have identified skill gaps - at all levels - that could have been addressed and ultimately enhance the development process.

The amount of information given to new developers in their mandatory two-hour induction courses came from multiple internal/external sources and was found to be inconsistent in content when presented. Training was therefore poor for developers both when joining Telco initially and later, once established. The cause of poor quality training was found, at least in part, to be due to the lack of communication between the project teams and other business units, facilitated by the culture of empowerment and, at a strategic level, centralisation of company processes. Thus company-wide consistency through centralisation and localised empowerment at Telco did not work concurrently as expected. Project management tools available on the intranet were either not known about or not used. This was due to a common lethargy felt towards courses on their use. The causes for this were found to be insufficient time and budget available and, their irrelevance to day to day tasks. The nature of some project management work was considered unsuitable for classroom-based learning and course material was found to be inconsistent and/or of poor quality, as described above. Internal, yet formal, training using appropriate methods should be provided for managers, therefore, which address these weaknesses.

Project managers did not seek necessary training which resulted in extra time spent on management tasks where their skills were lacking. There was limited awareness of the project management tools available on Telco's intranet and, when their existence was known, their value was often dismissed. This situation in general appeared to be a 'catch 22' situation, with no time for training which if taken, might have reduced the time taken to carry out the tasks that prevented the managers from attending the training courses. This contributed to (a) longer development times leading to increased cost (b) reduced quality of software delivered and (c) the postponement or cancellation of developers' staff reviews, as discussed above. One developer believed his project manager had the project under control, but the project's technical consultant - who had been the project manager previously - and two other developers, were uncertain. Project managers were expected to do a better job, however, if they knew they were to be audited at some later date. Awareness, then, should be raised in conjunction with the provision of appropriate funding and time, of the training available in the use of project management tools.

5.5 Evidence Collected Relating to the Organisation

Although the Code is not directed at organisations as a whole, the culture of the company and particularly influences from the US parent had an effect on the UK project teams which did not comply with many of the principles of the Code. The evidence collected shows that Telco at an organisational level did not comply with at least 6 principles of the IEEE/ACM Code. The principles are: (1) Principle 5.05 Ensure realistic time/cost/quality constraints are used, (2) Principle 3.01 Strive for high quality in the development process, (3) Principle 3.15 Conduct maintenance professionally, (4) Principle 3.11 Create appropriate documentation, (5) Principle 8.01 Increase knowledge in software development and, (6) Principle 6.03 Participate in professional organisations. The evidence from each is discussed below.

5.5.1 Project Planning - Principle 5.05

Principle 5.05 requires the use of realistic quantitative estimates of cost, scheduling, personnel, quality and outcomes on any project which is worked or proposed, with an uncertainty assessment of those estimates.

A lack of resources was the identified by the managers of all project teams and business units as a cause of development delays. These resources were invariably defined by time and/or money. A lack of general resource planning caused managers to use some of their limited time on unrelated project tasks. There was no central store for office furniture, for example. Anything needed was ordered new for each staff member from an office supplies catalogue provided by external suppliers. Nobody was aware of what happened to furniture, etc., after somebody left or no longer required it. Resource planning, therefore, was clearly lacking - which was surprising in an organisation which advocated frugality as one of its fundamental values (even prior to the widespread cutbacks). Despite the frugality measures in place, the offices were furnished with wavy keyboards and desks, (i.e. expensive) and, mobile telephones were still invariably used when - cheaper - landline telephones were in abundance. Even the relocation of project teams from Stockley Park to Reading involved the extended hire of crates - resulting in increased hire charges - due to the lack of sufficient shelving in Reading for the contents of the crates to be placed upon. Better resource management, therefore, would not only increase time necessary for other management tasks, but also make better use of the full range of resources available.

Operations Review Meetings were held weekly and typically lasted for one and a half hours - supporting the Meeting Mania culture described above. Project managers from many development teams and other business units in the UK and US were requested to attend - via conference calls - with a covert consensus that it was unnecessary for so many (disparate) managers to attend. Even then, the conference calls frequently ended before discussions had concluded due to insufficient time booked with the telephone company. One senior project manager believed the incessant need for meetings was due to a previous criticism of poor communication. . The culture of meeting mania incurred such a large time overhead that project managers had to reject some requests for their attendance to be able to concentrate on managing their projects. One meeting arranged by the US, for example, had no apparent set objectives or agenda. There were two project meetings a week with US managers for one particular project. The project manager privately stated to the author during one conference call to the US that they could not organise a party in a brewery (paraphrased). Occasionally personal attendance was required of UK managers at meetings held in the US, with some developers unaware of when their manager was leaving the UK or due to return. After one project manager had spoken by telephone to a manager in an adjacent Telco building also at Stockley Park, he concluded by stating he had achieved more in a five minute call than he could in a day of meetings. A large number of disparate managers were typically invited to attend conference calls, although not all attendees received PowerPoint presentations and/or emails, etc., distributed beforehand. Clearly meetings are needed to facilitate communication, leading to unambiguous and complete understandings. Too many meetings though caused confusion and took time away from day-to-day management tasks. A balance needs to be found between addressing the previous criticism of poor communication and the new criticism of having so many meetings that they do not always have an agenda and waste valuable time.

One project manager proclaimed that after the redundancy program had been implemented he was able to concentrate on real project work and not have to fight through bureaucracy, red tape and layers of unnecessary managers, and back, to get any answers - aptly describing the situation found prior to the cutbacks. Senior US management had increased workloads due to redundancies in the States also and, expected the remaining UK project managers to take on greater responsibilities to relieve them of the burden. This increased autonomy was exactly what the UK based project managers had previously sought - as advocated by Telco's culture.

Public holidays invariably fell on different days in the UK and the US, often with managers in each unaware of when they occurred in the other. The US celebrates Presidential Day, for example, which the UK does not. This resulted in difficulties in scheduling meetings involving multiple continents and the consequent amendment of Gantt charts and plans. In deed, a project manager entered Good Friday in his Gantt chart as a public holiday, but forgot to include Easter Monday. This then permitted an extra working day in the software - in theory - on the project which was not actually available. On another occasion, a project meeting went ahead in the US despite it being known beforehand that no-one would be in attendance from the UK due to it being scheduled on a public holiday. It could be argued that it was better for a meeting to take at all rather than not, but without the UK managers and developers present, its value could also be questioned. Physical localities should, therefore, be identified and accommodated in all mediums of communication to enable concurrent planning - as was lacking here between multiple continents.

During the redundancy program, senior US management told the project managers in the UK that three days holiday had to be taken by every employee in every quarter with immediate effect. This strategy had the intention of saving money as Telco would pay lower taxes and, had already been executed successfully in the US. As this action was believed to be illegal in the UK, one project manager spoke about conducting this exercise on paper only, without his developers' knowledge even, just to pacify US management. Communication between departments in different countries should therefore increase to facilitate a correct understanding of the legal requirements of each. These should then be respected by senior management to prevent - in this case, UK managers - having to consider fraudulent practices to satisfy both senior management in the US and local legislature.

5.5.2 Process of Development - Principle 3.01

Principle 3.01 requires all involved in the development of new IS to strive for high quality, acceptable cost and a reasonable schedule, ensuring significant tradeoffs are clear to and accepted by the client and employer and are available for consideration by the public.

As discussed above, the technical director insisted on providing boxed (generic) solutions to customers rather than solutions that matched their individual requirements (bespoke). The supply of generic applications may have had the intention of bringing companies around to Telco's way of doing things - resulting in some customers becoming trapped into a partnership - but it was accepted, however, that this practice was not best practice due to the distance created between what Telco wanted to develop and what customers individually required. The anomalies present in new IS developed as a boxed system once delivered were left to local consultancies to correct, at additional cost to customers. The cause of this approach is discussed below and can be summarised as: (a) development and maintenance costs were reduced, (b) project management was facilitated and, (c) training requirements for developers were also reduced.

The maintenance costs associated with a bespoke approach to software development prevented many companies from commissioning bespoke IS as they were calculated over an eight year period, although the initial development costs were identified as being the largest overhead - described by one project manager as dramatic. The cost of bug fixing in the development stage of new software was known to be small in comparison to bug fixing in the maintenance stage. Known bugs were not always fixed in development if the development team determined its impact at that time to be acceptable. This was despite consensus that the cost of fixing a bug post delivery increased almost exponentially. The other (logical) alternative was to sell bespoke systems at a comparative price to generic boxed systems. This was clearly not an option to Telco as it was financially infeasible. Generic product development - the provision of boxed solutions - was expected to reduce the bug fixing burden in development and maintenance as the process was believed to be streamlined over time, leading to a continual reduction in costs as the process was repeatable.

The corollary of the 'deliver it now and get it working later' development culture was noted by one senior project manager to be that situations existed when it was not completely known what functionality was required in new IS to be developed and, once developed, what functionality the new IS actually consisted of. The level of quality in new IS shipped to Telco's customers was thus found to be analogous to the quality of Telco's laboratories housed in the new Reading offices. Their availability for use was triumphantly proclaimed by the - graduate - developer responsible for setting them up as:

'The labs are now ready! Well, almost'

Educating customers is necessary to understand the nature of software development to reduce their high expectations of a perfect new IS on delivery. Telco did not facilitate this process however, by actually raising customer's expectations in an advertisement campaign implemented in January 2001 (also highlighting the importance of use of language - as identified earlier in Chapter 2), which included the company's desire not to provide 'customer support' but 'customer delight'. Customers should be educated, therefore, to understand (a) it is not possible to test software 100% due to the inability to predict how it will be used and what future needs will be placed on it and, (b) high quality software costs time and, therefore, money. Ideally cost, therefore, should not be a factor.

Salesmen sold new IS with functionality which did not exist, resulting in the consequential extended development time-scales exceeding the expected period for revenue generation. Salesmen additionally sold new IS to customers too quickly, with little or no regard for the information required by a project manager. The sale of pseudo bespoke systems to customers was practiced so that deals could be closed quickly - thereby securing the salesman's commission. Although this was evidence of a 'providing solutions' approach being exercised by salesmen - which was strongly advocated by one senior project manager - the project manager's own working practices did not enable him to follow this through and develop what was ideally required from a customer's viewpoint - i.e. affordable and bespoke new IS, as originally offered by the salesmen. The time-scales involved in bringing what new IS did not have on delivery onto the market exceeded the normal pay-back period - typically 12 months. Moreover, development costs had to be recovered before the market for a product died - typically twelve months after the first shipment, which was an additional constraint on time available for development. Salesmen should, therefore, be paid a fixed salary with the immediate cessation of commission payments - as noted above - to enable the implementation of ethical practices by offering customers new IS which are feasible in practice: technically, financially and timely.

Additional imposed deadlines created problems for the project teams, such as an instruction received on one occasion to reduce all testing by two weeks from the Quality Department. The US-based Marketing Department usually defined a project's end date, however, stating to project managers that to make projects financially viable something must be shipped on those dates regardless of the state of the code - reflecting the attitude of one project manager above. The salaries paid to developers and project managers were included in the fixed overheads of a project, although it was also not found to be in the interests of those in Sales and Marketing to make the costs of software development to look as small as possible in order to make the IS competitive - as they themselves were on commission. As a result, one project manager exasperated:

> 'Due to limited budgets, fixed delivery dates and a team of developers who command a given salary, the quality of any new IS suffers. It is not a logical argument, to ship something under these constraints because the only flexible attribute of the new system is its quality.'

Sufficient and realistic time should therefore be agreed at the outset of a new project (without external influence) and scheduled into the Gantt chart for both development and testing, which would enable testing to be both formally conducted and documented.

Consistent application of locally designed processes was identified by one senior developer to be inappropriate at corporate level. He stated that a process or method proposed might be best for some software development, for example, but not for Telco as a whole. With an American parent, some processes were not defined formally at all at a local level in the UK, as informed by one manager who stated that he did not have any processes formalised due to the processes provided being US centric and therefore unsuitable. What was overlooked by senior US management at a strategic level, rather than at a logistical or tactical level, was that some processes - including those for development and training - were best determined at a local level. For example, the quality manager found many test companies did not have procedures to inform the developers how to do integration testing as it was a mentoring, on the job, learning activity. Consequently, formal processes and templates provided were not always used, albeit recognised that this was not best practice.

Quality managers did not attend project meetings although one quality manager was told by several project managers that they usually created project documentation only after a system had been designed. The project managers questioned the justification for creating a complete set of documentation for small projects and added that they themselves did not usually follow or create project plans. Confusion thus arose over responsibility for carrying out specific tasks. One UK based developer was asked in a conference call by a US project manager to complete a task but the team leader refused, stating it was something a project manager should undertake. Managers who were able to make better progress were found to have clearer definitions and, consequently, a technical consultant in one team meeting stated that he wanted everyone to agree on the tasks to be done before they commenced work. This became increasingly important when all the developers in one project team became involved in dealing with customers so as to reduce the technical consultant's workload.

In addition, some developers did not receive information distributed due to their names being accidentally deleted from group email addresses. One project manager did not update his email address book for several weeks after a graduate joined and continued to email his old address, which were not redirected and led to consequent inaction. One meeting was scheduled between a project manager and a developer via software called Meeting Maker. When the developer was approached personally for confirmation of his attendance, he knew nothing about the meeting as the software had not been installed on his computer. Planning, via the use of Gantt charts and Work Breakdown Structures, for example, could have prevented the above from occurring, in conjunction with timely updates of developers' contact details on the intranet and notification of new team members. Even the name of a project team was expected to change from NSSU to VTC, but was updated on the intranet to Tandem. The team was therefore referred to by multiple names which caused confusion.

A lack of communication found in HR was caused by a long period without a senior manager in charge of the EMEA group (Europe, Middle East and Asia). Furthermore, information was occasionally not passed down to project teams when managers were absent from the office – attending meetings elsewhere or on sick leave or annual holiday. A form of communication then should be implemented when the usual information chain is broken, as was the case here with managerial absence - at both middle and senior levels.

Although the US parent had a culture referred to as Meeting Mania which they consistently attempted to impose on UK project managers - with conference calls between continents commonplace - many meetings ended abruptly due to insufficient time booked with the telephone company beforehand. Conference calls were still preferred by one project manager to meetings face-to-face though, as he was believed that more could be achieved in a call than a series of meetings. One conference call enabled a US manager to realise he needed to hold weekly meetings with his own staff to maintain project control. The achievement of consensual understanding was hampered further over an Easter bank holiday weekend when a project meeting was scheduled and went ahead in the US, despite the absence of the UK staff who usually attended. The geographical makeup of the development teams was acknowledged by the technical director as problematic and not ideal as it clearly hindered any decision-making process - which invariably involved uncertainty. Conference calls were a preferable form of communication involving multiple continents, therefore, where sufficient time was booked with respective telephone companies and an agreed agenda was adhered to. Time zones, public holidays and physical localities should additionally be identified and accommodated, therefore, in all conduits of communication.

5.5.3 Maintenance Issues - Principle 3.15

Principle 3.15 requires that all forms of software maintenance be treated with the same professionalism as new development.

New generic IS developed by Telco and purchased by small companies who could not afford bespoke IS were tailored by small, local consultancies to meet their individual needs, as discussed above. Telco developers then had to maintain code which had been altered by external consultants untrained in Telco methods. Even when knowledgeable developers were believed to be available, bug reports were known on occasion to take a long time to reach the correct team for resolution.

The technical director once stated that although one project team was going international with its IS product, the opening of one door lead to a bigger closed one - due to features and network protocols, for example, being country specific. This was despite the expansion originally anticipated to be relatively easy. The director went on to blame the massive number of bugs expected - due to previous experience - on system instability, caused by sudden growth. This view was supported by one project manager who added the implementation of any new technology would have many problems associated with it. Occasionally, as discussed above, customers would report a bug in there IS which, technically, was not a bug because the code for it did not exist in the software, i.e. the functionality which failed to execute was not actually in the new IS. The process of determining complete and accurate customer requirements should, therefore, be improved, with customers informed of what is actually included when installed.

A further cause of untimely bug resolution was that resources could not be accurately identified and assigned as the maintenance effort required could not be determined in advance. For example, it could not be determined beforehand how many customers would call or how many bugs per call would be reported. The scope and severity of the bugs could not be known, or if the developers available would have the necessary skills. How many developers would be required was also an uncertainty, as was how long it would take to correct the bug(s) - if at all. The location of relevant code needed to be established, in addition to the scope of impact across other modules of code. The nature of consequent testing would also need to be ascertained and, whether it was even financially viable to correct. All these factors needed to be known in advance for a project manager to make available adequate resources - if they were available at all. One project manager stated that he had not been able to provide maintenance support for one of his products for six months after its release. Another manager not expecting to provide any maintenance support until future releases of his software had been shipped. Without impractical deadlines and constraints imposed by others on the project teams, under-estimation of how long bugs would take to fix still occurred. This was due to numerous factors and uncertainties, which should have been accommodated in a risk plan. These factors and uncertainties included:

- Time - what could happen was always underestimated
- Unreliability of equipment used
- Its inter-operability with other modules of code
- Other pieces of equipment found to be needed
- Uncertainties in the workplace
- Computers might crash

A lack of financial resources was further identified as a contributor to the untimely resolution of bugs. This factor, in conjunction with those identified above, was exacerbated further by Telco's inability to systematically collect revenues from customers who received system upgrades. Even with all the necessary information provided and the necessary resources to fix bugs reported, visits were not made to some customers' sites if they were located too far away from their local Telco office.

5.5.4 Value of Documentation - Principle 3.11

Principle 3.11 requires adequate documentation, including significant problems discovered and solutions adopted, for all development projects.

The evidence collected showed that it was not always completely know what was required from new IS, with inevitable misunderstanding. This then led to situations where it was not always known what exactly was shipped to customers. One project team's technical consultant had previously handled all communication with his team's customers himself, but there came a time when there was too much work for him to handle the communication alone. From that point on, all developers in the team also became directly involved with customers. The misunderstanding, therefore, became unavoidable as requirement gathering in this case might better be described as a game of Chinese Whispers. The increased use of documentation might have provided a solution to this situation, but a project manager believed this not to be the case at all:

> 'The problem is writing it all down, I mean, all these documents. These are 60 page documents where you try to go into as much detail as possible. But there are always situations where you can't be absolutely sure of what the customer is asking for or what you're delivering' (Senior Manager)

It was accepted that there was too much project documentation for anyone to correctly comprehend in its entirety. It was further acknowledged that when documents were read, too much detail was provided which compounded the problem further still. It was clear that the level of detail required by some was too much for others and, the detail sought to ensure that correct, unambiguous and complete requirements were obtained was too great to facilitate the development process. Increased understanding was expected to lead to greater complexity, which would then increase the burden for an even greater understanding. The requirements provided by the Marketing Department to the project teams consisted of a different view of what was required in a new IS, with regard to what was needed and how it could be achieved, as Marketing requirements were at a much higher level of abstraction than that required by the development teams. This led to misunderstandings that were not always spotted until after new IS had been shipped. The only certainty project managers had in what was delivered to customers was with regard to the mandatory features supplied, such as Caller Display. Clearly the process of determining complete and accurate customer requirements should be improved. The value of quality documentation should be given a high profile to encourage developers and project managers to take its creation and timely updates - with appropriate distribution, seriously. Documents should be structured so that only pertinent sections are included and navigation is easy for the reader. Templates should be generic and tailorable, thereby enabling the completion of only necessary sections as appropriate for the task at hand.

On occasion IS requirements had to be re-defined in the latter stages of development. This was due to customers not stating what functionality they required because invariably they did not know - supporting the work of Paul (1994). Customers were found to be so busy trying to resolve existing work-related problems that they did not have any time to think about their actual IS needs. Furthermore, even in an ideal world where requirements are complete, unambiguous and constant, extravagant requirements were perceived to be nuisance in development. On these occasions, it was likely that insufficient analysis or testing would be conducted. It is now evident that requirements are not only defined by completeness, ambiguity and consistency but, also by degree of extravagance. With customers not able to correctly and completely identify their complete needs of a new IS, the process of requirement gathering needs to be greatly improved.

5.5.5 Provision of Training - Principle 8.01

Principle 8.01 requires all developers to increase their knowledge in the analysis, specification, design, development, maintenance and testing of software and related documents, together with the management of the development process.

The training of developers and managers was found to be informally conducted on the job, or courses were selected and attended invariably as a responsive activity only. Little evidence was found of training plans, either initiated by project managers or the developers themselves, despite Telco's culture of empowerment. Financial cutbacks following the economic downturn in the US resulted in severe restrictions being imposed on all spending, including training budgets. When presentations were arranged and given to train developers in the area of software quality, however, all developers were invited but minimal attendance was achieved. One project manager saw little value in the use of quality standards, except when he could use them to his own advantage, for example, to prevent the coerced use of a generic test plan by senior management in the US. This view was commonly held by the developers, which explains, at least in part, the poor attendance described above. Attendance was usually optional and, on one occasion, was non-existent. Training was identified as problematic not only in industry but, also in the alignment of material required and taught between industry and academia, respectively. This misalignment led, in part at least, to one project manager stating that he expected newly graduated developers to take up to twelve months before they became satisfactorily productive. Two developers - described as being the best two - from an outsourcing company arranged to visit the west London office to train the developers, but they admitted they did not know all the code - only the sections they had previously worked on. Relevant and timely training to developers and project managers should, therefore, be identified and included in project schedules and delivered via appropriate or preferred training mediums.

Developers and project managers did not attend training courses, in part, because they did not particularly appeal, as discussed earlier. They acknowledged there were many things they needed to learn and should have enrolled on the appropriate courses. But courses were not considered to be compelling - just something to do at some point in the future. Furthermore, the workload generally did not permit time off for any kind of training and, developers found training to be best obtained through learning on the job as against by attending taught courses. Developers were aware that additional training would be beneficial and, that not finding the time required to select and attend courses to meet individual needs may actually be partly their own fault. Improved time management for developers should be identified in staff reviews, therefore, to facilitate the identification of available time for appropriate training. A six month course in Switzerland that was compulsory for all new developers was considered by several HR staff, including the HR manager, to be so intensive that they were astonished developers did not return with nervous breakdowns.

The nature of software development was found to require the availability of timely online information as against untimely printed documentation - when provided. Project managers and the quality manager were in agreement about developers in general also preferring not to read documents - of any size. Training courses were believed to teach something that would only be remembered for a few days, as unless what was taught was used on a daily basis it would be forgotten quite quickly. Feedback from trainers was found to be either non-existent or untimely to queries about particular courses. Moreover, the period of heightened frugality meant courses were no longer affordable - where budgets existed at all for training purposes. A consequence of this lack of appeal/unsuitability/non-provision of training was that in a meeting of several project teams, only one developer present had been formally trained in testing and, that was with a previous employer (i.e. and not in alignment with Telco's GEM model). Courses should, therefore, be provided and pertinent to individual needs - in consultation with respective managers to ensure appropriate consistency. Timely electronic information should be made available, with hard copies structured to facilitate readability and navigation - in conjunction with timely feedback from trainers. The funding of courses should be addressed to enable the benefits from attending to be realised as and when needed and, not influenced by external factors.

Courses that were taught were considered time consuming and an alternative way to learn, such as reading, was preferred. For example, a presentation given by a quality manager lasted for one and a half hours and contained eighty five slides - with insufficient time available to show the last twenty. Software used in training sessions was itself described as buggy by one trainer, with requests for a new, bug free, release denied. Presentations were occasionally provided by a quality manager to train and educate developers and their managers - but with attendance options, few attended. The cause of this was, again, due to the limited time available of developers and project managers and courses general lack of appeal.

In one presentation given by a manager, there were no handouts available for those present to make notes on in the session, take away for perusal or, to pass on to those who were unable to attend. Presentations by the quality manager were given to train/educate developers and managers, usually with an open invitation, but few attended. On one occasion no-one attended and the course was cancelled. Training via presentations should therefore be of a reasonable length with adequate time made available for delivery of slides and questions. Handouts or manuals should be provided, at least for those requiring training even if they are unable to attend the session. Reading was preferred by some developers, but correct understanding cannot be assured through this medium without a test, for example, on completion. Learning on the job, as also preferred by some developers, again cannot ensure correct uniform understanding, uses no pre-approved documentation and takes longer due to its unstructured nature.

Material read in a developer's own time, however, was found to be more up to date than material studied on a course. Moreover, learning on the job was considered to be more beneficial than reading, due to each area read being generic and not wholly specific to what was needed. The generic - and voluminous - material found in books additionally made the material time-consuming to read, with one HR employee noting that he preferred to learn by actually not reading. This preference was found to be commonplace, with one developer noting that he preferred to learn by experimenting, as discussed above. Courses that had been provided on software development methodologies prior to the cutbacks were found by developers to be difficult with regard to transferring the simple examples provided to their day-to-day project work. The examples always matched with simple scenarios, but not when held up against multiple independent systems all working together, with problems spreading through the network and with no knowledge of how they could scale up - as was the reality. The quality of written text provided on courses was also found to be poor. Course material needs to be role specific, therefore, for the benefits of - increased - attendance to be fully realised, in addition to being timely and realistic. Simple examples should be used to introduce the material, but increased difficulty needs to be included for the course to be of practical use in developers' day-to-day roles.

The suitability of external training vendors was discussed by several managers and also found to be problematic with regard to consistency and quality. Inconsistency was identified as a problem when the Training Department based in the US attempted to centralise all training activities company-wide. It was then discovered that one training vendor in particular was being used by one business unit when another business unit had hired them previously and decided never to use them again:

> 'I was very disturbed to find they were still being used as I had long ago disqualified this company as someone with whom I would ever do business. I was highly annoyed by their supposed 'star instructor' and decided he would never teach here again' (Senior Manager)

The rehire of an unsuitable vendor was due to previous de-centralisation of training activities to promote empowerment and, the experience had not been shared with other business units. This is one instance where Telco's culture of empowerment and autonomy to both individual departments and managers hampered a more recent quest for quality and consistency across the company. A question was asked by the head of UK recruitment concerning the consistency and quality of material across internal courses. The answer was provided by a project manager when he exclaimed he was personally tired of always being different from the rest of Telco - he even enquired as to whether every development department in Telco had its own processes as he could not identify any consistency himself. As found above, consistency and empowerment did not work concurrently and each needs to be redefined to ensure uniform quality training is available to all managers and developers.

5.5.6 Approach to Professional Standards - Principle 6.03

Principle 6.03 requires developers to extend their knowledge in software development by appropriate participation in professional organisations, meetings and publications. Without the backing of management at all levels to establish and uphold the value of professional organisations and the standards they create to facilitate consistency in the IS profession, developers will be less likely to become members of those organisations.

One technical consultant believed if a company normally produces 'crap,' certification of any quality award only meant they are able to produce 'quality crap'. This view was generally shared by project managers. For any standard to be perceived as valuable and therefore, likely to achieve compliance to its requirements, they should be country specific where differences are identified. One project manager saw little value in the use of professional standards to aid project management practices, except when he could use them to his own advantage, for example, to prevent the coerced use of a generic test plan, as discussed above. Only a quality manager appeared to see value in the implementation of standards which were expected to lead to process improvement:

> 'The value of implemented standards results in a positive boost in attitude in the team, along with better team communication and increased customer satisfaction. Product quality is improved as the development process is defined and repeatable. Project managers should also be more conscientious of documentation for the

project - the discipline used for project management would be improved. Managers will be aware their practices will be reviewed in an audit and therefore more aware of meeting minutes, documentation and tracking project progress. Software developers who adhere to a standard will positively affect those who do not. Implemented standards directly influence customers' buying decisions, with the public interest supported through reliable internet connectivity' (Senior Manager)

5.6 Summary

The chapter discussed the causes of unethical practices identified in the literature and at Telco during a six month case study, which were aligned with the issues drawn out from Chapter 2.

The non-compliance of all the units of analysis to the principles of the IEEE/ACM Code was found to be due to many, and often intertwined, reasons. Difficulties in understanding both the requirements of new IS and the development process itself was common, in conjunction with customers who often did not know what they wanted exactly from a new IS. The creation and use of documentation was found to be poor, hampered by generic and inflexible templates. Communication at all levels was found to be difficult, due in part to the geographical spread of project teams. Conference calls facilitated the distribution of information but, consideration is needed for time differences between the continents and the existence of public holidays. Customers should be educated to understand resources are difficult to identify and quantify, in conjunction with the inability to test software 100%. Estimation of the time expected to complete known tasks was found to be more realistic when provided by the developers. Relevant and timely training in the estimation tools available on the intranet for project managers should be promoted to facilitate the process.

High quality ISD was prevented in part by the use of documentation templates which were generic and static in nature. As described above, developers whose mother tongue was not English were not consulted. Training budgets were not always available for training, with approved training frequently cancelled at short notice. Gantt charts were created with infeasible deadlines created by salesmen on commission. The contents of Telco's own development model - GEM - was largely unknown and, therefore, was invariably unreferenced. Formal models should be tailorable, with necessary compliance by developers and managers given a high profile. The few reasons given for not joining professional organisations were commonly held. Insufficient time available also prevented attendance as working into the evening was common. Senior management was found not to value membership of professional organisations. Developers were not able to conduct maintenance professionally due to factors occurring both prior and post delivery of new IS. A high number of bugs reported were found to be due to the short development time scales, reduced testing and invariably no time allocated for training. Maintenance activities were also hampered by the lack of information provided to the developers.

Project managers found it difficult to further educate their developers with appropriate training due to infrequent staff reviews, the nature of training provided and a general unawareness of standards. The recruitment of developers with appropriate skills per se was made difficult, in part, by the misalignment found between modules taught at university and Telco's specific needs. Staff reviews could have identified and addressed the misalignment, but they were conducted with great infrequency. Telco preferred to develop its own interviewing procedures which clearly contributed to the difficulties in recruiting quality developers. Learning on the job was preferred by many developers, with examples provided in training found to be simplistic and unhelpful. Moreover, the technical director acknowledged it was difficult for him to correctly define the processes needed for the large number of staff under his remit. Project managers were found to be unable to plan projects and estimate tasks effectively due to the nature and use of Gantt charts and the game playing of budgets. Gantt charts were used as part of the planning process but, with imposed deadlines their relative value was found to be limited. The culture of meeting mania took up much valuable management time which needs to be controlled. These meetings also need to be organised so that they accommodate time differences between the continents involved, enabling better attendance.

With the causes and reasons for non-compliance by the software developers and project managers to the Code established, the next chapter evaluates the IEEE/ACM Code itself against the strengths and weaknesses identified in the literature. Recommendations and their respective implications can then be identified.

Chapter 6. Critical Evaluation of the IEEE/ACM Code

6.1 Introduction

The literature search identified strengths and weaknesses of existing codes in IS and other professions, which were presented in chapters 1 and 2 respectively. This included research into how the other professions attempt to enforce the requirements of their own codes. The IEEE/ACM Code of Ethics is now evaluated to identify its own strengths and weaknesses in practical terms and, how enforcement of its requirements can be better implemented.

Section 6.2 evaluates the Code in practice and, compares its characteristics identified with those of codes found in the literature. The characteristics were categorised as strengths and weaknesses and, were described in Chapter 2. Resistance to the requirements of the Code is also discussed. Section 6.3 investigates how enforcement of the requirements of ethical codes can be enhanced.

6.2 The Code in Practice

As membership to the IEEE and ACM is voluntary and, at Telco membership to these organisations was found to be non-existent, the Code *per se* had no practical benefit in the context of managed software development or the relative areas of the other units of analysis participating in the research. Knowledge of the eight principles and their respective requirements was not found, therefore, and any correlation between the ethical practices found and the requirements of the Code was purely coincidental. Resistance to the implementation of codes and frameworks in the literature was also evaluated to identify whether the same reservations exist at Telco.

6.2.1 Practical Strengths & Weaknesses

The literature search in Chapter 2 identified strengths and weaknesses of codes and frameworks in general. The Code was evaluated to determine their presence, or, in the case of weaknesses, whether they had been addressed by the authors of the Code and were non-existent. Resistance found to codes and frameworks by developers was additionally identified in the literature and, whether the developers and project managers in the case study had the same, or other, reservations was identified and described.

The strengths of codes in general identified in the literature and found to be present in the Code were:

- The advocation of truth, co-operation, communication, self-control, goodwill to others, self reflection and charity
- Encouragement for continuous education
- The identification of stakeholders to be managers, clients, suppliers, communities, other professionals, other employees and society as a whole
- The consideration of knowledge in context and significance
- The identification of value categories as personal, organisational and professional
- A distinction between ethical reasoning and attitudes

More importantly, for the focus of this research, the weaknesses of codes identified in the literature and found to be present in the Code are provided below. The Code was found to address the weakness of being designed solely by outsiders or managers by the consultation period exercised by its authors in the design process.

- Ambiguity, with subsequent clarification compounding the problem further
- An arbitrary increase in workloads with regards to preparation/study for examinations (in the context of HR)
- Unclear/conflicting working practices, with private study being conducted during office hours resulting in the need to work until midnight
- Discipline, deterrence and prevention of cowboy presence were all lacking in the Code, with only an identification of inconsistency with being a professional developer being noted

Other weaknesses found in the Code are that it is long, consisting of eight principles each containing approximately ten sub areas, making a total of approximately eighty areas for a developer to bear in mind when working. In addition, the Code does not deal explicitly with the trauma that can be experienced by developers and managers when faced with change, as with the redundancy program implemented at Telco. Principle 4 only requires that developers give a fair hearing the concerns of other developers. Counseling was only provided by the organisation to those developers who were directly affected by the process and, only after their employment with Telco had ceased. The

Code needs to be reduced to become 'mentally' manageable, therefore, yet longer to better accommodate the needs of personal change which it does not in its present state. As with documentation discussed above, information provided will be perceived as excessive by some and insufficient to others. Furthermore, continuous education for developers is required in Principle 8 but is not stipulated as a requirement for project managers explicitly in Principle 5. This is necessary as evidence was collected of project managers unable to use software tools competently and, who were unaware of the existence of management tools available on the intranet for their use. Continuous education for project managers is required to be included, therefore, with skill gaps identified through periodic staff reviews with their respective directors.

Poor communication was found to be a frequent contributory factor of unethical practices identified and, therefore, non-compliance to the eight principles, but how poor communication should be addressed by practitioners is missing from the preamble provided at the outset of the Code. As with corporate policy at Telco, strategic goals of the Code are not implemented successfully at a tactical level. The generic application of the Code *per se* to all members of the IEEE/ACM organisations require its principles to be described at an abstract level to be appropriate to as many practitioners as possible, resulting in the ambiguity found in the principles as a consequence. The Code is produced in two forms - a short version and full length version - with a preamble supplied - with the latter consisting of more detail. The provision of this additional detail attempts to address the criticism of the Code of being too abstract to be implemented correctly at a local level with regard to the unclear working practices, but then the criticism arises of the Code being too prescriptive to be applicable - with the desired value - across the profession as a whole. The consideration of context and significance prior to application of the principles is addressed by the preamble provided at the outset of the Code which states that it cannot provide guidance relating to every practice of software development where ethical issues arise. It requires software developers and project managers to consider the context of a situation where such issues arise and, to make decisions based on the situation as a whole. The preamble clearly states that questions that arise with ethical issues attached can best be answered by thoughtful consideration to fundamental principles, rather than reliance on detailed regulations. The evidence collected, however, shows the practice of careful ethical consideration generally was not exercised.

The Code aims to enhance the development process, therefore, but with limited effect due to the minimal/non-existent membership of developers and managers - and other factors outside of their control. The professional organisations need to address how they are seen by practitioners, therefore, and review the real benefits that can be offered to encourage an increase in its membership. This would clearly lead to awareness of the Code and, subsequently, compliance to its requirements. Other departments with which project teams interact also need to comply with their own ethical codes of practice for the IEEE/ACM Code to be truly effective at a local level. HR has its own professional organisation - the CIPD - which created a code of ethics for its members. The HR employees participating in the research believed they were naturally ethical in the conduct of their working practices and, consequently, did not need to read the requirements of the CIPD code. A need for increased awareness of the code and its actual requirements is necessary for the members to become consistently and uniformly compliant. An ethical code is still needed for those employed in Sales, Quality and, for senior management. An ethical code is additionally required for customers who, in these competitive times, need to be educated to understand that quality IS takes time to develop and, therefore, costs money. Such a code would also educate customers to understand what is actually feasible - regarding technicality, quality, cost and time. This would prevent them from requesting impossible IS: highest quality, maximum functionality, minimum time and minimum cost.

Fair competition is still achievable with a consensus across the industry that ethical practices will be implemented by all those employed in the industry - requiring widespread agreement amoung all ISD professionals. Companies developing new IS should then be able to complete questionnaires to identify how ethical they actually are, enabling customers to make informed choices regarding the selection of which to commission to develop their new IS. A scheme similar to the gold stars awarded to hotels might be appropriate.

The scope of the Code should be increased to encapsulate organisations as a whole, made possible through company memberships, with the introduction of separate and relative requirements/principles for other departments, such as Sales, Recruitment, HR and Quality. When creating or amending organisational principles, guidelines from professional organisations should be considered carefully to facilitate consistency across the industry, but the ability to tailor the principles of the codes to reflect local needs should be exercised. These should be created in alignment with existing professional codes where available - such as the CIPD. Moreover, codes should be created in multiple languages to accommodate the diversity found in the workforce of large international IS companies conducting ISD.

Telco attempted to address the weaknesses of the IEEE/ACM Code by introducing its own in-house methodology for software development called GEM - the Great Engineering Model. GEM was expected to set a standard to which all software development created by Telco's developers and project managers would adhere to. Information about GEM was available in paper format and on the company's intranet, with training in its requirements available to all new developers via a six month training boot camp. In practice, developers were found to develop software as they chose on an individual level, due to the inappropriateness of the boot camp training. The course, in conjunction with those typically offered to veteran developers, was seen to be unsuitable for teaching software development due to the very nature of ISD. Learning at developer's desks or from reading was perceived to be more beneficial than attending classrooms where there was invariably insufficient time available to learn the module being taught in its entirety, with handouts not issued or were not representative of the actual work done by the developers. Senior management was in consensus in expecting developers to return from the boot camp with nervous breakdowns due to its intensity. To compound the problem further, developers were additionally known to be disinterested in reading project documentation, regardless of length. Many were found to prefer to learn by experimenting. A uniform understanding and consistency in application of the GEM model was therefore found to be minimal amoung the project teams.

6.2.2 Theoretical Strengths & Weaknesses

Ethical codes were found to have many common theoretical strengths and weaknesses, as identified earlier in Chapter 2. These are reiterated below, together with the evidence to confirm or refute their attribution to the IEEE/ACM Code. The theoretical strengths and weaknesses identified in the literature and found not to be present in the Code are also described.

The theoretical strengths identified and found in the Code were the advocation of: truth, co-operation, communication, self-control, goodwill to others, self-reflection and charity. The advocation of truth is found in Principle 1.06 which requires software developers to be fair and avoid deception. Co-operation is sought by Principle 1.05 which requires developers to co-operate in efforts to address matters of concern caused by software. The exceptions permitted, indicated by 'as appropriate' enable the truth to be circumvented when - as dictated by the *raison d'être* of the Code - the public good would otherwise be comprised. Principle 7.05 requires the fair hearing of opinions, concerns and complaints of others, providing evidence of the support for communication. Goodwill to others is supported by Principles 7.01 and 7.02 which requires developers to encourage their colleagues to adhere to the Code and assist them in professional development, respectively. Principle 1.08 advocates charity work with the requirement for developers to volunteer professional skills to good causes and to contribute to public education in the area of IS.

Stakeholders in the development of new IS were found in the literature to be: managers, clients, suppliers, communities, other professionals, other employees and society as a whole. The Code identifies the stakeholders as being software developers (Principles 1 to 8) and project managers (Principle 5 'Management') specifically. Principle 1 'Public' further identifies communities and society. Clients are recognised in Principle 2 'Client and Employer' with other professionals recognised in Principle 7 'Colleagues'. The only stakeholder not to be explicitly recognised by the Code is suppliers, although Principle 3.03 'Product' requires developers to address economic, legal and environmental issues, which implicitly incorporates suppliers. This is due to the initial selection process of a supplier consisting of, in part at least, an analysis of cost for services offered and the identification of projected value for money, with selection culminating with signed legal contracts for all parties concerned. The spirit of the Code, however, encompasses all parties with which a developer interacts - and expects him to conduct himself professionally with all those involved or related to his work.

Value categories were identified in the literature and presented in Chapter 2. The categories were personal, professional and organisational. Personal values held by software developers are identified and addressed by Principle 8 of the Code, entitled 'Self'. This principle describes the requirement for a continuous endeavour by developers to increase their knowledge in the area of IS development and maintenance, in addition to treating others fairly and, increasing their own understanding of the Code and pertinent legislation. Principle 6 'Profession' clearly seeks the advance of the integrity and reputation of the IS profession - with thirteen areas identified for developers and project managers to adhere to (consisting of more areas than all but one of the other principles in the Code). Principle 6 additionally seeks the development of ethical awareness in the developers' organisations, in conjunction with the requirements of Principle 2 'Client and Employer'. This latter principle further identifies a developer's clients as organisations, who should thereby be treated by a developer with the same ethical concerns as his own employer.

Ethical reasoning and attitudes are, at least implicitly, identified and addressed throughout all the principles the Code. For example, developers are required to moderate the interests of all those concerned with software projects in Principle 1.02. Principle 4.01 requires developers to consider those human values which need to be supported and maintained when making technical judgments, with project managers required to ensure developers are aware of standards before being held to them in Principle 5.02. These principles support the findings of Dozier *et al* (1996) and Johnson and Smith (1999) who identified the distinction between reasoning and attitude. Reactions to a process or outcome differ according to where the focus lies - on either or both stages. A project manager may focus on a process being ethical whereas a software developer may focus on the outcome, with each party possibly unaware of the other's (different) focus. This is important as an unfavourable outcome is more welcome when the process leading up to it is considered to be fair by the recipient of the outcome (Ambrose *et al* 1997). It is additionally possible for a number of forms of ethical reasoning to conclude with the same ethical attitude. This is achieved when there is input from both the project manager and his team of software developers in a decision-making process.

Ford *et al* (1996) found cultural factors to be considered when designing a new ethical code or framework, which were presented in Chapter 2. They are reiterated here to aid the reader and include: level of education, law, concept of time, social organisation, values and attitudes. The authors above believe the needs, wants and motivations of each should be determined and, their inclusion in the Code is now identified. A developer's level of education is addressed by the Code in Principle 5 where project managers are required to evaluate the education and experience of software developers who apply for a position within his team against the needs of the project. Developers are required in Principle 6 to obey all laws, albeit acceptable by the Code for developers and project managers to break the law in exceptional circumstances, such as those which are considered to be inconsistent with the upholding the public interest. Principle 3 requires the identification, defining and addressing of cultural and environmental issues, which includes the consideration of time differences. The consideration of the social organisation is addressed in Principle 2, which requires developers to identify and address issues of social concern present within the company. Principle 2 further requires the equipment belonging to an employer to be used only in ways properly authorised and with the employer's knowledge and consent. Values are addressed by Principle 4 which requires the support and maintenance of human values. Professional objectivity is additionally sought by Principle 4, addressing the requirement of a code to consider attitudes.

As identified in the literature, the richness of the English language is also its downfall due to the level of ambiguity it creates (Oliver 1998). The Code does not exclude ambiguity from its principles, although its authors did actively attempt to address the problem - as did the Law Society when compiling The Guide. For example, Principle 1.06 requires developers to be fair and avoid deception in all statements, which replaced the original requirement to be honest. Other areas addressed to reduce or even remove the ambiguity present actually compounded the problem yet further. Principle 3 'Product' requires developers to ensure their products meet the highest professional standards possible. This statement is supposedly clarified with, for example, a need to strive for high quality and a reasonable schedule. What high quality actually is, how it is measured, achieved and maintained - with consistency, accuracy and completeness - by all pertinent parties involved in a project, is left to the reader of the Code to determine. The problem of ambiguity is unavoidable due to the general nature of codes which are unable to be specific for individual contexts, supporting the findings of Walsham (1993) and Oliver (1998). The ambiguity present in the Code enabled the unethical practices for some principles to be identified through inference. For example, giving a fair hearing to concerns or opinions of a colleague as required by Principle 7 implies an objective thought process when decision making, as required by Principle 4. Some evidence collected, therefore, related to more than one principle, although the respective data itself may provide evidence for only one incident. Repetition in the requirements of the Code should thus be removed and, ambiguities in the principle descriptions provided should be clarified.

The use of alien language was found to be a weakness, as much evidence was collected in the case study showing problems with communication due to the English language not being the mother tongue of many of the developers. These developers were predominantly of Indian, Italian, Chinese and French origin - although predominantly based in the US. They might then describe the English language as alien, as it caused the deliberate lack of communication in project meetings which took place with developers present from multiple continents, as discussed above. As the UK based developers also did not have the mother tongue(s) used by these foreign nationals identified above, the UK based developers might additionally describe their (US based developers) languages as alien. An attempt to reduce, or remedy, the situation therefore, is to produce the Code - and project documentation - in all the mother tongue languages of the project managers and developers involved in a project. The instructions to project managers and software developers contained within the principles of the Code are intuitive to understand - to a native speaker of the English language. The generic nature of the Code necessitates the eight principles to consist of non-specific requirements - supposedly enabling the context of a situation to determine the timely relevance and degree of application for each.

As membership of the IEEE/ACM was found to be non-existent among the project teams participating in the research, the Code did not arbitrarily increase any workload. For the Recruitment and Human Resources departments, however, membership of their professional body - the Chartered Institute for Personnel Development (CIPD) - resulted in a significant amount of personal time needed to study for the CIPD examinations. This workload was found to be both significant and common for members of the CIPD, with membership found to be high in these two departments. Both membership and the successful completion of exams were considered a necessary prerequisite for professional progress as they were explicitly sought by Telco and employers generally when recruiting for positions in these areas.

The IEEE/ACM Code did not introduce any unclear or conflicting new working practices in ISD due to the general unawareness of its existence. The CIPD for the Recruitment and Human Resources departments introduced conflicting practices by requiring extensive and therefore time consuming study for its examinations, which was taken, in part, during work time. This situation was worsened during RiF (Reduction in Force) when the redundancy program was being implemented and employees in HR were found to work up to eighty hours a week, with some staff regularly working until midnight.

Neither the IEEE/ACM Code nor the CIPD were found to threaten jobs at Telco. Although 8,500 employees in the company were made redundant globally, this was not the consequence of any professional code being implemented. The redundancy program was the (arguable) result of the downturn in the US economy, followed by a downturn in the UK economy and other countries worldwide.

The literature additionally found although the cultural factors of family, class and race have an impact early on in IS professionals' lives, competent and professional project managers are what is ultimately needed (Dozier *et al* 1996). Such managers would also be ethical, according to the defined principles of the IEEE/ACM Code, but the complex social, financial and political environment within which a project manager has been shown to have to function overpower his ability to conduct project management in a completely ethical manner.

6.2.2 Resistance to the Code

The literature review found several reasons for resistance to frameworks and codes in general. These are reiterated below, along with confirmation, or otherwise, of their perceived existence at Telco.

Membership of the IEEE or ACM was not perceived as appealing by developers or project managers and, was not expected to have an affect on their job status either, positively or otherwise. Job status was found to be protected and even enhanced via membership of the CIPD and the sitting of its examinations, however, as described above. This existed in the Recruitment and HR departments, with membership to the CIPD high in both, caused, at least in part, by a perception that membership was compulsory for those seeking career progression in these fields.

An arbitrary increase in workloads was identified in the literature as a cause of resistance to codes. This was experienced by both the Recruitment and Human Resource departments which were attributed to the CIPD, with regard to the amount of study required to pass the - numerous - CIPD examinations. No increase was identified in the day-to-day work tasks of its members as a consequence of meeting the requirements of its own code. The passing of exams in order to progress in one's career was viewed so strongly that the substantial amount of time required for study was generally accepted - confirmed further still by the need to study in both personal time and during working hours. In a separate context, during the plans for ISO 9000 to be implemented, the value of the implementation was found to be questionable. One project manager actually only found value in documentation for an existing code by using it to prevent an increase in his workload, stipulating that additional work being requested by a senior manager was not a requirement in the documentation and, therefore, he was not obliged to accept the request. Most developers and project managers expected an increase in paperwork and possibly some changes to the way they currently developed and maintained software. No-one knew exactly knew what would be required, supporting the known weakness of codes described above. The resistance was exacerbated amongst management who consensually and explicitly expressed the opinion of a lack of tangible benefits existed in the implementation of ISO 9000. As one project manager noted,

'If you produced crap before, after ISO 9000 you will produce quality crap' (Senior Manager)

The literature found the need to work with new people was justification for resisting the implementation of a new code. Neither the software developers and project managers or the other business units taking part in the case study, however, identified the need to work with new people as problematic.

6.3 Enforcement of Principles

Ethical practices have been defined for the purposes of this research by the eight principles of the IEEE/ACM Code and, research was conducted via a case study to identify the unethical practices being conducted at the individual, team, management and, organisational levels. Other professions were also examined to establish how they define and enforce ethical practices in their respective workplaces, i.e. UK Law, Retail, Finance and, the wider context of Law in the European Union. It is now possible to reflect on the enforcement of ethics within each of these units of analysis, with the objective of achieving greater compliance to the principles of the IEEE/ACM Code of Ethics and, thus reducing or omitting the problems in IS described above.

6.3.1 Achieving Individual Compliance

Ethical frameworks and codes were found in the literature to be best at institutionalising ethics due to the level of detail they provided. The IEEE and ACM jointly created a code of ethics to institutionalise ethical IS practices at a local level for software developers, but with limited effect. This was due to (a) membership of professional organisations found to be minimal or non-existent, (b) where membership existed, awareness of codes in place was scarce and, in any case, (c) compliance to its requirements was optional. These findings would, therefore, suggest that institutionalisation of ethics through frameworks is not the best method, unless these restrictive factors can be addressed. Membership of organisations can be increased with the provision of tangible benefits for its members, in conjunction with timely relevance to the day to day responsibilities of developers and project managers.

Discipline of developers who commit unethical practices is addressed by principles 6.11 'Profession' and 8.09 'Self'. Both stipulate any action taken by a developer which violates the requirements of the Code is inconsistent with being a professional software developer. There is no disciplinary procedure provided by the Code, only the identification of the unethical practice as just described. The Code was found, therefore, not to be a deterrent against unethical practices being implemented. This is reflected in amount of evidence collected of unethical practices at Telco and presented in Chapter 4, thus supporting the findings of Gotterbarn (1999a).

Compliance to the eight principles is only requested by the Code and is not mandatory. With membership to professional organisations found to be optional and an increased membership desired by such organisations, the threat of tangible disciplinary action would not be perceived by potential new members as a benefit of joining. Thereby a code should be used as a benchmark for determining what is and what is not ethical practice expected of developers, but total adherence to the Code would fail its overall objectives of upholding the public interest. This was shown to be evident above with the identification of allowable unethical practices in the case study - considered necessary for the overall success of some projects.

An acceptance of additional responsibility is required of solicitors when working with a colleague whose mother tongue is not English. If additional time and patience cannot be assured from the outset of such a relationship, then the solicitor must not accept the work. The evidence collected in the case study found software developers working with other developers whose mother tongue was not English actually made less time available to interact with them. Deliberate acts of reduced communication were practiced by some developers to save time providing explanations to those developers with difficulty understanding what was being discussed. In deed, some issues were not raised at all so that meetings would not last longer than absolutely necessary. It is clear that in these instances the meeting chair should have ensured all pertinent issues were raised, discussed and concluded to the satisfaction of everyone present. Omission of issues requiring clarification or consensual agreement could be viewed as negligent on behalf of those developers deliberately not communicating these issues. Professionalism was clearly lacking in these instances on the part of both the developers and, the meeting chair.

The legal profession is governed by the Law Society which publishes the behaviour expected of solicitors and requires membership to the society to be renewed annually. Membership renewals require signatures from two other solicitors who are not connected to the applicant - professionally or personally - to facilitate the membership of reputable persons only. The IEEE/ACM do not require any additional signatures for annual membership to be renewed. The desired working practices of solicitors are enforced by legislation, following definition by the Law Society. Although there is very little in the IEEE/ACM code regarding inter and intra team communication and working practices, the EU Code of Practice for Solicitors - through legislation - additionally requires a spirit of trust and co-operation from its members.

Legislation has recently been introduced into the field of IS, with, for example, the Computer Misuse Act and Data Protection Act. Other acts of parliament are in force which are pertinent to IS, but are still very few in comparison. Clearly IS professionals cannot be left to exercise their choice of behaviour in IS practice as the evidence collected in the case study clearly shows unethical practice to be in abundance. Annual membership to an IS body in conjunction with compliance to (existing and new) legislation or, accept possible expulsion and the loss of right to practice are necessary for the IEEE/ACM Code to achieve greater success in its objectives. The Code currently does not have a 'champion' in parliament to ensure legislation is drafted and passed onto the statute books. The current de facto requirements are unlikely to progress to anything further until this situation changes. Legislation can be used to punish solicitors who do not implement the required ethical practices, but the use of formal disciplinary action is viewed by the Law Society as a last resort. An informal channel of communication is preferred initially, in part to protect the public image of the legal profession. Complaints, regulation and discipline of solicitors are controlled by several non-statutory bodies, including the Law Society, Compliance and Supervision Committee, Solicitors Disciplinary Tribunal, Common law and Master of the Rolls. This tight regulation for administering discipline among the membership is missing in the IS profession, where membership of professional bodies is both optional and subsequently minimal, with discipline scarce so as not to risk a reduction in membership to an even smaller number.

Solicitors cannot practice unless they have been admitted to the Law Society. Admittance requires the successful completion of appropriate examinations, the ownership of a practicing certificate and, are registered on the Roll of Solicitors. Developers have no such obligatory requirements to the IEEE/ACM bodies or their code of conduct. If the ability to practice for developers required mandatory qualifications and annual registration as exists for solicitors, the notoriety of the IS profession should be reduced, if not removed altogether. These mandatory requirements of the Law Society are accepted and enforced through legislation. In deed, anyone practising as a solicitor without the appropriate qualifications can be imprisoned for up to two years. The threat of imprisonment for unqualified IS professionals would undoubtedly facilitate a reduction in the presence of 'cowboys' in the industry and consequently, the reduction or removal of notoriety from the profession. Furthermore, qualified developers should not accept work where they consider themselves not to be competent. They should refer customers to other developers who they feel is - or would know someone who is - competent in that area. In this vein, delegation of development should only be permitted to suitably qualified developers.

In the financial sector, professional behaviour at a personal level is both defined and required by principles created by the FSA. These principles require professionals employed in the financial sector to be open, honest, responsive and accountable. Colleagues and customers must be respected and work should be conducted with a commitment to competence, responsibility and reliability. Unlike the IEEE/ACM organisations which only state the practices required, the FSA created a series of questions relating to the values sought to enable the spirit of their principles to be perceived. Furthermore, the FSA created a series of hypothetical situations which require thought for decision making, with the desired values described above providing a basis for discussion. The process is expected to conclude with solutions provided by the members that would be approved by the FSA.

6.3.2 Achieving Team Compliance

It has been implicitly assumed so far that the environment in which software is developed is ethical and facilitates the development process. The data from the case study shows this not to be the case, however, and that the environment surrounding the development of new IS also exhibits unethical practices just as the development process itself was found to do. The case identified four areas of the environment in which developers have to interact and, which were themselves found to be unethical in their own respective practices. These other departments were Sales, HR, Recruitment, Quality and, Telco at an organisational level. Unless these other departments also comply to ethical codes reflecting their respective working practices, a requirement of mandatory compliance from developers would be analogous to sending sheep to graze among wolves. HR does have a code of ethics in place provided by the CIPD, however, albeit not actively promoted in communication with members or seen as necessary by them. Furthermore, the existence of unethical tendencies should be identified in the recruitment process of new developers, identifying the need for recruiters to become ethically aware - in addition to developers and other IS professionals.

The workforce at M & S complied with the requirements of head office as a result of being listened to by senior management in conjunction with two way communication on a regular and transparent basis. Secondly, training was provided and appropriate and, recognition and financial rewards relating to individual/team/corporate performance were given. Career progression was discussed annually with respective line managers to facilitate job satisfaction. Good time management further enabled time to be available for meetings and two way communication via, for example, periodic in-house magazines. The staff felt valued and involved as a result. Many staff are also shareholders of the company and so have a vested interest in company productivity and subsequent profit.

6.3.3 Achieving Management Compliance

Principle 5 of the IEEE/ACM Code is for project managers and team leaders, but the title of the Code specifically targets only software developers. The awareness of the principle by team leaders and managers, therefore, is perhaps just serendipitous.

Senior management in the US had a large affect on UK managers, for example, by setting infeasible deadlines for project deliverables. The level of power able to be exercised by US management dictated what practices were, and were not, to be implemented by the project teams - and not from any ethical basis. Competitive forces in conjunction with business economics influenced the decision making of senior US management. This identifies a need for an industry-wide code of ethics - or series of codes - to be adopted and enforced in a practical and acceptable manner. The myopic viewpoint of US management seeking to maximise corporate profit at each and every quarter without regard for long term goals or relationships with either employees or customers, lead to infeasible deadlines being imposed with a culture of meeting mania to monitor and ensure the deadlines were achieved. The project managers bought into the deadlines, but without any assurances offered of quality when new IS was shipped. Even maintenance suffered with support contracts on occasion not being offered to customers for several months after delivery, or even until the next available release. The cost, impact and frequency of failure are notorious in this industry and, therefore, priorities have to change for the causes of failure to be reduced or even eliminated. For example, estimates are frequently taken as absolutes but should be treated as variables which become more accurate as projects progress. Planning needs to include additional processes necessary to handle the changes that are inevitable in any project.

Maximising corporate profit at every opportunity - which in the bigger picture is over shadowed by larger costs at some later date to remedy the problems - needs to be replaced with a balance which considers ethical development and management practice. Senior management in addition to Sales, Recruitment, HR and Quality have to work together ethically for the overall benefits to be realised. Developers were found to desire following the spirit of the IEEE/ACM Code even without knowing of its existence - for reasons discussed previously - but the hierarchical levels of management above prevented many individual preferences from being implemented.

Certainly high level sponsorship within a company ensures a greater to compliance to codes of practice, as happened at M & S. With a chairman spear-heading a campaign throughout the company, M & S not only joined ethical trading initiatives but also the FTSEA4Good Ethical Company index. A champion at boardroom level enthused the workforce to support his policies and achieve the success he desired. A charismatic leader clearly influenced the success of this campaign. In practical terms for project managers, the constraints imposed by tight deadlines and budgets invariably viewed as almost impossible, would not permit any additional meetings/paperwork/reviews for additional 'work'. This was apparent when a meeting was organised in the case study by a manager responsible for quality control for all project managers and developers and, was cancelled due to nil attendance. The quality manager concerned did not have authority over the project managers and additionally had no champion amoung senior management in the boardroom enthusing the workforce - as was the case at M & S.

6.3.4 Achieving Organisational Compliance

Controlled subscriptions to the professional organisations should be paid for by employers, persuaded by the (new) tangible benefits of membership. This should include a tailored version of the Code to individual companies - if necessary - edited jointly by the professional organisations and the employer, to ensure compliance to the principles is beneficial to the employer and appropriate in respective development environments. This would address the issues of ambiguity of requirements and inappropriateness regarding implementation at a local level. The image perceived by practitioners of the leaders of such organisations should be addressed, to eliminate the 'old fashioned fuddy duddies' (Senior Manager) image currently held. With membership on the increase, the professional organisations need to be proactive in promoting their ethical code, with support from all levels of management within the company. Membership to professional organisations and compliance to a code by senior management would encourage developers to do the same. Mandatory membership and compliance would facilitate the aims of the code authors to be achieved, i.e. to instill ethical working practices into the managed development of software, and, in conjunction with the recommendations described, ensure acceptance at local levels.

The purpose of the Law Society's Code of Conduct shares common ground with that of the IEEE/ACM's Code. It aims to promote professional improvement and to facilitate the acquisition of professional knowledge through rules of behaviour created by their respective members. The IEEE/ACM could benefit, however, from the status given to The Guide, regarded as the hallmark of the legal profession and vital to the administration of justice. The IEEE/ACM Code is currently neither of these in the context of IS but would enhance the profession if regarded as such. The acceptance of The Guide is due - at least in part - to the evolving nature of its requirements of members and their input into its contents. The IEEE/ACM Code is created this way but, is not seen by the members of the IS profession to be vital to the development of new software projects. The very nature of software development is clearly very different from that of law practice. Software development (IS) is a nascent profession, with new tools and methodologies being continuously created and/or amended with the creativity required of developers and mangers part of the appeal of choosing a career in this field. In deed, creativity is argued to be the life blood - and therefore a necessity - of development, since bespoke development demands it. Solicitors, conversely, have a profession which is centuries old and dependent on legislation for both definition and subsequent enforcement of its required legal practices.

The requirements of professional conduct for solicitors come from legislation, rules, orders, regulations, codes, guidance and principles. These – numerous – sources and respective requirements are encompassed in The Guide for reference by solicitors as required. With one project manager acknowledging that 'large documents don't get read,' how many of the 900 pages which make up The Guide are actually read by solicitors, however, is unknown. This is where the Law Society code contrasts greatly with the IEEE/ACM code – which has less than 20 pages. Of course no level of documentation will prove to be a panacea, with some IS professionals always wanting more detail and, others less. The evidence collected in the case study indicates that in this context, less is definitely more. The opposite clearly applies to the Law Society. Furthermore, with process improvement practices formally acknowledged by Telco as having tangible benefits, the benefits of encouraging and supporting an ethical workforce - rather than simply stipulating the requirements of a code - would result in greater compliance.

Membership of the EU continues to grow and in response to the need for many solicitors to work in a wider geographical context than just at a local/national level, rules have been created reflecting common ground between the member states. These have been accepted by the respective legal bodies and aim to provide consistency in the application of justice throughout the courts of Europe. Where inconsistencies arise, respective national requirements prevail over any European requirements. Again, the legal profession is ahead of the IS profession, with requirements for European legal practice defined in legislation which started to become effective almost 30 years ago. The legal profession has identified a geographical boundary of its remit by acknowledging that it is neither possible nor, as a consequence, desirable to seek legislation for general application to all EU member states.

Ethical trading is achieved by M & S both internally and with suppliers by working with the parties concerned to agree on what are, and are not, acceptable working practices.

M & S hold inspections and commission an independent company to hold additional inspections to ensure these agreed working practices are adhered to by employees and suppliers at all levels. The suppliers are further encouraged to hold internal inspections and inspect their own suppliers. Furthermore, these agreed working practices are viewed as the minimum standard to be achieved and are expected to be exceeded wherever possible.

The finance industry is regulated by the FSA which, in reaction to large public scandals, created a model representing the nature of the relationship between the FSA and companies exhibiting particular values. The model focuses specifically at organisational levels and, shows that the communication between a company and the FSA is much greater - with a relationship similar to a teacher with a stick and student - for those which exhibit few or none of the desired characteristics. The FSA would appear, therefore, to have the necessary (unlimited) resources to respond to the needs of individual companies with regard to enforcing compliance to the financial regulations - irrespective of need. Clearly such resources do not exist within the IEEE/ACM bodies. Funded by its members, these two organisations have an income relative to the size of membership. The FSA obtains its funding from the government and, therefore, could be argued to have an infinite supply. As with the law society, the FSA is highly regulated with legislation to enforce its requirements on companies and, not a de facto code seeking optional compliance from its members with little legislative backing.

6.4 Summary

The Code was analysed to determine the presence of strengths and weaknesses of existing codes in IS as identified in the literature search. The causes of resistance to codes identified in the case study were discussed with the aim of facilitating acceptance and subsequent compliance. Enforcement of the principles of the Code was also discussed, with contributions from other professions and how they achieve compliance from their respective members.

The weaknesses of codes in general identified in the literature and found to be present in the IEEE/ACM Code included ambiguity, with subsequent clarification compounding the problem yet further. An arbitrary increase in workloads with regards to preparation/study for examinations (in the context of HR and Recruitment) was found, in addition to conflicting working practices, with private study being conducted during office hours resulting in the need to work until midnight. Discipline, deterrence and prevention of cowboy presence were all lacking in the Code, with only an identification of inconsistency with being a professional developer being noted.

Other weaknesses identified in the Code were that it is too long, consisting of eight principles each containing approximately ten sub areas, making a total of approximately eighty areas. Evidence was collected of managers unable to use software tools competently and, who were unaware of the existence of management tools available on the intranet. The explicit requirement for continuous education for project managers is needed, therefore, with skill gaps identified through periodic staff reviews with their respective directors.

The literature found resistance to codes to be common, typically due to protection of job status, the need to work with new people, arbitrary increases in workloads and clarification of new working practices. The protection of job status as possible members of the IEEE/ACM was not seen to be affected by the developers, although employees in HR perceived their status to be greater as members of the CIPD. None of the participants in the case study experienced or expected to experience a need to work with new people or, saw that as a problem.

Poor communication was found to be a frequent contributory factor of unethical practices identified and, therefore, non-compliance to the Code, but how poor communication is addressed on a practical level is missing from the preamble. As with corporate policy at Telco, strategic goals of the Code are not implemented successfully at a local level. Management should be perceived to consider the Code as valuable and be seen to comply with its principles to encourage developers to do the same.

When creating or amending company procedures, guidelines from professional organisations should be considered carefully to enable consistency across the industry, but the ability to tailor procedures and codes to reflect local needs should be exercised. The scope of the Code should be increased to encapsulate companies as a whole, made possible through company memberships, with the introduction of separate and relative requirements/principles for other departments, such as Sales, Recruitment, HR and Quality. These should be created in alignment with their own organisational codes where available. Codes should be created in multiple languages to reflect the diversity found in the workforce of large international IS companies conducting ISD. A code should also be created for customers so they understand what is actually feasible - regarding technicality, quality, cost and time. This would prevent them from requesting impossible ISD: high quality, maximum functionality, minimum time and minimum cost.

Conclusions from the research are presented in the next chapter, and recommendations for further work in the area.

Chapter 7. Summary and Conclusions

7.1 Introduction

Due to the frequent failure of software projects - defined as over budget, late delivery and/or poor quality - the research investigated the role and value of ethics in the development of managed software projects. A definition of ethical practices for this context was provided by the IEEE/ACM Code of Ethics and Professional Conduct for Software Developers. Interpretive research was identified as most appropriate to achieving the objectives with the use of a case study to collect predominantly qualitative data. The software development environment was thus considered in at the individual, team, management and organisational levels, within a large international IS company based in west London. Although many instances of non-compliance to the IEEE/ACM Code were identified by the research methods used, these were not unique to the project teams alone. The other business units participating in the research were all found to be non-compliant - in the context of their own business areas. Although teams of unethical developers and unethical managers working in an ethical environment are what the current climate of blame expects, at least in part, to cause project failure, the cause was found to be due to many factors which were predominantly found at the organisational level. Implications for the IEEE/ACM Code in practice are thus described, with contributions from research into how other professions enforce ethical practices contained in their respective codes and frameworks. Lastly, recommendations and areas for future work are provided.

Section 7.2 reiterates the research area as the role of ethics in the context of managed software development. A definition of ethical practices in managed software development was provided by the eight principles of the IEEE/ACM Code and is briefly described. Section 7.3 summarises the justification for the interpretive research approach, which was to gain an understanding of how and why unethical practices occurred, with an in-depth case study identified as being most appropriate to meeting the research objectives. This approach enabled the data collection techniques of interviews, document analysis and participation observation to be used. Section 7.4 describes the unethical practices found relating to the main issues drawn out of Chapter 2 and, how they occurred. Section 7.5 discusses the causes of the unethical practices identified. Section 7.6 reflects on the enforcement of ethical principles in practice. Lessons learned from the enforcement of codes in other professions are included. Section 7.7 provides a critical evaluation of the Code, followed by Section 7.8 which summarises the contribution made. Lastly, further research in the area is described in Section 7.9 for the interested reader.

7.2 The Research Area

Ethics was introduced through its contribution from philosophy and academia and, the unethical practices currently found in software development were summarised. Management training was found to contribute to the lack of awareness/concern of ethics in ISD, in conjunction with the reward structure of IS professionals in Western Europe based primarily on individual effort and achievement. Unique knowledge was found to be frequently rewarded with individual bonuses, salary increases and promotions. The consequence is an anti-sharing environment, resulting in a 'knowledge guru' status for those with esoteric knowledge. The strengths and weaknesses of current codes and frameworks were then identified in the literature search. The IEEE/ACM Code of Ethics was examined to determine its structure and objectives. The eight principles of the Code were described to provide a definition of ethical practices in managed software development.

The focus of management education was found to be primarily on corporate strategy and profit. A need for change was identified by (a) ethical considerations being an integral part of IS management decision-making (b) a project manager's claim on ethical authority in the workplace is questionable, (c) the constant presence of dispute and conflict in the business environment, as well as the failing of traditional compliance-based management control and, (d) ethical awareness and adherence to a code is essential by project managers if software developers are to take ethical practices seriously.

Frameworks were found to be the most important and most common way of institutionalising ethics and, have been introduced in conjunction with codes and legislation in an attempt to overcome the problems identified in software development. Limitations and weaknesses of current frameworks and codes were identified in the literature as (a) being highly ambiguous, (b) arbitrarily increasing workloads, (c) introducing unclear/conflicting new working practices, (d) a need to work with new people (e) threatening jobs, (f) using alien language, (g) over-simplifying instructions,

(h) designed solely by managers or outsiders and, (i) not addressing discipline, deterrence or cowboy presence. The creation and implementation of codes were found to be intertwined and, not separate activities, with open communication necessary for codes to gain acceptance. When a code is being designed, a software developer could be responsible for forming strategy content, understanding its context and for facilitating any necessary strategic change. He could additionally be responsible for reviewing any vision statements to identify the values which act as its foundation and, the organisations involvement in creating the mission statement.

The umbrella strategy - setting guidelines without detailed plans - was identified as being the most appropriate method to institutionalise ethics in other departments in organisations, such as Recruitment, Human Resources and Quality. Each department would need the ethical standards to firstly be defined and agreed in a framework for reference, as is provided by the CIPD for professionals working in Human Resources. Designers typically determine the contents of new codes, which should be prevented through open communication across all levels prior to its completion. Frameworks are useful on an individual level due to the amount of detail provided. Strict adherence to a framework is not ideal and a framework is best used as a valuable tool and not a rigid structure.

The preamble to the IEEE/ACM Code of Ethics stipulates it should not be read as a finite guide to ethical behaviour and that it: (a) provides guidance on ethical principles, (b) provides a decision-making strategy, (c) addresses conflict in ethical values, (d) addresses three levels: humanity, professionalism and professions, (e) educates software developers into promoting and protecting positive values, without encouraging any whistle-blowing activities and lastly, (f) promotes education, training, support, guidance and inspiration. Language problems typically exist in codes and the IEEE/ACM Code is both ambiguous and repetitious in its requirements. The definitions provided to reduce the ambiguity present, however, actually increase it. The IEEE/ACM organisations do not now hold a monopoly on ethical behaviour, with other professionals in existence, many with their own ethical codes.

7.3 The Research Method

A non-scientific methodology was identified as being most appropriate to enabling the research objectives to be met, with Burrell and Morgan's four sociological paradigms analysed. A justification of why an interpretive approach was taken was provided, followed by a critique of the descriptive case study strategy used. A six month in-depth single case study was conducted as this was necessary for the environment within which project managers and software developers function to be identified, understood and described. The case study strategy was described further to facilitate the research design and, a single embedded case study was identified to be the most appropriate way of enabling the aims and objectives of the research to be met. The activities and deliverables considered essential for a case study were then described and put into context. These included the case study protocol which is a major tactic in proving the reliability of a case study. The reliability and construct validity of the case study was found to be increased by adhering to three principles: the use of multiple sources of evidence, the creation of a case study database and, maintaining a chain of evidence. Participation-observation, interviews and document analysis were techniques described and subsequently selected to collect both quantitative and qualitative data for analysis. These factors determined the suitability of the IEEE/ACM Code of Ethics as an ethical framework to evaluate the practices implemented by software developers and their project managers. A critique of repertory grids was then provided, with a major advantage identified as their 'tailorability' - and with no prior expert knowledge in their use required. A version of repertory grids was thereby chosen as most appropriate for the entering of data collected, thereby enabling a large amount of qualitative data to be transformed into - manageable - quantitative data for analysis.

In an interpretive context, the conclusion(s) drawn are neither right or wrong, nor correct or incorrect - just simply interesting to some readers and not so interesting to others. This is the corollary of adopting the interpretivist research approach. The conclusions are still of value to a wider community than just the author, as they enable both written and verbal debate leading to broader judgments of value to be ascertained. Conclusions can then be compared, evaluated and improved in this manner. Additionally, they provide a vehicle for communicating this knowledge to others - facilitating learning and understanding.

7.4 Evidence Collected of Unethical Practices

The case study conducted enabled an understanding of the environment within which project managers and software developers have to function to be identified and described - including the business units with which they interact. These business units were Recruitment, Human Resources, Quality and, Telco at an organisational level – particularly as it had an influential American parent company. The unethical practices found were categorised by the eight principles of the IEEE/ACM Code of Ethics to enable the research objectives to be met.

The evidence collected showed widespread non-compliance to the principles of the Code - by software developers, project managers and the other business units - relatively as appropriate. As an objective of the case study was to identify and understand the unethical practices specifically in managed software development, it was expected that a higher number of incidents of non-compliance to Principle 3 'Product' and Principle 5 'Management' were recorded than the other six principles. The evidence collected included the following unethical practices which occurred at Telco during the case study:

- A general lack of understanding regarding what exactly was required to be developed
- Estimates of the time required to develop new IS were unrealistic
- Gantt charts were frequently unused, complex and/or poorly maintained
- Training for developers was invariably inappropriate, poorly attended, unplanned or unavailable
- Structured career development was typically unplanned and unmonitored
- Membership of professional organisations was almost non-existent

The synthesis of literature and data from the case study show the theoretical characteristics of ethical practices were desired by the developers and managers participating in the research. The restrictions on ethical practices from being implemented were often found to be caused by factors in the development environment, typically from senior management, salesmen and organisational factors. Senior management in the US were found to set unrealistic deadlines which were imposed on UK project managers who were additionally swamped in a management culture known as meeting mania. Consequently, a development culture of 'deliver it now and get it working later' prevailed. Salesmen were keen to secure their commission and therefore sold new IS which were unrealistic in terms of technical feasibility, quality and/or time. Many new proposals needing estimates of time, cost and quality to be calculated, further reduced management time for concurrent projects.

7.5 Causes of Non-Compliance

The causes of unethical practices identified in the literature and which occurred at Telco during the six month case study were discussed, in terms of individual, team, management and organisational levels.

Developers were found to be non-compliant to Principle 3 which requires developers to strive for high quality. This was due to the development culture, the process of development and testing and, generic organisational influences. High quality development was further prevented by the use of documentation templates which were generic and static in nature, as described above. Documentation should, therefore, be created with only pertinent sections included, with the final document easy to navigate. The development practices described in Telco's own development model - GEM - were largely unknown and, therefore, was invariably unreferenced. Formal models should be tailorable, with necessary compliance given a high profile. Developers whose mother tongue was not English were not consulted in full and pertinent issues went undiscussed. Documentation created in multiple languages is therefore needed in addition to facilitate the understanding and contribution of all team members.

Maintenance activities were hampered by the lack of information provided to the developers. This was found to typically include: (a) what triggered the bug(s) to occur, (b) what had happened that should not - and vice versa, (c) whether the customer had recently upgraded from a version which was working - or not, (d) whether the bug was repeatable, (e) what network topology, i.e. manufacturer/model of equipment, was involved, and (f) whether there were any other bugs reported which might be related - by the customer or partners connected. These points should be included in document templates for testing to ensure consistency in structure but the document should be tailorable, however, to reflect the need of each new IS and enable the omission of redundant information. Communication with customers was not sufficient for developers to create test environments which exactly reflected the environment in which the new IS would eventually be used. Better descriptions are therefore needed to describe a customer's site to facilitate accurate and realistic testing. Moreover, developers were untrained in testing methods as it was assumed they had been trained elsewhere. In addition, there was no information on how to conduct testing on the intranet. The creation and availability of courses on testing are thus necessary to facilitate the achievement of quality development and maintenance.

Training was found to be unplanned and, severely reduced following the downturn in the US economy. Developers were expected to create and execute their own training plans, subject to approval by respective project managers. Budgets were not always available, with approved training frequently cancelled at short notice. Structured training programs are needed, therefore, funded separately from individual and vulnerable project budgets to enable completion. Time constraints also prevented training courses from being taken, for the same reasons as described above: infeasible deadlines, unknown tasks occurring and, the unavailability of training budgets. Training would be easier to accommodate if time management by developers per se was improved, which should be addressed in staff reviews. Learning on the job was preferred by many developers as courses were found to take too long, be generic in content and, handouts were not always available. These were desired for making notes, later perusal to clarify understanding, reference at a later date or, to take away for other developers unable to attend the course. In addition, the examples provided were simplistic and unhelpful when compared to the difficulties experienced in developers' day-to-day roles. Courses should, therefore, be provided which are pertinent and realistic to individual needs - in consultation with respective managers to ensure appropriate consistency. Timely information should be available. Feedback from trainers should also be provided in a timely manner. Where external vendors are used, increased internal communication is needed between departmental managers to ensure valuable resources are not wasted hiring vendors which were previously found to be poor.

Project managers used Gantt charts created with infeasible deadlines defined by salesmen motivated by commission payments. Salesmen should be paid a fixed salary, therefore, to facilitate the delivery of new IS which is feasible to be designed, developed, tested and documented in the timescales agreed with customers. Realistic plans can then be created and adhered to regarding both development and testing. Gantt charts were used as part of the planning process but, with imposed deadlines from US management and/or unrealistic deadlines from salesmen, their relative value to individual projects was found to be limited. Deadlines and deliverables need to be agreed in consultation between US management or salesmen and, customers and project managers. A balance needs to be determined between the empowerment awarded to salesmen and management and, what is realistically possible as an acceptable standard - company wide with consistency assured. Gantt charts would receive a higher profile as a result of their improved value in practice, leading to increased awareness of their existence and use by all project members. This would be facilitated by timely updates and re/distribution as appropriate, with time allocated for staff reviews, training and other duties. Training in the use of software, for example MS Project, would further enhance their value by increased awareness and understanding of the features available in the software. Training in other project management tools - which were available on Telco's intranet - would additionally facilitate the planning process as the tasks could be completed sooner and/or better. The culture known as meeting mania took up valuable management time which needs to be controlled. Telco was thought to have been previously criticised for poor communication, yet is now criticised for too much. Numerous meetings are organised in conjunction with a high number of apparently disparate managers required to attend. These meetings need to be managed efficaciously and, need to accommodate the time differences between the continents involved, enabling attendance by all pertinent managers.

The game playing of budgets was found to be necessary by some project managers on occasion to facilitate the overall success of their projects. This was due, in part, to the empowerment bestowed upon the technical director who was found to cancel large purchases which he had previously approved. Unexpected sources of revenue were then used by managers to pay for purchases, without the technical director being notified of their existence. Furthermore, to increase the chance of meeting imposed deadlines set by the US, lists of developers who could be made redundant with minimum impact to the company were edited to enable projects to continue with fewer redundancies and, therefore, reduced impact. Moreover, developers were hired as contractors as they were not included in the head count of projects, enabling a larger team to be employed than would otherwise be possible. Again it is necessary for project contracts to be awarded with realistic constraints on time and money. That was not happening at Telco, hence some of the unethical practices found. Communication between departments should, therefore, increase to facilitate a correct understanding of what each department requires and when that information is needed. As a result of the improved relationships which would ensue, favours could be done and later reciprocated as required. The increased understanding (at an organisational level) should result in open and honest practices being conducted which aim to facilitate the achievement of organisational goals and not only local goals.

The non-compliance found to Principle 5 requiring project managers to recruit developers with appropriate skills and further their development was due to a general unawareness of standards, infrequent staff reviews and the nature of training provided. The recruitment of developers with appropriate skills was made difficult with the misalignment found between modules taught at university and industry's specific needs. Even for project managers, the Gantt charts used at university were simplistic compared to those in use by the project teams. Collaboration had been sought previously by both parties on more than one occasion in an attempt to bridge the gap, but without success. A relationship clearly exists between industry and academia - hence the case study and concurrent presence of several placement students. For graduates to be better equipped for industry a continued effort is required for the gap to be reduced and, ultimately eliminated. Project managers might then not have to wait 18 months to find developers with experience gained at other similar companies as was found. Staff reviews could have identified and addressed the misalignment between academia and industry - and other skill gaps - but they were conducted with irregularity. Time constraints on projects were found to be the most common reason for postponement or cancellation of staff reviews. Infeasible deadlines were set for new ISD by both salesmen and US management, in conjunction with the existence of tasks unknown tasks at the outset.

The non-compliance found to Principle 3 requiring developers to conduct maintenance professionally was due to many factors controlled at an organisational level, occurring both prior and post delivery of new IS. A high number of bugs reported was found to be due to the short development time scales allocated to projects, the lack of training in test procedures and the lack of time allocated for testing *per se*. Training should therefore be provided in formal testing procedures, including the writing of test cases. These should be written prior to and during the testing process as initial design documents were found to be incomplete, ambiguous and misunderstood. Sufficient time should be allowed in Gantt charts for testing to be conducted thoroughly, with customers educated to understand the nature of software and limitations of testing. Documentation was also inappropriate, with records identified of excessive length, containing misuse of terms and which were difficult to navigate. The corollary to these anomalies was documentation went unread. Documentation is clearly a difficult issue, with the information provided seen as insufficient for some and excessive for others. This leads again to the need for tailorable templates so that information provided can be specific to the project at hand.

Although formal interviewing procedures were created by the CIPD to ensure fairness, legal compliance and consistency across the industry, Telco developed its own interviewing procedures, however, or rather the President/CEO of Telco. He created his own interview technique which was then implemented throughout the company. Moreover, the head of UK recruitment was hired without any experience in the industry which led, in part, to a developer not being offered position initially, until spotted serendipitously by a project manager at some point later. Clearly the recruitment department at Telco contributed to the difficulties in recruiting quality developers.

7.6 Evaluation of the Code

The practical value of the IEEE/ACM code and other codes in existence is evident from the current state of the IS profession. Large scale failure of new IS projects is frequently in the media with millions of pounds regularly lost and enormous damage caused to the reputation of the profession as a whole. Numbers of students enrolling on IS courses are falling and the appeal that once existed for a career in this area no longer exists. The blame cannot be the sole responsibility of the codes or the professional organisations which create them. Research has shown, however, that other professions do not suffer the same notoriety as the field of IS and, have other methods of achieving - greater - compliance to their own codes of practice. The Code provides a starting point, however, which enables debate and consequent refinements, which should lead to a code of increased value (Dodd and Lycett 2003).

The requirements of the IEEE/ACM Code are described in approximately 20 pages with minimal legislation in existence to formally enforce them. The 900 page document created by the Law Society for similar purposes may appear extreme in length, but enforced through a plethora of legislation results in almost total compliance from the Society's members. The Code needs to be developed in a similar fashion further, therefore, to include: (a) situations in which its members are likely to find themselves in, (b) the detail of those situations and, (c) what is required of developers and managers in those contexts. The eight principles as they stand are short and simplistic and not applicable to any great effect in the real working environments in which developers and project managers find themselves in today. The amended Code should be structured as a reference document to be useful in practical terms and, not left unread on a shelf. The document should then be used as the definition for new legislation which would ensure its requirements are implemented by the membership.

Few reasons were given for not joining professional organisations, such as the IEEE and ACM but, those given were commonly held: (a) there was no need to join, (b) the organisations were not terribly relevant, (c) there were no particular benefits, (d) developers could not being bothered to join and, (e) non-membership was perceived not to hinder one's work. Insufficient time available also prevented attendance to meetings as working into the evening was common. Increased working from home would enable greater autonomy of working patterns and, in conjunction with abolition of flexi-time, would facilitate attendance of meetings. Feasible timescales allocated to development would additionally facilitate meeting attendance. Senior management were found not to value the membership of professional organisations, but for such membership - leading to compliance to their codes - to be seen as valuable by developers, it is necessary for management - or at least be perceived - to see them as valuable too. The consistent requirement of criteria to join would facilitate the value of membership further and, payment of fees by companies would encourage wider membership, with the levels of membership previously consulted and agreed between senior management and the professional organisations for all employees involved.

For the Code to have an impact on the IS profession as a whole and remove or reduce the notoriety currently experienced, its effects needs to be far reaching. The level of membership of the professional organisations needs to substantially increase so as to affect as many developers and project managers as possible. Furthermore, the membership needs to be extended from developers and managers to all those employed in the IS profession, including contractors, the self-employed, consultants and recruiters. Compulsory membership in conjunction with mandatory exams and the issue of annual practicing certificates is recommended. In addition, the organisations need to grow substantially in authority. Without the authority to enforce its requirements with disciplinary action, the requirements will continue to be perceived as optional by the members. A professional organisation was found not to exist for the recruitment industry, with information and guidance sought from the organisation created to oversee Human Resources, the CIPD. Accommodation of the recruitment industry needs to be formally addressed by the CIPD, another organisation or, a new one specifically for this industry. In addition, confidential advice and support services should be set up by the professional organisations to help with members' personal and professional problems.

Members of professional organisations should be required to behave in both their professional and private lives in a way which is not fraudulent, deceitful or otherwise harmful to the profession and, join a mandatory scheme of compulsory continuing professional development. The professional bodies should receive annual payments from members for the running costs and, lobby government to make requirements of the Code legal obligations. A minimum 3 years of registered practice should be expected before a member can become a project manager, contractor or, set up his own consultancy business.

The discipline required for the Code to be taken seriously consists of much more than an 'identification that their actions are inconsistent with those of a professional' as the code currently states. The annual practicing certificates should be issued with restrictions or comments to ensure the holder considers his actions and takes appropriate corrective action, as happens with the law society. The practicing certificate should be withdrawn altogether in extreme circumstances. This is one method of enforcement. Others are summarised in the following section.

7.7 Enforcement of Ethical Codes

The research found enforcement of ethical codes at individual, team, management and organisational levels to be disparate in both implementation and effect across the professions investigated. In law, compliance is achieved through legislation and the administration of practicing certificates. In finance, compliance is achieved with an appropriate relationship between the FSA and organisations which reflects their current practices and requirement to change. In the retail sector, M & S achieved great success through its charismatic leader, financial reward scheme and inspections. The IS profession was found to have little in the way of legislation, no charismatic leader, inconsistent reward schemes and, both optional membership and compliance to professional IS organisations and their respective codes.

Any penalty awarded to developers and project managers who do not comply with the IEEE/ACM Code's requirements needs to be administered and monitored to be effective, requiring collaboration between the professional organisations and employers. Other difficulties then become evident. Such a task would require an infeasible administrative effort regarding manpower and paperwork. The professional organisations need to be seen as credible by employers for them to join their efforts in enforcing the Code's requirements. In addition, with membership to professional organisations found to be almost non-existent at Telco, the professional organisations cannot afford to risk upsetting those IS professionals who are members for fear of having their membership reduced further still. The case study found senior management to not hold respect for IS professional bodies and/or their leaders and are, therefore, unlikely to support any efforts to reprimand developers and project managers for non-compliance. The weaknesses can be seen, therefore, to be introduced through the practical implementation of the code.

The Code requires developers to break the law, however, when they believe it to be inappropriate, but they cannot hold the authors of the Code (the IEEE and ACM) responsible for any subsequent legal action taken against them. Mandatory compliance would be difficult to enforce with a lack of formal support by the professional organisations in such instances.

7.8 Summary of Contribution

Allowable unethical practices found necessary to contribute to overall project success were identified. These involve (a) non-disclosure of unexpected sources of revenue to senior management by project managers which are then used to pay for project purchases, (b) the editing of lists of developers who can be made redundant with minimum impact to the company to enable projects to continue with reduced impact and, (c) the hiring of developers as contractors as they are not included in the head count of projects, enabling a larger team to be employed than would otherwise be possible. The accommodation of allowable unethical practices needs to be incorporated into ethical codes facilitate the achievement of overall project success.

Developers are prevented from moderating the interests of all parties involved in ISD as required by the Code due to development approaches imposed by senior management.

This requirement should be amended to acknowledge that compliance in practice is only achievable within a developer's remit of authority.

Developers are prevented from striving for high quality development due in part to the documentation templates which were generic and static in nature. Templates need to be tailorable to meet the needs of individual projects. Project documentation should be easy to navigate and of a manageable size to encourage readership. Furthermore, documentation should be reproduced in multiple languages to facilitate correct understanding and contribution among the developers (and project managers) whose first language is not English.

Developers are prevented from conducting maintenance professionally due in part to the lack of information provided in bug reports. This is codified as: (a) what triggered the bug(s) to occur, (b) what had happened that should not - and vice versa, (c) if the customer had recently upgraded from a version which was working - or not, (d) if the bug was repeatable, (e) what network topology, i.e. manufacturer/model of equipment was involved, and (f) if there were any other bugs reported which might be related - by this customer or partners connected. Document templates should incorporate these points but, be tailorable to reflect the nature of each bug reported. In addition, better descriptions are needed of customers' sites, regarding for example, hardware topologies and other IS in use with which they must connect, to facilitate accurate and realistic testing in the offices where the tests are to be conducted.

Estimation of the time expected to complete known tasks was found to be more realistic when provided by developers and as such, should be formally required to contribute to the estimation activities in a development model/conducted by a project manager.

The use of external and pertinent consultants should be adopted to inform project teams and their managers of developments in the field, as found to work successfully in Human Resources. This would eliminate the burden to identify, evaluate, select and address issues of possible concern due to the lack of time commonly found.

The financial reward structure of IS professionals should be based on team successes and not individual effort and/or achievement. This would facilitate a reduction in the existence of esoteric knowledge which results in an anti-sharing environment. Information sharing, therefore, needs to become a component of a developer's performance. Commission payments to salesmen should cease, replaced by fixed salaries to facilitate the delivery of new IS which is feasible to design, develop, test and document in the time scales agreed with customers.

The strengths of codes in general identified in the literature and found in the IEEE/ACM Code were codified as: (a) the advocation of truth, co-operation, communication, self-control, goodwill to others, self reflection and charity, (b) encouragement for continuous education, (c) the identification of stakeholders to be managers, clients, suppliers, communities, other professionals, other employees and society as a whole, (d) the consideration of knowledge in context and significance and, (e) the identification of value categories as personal, organisational and professional. The weaknesses of codes in general identified in the literature and found in the Code were codified as: (a) being highly ambiguous, (b) over-simplifying instructions and, (c) not addressing discipline, deterrence or cowboy presence. In addition, the Code was repetitious and ambiguous in its requirements and, the application of a requirement to communicate effectively and efficiently at all levels in practice is missing. A requirement for project managers to embark on lifelong learning is also needed - as currently required for developers.

Reasons for not joining professional organisations, such as the IEEE are codified as: (a) there is no need to join, (b) the organisations are not terribly relevant to day-to-day roles, (c) there are no particular benefits, (d) developers cannot being bothered to join and, (e) non-membership is perceived not to hinder one's work.

A code is needed for customers to comply with so that they understand what is feasible from new ISD regarding technicality, quality, cost and time. This would prevent customers from requesting impractical ISD: high quality, maximum functionality, minimum time and minimum cost.

The creation of a support group is needed offering confidential advice and support on all professional and private matters. This service should be available to all IS professionals, funded by contributions from the members of professional organisations.

A professional organisation does not exist for the Recruitment industry, with information and guidance currently sought from the organisation created to govern Human Resources - the CIPD. Accommodation of the Recruitment industry needs to be formally addressed by the CIPD, another organisation or, a new one specifically for the industry. Who should create it and what it should consist of needs to be established, with appropriate consultation.

7.9 Further Research

Research is needed into ethical working environments which promote business advantage. A model of contingent hierarchy is needed for information to be distributed in a timely manner when project members are absent which prevents the normal hierarchy of information distribution from being executed. Tailorable templates of project documents are needed in multiple languages, the contents of which need to be identified, with tailorable and mandatory/optional sections determined. A development model incorporating developers into the estimation process is required and, how this is best implemented needs further research.

The funding of training needs investigation to establish how training can be identified and implemented without (negative) external influence to enable identified skill gaps to be filled. Formal courses in testing are needed to increase the level of professionalism in this area. The content and structure of such courses need to be ascertained.

Pay structures of salesmen, developers and other IS professionals where fixed salaries are paid should be investigated to establish whether this method of payment facilitates ethical development practices. Research needs to be conducted to establish whether the abolition of flexi-time would facilitate the attendance of meetings hosted by professional organisations.

References

ABI-RAAD, M. June 1999: Code of ethics? Which one?. *SIGCSE Bulletin*, vol 31 no 2, 73-77, ACM, USA

AMBROSE, M., NOEL, T., SCHMINKE, M. October 1997: The effect of ethical frameworks on perceptions of organisational justice. *Academy of Management Journal*, vol 40 no 5, 1190. 7 August 2000, http://web6.infotrac.london.galegroup.c.dyn=47!xrn_10_0_A20122522?sw_aep=bui.

ANDERSON, G. January 5th, 2001: Presentation to team members, Crown Plaza Hotel, London.

ARMSTRONG, A. 1993, 3rd Edition: *How to be an even better manager.* London, UK: Kogan Page Ltd.

AVISON, D. 2001: research methods course, Brunel University

BALDWIN, L. March 3, 2000, Interview at Brunel University

BELBIN, M. 1996: *The coming shape of organisation.* Butterworth Heinemann, London. In *Organisational Behaviour an introductory text,* BUCHANAN, D., HUCZYNSKI, A. 1997. Prentice hall, UK. 3rd Edition

BENBASAT, I., GOLDSTEIN, D., MEAD, M. 1987: The case research strategy in studies of information systems. *MIS Quarterly,* 11, number 3, pp369-386.

BENYON-DAVIES, P., BLYTH, A., OWENS, I., LLOYD-WILLIAMS, M. 2000: *IS failure,evaluation and organisational learning.* In the Proceeding of the 5th UKAIS Conference, Cardiff, eds. P. Benyon-Davies, M. Williams and I. Beeson. Pp444-452. McGraw-Hill, England

BOLAND, R. 1985: Phenomenology: a preferred approach to research on information systems. *Research methods in information system.* In Walsham, G. 1993: *Interpreting information systems in organisations*. Wiley and Sons, England

BOLAND, R. 1989: Beyond the objectivist and subjectivist: learning to read accounting `as text. *Accounting, Organisations and Society*, 14, number 5/6, pp591-604.

BOTT, F., COLEMAN, A., EATON, J., ROWLAND, D. 1996: *Professional issues in software engineering*. UCL Press Limited

BOTTON, A. May 2000: *A guide to happiness*. Channel 4, 'Nextstep', 8pm

BUCHANAN, D., HUCYZNSKI, A. 1997: *Organisational behaviour an introductory text*. Prentice Hall, 3rd Edition

BUERK, M. June 4 2000: *Soul of Britain*. BBC1, 10.35pm

BURNS, T., STALKER, G. 1961: *The management of innovation*. Tavistock, London.

BURRELL, G., MORGAN, G. 1979: *Sociological paradigms and organisational analysis*.Heinemann Educational Books Limited

CAMPBELL, H., MARSHALL, R. September 1999: Ethical frameworks and planning theory. International Journal of Urban and Regional Research. Vol 23 no 3, 464. 7 August 2000, http://infotrac.london.galegroup.c.&dyn=sig!3?sw_aep=bu_i

CAVEDON, L., ANAND, R., SONENBERG, L., TIDHAR, G. 1997: Teamwork via team plans in intelligent autonomous agent system. *Worldwide Computing and its Applications International Conference*, WWCA 1997 Proceedings, Japan, 106-121

CAVAYE, A. 1996: *Case study research: a multi-faceted research approach for IS*. Information Systems Journal, no 6, 227-242

CHECKLAND, P. 1981: *Systems thinking, systems practise*.Wiley, Chichester.

CLIST, May 31, 2000. Centre for Living Information Systems Thinking workshop, Brunel University, UK

COLLINS GEM, 1992: *English dictionary*. Harper Collins Publishers

COMPUTING, 12 October 2000a: *R and D information systems*, p106

COMPUTING, 12 October 2000b: *Three secrets for smart people*, p76

COOPER, R. 1986: Organisation/disorganisation, *Social science information,* 25, number 2, pp299-335.

COOPER, R., FOX, S. 1989: Two modes of organisation, *Frontiers of management,* Routledge, London.

COPELAND, L. 2003: The Maturity Maturity Model (M3) – Guidelines for Improving the Maturity Process. www.stickyminds.com/sitewide.asp?sid=1&sqry=%2A, 09-04-2004

CURRIE, W. 1989: The art of justifying new technology to top management, *Omega*, 17, number 5, pp409-418.

CZAJA. R., BLAIR, J. 1996: *Designing surveys: A guide to decisions and procedures.* Pine Forge Press, California.

DAKIN, K. July 1996: Are engineers morally challenged? *IEEE Software*, vol 13 no. 4, 20-27

DAVIES, H. (2002): An Ethical Framework for Financial Services. Financial Services Authority Discussion Paper 18, October 2002

DAVIS, M. April 1998: Computers, ethics and society. *Ethics*. Vol 108, no 3, 636, 2nd Edition. 7 August 2000, http://web6.infotrac.london.galegroup.c&dyn=28!xrn_8_0_A20548408?sw_aep=bui

DODD, S. 1999: *The implications of using the internet for teaching*. Unpublished Final Year Project

Human Services Evaluation, Kluwer-Nijhoff, Boston.

FORD, J., HONEYCUTT, E., KURTZMAN, L., winter 1996: Potential problems and solutions when hiring and training a worldwide sales team. *Journal of Business and Industrial Marketing.* Vol 11 no 1, 42. 7 August 2000, http://web6. infotrac. london.galegroup.cdyn=53!xrn_19_0_A18377741?sw_aep=bui.

FRANSELLA, F., BANNISTER, D. 1977: *A manual for repertory grid technique.* Academic Press. In NORRIS, M. (date unknown): *Beginner's guide to the use of repertory grid in research.* University of Surrey. SBN 0 907928 01.3

FREEMAN, H. May 20, 2000: *Are you a really good team player?* The Guardian, 5

GALLIERS, R. 1992: *Information systems research: issues, methods and practical guidelines.* Blackwell Scientific Publications, Oxford.

GIDDENS, A. 1984: *The constitution of society,* Polity Press, Cambridge.

GOBOLD, A. April 1999: What has ethics got to do with me? I'm an engineer. *Engineering Management Journal*, 81-86

GOTTERBARN, D. 1999a: Specifying the Standard – Make it right: a software engineering code of ethics and professional practice. *Computers and Society*, vol 29 no 3, 13-16

GOTTERBARN, D. 1999b: How the new software engineering code of ethics affects you. *IEEE Software*, vol 16 no 6, 58-64

GRODZINSKY, F. March 1999: The practitioner from within: revisiting the virtues. *Computers and Society*, 9

HALANG, W.A. 1998: Real-time systems education: What is really essential? *Proceedings Real-Time Systems Education III*, IEEE Computer Society, Poland

HARRIS, G. 2000. Project manager, ID Research Ltd, London. Private communication

HASNAS, J., SMITH, J. March 1999: Ethics and information systems: the corporate domain. *MIS Quarterly*, vol 23 no 1, 109. 7 August 2000, http://web6.infotrac.london.galegroup.cdyn=9!xrn_17_0_A54711025?sw_aep=bu_i.

HEBEL, M. 2000. Lecturer, Brunel University, private communication.

HEIDEGGER, M. 1962: *Being and time.* Harper and Row, New York

HIRSCHHEIM, R., KLEIN, H. 1989: Four paradigms of information systems development. *Communications of the ACM,* vol 32, no 10, pp1199-1216

HIRSCHHEIM, R., SMITHSON, S. 1988: A critical analysis of information systems evaluation. In *IS Assessment: Issues and Challenges,* North Holland, Amsterdam. In Walsham, G. 1993: *Interpreting information systems in organisations*. Wiley and Sons, England

HIRSCHHEIM, R., KLEIN, H., LYYTINEN, K. 1995: *Information systems development and data modeling – conceptual and philosophical foundations.* Cambridge University Press, UK

HOUSDEN, M. 1992: *Successful market research.* Hodder and Stoughton PLC, London

HURST, J. June 9, 2000. Interview at The Rectory, Langley, UK

ICSE 2000 (International Conference on Software Engineering): *advance call for participation*. http://www.csr.uvic.ca/icse2001/acfp.html, 18 April 2000

IFIP ETHICS TASK GROUP. August 1995: Regarding codes of conduct for computer societies. Recommendations to the International Federation for Information Processing. 7 August 2000, http://courses.cs.vt.edu/~cs3604/lib/worldcodes/[10].recommendations.html

INTRONA, L.D., POULOUDI, A. 1999: Privacy in the information age: stakeholders, interests and values. *Journal of Business Ethics* 22, 27-38

IRANI, Z. May 31, 2000. Centre for Living Information Systems Thinking (CLIST) Workshop

JSB SOFTWARE TECHNOLOGIES PLC, June 22, 2000. Advert in Financial Times

JOHNSON, P., SMITH, K. 1999: Contextualizing business ethics: anomie and social life. *Home Relations*, vol 52 no 11, 1351. 7 August 2000, http://web6.infotrac.london.galegroup.c5&dyn=3!xrn_5_0_A59106659?sw_aep =bui

KANER, C., BACH, J., PETTICHORD, B., 2002: Lessons Learned in Software Testing: A Context Driven Approach. Wiley & Sons

KAWANO, R. 1997: Steps toward the realisation of "Human-centred systems". *IEEE Sixth Annual Human Factors Meeting*, Florida

KELLY, G. 1955: *The psychology of personal constructs.* (2 vols.) Norton. In NORRIS, M. (date unknown): *Beginner's guide to the use of repertory grid in research.* University of Surrey. SBN 0 907928 01.3

KELLEY, R. Oct 1999: How to be a star engineer. *IEEE Spectrum*, vol 36 no 10, 51-58, IEEE USA

KLEIN, H., HIRSCHHEIM, R., NISSEN, H. 1990: Information systems research: contemporary approaches and emergent traditions. *Proceedings of the [10] TC8/WG8.2 working conference on the IS research arena of the 90's.* Elselver Science Publications, 327-345.

KLING, R., IACONO, S. 1989: The institutional character of computerized information systems. *Office, Technology and People,* 5, number 1, pp7-28.

KNIGHTS, D., MORGAN, G. 1991: Corporate strategy, organisations and subjectivity: a critique. *Organisation Studies,* 12, number 2, pp251-273.

KUMAR, K. 1990: Post implementation evaluation of computer-based IS: current practices, *Communications of the ACM,* 33, number 2, pp203-212.

KYNG, M. 1991: Designing for co-operation: co-operating in design. *Communications of the ACM,* vol 34, no 12, pp65-73

LAWRENCE, P., LORSCH, J. 1967: Differentiation and integration in complex organisations. *Administrative Science Quarterly, 12,* number 1, pp1-47.

LEEDY, P. 1997: *Practical research – planning and design.* Prentice Hall Inc.

LEGGE, K. 1984: *Evaluating planned organisational change,* Academic Press, London.

LEMBKE, S., WILSON, M. 1998: Putting the "team" into teamwork: alternative theoretical contributions for contemporary management practice. *Human Relations*, vol 51, no 7

LEWIS, T. 2000: answer given to question on how ethics might help development. http:www.askme.com/viewanswer.asp?vid=681866&ex

MACREDIE, R. 1998, Lecturer, Brunel University, BSc Computer Science Level 3

MACREDIE, R. 1999, private communication

MARKUS, M. 1983: Power, politics and MIS implementation. *Communications of the ACM,* 26, number 6, pp430-445.

MILES, M., HUBERMAN, A. 1994: *Qualitative data analysis*. Sage Publications, UK. 2nd Edition

MILLER, S. 2004: www.pragmaticsoftware.com 12-04-04

MINTZBERG, H., WATERS, J. 1985: Of strategies, deliberate and emergent. *Strategic Management Journal,* 6, number 3, pp257-272.

MORGAN, G. 1986: *Images of organisation.* Sage, Beverly Hills.

MURRAY, F. 1989: The organisational politics of information technology: studies from the UK financial services industry. *Technology analysis and strategic management,* 1, number 3, pp285-298.

NICHOLLS, D. 1987: *Introducing SSADM.* NCC Publications

NISSEN, H., KLEIN, H., HIRSCHHEIM, R. 1990: Information systems research: contemporary approaches and emergent traditions. *Proceedings of the IFIP TC8/WG 8.2 Working Conference on the Information Systems Research Arena of the 90's, Denmark.*

NORRIS, M. (date unknown): *Beginner's guide to the use of repertory grid in research.* University of Surrey. SBN 0 907928 01.3

OLIVER, M. 1998: *Hamlyn history philosophy.* Octopus Publishing Group

ONLINE ETHICS 2000 (Online Ethics Center for Engineering and Science): *Ethics center education.* wysiwyg://55/http://www.onlineethics.com 11 April 2000

ORLIKOWSKI, W., BAROUDI, J. 1991: *Studying information technology in organisations: research approaches and assumptions.* Information Systems Research, vol 2, no1, 1-28

ORLIKOWSKI, W. 1992: The duality of technology: rethinking the concept of technology in organisations, *Organisation Science*, 3, number 3, pp398-427.

OXFORD CONSICE DICTIONARY 1996, Oxford, UK, Oxford University Press, 463, 9th Ed

PATRICK, F. 2004: Promises and prescription: how the theory of constraints can help cure common project ailments. www.stickyminds.com 1-5-2004

PATTON, M. 1987: *How to use qualitative methods in evaluation.* Sage, USA. In YIN, R. 1994: *Case study research: design and methods.* Sage Publications, UK.

PAUL, R.J. 1994: Why users cannot get what they want. *International Journal of Manufacturing System Design*, vol 1 no 4, 389-394

PC WEEK. May 1999: *Brainstorming: before opening the floodgates to new KM technologies, IT managers should make sure users are ready, willing and able to share what they know.* v16 i22, 65

PETTIGREW, A. 1990: Longitudinal field research on change: theory and practice. *Organisation science,* 1, number 3, pp267-292.

POULOUDI, A. April 1997: Focus: conflicting concern over the privacy of electronic medical records in the NHSnet. *Business Ethics*, vol 6 no 2, Oxford UK: Blackwell Publishers

POULOUDI, A. 1999: Aspects of the stakeholder concept and their implications for information systems development. *Proceedings of the 32nd Hawaii International Conference on System Sciences*

POULOUDI, A. March 3, 2000, Interview at Brunel University, UK

PRAXIS 1999: *Ethics in the workplace – objectives.* http://www.tiac.net/users/praxis/index.html, 13 April 2000

PREECE, J. 1996: *Human computer interaction.* Addison Wesley, England

QUINN, J. 1980: *Strategies for change: logical incrementalism.* Irwin, Homewood.

REMEMYI, D., March 2, 2000. Research Methods Course, Brunel University

ROBSON, C. 1996: *Real world research.* Blackwell Publishers, Oxford.

ROGERSON, S. 2004: A week is a long time in computer ethics. IMIS Journal, vol 14 no 2, from www.ccsr.cse.dmu.ac.uk 8 April 2004

ROGERSON, S., Prior, M. 2003: Is it ethical? ETHICOMP 2002 survey of professional practice. IMIS Journal vol 13, no 6

RYLE, A. 1975: *Frames and cages, the repertory grid approach to human understanding.* Sussex University Press, England.

SALT, J. 1998. Lecturer, Brunel University, BSc Computer Science Level 2

SAUER, C. 1993: *Why information systems fail: a case study approach.* Alfred Waller Ltd., England

SCHNETTLER, M. 2000: answer given to question on how ethics might help development problems. http://www.askme.com/viewanswer.asp?ja=1&vid=67072, 21 April 2000

SEDLET, S. July 1999: Computers, ethics, law and society: What do we teach undergraduates? International Symposium on Technology and Society – Women and Technology: Historical, Societal and Professional Perspectives, Proceedings, *IEEE Networking the world*, USA, 249-253

SINGER, J., VINSON, N. 1998: empirical software engineering research ethics. *Proceedings of the sixth IEEE International Symposium on Software Metrics.* http://church.computer.org/proceedings/metrics/0403/04030326abs.html, 11 April 2000

SOMMERVILLE, J., DALZIEL, S. 1998: Project team building – the applicability of Belbin's team-role self-perception inventory. *International Journal of Project Management*, vol 16 no 3, 165-171, Pergamon, UK

SMIRCICH, L., STUBBART, C. 1985: Strategic management in an enacted world. *Academy of Management Review,* 10, number 4, pp724-736.

STAKE, R. 1994: *Case studies.* In Denzin, N. and Lincoln, Y. (ed.), *Handbook of qualitative research*, London, SAGE Publications

TIMKA, T. June 1999: Professional ethics for system engineers in health care. *Methods of Information in Medicine*, vol 38 no. 2, 144-146

VAN DER VEN, J. 1998: A formation of the moral self. Eeedmans Publishing Company, Michigan. In *Ethics*, July 1999, vol 109 no 4, 939. 7 August 2000, http://web6.infotrac.london.galegroup.c&dyn=6!xrn_11_0_A55266897?sw_aep =bui

WALSHAM, G. 1993: *Interpreting information systems in organisations*. Wiley and Sons, England.

WOODALL, J. November 1996: Managing culture change: can it ever be ethical? *Personnel Review*. Vol 25 no 6, 26. 7 August 2000, http://web6.infotrac.london. galegroup.cdyn=50!xrn_14_0_A57829295?sw_aep=bui.

YEATES, D., CADLE, J. 1996: *Project management for information systems*. Pitman, 2nd Edition, London

YIN, R. 1984: *Case study research: design and methods*. SAGE Publications, UK

YIN, R. 1994: *Case study research: design and methods*. SAGE Publications, UK. 2nd Edition

YOGALINGAM, K. June 2000, Interview at Ms Yogalingam's home

YOUNG, E. 1989: On the naming of the rose: interests and multiple meanings as elements of organisational culture. *Organisation Studies,* 10, number 2, pp187-206.

ZUBOFF, S. 1988: *In the age of the smart machine.* Basic Books, New York.

Appendix A. Alternative Research Approaches

For the research objectives to be met, an appropriate research methodology needed to be selected. The philosophical stance adopted by the author had to be determined, enabling the research approach used and associated methods adopted - for data collection and analysis - to then be selected and described.

Sociological approaches can be categorised into just four paradigms (Burrell and Morgan 1979, Hirshheim *et al* 1995, Walsham 1993). These paradigms are: Functionalism, Radical Structuralism, Radical Humanism and Interpretivism. As the latter, Interpretivism, was described in Chapter 3, the remaining paradigms are described below, including a brief description of the philosophical stance taken.

A.1 Functionalism

Functionalism assumes reality can be explained, predicted and controlled by laws or patterns. In the context of this research, software developers cannot be involved in any issues related to power, conflict or system goals. The nature of the social world is regarded not as conflicting, but ordered, to overcome 'fuzzy' issues. Relatively concrete artifacts and relationships which make up the social world are assumed, enabling it to be modeled and understood by mechanical and biological analogies. The resolution of difficult issues is assumed to be the prerogative of management. Objectives are also decided by management and are believed to be clear, agreed and without conflict. Any software developer's conflicts are about the means to the ends, not the ends *per se*. His choice of tools for software development maximize formality so as to reduce reliance on intuition, politics or judgment - politics is viewed as irrational as it is believed to interfere with maximal efficacy. Structured analysis formalizes analysis procedures to reduce the disruption caused by politics and other influences, i.e. it is a simple technical process. Functionalism assumes there is only one reality - economic - consisting of objects, properties and processes which can be directly identified, described and measured - usually in the form of quantitative financial goals and system performance characteristics. The ultimate organisational goal is to have technical control and to maximize profits, with this economic reality believed to be essentially the same for everyone, enabling consistent user-requirements to be obtained. Any change management necessary is expected to have deterministic qualities. The emphasis is on fixed functional specialization and standardization of jobs and responsibilities, with the hierarchical and organisational arrangements monitoring and controlling software development. The ontology is of realism: an empirical organisational reality exists, independent of its perceiver or observer.

Unfortunately, the objectives of a new information system are invariably not clear, agreed or without conflict, as this approach assumes. Realistically, new IS objectives are the converse: ambiguous, controversial and the subject of debate and disagreement. Functionalism is not able to understand organisational or societal life as both entail the study of people - known to be difficult with positivistic means.

A.2 Radical Structuralism

The radical structuralist paradigm believes in an objective, economic reality. A fundamental social conflict is naturally associated to society, in this context mainly between the software developers and project managers. Information systems should facilitate the collection of wealth by developers in return for their efforts. Legitimate system objectives are considered to be the improvement of developers' working conditions, as against maximizing profits for the shareholders. Social conflict follows a pattern of exploitation, revolt and synthesis, where synthesis takes the form of a new political order and ideology. This paradigm is committed to radical change, emancipation and man's potentiality, by striving to break away from a capitalist market economy towards a collectively planned and managed economy. The ontological stance is that of realism, reflecting in the belief of a pre-existing empirical reality. The epistemological stance is that of a dialectical enquiry in the specific form of a materialistic view of history and society. A reflection of the status quo is reflected, with the aim of providing the rationale for radical change. As all knowledge is attested to be related to human interests, a neutral science is not possible. Culture, knowledge and human interests are believed to be intimately related - with subtle cultural norms and values which are effective at controlling personal behaviour. A participative approach is supported, but only with software developers and not project managers - the latter are seen to be biased in favour of shareholders. A higher price for software as a result of rewarding developers according to their efforts is considered inconsequential. Common to most theorists supporting this paradigm is the view that contemporary society is characterized by fundamental conflicts which generate radical change through political and economical crises. It is this very conflict which enables mankind to be emancipated from the social structures in which he lives. All analysis emphasizes structural conflict, contradiction, deprivation and modes of denomination. The stance taken when approaching these issues is generally of a realist/determinist/positivist or nomothetic.

The weaknesses of this approach include the acceptance of only a classless society. Conflict is seen as a better way forward than co-operation and negotiation. A state of conflict is preferred otherwise the status quo is considered to be satisfactory. The identification here of only 'software developers versus management' would be a simple reflection of the real world. Unions, group leaders, co-optation and differences between members of each all have an influence. All events are viewed as the results of evolutionary conflict.

A.3 Radical Humanism

'The truth will set us free' summarises this paradigm. This is thought possible through rational discourse, overcoming injustice and social domination - culminating in freedom and self-realization for all. This is similar to the Age of Enlightenment which freed us from unfounded dogmas and superstitions. Prevention comes in the form of physical constraints, social conventions and internal psychological compulsions. The latter include a weak personality - addressed with counseling and/or group therapy. From an ontological stance, this paradigm acknowledges three worlds: firstly, an external world of objects, events and processes; secondly, a world of language and shared meanings and lastly, a world of subjective meanings and emotional states for each individual. Compromise is acceptable as a conduit towards emancipation for all, i.e. on a global basis. Personal understanding comes from a person's background which is interpreted and changed continuously. This paradigm realizes the world is complicated and sometimes threatening, but believes consensual commitment is necessary to regulate all human affairs: individually, family, locally, nationally, internationally and globally. As such, radical humanism aims to remove any unwarranted restrictions on personal or social growth. Emphasis is placed on radical change, emancipation, deprivation, potentiality and modes of domination. Typical critical/emancipatory sciences include: critical reflection, psychoanalysis, historical reflection, analysis of ideology and comparative critique of literature. IS development is believed to be able to support emancipation, work and social interaction, but a software developer must be aware of the external compulsions: firstly, a good or bad societal island can be created by large companies and secondly, he must be able to work to realistic goals within imposed limits. Emancipation can be facilitated if major issues are raised resulting in an enlightened system. Organisational compulsions need to be overcome, such as (a) influence of prevailing organisational culture and any incentive system, (b) social differentiation, (c) information withheld or distorted to protect the use of hierarchical power, (d) mis-allocation of time preventing democratic access to information and, (e) peer opinion pressure suppressing validity checks and possible criticism.

It is possible under this paradigm for legitimate system objectives to be obtained from open rational discourse. This is an ideal speech situation which is founded on reason and free from all socially initiated pressures. Ideally, participation from all relevant parties takes place, but the principle obstacle to this is hierarchical power. Radical humanism has three fundamental IS development objectives: continued emancipation (anti-positivism), better mutual understanding (anti-positivism) and improved technical control (positivism). The epistemology of truth is what prevails - maximally consistent with evidence available and least inconsistent with maximal criticism. The ontological stance is of (socially constructed) nominalism and realism, respectively. Emphasis is on what could be, as against what is - as with Functionalism. A change in modes of cognition and consciousness is sought, in an attempt to change the world. The goal is the release of consciousness and experience from the domination of various aspects of the ideological superstructure of the social world in which we live. This is believed to be possible via a subjectivist view towards change. As a consequence, theorists have created a (nascent) anti-organisation theory.

Appendix B. IEEE/ACM Code of Ethics for Software Developers

B.1 Introduction

The IEEE/ACM Code of Ethics and Professional Conduct for Software Developers was chosen for this research as the IEEE is one of the oldest and largest professional organisations for computing professionals - dating back to 1946. The IEEE Computer Society is dedicated to advancing the theory, practice and application of computer and information processing technology. The Society promotes an active exchange of information, ideas and technological innovation among its members through its conferences, journals, distance learning campus, technical committees and standards working groups. Over 40% of its members live and work outside the US. Further information about the IEEE Computer Society can be found at http://www.computer.org/csinfo. The Code is produced in two formats, a short version and full version. Both versions are provided below.

B.2 Short Version

Principle 1: PUBLIC. Software engineers shall act consistently with the public interest. In particular, software engineers shall, as appropriate:

1.01. Accept full responsibility for their work

1.02. Moderate the interests of the software engineer, the employer, the client and the users with the public good

1.03. Approve software only if they have a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish quality of life, diminish privacy or harm the environment. The ultimate effect of the work should be to the public good

1.04. Disclose to appropriate persons or authorities any actual or potential danger to the user, the public, or the environment, that they reasonably believe to be associated with software or related documents

1.05. Cooperate in efforts to address matters of grave public concern caused by software, its installation, maintenance, support or documentation

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

DODD, S., LYCETT, M. 2003: *Ethical frameworks and software project management: a case study to identify the role and importance to practitioners of the IEEE/ACM Code of Ethics and Professional Conduct for Software Developers*. UKAIS Conference 'Co-ordination and Co-opetition: the IS role', Warwick, UK

DODD, S., LYCETT, M. 2002: *Ethical frameworks and software project management: acase study to identify constraints on their implementation*. UKAIS Conference 'Information Systems Research, Teaching and Practice', Leeds, UK

DODD, S., LYCETT, M. 2001: *Ethical considerations in software development: a review*. UKAIS Conference 'Information Systems - in the Digital World', Portsmouth, UK

DOZIER, J., HUSTED, B., McMAHON, J., KATTAN, M. summer 1996: The impact of cross-national carriers of business ethics on attitudes about questionable practices and forms of moral reasoning. *Journal of international business studies*. Vol 27 no 2, 391. 7 August 2000, http://web6.infotrac.london.galegroup.cdyn=50!xrn?_17_0_A18622680?sw_aep=bui.

ELZER, P. 1988: Management of large software projects in control. *Software Engineering of Large Scale Process Control Systems*. Proceedings of the 14th Annual Advanced Control Conference, Indiana, 1-13

FAIRWEATHER, B. 2004: Commentary on the 'Ten commandments for computer ethics'. www.ccsr.cse.dmu.ac.uk/resources/professionalism/codes/cei_command_com.html

FELDMAN, M. MARCH, J. 1981: Information in organisations as signal and symbol. *Administrative Science Quarterly,* 26, number 2, pp171-186.

FLODER, R., WEINER, S. 1983: Rationality to ritual: the multiple roles of evaluation in governmental processes, *Evaluation Models: Viewpoints on Educational and*

STEPHENS, R. 2000: *Authenticity and ethics in IS: the limits of theoretical thinking*. In the Proceeding of the 5th UKAIS Conference, Cardiff, eds. P. Benyon-Davies, M. Williams and I. Beeson. Pp341-348. McGraw-Hill, England

SVENSEN, N. 1998: Knowledge sharing. *Knowledge Management*, Issue 6 Vol 1, June/July, pp21-24

SWECC 2000 (SoftWare Engineering Co-ordinating Committee): *software engineering code of ethics and professional practice*. http://church.computer.org/tab/swecc/code.html, 11 April 2000

SWETNAM, D. 1998: *Writing your dissertation*. How To Books Ltd, Oxford UK. 2nd Ed.

SYMONS, V. 1990: *Evaluation of information systems: multiple perspectives.* Unpublished PhD thesis, University of Cambridge.

SYMONS, V., WALSHAM, G. 1988: The evaluation of information systems: a critique, *Journal of Applied Systems Analysis,* 15, pp119-132.

TASHAKORRI, A., TEDDLIE, C. 1998: *Mixed methodology: combining qualitative and quantitative approaches*. Sage Publications, UK

TAYLOR, N. 1999: The Guide to the Professional Conduct of Solicitors, 8th ed., Law Society Publishing, London

TELCO, January 6th, 2001: TV advert, channel 4, 8.40pm.

THEMISTOCLEOUS, M., May 31, 2000. Centre for Living Information Systems Thinking (CLIST) workshop

THOMPSON, N. 2003: best practices and context driven: building a bridge. International conference on software testing, analysis and review. Stareast 2003, Florida, USA.

1.07. Consider issues of physical disabilities, allocation of resources, economic disadvantage and other factors which can diminish access to the benefits of software

1.08. Be encouraged to volunteer professional skills to good causes and to contribute to public education concerning the discipline

Principle 2: CLIENT AND EMPLOYER. Software engineers shall act in a manner that is in the best interests of their client and employer, consistent with the public interest. In particular, software engineers shall, as appropriate:

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

2.02. Not knowingly use software that is obtained or retained illegally or unethically

2.03. Use the property of a client or employer only in ways properly authorised, and with the client's or employer's knowledge and consent

2.04. Ensure that any document upon which they rely has been approved, when required, by someone authorised to approve it

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

2.06. Identify, document, collect evidence and report to the client or employer promptly, if, in their opinion, a project is likely to fail, to prove too expensive, to violate intellectual property law, or otherwise to be problematic

2.07. Identify, document, and report significant issues of social concern, of which they are aware, in software or related documents, to the employer or the client

2.08. Accept no outside work detrimental to the work they perform for their primary employer

2.09. Promote no interest adverse to their employer or client, unless a higher ethical concern is being compromised; in that case, inform the employer or another appropriate authority of the ethical concern

Principle 3: PRODUCT. Software engineers shall ensure that their products and related modifications meet the highest professional standards possible. In particular, software engineers shall, as appropriate:

3.01. Strive for high quality, acceptable cost and a reasonable schedule, ensuring significant tradeoffs are clear to and accepted by t eh employer and the client, and are available for consideration by the user and the public

3.02. Ensure proper and achievable goals and objectives for any project on which they work or propose.

3.03. Identify, define and address ethical, economic, cultural, legal and environmental issues related to work projects

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

3.05. Ensure that an appropriate method is used for any project on which they work or propose to work

3.06. Work to follow professional standards, when available, that are most appropriate for the task at hand, departing from these only when ethically or technically justified

3.07. Strive to fully understand the specifications for software on which they work.

3.08. Ensure that specifications for software on which they work have been well documented, satisfy the users' requirements and have the appropriate approvals

3.09. Ensure realistic quantitative estimates of cost, scheduling, personnel, quality and outcomes on any project on which they work or propose to work and provide an uncertainty assessment of these estimates

3.10. Ensure adequate testing, debugging, and review of software and related documents on which they work

3.11. Ensure adequate documentation, including significant problems discovered and solutions adopted, for any project on which they work

3.12. Work to develop software and related documents that respect the privacy of those who will be affected by that software

3.13. Be careful only to use accurate data derived by ethical and lawful means, and use it only in ways properly authorised

3.14. Maintain the integrity of data, being sensitive to outdated or flawed occurrence

3.15. Treat all forms of software maintenance with the same professionalism as new development

Principle 4: JUDGMENT. Software engineers shall maintain integrity and independence in their professional judgment. In particular, software engineers shall, as appropriate:

4.01. Temper all technical judgments by the need to support and maintain human values

4.02. Only endorse documents either prepared under their supervision or within their areas of competence and with which they are in agreement

4.03. Maintain professional objectivity with respect to any software or related documents they are asked to evaluate

4.04. Not engage in deceptive financial practices such as bribery, double billing, or other improper financial practices

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

4.06. Refuse to participate, as members or advisors, in a private, governmental or professional body concerned with software related issues, in which they, their employers or their clients have undisclosed potential conflicts of interest

Principle 5: MANAGEMENT. Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance. In particular, those managing or leading software engineers shall, as appropriate:

5.01. Ensure good management for any project on which they work, including effective procedures for promotion of quality and reduction of risk

5.02. Ensure that software engineers are informed of standards before being held to them

5.03. Ensure that software engineers know the employer's policies and procedures for protecting passwords, files and information that is confidential to the employer or confidential to others

5.04. Assign work only after taking into account appropriate contributions of education and experience tempered with a desire to further that education and experience

5.05. Ensure realistic quantitative estimates of cost, scheduling, personnel, quality and outcomes on any project on which they work or propose to work, and provide an uncertainty assessment of these estimates

5.06. Attract potential engineers only by a full and accurate description of the conditions of employment

5.07. Offer fair and just remuneration

5.08. Not unjustly prevent someone from taking a position for which that person is suitably qualified

5.09. Ensure that there is fair agreement concerning ownership of any software, processes, research, writing, or other intellectual property to which a software engineer has contributed

5.10. Provide for due process in hearing charges of violation of an employer's policy or of this Code

5.11. Not ask a software engineer to do anything inconsistent with this Code

5.12. Not punish anyone for expressing ethical concerns about a project

Principle 6: PROFESSION. Software engineers shall advance the integrity and reputation of the profession consistent with the public interest. In particular, software engineers shall, as appropriate:

6.01. Help develop an organisational environment favourable to acting ethically

6.02. Promote public knowledge of software engineering

6.03. Extend software engineering knowledge by appropriate participation in professional organisations, meetings and publications

6.04. Support, as members of a profession, other software engineers striving to follow this Code

6.05. Not promote their own interest at the expense of the profession, client or employer

6.06. Obey all laws governing their work, unless, in exceptional circumstances, such compliance is inconsistent with the public interest

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

6.08. Take responsibility for detecting, correcting, and reporting errors in software and associated documents on which they work

6.09. Ensure that clients, employers, and supervisors know of the software engineer's commitment to this Code of ethics, and the subsequent ramifications of such commitment

6.10. Avoid associations with businesses and organisations which are in conflict with this code

6.11. Recognise that violations of this code are inconsistent with being a professional software engineer

6.12. Express concerns to the people involved when significant violations of this Code are detected unless this is impossible, counter-productive or dangerous

6.13. Report significant violations of this code to appropriate authorities when it is clear that consultation with people involved in these significant violations is impossible, counter-productive or dangerous

Principle 7: COLLEAGUES. Software engineers shall be fair to and supportive of their colleagues. In particular, software engineers shall, appropriate:

7.01. Encourage colleagues to adhere to this Code
7.02. Assist colleagues in professional development
7.03. Credit fully the work of others and refrain from taking undue credit
7.04. Review the work of others in an objective, candid, and properly documented way
7.05. Give a fair hearing to the opinions, concerns or complaints of a colleague
7.06. Assist colleagues in being fully aware of current standard work practices including policies and procedures for protecting passwords, files and other confidential information, and security measures in general
7.07. Not unfairly intervene in the career of any colleague; however, concern for the employer, the client or public interest may compel software engineers, in good faith, to question the competence of a colleague
7.08. In situations outside their own areas of competence, call upon the opinions of other professionals who have competence in that area

Principle 8: SELF. Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession. In particular, software engineers shall continually endeavour to:

8.01. Further their knowledge of developments in the analysis, specification, design, development, maintenance and testing of software and related documents, together with the management of the development process
8.02. Improve their ability to create safe, reliable, and useful quality software at reasonable cost and within a reasonable time
8.03. Improve their ability to produce accurate, informative, and well-written documentation
8.04. Improve their understanding of the software and related documents on which they work and of the environment in which they will be used
8.05. Improve their knowledge of relevant standards and the law governing the software and related documents on which they work

8.06. Improve their knowledge of this Code, its interpretation, and its application to their work

8.07. Not give unfair treatment to anyone because of any irrelevant prejudices

8.08. Not influence others to undertake any action that involves a breach of this Code

8.09. Recognise that personal violations of this Code are inconsistent with being a professional software engineer

B.3 Full Version

ACM Code of Ethics and Professional Conduct

Adopted by ACM Council 10/16/92.

From: http://www.ccsr.cse.dmu.ac.uk/resources/professionalism/codes/acm.html, (10-10-2000)

Preamble. Commitment to ethical professional conduct is expected of every member (voting members, associate members, and student members) of the Association for Computing Machinery (ACM).

This Code, consisting of 24 imperatives formulated as statements of personal responsibility, identifies the elements of such a commitment. It contains many, but not all, issues professionals are likely to face.Section 1 outlines fundamental ethical considerations, while Section 2 addresses additional, more specific considerations of professional conduct. Statements in Section 3 pertain more specifically to individuals who have a leadership role, whether in the workplace or in a volunteer capacity such as with organisations like ACM. Principles involving compliance with this Code are given in Section 4.

The Code shall be supplemented by a set of Guidelines, which provide explanation to assist members in dealing with the various issues contained in the Code. It is expected that the Guidelines will be changed more frequently than the Code.

The Code and its supplemented Guidelines are intended to serve as a basis for ethical decision making in the conduct of professional work. Secondarily, they may serve as a basis for judging the merit of a formal complaint pertaining to violation of professional ethical standards.

It should be noted that although computing is not mentioned in the imperatives of Section 1, the Code is concerned with how these fundamental imperatives apply to one's conduct as a computing professional. These imperatives are expressed in a general form to emphasize that ethical principles which apply to computer ethics are derived from more general ethical principles.

It is understood that some words and phrases in a code of ethics are subject to varying interpretations, and that any ethical principle may conflict with other ethical principles in specific situations. Questions related to ethical conflicts can best be answered by thoughtful consideration of fundamental principles, rather than reliance on detailed regulations.

Contents

Guidelines

1. General Moral Imperatives.

As an ACM member I will ....

1.1 Contribute to society and human well-being.

This principle concerning the quality of life of all people affirms an obligation to protect fundamental human rights and to respect the diversity of all cultures. An essential aim of computing professionals is to minimize negative consequences of computing systems, including threats to health and safety. When designing or implementing systems, computing professionals must attempt to ensure that the products of their efforts will be used in socially responsible ways, will meet social needs, and will avoid harmful effects to health and welfare.

In addition to a safe social environment, human well-being includes a safe natural environment. Therefore, computing professionals who design and develop systems must be alert to, and make others aware of, any potential damage to the local or global environment.

1.2 Avoid harm to others.

"Harm" means injury or negative consequences, such as undesirable loss of information, loss of property, property damage, or unwanted environmental impacts. This principle prohibits use of computing technology in ways that result in harm to any of the following: users, the general public, employees, employers. Harmful actions include intentional destruction or modification of files and programs leading to serious loss of resources or unnecessary expenditure of human resources such as the time and effort required to purge systems of "computer viruses."

Well-intended actions, including those that accomplish assigned duties, may lead to harm unexpectedly. In such an event the responsible person or persons are obligated to undo or mitigate the negative consequences as much as possible. One way to avoid unintentional harm is to carefully consider potential impacts on all those affected by decisions made during design and implementation.

To minimize the possibility of indirectly harming others, computing professionals must minimize malfunctions by following generally accepted standards for system design and testing. Furthermore, it is often necessary to assess the social consequences of systems to project the likelihood of any serious harm to others. If system features are misrepresented to users, coworkers, or supervisors, the individual computing professional is responsible for any resulting injury.

In the work environment the computing professional has the additional obligation to report any signs of system dangers that might result in serious personal or social damage. If one's superiors do not act to curtail or mitigate such dangers, it may be necessary to "blow the whistle" to help correct the problem or reduce the risk. However, capricious or misguided reporting of violations can, itself, be harmful. Before reporting violations, all relevant aspects of the incident must be thoroughly assessed. In particular, the assessment of risk and responsibility must be credible. It is suggested that advice be sought from other computing professionals. See principle 2.5 regarding thorough evaluations.

1.3 Be honest and trustworthy.

Honesty is an essential component of trust. Without trust an organisation cannot function effectively. The honest computing professional will not make deliberately false or deceptive claims about a system or system design, but will instead provide full disclosure of all pertinent system limitations and problems.

A computer professional has a duty to be honest about his or her own qualifications, and about any circumstances that might lead to conflicts of interest.

Membership in volunteer organisations such as ACM may at times place individuals in situations where their statements or actions could be interpreted as carrying the "weight" of a larger group of professionals. An ACM member will exercise care to not misrepresent ACM or positions and policies of ACM or any ACM units.

1.4 Be fair and take action not to discriminate.

The values of equality, tolerance, respect for others, and the principles of equal justice govern this imperative. Discrimination on the basis of race, sex, religion, age, disability, national origin, or other such factors is an explicit violation of ACM policy and will not be tolerated.

Inequities between different groups of people may result from the use or misuse of information and technology. In a fair society,all individuals would have equal opportunity to participate in, or benefit from, the use of computer resources regardless of race, sex, religion, age, disability, national origin or other such similar factors. However, these ideals do not justify unauthorized use of computer resources nor do they provide an adequate basis for violation of any other ethical imperatives of this code.

1.5 Honor property rights including copyrights and patent.

Violation of copyrights, patents, trade secrets and the terms of license agreements is prohibited by law in most circumstances. Even when software is not so protected, such violations are contrary to professional behavior. Copies of software should be made only with proper authorization. Unauthorized duplication of materials must not be condoned.

1.6 Give proper credit for intellectual property.

Computing professionals are obligated to protect the integrity of intellectual property. Specifically, one must not take credit for other's ideas or work, even in cases where the work has not been explicitly protected by copyright, patent, etc.

1.7 Respect the privacy of others.

Computing and communication technology enables the collection and exchange of personal information on a scale unprecedented in the history of civilization. Thus there is increased potential for violating the privacy of individuals and groups. It is the responsibility of professionals to maintain the privacy and integrity of data describing individuals. This includes taking precautions to ensure the accuracy of data, as well as protecting it from unauthorized access or accidental disclosure to inappropriate individuals. Furthermore, procedures must be established to allow individuals to review their records and correct inaccuracies.

This imperative implies that only the necessary amount of personal information be collected in a system, that retention and disposal periods for that information be clearly defined and enforced, and that personal information gathered for a specific purpose not be used for other purposes without consent of the individual(s). These principles apply to electronic communications, including electronic mail, and prohibit procedures that capture or monitor electronic user data, including messages,without the permission of users or bona fide authorization related to system operation and maintenance. User data observed during the normal duties of system operation and maintenance must be treated with strictest confidentiality, except in cases where it is evidence for the violation of law, organisational regulations, or this Code. In these cases, the nature or contents of that information must be disclosed only to proper authorities. (See1.9)

1.8 Honor confidentiality.

The principle of honesty extends to issues of confidentiality of information whenever one has made an explicit promise to honor confidentiality or, implicitly, when private information not directly related to the performance of one's duties becomes available. The ethical concern is to respect all obligations of confidentiality to employers, clients, and users unless discharged from such obligations by requirements of the law or other principles of this Code.

2. More Specific Professional Responsibilities.

As an ACM computing professional I will ....

2.1 Strive to achieve the highest quality, effectiveness and dignity in both the process and products of professional work.

Excellence is perhaps the most important obligation of a professional. The computing professional must strive to achieve quality and to be cognizant of the serious negative consequences that may result from poor quality in a system.

2.2 Acquire and maintain professional competence.

Excellence depends on individuals who take responsibility for acquiring and maintaining professional competence. A professional must participate in setting standards for appropriate levels of competence, and strive to achieve those standards. Upgrading technical knowledge and competence can be achieved in several ways:doing independent study; attending seminars, conferences, or courses; and being involved in professional organisations.

2.3 Know and respect existing laws pertaining to professional work.

ACM members must obey existing local, state,province, national, and international laws unless there is a compelling ethical basis not to do so. Policies and procedures of the organisations in which one participates must also be obeyed. But compliance must be balanced with the recognition that sometimes existing laws and rules may be immoral or inappropriate and, therefore, must be challenged. Violation of a law or regulation may be ethical when that law or rule has inadequate moral basis or when it conflicts with another law judged to be more important. If one decides to violate a law or rule because it is viewed as unethical, or for any other reason, one must fully accept responsibility for one's actions and for the consequences.

2.4 Accept and provide appropriate professional review.

Quality professional work, especially in the computing profession, depends on professional reviewing and critiquing. Whenever appropriate,individual members should seek and utilize peer review as well as provide critical review of the work of others.

2.5 Give comprehensive and thorough evaluations of computer systems and their impacts, including analysis of possible risks.

Computer professionals must strive to be perceptive, thorough, and objective when evaluating, recommending, and presenting system descriptions and alternatives. Computer professionals are in a position of special trust, and therefore have a special responsibility to provide objective, credible evaluations to employers, clients, users, and the public. When providing evaluations the professional must also identify any relevant conflicts of interest, as stated in imperative 1.3.

As noted in the discussion of principle 1.2 on avoiding harm, any signs of danger from systems must be reported to those who have opportunity and/or responsibility to resolve them. See the guidelines for imperative 1.2 for more details concerning harm,including the reporting of professional violations.

2.6 Honor contracts, agreements, and assigned responsibilities.

Honoring one's commitments is a matter of integrity and honesty.For the computer professional this includes ensuring that system elements perform as intended. Also, when one contracts for work with another party, one has an obligation to keep that party properly informed about progress toward completing that work.

A computing professional has a responsibility to request a change in any assignment that he or she feels cannot be completed as defined. Only after serious consideration and with full disclosure of risks and concerns to the employer or client, should one accept the assignment. The major underlying principle here is the obligation to accept personal accountability for professional work. On some occasions other ethical principles may take greater priority.

A judgment that a specific assignment should not be performed may not be accepted. Having clearly identified one's concerns and reasons for that judgment, but failing to procure a change in that assignment, one may yet be obligated, by contract or by law, to proceed as directed. The computing professional's ethical judgment should be the final guide in deciding whether or not to proceed. Regardless of the decision, one must accept the responsibility for the consequences.

However, performing assignments "against one's own judgment" does not relieve the professional of responsibility for any negative consequences.

2.7 Improve public understanding of computing and its consequences.

Computing professionals have a responsibility to share technical knowledge with the public by encouraging understanding of computing, including the impacts of computer systems and their limitations. This imperative implies an obligation to counter any false views related to computing.

2.8 Access computing and communication resources only when authorized to do so.

Theft or destruction of tangible and electronic property is prohibited by <u>imperative 1.2</u> - "Avoid harm to others." Trespassing and unauthorized use of a computer or communication system is addressed by this imperative. Trespassing includes accessing communication networks and computer systems, or accounts and/or files associated with those systems, without explicit authorization to do so. Individuals and organisations have the right to restrict access to their systems so long as they do not violate the discrimination principle (see 1.4). No one should enter or use another's computer system, software, or data files without permission. One must always have appropriate approval before using system resources, including .rm57 communication ports, file space, other system peripherals, and computer time.

3. Organisational Leadership Imperatives.

As an ACM member and an organisational leader, I will ....

BACKGROUND NOTE: This section draws extensively from the draft IFIP Code of Ethics,especially its sections on organisational ethics and international concerns. The ethical obligations of organisations tend to be neglected in most codes of professional conduct, perhaps because these codes are written from the perspective of the individual member. This dilemma is addressed by stating these imperatives from the perspective of the organisational leader. In this context"leader" is viewed as any organisational member who has leadership or educational responsibilities. These imperatives generally may apply to organisations as well as their leaders. In this context"organisations" are corporations, government agencies,and other "employers," as well as volunteer professional organisations.

3.1 Articulate social responsibilities of members of an organisational unit and encourage full acceptance of those responsibilities.

Because organisations of all kinds have impacts on the public, they must accept responsibilities to society. Organisational procedures and attitudes oriented toward quality and the welfare of society will reduce harm to members of the public, thereby serving public interest and fulfilling social responsibility. Therefore, organisational leaders must encourage full participation in meeting social responsibilities as well as quality performance.

3.2 Manage personnel and resources to design and build information systems that enhance the quality of working life.

Organisational leaders are responsible for ensuring that computer systems enhance, not degrade, the quality of working life. When implementing a computer system, organisations must consider the personal and professional development, physical safety, and human dignity of all workers. Appropriate human-computer ergonomic standards should be considered in system design and in the workplace.

3.3 Acknowledge and support proper and authorized uses of an organisation's computing and communication resources.

Because computer systems can become tools to harm as well as to benefit an organisation, the leadership has the responsibility to clearly define appropriate and inappropriate uses of organisational computing resources. While the number and scope of such rules should be minimal, they should be fully enforced when established.

3.4 Ensure that users and those who will be affected by a system have their needs.

clearly articulated during the assessment and design of requirements; later the system must be validated to meet requirements.

Current system users, potential users and other persons whose lives may be affected by a system must have their needs assessed and incorporated in the statement of requirements. System validation should ensure compliance with those requirements.

3.5 Articulate and support policies that protect the dignity of users and others effected 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 personal dignity.

3.6 Create opportunities for members of the organisation to learn the principles and limitations of computer systems.

This complements the imperative on public understanding (2.7). Educational opportunities are essential to facilitate optimal participation of all organisational members. Opportunities must be available to all members to help them improve their knowledge and skills in computing, including courses that familiarize them with the consequences and limitations of particular types of systems.In particular, professionals must be made aware of the dangers of building systems around oversimplified models, the improbability of anticipating and designing for every possible operating condition, and other issues related to the complexity of this profession.

4. Compliance with the Code.

As an ACM member I will ....

4.1 Uphold and promote the principles of this Code.

The future of the computing profession depends on both technical and ethical excellence. Not only is it important for ACM computing professionals to adhere to the principles expressed in this Code, each member should encourage and support adherence by other members.

4.2 Treat violations of this code as inconsistent with membership in the ACM.

Adherence of professionals to a code of ethics is largely a voluntary matter. However, if a member does not follow this code by engaging in gross misconduct, membership in ACM may be terminated.

This Code and the supplemental Guidelines were developed by the Task Force for the Revision of the ACM Code of Ethics and Professional Conduct: Ronald E. Anderson, Chair, Gerald Engel, Donald Gotterbarn, Grace C. Hertlein, Alex Hoffman, Bruce Jawer, Deborah G. Johnson, Doris K. Lidtke, Joyce Currie Little, Dianne Martin, Donn B. Parker, Judith A. Perrolle, and Richard S. Rosenberg. The Task Force was organised by ACM/SIGCAS and funding was provided by the ACM SIG Discretionary Fund. This Code and the supplemental Guidelines were adopted by the ACM Council on October 16, 1992.

Last update
Wed Mar 11 1998

Ambiguity in the IEEE/ACM code is identified by the author as present. Ambiguity contained in The Guide is recognised and addressed per se by statements such as, 'for the avoidance of doubt it is hereby declared that in sections 20, 22 and 23 references to unqualified persons and to persons include references to bodies corporate'. The interpretation of the regulations is formalised for the Society by the Interpretation Act 1978. For example, a 'practice' means a solicitor's practice and includes a multi-national partnership, a recognised body, the in-house legal department of a non-solicitor and an overseas partnership or corporate practice of solicitors and other solicitors.

Several services are available to certified solicitors for personal and/or professional matters which may affect their work, with contact details included in The Guide. The Solicitors Assistance Scheme (SAS) provides volunteer solicitors to help with any problem. Callers are listened to, helped and advised. Callers can contact the service anonymously if they wish. The scheme is administered by the Professional Ethics department of the Law Society and assisted by local law societies. The members are independent, experienced and sympathetic solicitors. No charge is made for an initial interview. The Law Society's Support Group is a self-help group offering confidential help to solicitors with an alcohol or drugs problem and operates independently of the Law Society. Sol Care offers a free confidential advice for solicitors, their families and staff about stress, depression, alcohol, drugs, eating disorders and other health issues. SolCare is a registered charity independent of the Law Society. Support is clearly available to those solicitors who are experiencing difficulties. IT/IS professionals do not have similar help schemes perhaps other than their immediate colleagues or project managers.

C.2 European Law

Fees charges to clients by solicitors must be fair and reasonable. Percentage fees calculated on successful outcomes are not permitted to be arranged in the final stages of a case. This might prevent the best interests of justice from being sought. Fees charges are subject to regulation in the rules of the society to which he belongs. Where he belongs to more than one association, the one most closely connected to the case should be chosen. If a payment on account is required, this must not exceed a reasonable estimate of the fee to be charged. If the payment on account is not paid by a client, the solicitor may withdraw from the case – after safeguarding the client's interests as described above. Client's monies deposited with a solicitor must be repaid on demand or

Appendix C. Codes in Other Professions

Ethical codes of conduct implemented in other professions were researched to identify how their requirements were enforced. Additional information obtained is provided below for each of the professions: UK Law, European Law, the Retail Sector and the Financial Services Sector.

C.1 UK Law

Civil justice reforms impose additional requirements on practicing solicitors. Serving a client's best interests is tempered by a new requirement that cases should only be pursued in a way which is proportionate to the likely benefit. Courts will no longer allow every point to be pursued – some procedural tactics are no longer permitted.

The requirements of professional conduct arise from many statutory and non-statutory sources. Statutory sources include:

- Legislation such as the Solicitors Act 1974 and the Courts and Legal Services Act 1990
- Other statutes such as the Financial Services Act 1986
- Rules, orders and regulations made under statute such as the Solicitors' Practice Rules 1990
- Codes and guidance made under statutory rules such as the Solicitors' Publicity Code 1990
- Principles, rules and codes applied under statutory rules such as the Statement of Principle of the FSA

Non-statutory sources include:

- Law Society
- Compliance and Supervision Committee
- Solicitors Disciplinary Tribunal
- Common law
- Master of the Rolls

upon such actions as the client may authorise. Personal experience of the researcher has found this not to be the case in practice, where it was necessary to write and telephone his solicitor on many occasions to have his money repaid. It was returned after much frustration – clearly not reflecting the requirements of this Code. Commission can be paid but only where the client's best interests can be maintained. In the context of IS and this research, the data collected shows that commissions are paid to employees who clearly consider their financial gain over the requirements of the client. Solicitors fees are controlled and, referral fees are prohibited to prevent additional monies being received by a solicitor. Where a new solicitor is appointed to take over a case, he should not commence work on it until he is satisfied arrangements have been made for the previous solicitor to be paid what he is entitled to. The new solicitor does not become responsible for those fees. Again, the EU Code acknowledges that misunderstandings about responsibility for unpaid fees are a common cause of difference between solicitors of different member states. In an attempt to reduce misunderstandings between EU members, solicitors are encouraged to offer training to new solicitors from other states.

Professional indemnity insurance insures solicitors against claims based on professional negligence 'to an extent which is reasonable having regard to the nature and extent of the risks which solicitors incur in practice' (p216). He must seek extended insurance cover for work conducted in other EU members on his own policy or, obtain insurance in the host member. If this is not possible he must notify his client. Solicitors are never permitted to knowingly give false or misleading information to a court. Setting the interests of the profession against those of justice or those who seek it can never be justified.

Ambiguity exists in the term 'confidential' used in the codes, which, considering its importance in the conduct of practicing solicitors is interesting. The Law Society defines confidentiality as keeping private communication between a solicitor and his client. In many EU countries it is defined as keeping private communication between all the solicitors involved in a case – i.e. the non-sharing of information received by solicitors with respective clients. Communication between solicitors is not normally admissible by a court as evidence and, the solicitor submitting this type of evidence would be considered to be breaching the rules of professional conduct. An exception to this rule applied in Germany where the communication between solicitors can be shared with clients and submitted as evidence to a court. The EU recognise this and admit 'misunderstandings between solicitors often arise' (p219).

C.3 The Retail Sector

An international template for best practice created by the United Nations was used by M & S in creating their own key areas under which performance is monitored and pursued. The template is called The United Nations Global Reporting Initiative (GRI) was used by M & S as a basis for creating its CSR objectives.

The Global Reporting Initiative (GRI) is a multi-stakeholder process and independent institution whose mission is to develop and distribute globally applicable Sustainability Reporting Guidelines. These guidelines are for voluntary use by organisations for reporting on the economic, environmental and social dimensions of their activities, products and services. The GRI incorporates the active participation of representatives from business, accountancy, investment, environmental, human rights, research and labour organisations from around the world. Started in 1997 by the Coalition for Environmentally Responsible Economies (CERES), the GRI became independent in 2002 and is an official collaborating centre of the United Nations Environment Programme (UNEP) and works in cooperation with UN Secretary-General Kofi Annan's Global Compact.

C.4 Financial Services

The value of discussion of practical day-to-day examples is recognised. These can tease out the kind of issues that people have to face and how they might deal with them and the value judgements they consider in doing so. A variety of hypothetical situations were developed by the FSA to illustrate where it might it be more difficult to decide on the right course of action. The situations are intended to initiate thought and encourage a deeper understanding about the ethical behaviour expected of its members. Two others not included in the main thesis are provided below.

Scenario 2: You run a successful but small treasury operation for a bank with a good reputation for ethical practice. A close family friend is worried about his son's job prospects and asks you to take him on even though there are many more suitably qualified candidates for the advertised role. Question: What would you say to your friend?

Scenario 3: You are the journalist for the city department of an evening newspaper. Pickings are a bit light one day and your city editor tells you that he heard rumours of a big city takeover involving two large city banks. You explain that you have good, reliable contacts with these firms and have heard nothing. Your editor conceded that there really is no rumour and, nonetheless, that you run a speculative headline and are told to make it sound real. Questions: (a) What do you say to your boss? (b) If he does not listen, what do you do next?

Appendix D. The Use of Repertory Grids

D.1 Introduction

Much of the work below is taken from Norris (date unknown) and Robson (1996). The development of a new coding system (scheme) is both difficult and time consuming. As there are already a plethora of coding systems developed, tried and tested, it makes sense to use one already proven. The use of a version of Repertory Grids was considered to be most appropriate for this research, with a description and justification of this coding system given below.

Although many types of grid exist, such as rated grids, implication grids, interactional grids, resistance to change grids, dependency grids, etc., repertory grids are most suited to measuring and/or comparing attitudes of individuals and/or groups of individuals over a brief time period or at intervals or over any time period, though it has other uses in addition. Computer analysis of repertory grids is fast and interactive which may also prove beneficial. Its ability to measure and compare complex attitudes makes the techniques especially useful when small groups are being studied or there is a limitation on time available for the study. As such, repertory grids can provide a large amount of data in a structured format - e.g. transforming qualitative data into quantitative data - facilitating comparisons to be made and conclusions to be drawn. Grids can be used alone or in conjunction with other methods, such as interviews and questionnaires and, be used safely by relative newcomers to the technique. As such, repertory grids - in particular rank-order grids - are considered most suited to this research and, will be described in more detail below, particularly in relation to practical considerations, multi-dimensionality, reliability and validity, design and analysis.

D.2 The Multi-Dimensionality of Grids

Essential to the design of repertory grids is an element of bi-polarity, i.e. scales used offering a choice to a respondent of 'always, usually, never'. This differs to one dimensional scales frequently used, such as Likert scales, offering respondents the choice of 'strongly agree, agree, don't know, disagree, strongly disagree'. Categorizations are usually graded from positive to negative feelings - about something in the area of interest to the researcher. In repertory grids, each variable is a bi-polar categorization and, the variables (or constructs) are related. Many inter-related grids compose a complex 'construct world' from which every respondent, consciously or otherwise, evaluates the world and consequently acts - or is prevented from acting. Repertory grids then, focus on this dynamic aspect in making sense of the (respondent's) world. The results can be plotted on a three dimensional diagram, or used to make profiles or responses. Behaviour may be determined as to whether it varies between individuals whose evaluative ratings towards, for example, an object, are similar. The findings can be used to identify if concepts have different meaning for different individuals or groups of people. Grids can be used about objects, concepts, others or oneself. Therefore repertory grids can explore multiple dimensions of attitudes concurrently. Thereby grids offer dignity to the respondents taking part, due to the underlying theoretical assumptions, by obtaining their views based on their own personal meanings. In practice, this is the design of grids with no assumption about the number of dimensions from which the respondents construct their notion of reality and the research itself is not limited by a pre-specified number of dimensions.

D.3 Reliability and Validity

For repertory grids to be used, the researcher needs to be satisfied about the reliability and validity of the technique, to be able to make a comparison to others. Grids are no better or worse that other techniques, argues Kelly (1955), as both the researcher and respondents may attribute different meanings to words used in the grid. This is also applicable to questionnaires and interviews. Consequently, all these tests some other point of reference - such as observation or document analysis - to ascertain that they are measuring what is intended, to establish validity. Repertory grids are unique in demonstrating internal validity: associations within a grid can be demonstrated to have patterns - which are not random - which can be interpreted and which may also recur when the grids are replicated. Reliability, in the conventional sense, is often questioned with this technique, due to its dynamic nature. It is possible, however, to calculate the constancy of grid relationships over time and to show that some relationships change and others do not. Constancy in the creation of elements and constructs over time can also be shown. Kelly further argues, along with Fransella and Bannister (1977), that grids need to be considered somewhat differently from other, conventional, techniques.

D.4 Practical Considerations

Grids are usually completed by the respondents on a one-to-one basis with a researcher, but this can also be done in groups and also by a researcher himself from data collected, for example, via an interview. The length of time required to both complete the grid(s) and analysis the information provided needs to be considered before adopting this technique. The design of the grid itself varies according to its application, with a bespoke grid for each research project, although a general size used in clinical trials, for example, is a ten by ten matrix. A grid this size, being completed by a researcher as a respondent answers his questions, could be completed in about forty minutes. A fifteen by fifteen grid could take around ninety minutes to complete. No special abilities are required of a researcher using repertory grids, other than the ability to build a rapport with any respondents - if he is not completing the grid himself having already completed the data collection.

The practical benefits of using grids include flexibility and the ability to produce more subjective material than a questionnaire, for example - when used to collect data. Interviews can obtain nuances of meaning which can be shown in a grid and, coupled with the multi-dimensional nature of a grid, it can gather a huge amount of information in a relatively short amount of time - again, compared to a questionnaire, for example. The data provided reflects that obtained, or elicited, from meanings which are personal to the respondent(s) and, consequently a researcher's task of completing the grid is generally an enjoyable task. Grids are additionally beneficial in identifying redundant information, which can be quickly omitted/ignored from the analysis. As such, the design of a grid and the planning of analysis on the data it provides are carried out together. Grids can be analysis manually, although computer programs exist which expedite the process. Researchers with a mathematical background may find them more intuitive to use than researches from other disciplines.

The researcher of this research completed the grids himself from data collected via questionnaires and interviews as it was considered obtrusive for the respondent to complete them after each question put to them by the researcher and therefore, would impede the rapport with the respondent - resulting in 'poorer' data being collected.

D.5 Grid Design

The design of a repertory grid is based on a sorting procedure - in this case, the eight principles of the IEEE/ACM Code of Ethics and Professional Conduct for Software Developers. Data is recorded in a matrix, with the number of rows and columns needed reflecting the sorting method and the number of items to be sorted: eight principles consisting of approximately ten points indicate eight grids are necessary. The purpose of the grid is to enable the examination of the relationship between individual and personal ideas, concepts and 'constructs' obtained from respondents. Its form enables basic statistical tests to be performed on a particular area of interest. As each grid is bespoke - created specifically for each research project - there is no standard grid form.

The two essential components in grid design are the 'items' (called constructs) to be sorted and the bi-polar attributes (called elements) to categorise the items. Items have previously included: people, photographs, coastal towns, political parties, emotions and products. Bi-polar attributes have included: successful/unsuccessful and shy/confident. A grid enables the relationships between all constructs and all elements or, between any sets of elements and constructs, to be examined. For example, a pile of cards each containing the name of someone known to the researcher could be sorted into two sub-piles: one for confident people and one for shy people. These could then be sorted into two more sub-piles: one for successful people and one for unsuccessful people. If all the people in the shy pile were also in the unsuccessful pile on the second sort, there would be a perfect correlation for the researcher between the constructs of shyness and lack of success.

Constructs are usually adjectives or adjectival phrases, in practice, used as sorting categories. Sorting may be by allocation to the two poles of a construct, or constructs may be scales upon which elements can be scaled or graded. It is important that the researcher does not impose his own meanings or interpretations into the identification of constructs. (This is ensured in this research as they are taken from the eight principles within the IEEE/ACM Code of Ethics.) The elements can be provided by respondents or, as in this case, taken from interview transcripts, questionnaire responses and observation notes - to show consensual meaning, as closely as possible. Having established the constructs and elements the data then needs to be categorized, and ranking is usual, although data can also be graded or dichotomized. The use of cards is recommended, with the results transferred into a matrix once the process is completed. Ranking is the identification of an 'extreme' element and placing it at the head of a column (or row, depending on the matrix design). The opposite 'extreme' element is then identified and placed at the other end of the column, with the remaining elements inserted accordingly between the two opposite extremes. When all the elements have been ranked for each construct, the card numbers are entered, as raw data, into the matrix, in the form they are laid out. The process is repeated for each construct. Any difficulties in ranking the constructs suggests they might be inappropriate and may need replacing with synonyms to become relatable to the meanings used by respondents. In some cases the elements may just be inappropriate, such as smoker/non-smoker when ranked for a construct about babies. In this instance, it would be necessary to remove the element from the grid and place it accordingly. Where no difficulty in ranking is experienced, the construct should be kept as it has meaning to the respondent.

The research aim objectives are to: review the value and importance of ethical behaviour in the context of software development; analyse the IEEE/ACM Code of Ethics as a framework able to facilitate decision-making from an ethical viewpoint; examine ethical software development in practise and, reflect upon the differences found between ethical theory and practise in order to improve an ethical approach to software development. As such, the grid design used supplied constructs of interest from the IEEE/ACM code, elicited constructs (from the interview/questionnaire data collected) to establish the justification of the supplied constructs - and also to identify differences between the different software development teams, and business units studied and, elements of 'self'. The points identified in the data collected and included refer to examples of compliance and/or non-compliance to the Code, which may have been carried out either by the interviewee, for example, or another person within Telco to whom the interviewee is referring.

D.6 Grid Analysis

As the purpose of every grid is unique, guidelines on analysis are vague, just as they are on their design. A major strength in the use of grids is their flexibility and adaptability, but this leads to their weakness. This versatility prevents the creation of a single all all-purpose grid or single set method for analysis. The rank-order grid described here is the often used by academics and clinicians, but can still be analysed in a number of ways. For example, the grid could be analysed by hand, or by a computer program such as SPSS (a Statistical Package for the Social Sciences) for factor or principal component analysis. Clearly the nature and extent of any analysis undertaken varies according to the area of interest.

When a grid is designed to be used as a measure, results should be predicted before it is administered. Hypotheses can then be examined when the analytical results are known showing (or otherwise) matrix correlations to establish whether anticipated relationships exist. Computer programs can generate so much data that a researcher may be tempted to make unscientific post-hoc explanations of sets of relationships. (The hypotheses here are described in Chapter 2.)

It is difficult, however, to compare individuals in absolute terms due to the size of their own construct world. One person may have minute size world whereas another may have a loosely constructed huge world. It is also possible to analyse by hand by dichotomizing, by using ticks and blanks for example. Constructs can be dealt with in pairs, and the ticks and blanks counted for elements allocated to each of the poles provided.

For the purposes of this research, data collected in the case study will be analysed in a series of grids similar to repertory grids. They are similar in that they have a horizontal axis consisting of elements, a vertical axis consisting of constructs and are two dimensional. The use of flexible grids is recommended as being highly valuable and as such, is one of its great strengths (Ryle 1975). The differences between are:

1. The constructs to be used are predefined by the eight principles of the IEEE/ACM Code of Ethics, and not by the project manager and software developers (who make up the elements), as would normally be the case. This variant is necessary as one objective of the research is to identify adherence, or not, to the IEEE/ACM Code *per se*,
2. The elements consist of the project manager and the software developers in his team. Therefore the manager cannot choose the people to be elements as is normally the case - here they are determined by those developers in his team,
3. Construct 5 covers Principle 5 of the IEEE/ACM Code which was created only for project managers, and not all elements on the horizontal axis of the grid, i.e. the software developers. These elements, the software developers, are known as 'non-applicable'.

Although the constructs are predetermined in this instance, it is not uncommon and is known to facilitate analysis. An objective of the research may be to try and understand someone else's way of making sense of the world, which might make the use of predetermined constructs appear inappropriate. The possible scope, however, of areas covered by all participants would be huge and certainly beyond the scope of this research, if, in deed, the research objectives were able to be met at all from the eclectic data collected (Ryle 1975). A series of grids are needed as four other business units were included in the research, in addition to the principle software team and three others, making a total of eight units in all. Eight grids will therefore be used. In each of the cells will be entered the codes identifying the type and location of data collected as evidence of adherence or non-adherence, as appropriate. An example of the grid structure to be used is shown in Table 24 below. The structure used might appear simple when compared to other repertory grid structures, but with approximately 80 constructs to be used (8 principles each covering approximately 10 issues) as against the recommended maximum of 25, the ultimate analysis needs to be manageable (Ryle 1975). The code shown in the first data cell shows the evidence to support this is from an interview (I) with a manager whose initials are JF, and can be found on page 20 at line 4. (Obviously the initials entered into the grids are coded to preserve the confidentiality agreement with the participants.) The final digit of the code represents the importance of the evidence in supporting or not, the adherence to the principle, with 1 being least important to 3 being most important.

	Project Manager	Developer 1	Developer 2	Developer 3	Grand Total
Principle 1.1 Adherence	I-GA-20-4-1		D-23-14-1		1
Principle 1.1 Non-Adherence					
Principle 1.2 Adherence					
Principle 1.2 Non-Adherence					

Table 24. The Grid Structure for Data Analysis

Analysis of grids will identify individual project team adherence, or non-adherence, to the Code of Ethics. However, grids are frequently compared, and this is done in its simplest form by direct comparison of particular constructs or by inspection, both of which can be done manually by a researcher. A computer program will not be used in this instance as the constructs and elements are not the same for all grids, which is a prerequisite - and is also its weakness. Three other business units are included in the research, in addition to the four software development teams, and therefore not only are the elements are different but the constructs also need transposing to each unit's respective professions. This was achieved during the interviews when the questions were reworded by the researcher to be appropriate to the interviewees, for each business unit. Another concern was that the participants would not admit to being unethical, just as one would not admit to beating one's wife (interview with TW). Therefore the participants were not asked directly if, for example, they lie at work. This concern is supported by Ryle (1975) and was overcome by asking 'softer' questions and comparing interview transcripts to identify consistency. This supports the argument that repertory grids can be employed equally well in other areas regarding practices and theories, to those areas identified by Kelly.

There is no general validation available for repertory grid analysis. Due to its individual and adaptable nature, an extensive range of data is available and is open to interpretation in a number of ways, in addition to the technique being used for a variety of purposes. The interview transcripts were analysed and recorded by both frequency and weighting. The weight scale to be used is a three-point Likert scale: 3 for Code adherence/important, 2 for relevance, and 3 for interest. The sense to be gained from a repertory grid approach is largely common sense, although this might only be a deceiving label for the researcher's preconceptions (Ryle 1975). As such, more importance is placed on the triangulation of data collected, as advocated by Yin (1984, Yin 1994), to ensure a rigorous case study.

The tables used to capture the data not included in the IEEE/ACM Code represented the headings generated from the identified themes in the interview transcripts. The points identified under each heading were entered into the tables in such a way as to allow negative incidences or non-truths of each to be recorded as a negative factor, i.e. represented with a 'N', and conversely a 'Y' for when statements were found to be true or incidents occurred. This deliberate manipulation of the wording to describe each point was considered necessary to facilitate subsequent analysis of the tables.

D.7 Statistical Scales

In order to make an informed decision about the appropriateness of the use, or non-use, of statistical methods in this research, it was necessary to know about them first. Much of the work described below was taken from Galliers (1992).

As described in earlier in the chapter, a questionnaire was initially used, but then discarded after refining the direction of the research. The (predominant) data collected from transcribing the interview transcripts and log book were entered into a version of repertory grids and so some statistical knowledge was required. This was not only necessary for the analysis of the data once collected, but also for the construction of the initial questionnaire itself. Many questions required an answer in the form of a tick against a number or word forming part of what is called a measurement scale such as, for example, from 1 to 5 or from 'excellent' to 'poor'. All scales were at least one of the following four types: ratio, interval, nominal and ordinal.

Ratio measurement scales incorporate all the attributes of the other three scales plus the addition of a zero origin, representing the absolute amounts of any variable. Examples of this type include number of customers or sales expected or achieved. All statistical descriptive measures are applicable, as are all inferential techniques. A typical question of this type might request the number of hours worked on a particular activity per week.

Interval measurement scales incorporate the attributes of the nominal and ordinal scales plus the concept of equality. The mean average from arithmetic is used to identify any centre and the standard deviation is invariably used to measure dispersion.

Nominal measure scales are the easiest scales to use and are used probably more frequently than the other types. The assigned numbers or words have no value and are viewed only as labels. No order can be derived, nor distance relation or origin. Examples of nominal scales include 1 for green, 2 for blue, or A for male, B for female, etc. As such, when used in this way, nominal scales have no quantitative value. The researcher could also ask a closed question resulting in a Yes or No answer from a respondent. The data collected would then be nominal.

Ordinal measurement scales have been used in IS research in three main ways. Firstly, and most frequently of the three types, by ranking items such as greater or less than another. For example, a respondent might be asked to select an answer from Rarely, Usually and Always. Secondly, and fuelling a debate in some quarters, by creating other complex scales as a result of manipulating the rating giving to various items by questionnaire respondents. Thirdly, and most controversially, by rating characteristics. Here the researcher interprets ordinal measures with interval characteristics simultaneously. For example, a researcher might assign a number to an item to reflect its relative rating and then interpret these numbers to find relative differences.

D.8 Techniques for Data Analysis

Yin (1994) describes four main techniques for analyzing case study evidence, within an overall strategy of prioritizing what evidence to analyze and why it is relevant. Yin terms a descriptive case study as 'developing a case description' where the original purpose of a case study is descriptive, as is the case here and, where theoretical propositions may not be available. This strategy was adopted by the researcher as was considered to be most appropriate. As an alternative, where a hypothesis is proposed, Yin terms this strategy 'relying on theoretical propositions'. He also describes three other techniques for data analysis, but stresses they are incomplete on their own and if used, need to be carried out in conjunction with at least one of the four main techniques noted above - most likely at the embedded unit and case level, respectively. These three are the case survey approach, the analysis of embedded units and repeated observations. Each is briefly described below, following the four main techniques now described.

The four main techniques noted above are: pattern matching, explanation building, program logic models and time-series analysis. All four are (a) applicable to both single and multiple case studies and, (b) attempt to address one of the weaknesses of the case study research strategy: data analysis is least developed and difficult. Yin warns, however, of manipulating data unaware of bias being incorporated. The ultimate goal at this stage is to treat all data collected fairly, to produce compelling conclusions and to rule out alternative interpretations. Each is now briefly described to ascertain its suitability. (The terms 'pattern' and 'time series' are used and refer to chain of events and, over time, respectively.)

Pattern matching can strengthen the internal validity of a case study when a comparison of an empirically based pattern with an anticipated pattern coincide. Explanatory case studies may find the pattern relates to in/dependent variables, or both. For descriptive case studies, pattern matching is still relevant as long as the anticipated pattern from specific variables is defined before any data is collected. This research design is more specifically termed a 'non-equivalent, dependent variables design' due to its inclusion of multiple independent variables - i.e. a variety of outcomes. This is because the following (each representing different dependent variables) can be derived from the original hypothesis:

1. Ethics will be viewed as a scalable attribute by developers and project managers
2. The project manager will consistently consider ethics when making decisions
3. Team members will carry out their tasks with a greater concern for ethical issues
4. Current problems in IS will be reduced as a result of this increased concern for ethics
5. The IEEE/ACM Code will be evaluated as an ethical framework
6. The IEEE/ACM Code will be perceived as a valuable framework by project managers and their respective team members

These outcomes were assessed with different measures and techniques. This identified non-equivalent dependent variables. If the results are as predicted here, a solid conclusion can be drawn about the effects of the IEEE/ACM Code and the alignment of desires of the project managers, their software developers and other business units. If the results are not as predicted, however, the proposition(s) would need to be questioned. This concern is reduced when the researcher is aware of the existence of threats to this logic, such as the replacement of key staff or external autonomous influences - such as mass redundancies, including those participating in the research (as was the case at Telco). This threat is accommodated by a subset of the initial dependent variable being created to refute any possible counter-argument(s) to the pattern(s) identified in the analysis. This type of pattern matching is appropriate here, as seen by the development of the propositions above, but two further types also appropriate are 'rival explanations' and 'simpler patterns'. What is important though, is acknowledging the degree of precision of pattern matching. As there are no precise pattern matching procedures for comparing fundamentals, an amount of interpretive discretion asserted by the researcher when comparing predicted behaviour with actual behaviour, is allowed. Yin (1994) recommends only patterns identified with 'stronger than subtle' differences are postulated by researchers.

Explanation building attempts to analyze the case study data by developing an explanation about the case itself - its goal is not to conclude the study but to create and develop ideas for further study. The explanation consists of a set of causal links, similar to the independent variables used in pattern matching and, are usually complex and difficult to measure. Explanation building is commonly in narrative form and consequently cannot be precise, but can ultimately lead to recommendations for future policy changes or contribute to theory building. Although this analytical strategy is not well documented, it is known to be an iterative process. This iteration consists of: make a theoretical statement or proposition, compare findings of initial case, revise the statement or proposition, compare other details of the case, revise again, compare findings of further cases, repeat as necessary. This strategy differs from pattern matching in that the final proposition(s) or statement(s) may not have been known in full at the outset and, is also considered one of the more 'dangerous' strategies for analyzing case study evidence.

Program logic models are a combination of time series analysis and pattern matching. The pattern being matched is the main cause-effect pattern between independent and dependent variables. The analysis of a complex chain of events (the pattern) over a period (time series) results in the linking of all the claimed existence of cause and effect sequences - the more complex the better - strengthening, or otherwise, the hypothesis of a pattern match with the events covered over time.
Three other types of data analysis techniques can be used, argues Yin (1994), but if selected, this should be done with at least one of the main four techniques above - most likely at the unit and case level, respectively. These three 'lesser' techniques are: repeated observations, analysis of embedded units and the case survey approach. Each relevant technique to the research is now described.

Repeated observations are a special type of time series analysis as although carried out over time, the repeated observations can be made on a cross sectional basis, i.e. on the project team, management and HR staff. Although demanding on resources and the ability to collect a large amount of data exists, repeated observations remains a 'lesser' research approach as it cannot readily focus on all the concerns of any particular case study.

Analysis of embedded units is relevant here as the research is focused on embedded units in the case study. These include the responses from questionnaires completed by the project manager and software developers. Although analysis of embedded units is important, caution must be given, however, to keep the main interest on the case being studied as a whole – rather than allowing any embedded unit to become most important. This strategy can get complex when analyzing embedded units across a multiple case study, but that is not the case here, with a single case study and, its use is appropriate.

Whichever analytical approach is adopted, Yin (1994) identifies four key principles to ensure high quality data analysis:

1. All relevant data is analyzed
2. All major rival interpretations are analyzed
3. The analysis concentrates on most important aspect of the case study
4. The researcher should include his own experience(s) and knowledge where appropriate

Finally, Yin recommends that analysis should be seen to be carried out with expertise. For example, use care in the organisation and presentation of the case and facilitate readability of the narrative given. This will help in addition to any methodology and technique(s) used, which he argues are difficult to use, making data analysis the hardest stage of conducting a case study. He warns novice researchers to expect trouble.

Observers can develop narrative accounts whilst on a case study, but it could be difficult for a researcher to be objective; a better approach might be the use of a coding scheme (Robson, 1996). Reliability and validity are still essential issues with coding schemes. Reliability is achieved when several observers are able to develop the same, or similar, coding scheme when presented with a situation. For the purposes of this researcher, reliability is ensured with the use of a coding scheme - the eight principles - created previously and used in the IEEE/ACM Code of Ethics. As the researcher is working alone, no other researchers are available to create coding schemes of their own to test reliability of the observations. This is also desired as the case study almost dictates unstructured observations, compared to the structured observations carried out on relative certainties and simplified complexities presented in a lab experiment.

D.9 Allocation of Unethical Practices to Principles

As identified earlier, there are 3 generic areas of unethical practices in the workplace, i.e:

1. Six categories of questionable practices consisting of:
    - Project manager's or software engineer's behaviour affecting the company
    - Company or project management practices affecting software engineers
    - Software engineer's behaviour affecting other software engineers
    - Company practices affecting customers
    - Company practices affecting shareholders
    - Company practices affecting the general public and the community

2. Common worst practices consisting of:

    Unsafe working conditions, pollution and waste, sexual harassment, race discrimination, sex discrimination and favouritism

3. Current problems/unethical behaviour/bad practice consisting of:

    Hacking, Invasion of privacy, Supply of inaccurate/untimely information, Myopic viewpoint, Lost investments, Software released with known bugs, Programs copied for personal-use, Contractors hopping between jobs for the highest hourly rate, Contractors deliberately inserting erroneous code to ensure their contracts are extended, Inaccurate progress reports/time sheets submitted, Use of Internet for personal use, Unauthorized access to files, Programs deleted from networks, Bulk emails, Viruses, Website material copied, Sexual harassment, Sexual discrimination, Race discrimination and Unsafe working practices

Each will now be considered with data collected from the case study to either support or refute their existence, in the context of ethical practices in IS development. This is defined by the eight principles of the IEEE/ACM Code of Ethics for Software Developers with consequent identification of compliance or non-compliance to the areas covered by each of the principles. To facilitate brevity the three areas will be treated jointly as appropriate - due to their synonymity. The three areas above will therefore be merged and analysed as shown in Table 25 below:

Area Covered	Principle Heading	Principle Description
Software engineer's behaviour affecting other software engineers Sexual harassment	Principle 7: Colleagues	7. Developers will be fair and supportive toward their colleagues
Software engineer's behaviour affecting the company Hacking Supply of inaccurate/untimely information Programs copied for personal-use Contractors hopping between jobs for the highest hourly rate Contractors deliberately inserting erroneous code to ensure their contracts are extended Inaccurate progress reports/time sheets submitted Use of Internet for personal use Pornography Unauthorized access to files Programs deleted from networks Bulk emails Viruses Website material copied Plagiarism	Principle 2: Client and Employer Principle 3: Product Principle 4: Judgment Principle 6: Profession Principle 8: Self	2. Client and Employer. Engineers will always behave in the best interest of their employer and client – as long as it is consistent with public interest 3. Product. The highest professional standards possible will be sought by engineers for their products and their respective modifications 4. Judgment. Engineers will exercise integrity and independence when making professional judgments 6. Profession. The integrity and reputation of the profession will be increased - consistent with public interest 8. Self. Engineers will take part in life-long learning of the profession's practices and promote an ethical approach to the practices used
Project manager's behaviour affecting the company Company or project management practices affecting software engineers Race discrimination Sex discrimination Favouritism Unsafe working conditions Invasion of privacy Company practices affecting shareholders	Principle 5: Management	5. Management. An ethical approach will be promoted and exercised by engineers in the management of software development and maintenance
Company practices affecting customers Myopic viewpoint Lost investments Deals disrupted Businesses destroyed Software released with known bugs Company practices affecting the general public and the community Pollution and waste	Principle 1: Public	1. Public. Software engineers shall always behave appropriately to public interest (health, safety and welfare)

Table 25. Principles Covering Area to be Analysed

D.10 Repertory Grids from the Case Study

Many repertory grids were created and used to store and organise data collected from the case study. This process enables a large amount of qualitative data to be analysed quantitatively leading to conclusions and recommendations. The repertory grids included here contain the data collected for (1) all level 3 non-compliance, (2) most frequent non-compliance, (3) all non-compliant data collected for the main project team participating in the research and, (4) all non-compliance data for Telco as a whole.

D.10.1 Level 3 Non-Compliant Data Collected

The total number of incidents found in the case study of non-compliance to the IEEE/ACM Code of Ethics and with the highest level of importance - Level 3 - was collected in a repertory grid and can be found in Figures 5, 6 and 7 below.

Principles				Teams					Total
	DT1	DT2	DT3	DT4	Quality	Recruitment	HR	Cisco	
Management									
5.01 N	7			2		1	1	1	**12**
5.02 N									
5.03 N									
5.04 N	1	1		1		3	1		**7**
5.05 N	5			1			1	1	**8**
5.06 N	2	1		1					**4**
5.07 N			1			1	3		**5**
5.08 N									
5.09 N								1	**1**
5.10 N									
5.11 N									
5.12 N									
Total	**15**	**2**	**1**	**5**		**5**	**6**	**3**	**37**
Profession									
6.01 N				1					**1**
6.02 N		2							**2**
6.03 N	4	3	1	5		1			**14**
6.04 N									
6.05 N	1		1	1					**3**
6.06 N	3	1				1			**5**
6.07 N									
6.08 N	1		1						**2**
6.09 N									
6.10 N								2	**2**
6.11 N									
6.12 N									
6.13 N									
Total	**9**	**6**	**3**	**7**		**2**		**2**	**29**
Colleagues									
7.01 N									
7.02 N	2	1							**3**
7.03 N									
7.04 N	1	1							**2**
7.05 N	1			1	2		1		**5**
7.06 N				3					**3**
7.07 N									
7.08 N							1		**1**
Total	**4**	**2**		**4**	**2**		**2**		**14**

Figure 6. Summary of Level 3 Non-Compliance Found

Principles				Teams					Total
	DT1	DT2	DT3	DT4	Quality	Recruitment	HR	Cisco	
Self									
8.01 N	1	1	1			1		1	5
8.02 N	1			1					2
8.03 N	1	2							3
8.04 N	1								1
8.05 N	3		1	1		1			6
8.06 N									
8.07 N									
8.08 N									
8.09 N									
Total	7	3	2	2		2		1	17

Figure 7. Summary of Level 3 Non-Compliance Found

D.10.2 Most Frequent Non-Compliant Data Collected

The number of incidents found in the case study of non-compliance to the IEEE/ACM Code of Ethics was collected in a repertory grid in order of frequency and can be found in Figure 8 below.

Principle				Teams					Total
	DT 1	DT 2	DT 3	DT 4	Quality	Recruitment	HR	Cisco	
5.01 N	45			5		2	4	12	**68**
3.05 N	9	3	4	3	1	4		6	**30**
3.06 N	14		1	1	2	4	1	7	**30**
3.10 N	8	6	1	1	1			6	**23**
3.11 N	10	2	1	2				8	**23**
5.05 N	11		1	1	1		2	6	**22**
4.04 N	7				2	1	1	8	**19**
6.03 N	4	3	1	5		3	1	1	**18**
5.04 N	5	1		1	1	6	2		**16**
2.03 N	7							8	**15**
3.08 N	9	1		2				3	**15**
3.15 N	4		1	1		1		8	**15**
5.07 N	6	1	1		1	2	4		**15**
7.05 N	5			1	2		3	3	**14**
1.06 N	5	1			1		2	4	**13**
3.09 N	5			3			2	1	**11**
4.03 N	3			1	1	4		1	**10**
5.06 N	6	1		1			1	1	**10**
6.05 N	6		1	1			2		**10**
8.05 N	6		1	1		1	1		**10**

Figure 8. Summary of Most Frequent Incidents of Non-Compliance

D.10.3 Non-Compliant Data Collected for the Main Project Team

The number of incidents found in the case study - for the main development team to which the author was assigned - of non-compliance to the IEEE/ACM Code of Ethics was collected in a repertory grid in order of frequency and can be found in Figure 9 on the following pages. The incidents relating to compliance to the code are also provided to show the development team additionally exhibited ethical behaviours.

Figure 9. Main Project Team Details of Non-Compliance

Principles	Documentation	Log Book	Manager	M TOT	Developer 1	Developer 2	Developer 3	Developer 1	TM TOT	Total	3'S
Public											
1.01 Y	D-205-10-1			0					1	1	
1.01 N				0					0		
1.02 Y			I-GA-3-15-3, I-GA-4-15-3, I-GA-31-25-2	3		I-ML-1-27-2, I-ML-2-22-3			2	5	3
1.02 N				0					0		
1.03 Y				0			I-SY-3-9-2, I-SY-4-24-3, I-SY-6-4-3, I-SY-15-25-1		4	4	2
1.03 N				0					0		
1.04 Y				0					0		
1.04 N				0					0		
1.05 Y				0			I-SY-6-16-2		1	1	
1.05 N				0					0		
1.06 Y			I-GA-19-12-3	1			I-SY-14-21-2	I-GL-8-11-3, I-GL-10-29-3	3	4	3
1.06 N	D-5-5-2	L-22-20-2, L-102-10-2, L-160-10-2, L-171-30-2		0					5	5	
1.07 Y		L-16-10-2	I-GA-3-7-3, I-GA-19-28-3,	2				I-GL-17-25-2, I-GL-35-20-3	3	5	3
1.07 N		L-12-25-1		0					1	1	
1.08 Y	D-248-15-1, D-308-1-1			0				I-GL-17-28-1	3	3	
1.08 N				0					0		
C & E											
2.01 Y			I-GA-1-12-2, I-GA-6-9-2, I-GA-7-1-2, I-GA-7-24-3	4	I-TC-12-3-3			I-GL-29-12-3	2	6	3
2.01 N				0					0		
2.02 Y				0					0		
2.02 N				0					0		
2.03 Y				0					0		
2.03 N	D-343-25-1	L-47-20-2, L-63-30-3, L-94-1-2, L-152-15-2, L-174-20-3, L-177-20-1		0					7	7	2
2.04 Y				0			I-SY-2-1-3		1	1	1
2.04 N				0					0		
2.05 Y				0					0		
2.05 N				0					0		

Principles	Documentation	Log Book	Manager	M TOT	Developer 1	Developer 2	Developer 3	Developer 1	TM TOT	Total	3'S
2.06 Y		L-107-20-3	I-GA-20-5-3, I-GA-20-33-2	2			I-SY-1-26-3, I-SY-2-1-3, I-SY-14-21-3, I-SY-15-16-3	I-GL-10-32-2, I-GL-35-2-3	7	9	7
2.06 N	D-344-30-3			0					1	1	1
2.07 Y				0					0		
2.07 N				0					0		
2.08 Y				0					0		
2.08 N				0					0		
2.09 Y			I-GA-37-21-3, I-GA-41-33-3	2					0	2	2
2.09 N				0					0		
Product											
3.01 Y		L-137-30-1	I-GA-4-18-3, I-GA-8-1-3, I-GA-10-26-3, I-GA-17-29-3, I-GA-18-27-3, I-GA-19-28-2	6	I-TC-3-2-3	I-ML-5-33-3, I-ML-6-21-3			4	10	8
3.01 N		L-102-5-3		0					1	1	1
3.02 Y				0		I-ML-2-4-3	I-SY-26-14-3		2	2	2
3.02 N		L-74-15-2		0					1	1	
3.03 Y	D-70-1-1, D-154-1-1, D-345-30-1	L-60-10-1	I-GA-6-27-1	1					4	5	
3.03 N				0	I-TC-12-20-3				1	1	1
3.04 Y				0					0		
3.04 N				0					0		
3.05 Y		L-104-15-1		0			I-SY-10-1-3, I-SY-11-30-2, I-SY-14-5-3		4	4	2
3.05 N	D-36-2-1	L-3-30-2, L-47-10-3, L-100-1-1, L-104-5-2, L-108-5-3, L-125-15-2		0	I-TC-8-27-3			I-GL-34-26-3	9	9	4
3.06 Y	D-200-30-1, D-287-10-2, D-334-30-1	L-42-15-2	I-GA-9-6-3, I-GA-9-15-3, I-GA-11-20-1, I-GA-13-32-3, I-GA-22-4-3, I-GA-22-20-3	6			I-SY-1-28-3, I-SY-10-33-3	I-GL-3-8-3O, I-GL-14-15-3, I-GL-15-17-2, I-GL-27-33-3, I-GL-32-33-3	11	17	10

Principles	Documentation	Log Book	Manager	M TOT	Developer 1	Developer 2	Developer 3	Developer 1	TM TOT	Total	3'S
3.06 N	D-370-20-1	L-38-15-2, L-38-28-1, L-39-5-2, L-81-25-2, L-101-10-2, L-102-15-2, L-158-20-3		0	I-TC-3-1-3, I-TC-5-17-3	I-ML-7-26-3, I-ML-8-17-3	I-SY-11-20-3	I-GL-33-21-3	**14**	14	7
3.07 Y			I-GA-15-7-3	**1**	I-TC-2-19-3		I-SY-12-8-3, I-SY-24-18-3, I-SY-26-12-3		**4**	5	5
3.07 N		L-28-25-2		**0**					**1**	1	
3.08 Y			I-GA-15-8-3	**1**			I-SY-1-25-3, I-SY-2-1-3, I-SY-2-20-3		**3**	4	4
3.08 N	D-292-1-1, D-344-30-3	L-101-25-1, L-168-30-2	I-GA-10-32-3, I-GA-11-6-3, I-GA-15-18-3, I-GA-18-23-3, I-GA-19-31-2	**5**					**4**	9	5
3.09 Y				**0**			I-SY-20-25-3		**1**	1	1
3.09 N		L-31-25-2, L-33-15-2, L-49-20-2, L-50-12-1	I-GA-30-14-3	**1**					**4**	5	1
3.10 Y			I-GA-10-8-3, I-GA-10-28-3, I-GA-11-14-3, I-GA-11-28-3, I-GA-11-26-3, I-GA-17-23-3	**6**		I-ML-1-22-3	I-SY-8-31-3		**2**	8	8
3.10 N		L-28-20-3, L-79-2-2, L-99-25-2, L-101-30-1, L-103-10-2, L-109-3-1	I-GA-13-28-2,	**2**		I-ML-5-1-3			**6**	8	2
3.11 Y	D-36-10-2		I-GA-28-16-2	**1**			I-SY-11-7-3, I-SY-15-6-3		**3**	4	2
3.11 N	D-73-20-2, D-293-10-2, D-337-1-1	L-30-10-2, L-47-25-1, L-101-20-3, L-102-10-2, L-102-30-1, L-151-5-3		**0**	I-TC-5-17-3				**10**	10	3
3.12 Y				**0**					**0**		
3.12 N				**0**					**0**		
3.13 Y				**0**					**0**		
3.13 N				**0**					**0**		
3.14 Y				**0**					**0**		
3.14 N				**0**					**0**		

Principles	Documentation	Log Book	Manager	M TOT	Developer 1	Developer 2	Developer 3	Developer 1	TM TOT	Total	3'S
3.15 Y			I-GA-8-25-3, I-GA-17-2-3, I-GA-18-4	3		I-ML-4-20-2		I-GL-14-25-1	2	5	2
3.15 N	D-58-30-1	L-50-28-1, L-99-25-2, L-106-15-2		0					4	4	
Judgment											
4.01 Y			I-GA-5-6-3, I-GA-33-20-3	2					0	2	2
4.01 N		L-37-5-1		0					1	1	
4.02 Y			I-GA-6-5-2	1					0	1	
4.02 N				0					0		
4.03 Y		L-82-25-3	I-GA-5-10-3, I-GA-6-9-2, I-GA-7-1-2, I-GA-7-24-2, I-GA-28-11-1, I-GA-41-33-3	6			I-SY-7-15-2, I-SY-13-3-3, I-SY-13-24-3	I-GL-11-6-3, I-GL-32-3-3	6	12	7
4.03 N		L-80-25-2, L-161-10-2		0	I-TC-2-33-3				3	3	1
4.04 Y				0				I-GL-1-33-2	1	1	
4.04 N		L-3-20-3, L-12-30-2, L-20-30-1, L-73-30-3, L-82-1-3, L-137-5-2, L-141-20-1		0					7	7	3
4.05 Y			I-GA-33-24-3	1					0	1	1
4.05 N		L-11-10-1		0					1	1	
4.06 Y				0					0		
4.06 N				0					0		
Management											
5.01 Y	D-46-5-1, D-71-5-1, D-71-30-1, D-127-1-1, D-128-1-1, D-130-1-1, D-334-30-1, D-345-30-2	L-132-20-1	I-GA-2-7-3, I-GA-11-22-1, I-GA-33-13-3, I-GA-36-17-3, I-GA-37-22-3	5					9	14	4

Principles	Documentation	Log Book	Manager	M TOT	Developer 1	Developer 2	Developer 3	Developer 1	TM TOT	Total	3'S
5.01 N	D-2-1-1, D-71-30-1, D-73-1-1, D-115-10-2, D-205-1-1, D-229-1-1, D-285-5-3, D-289-1-2, D-309-2-1, D-311-15-2, D-315-5-2, D-370-30-1	L-10-30-2, L-12-20-2, L-20-15-2, L-22-25-2, L-26-25-2, L-26-28-2, L-33-25-2, L-37-28-1, L-38-20-1, L-42-7-1, L-44-10-1, L-44-12-2, L-55-1-2, L-83-5-1, L-84-28-3, L-96-15-3, L-97-12-1, L-98-1-3, L-98-28-3, L-104-30-1, L-107-15-1, L-127-28-1, L-129-30-2, L-138-25-2, L-157-30-1, L-158-20-1, L-158-20-2, L-161-25-2, L-164-25-1, L-168-25-1, L-175-5-1, L-177-15-3		**0**	I-TC-9-10-3				**45**	45	7
5.02 Y	D-114-1-3		I-GA-9-15-3	**1**					**1**	2	2
5.02 N				**0**					**0**		
5.03 Y				**0**					**0**		
5.03 N				**0**					**0**		
5.04 Y		L-53-10-1, L-103-5-1	I-GA-2-16-2	**1**					**1**	2	
5.04 N	D-57-5-2	L-97-5-2, L-128-15-1		**0**	I-TC-9-10-3, I-TC-11-8-2				**5**	5	1
5.05 Y		L-24-5-1	I-GA-30-18-2, I-GA-34-11-3,	**2**					**1**	3	1
5.05 N	D-31-5-3, D-74-1-3, D-149-5-3, D-341-15-2, D-344-5-1, D-372-1-1	L-44-12-2, L-74-7-3, L-79-5-2, L-81-20-2	I-GA-7-13-3	**1**					**10**	11	5
5.06 Y				**0**	I-TC-1-17-2				**1**	1	

Principles	Documentation	Log Book	Manager	M TOT	Developer 1	Developer 2	Developer 3	Developer 1	TM TOT	Total	3'S
5.06 N	D-288-15-2, D-343-10-3	L-97-2-2, L-97-28-2, L-99-5-1		**0**				I-GL-16-10-3	**6**	6	2
5.07 Y	D-80-5-1, D-324-5-2	L-53-12-1, L-57-20-1, L-106-5-1, L-127-30-1		**0**	I-TC-1-17-2				**7**	7	
5.07 N	D-80-5-1, D-205-1-1	L-10-25-2, L-11-18-2, L-53-20-1, L-152-2-2		**0**					**6**	6	
5.08 Y			I-GA-37-33-2	**1**					**0**	1	
5.08 N				**0**					**0**		
5.09 Y		L-101-10-1		**0**					**1**	1	
5.09 N		L-4-8-2		**0**					**1**	1	
5.10 Y				**0**					**0**		
5.10 N				**0**					**0**		
5.11 Y			I-GA-34-23-3, I-GA-38-7-3	**2**		I-ML-9-8-2			**1**	3	2
5.11 N				**0**					**0**		
5.12 Y			I-GA-38-7-2	**1**					**0**	1	
5.12 N				**0**					**0**		
Profession											
6.01 Y	D-248-5-1, D-338-10-1	L-3-20-1	I-GA-8-19-3, I-GA-23-20-3, I-GA-38-5-2	**4**				I-GL-14-22-3, I-GL-32-33-3	**4**	8	4
6.01 N	D-368-10-2			**0**				I-GL-9-17-1	**2**	2	
6.02 Y			I-GA-29-14-2, I-GA-29-22-3	**2**					**0**	2	1
6.02 N				**0**					**0**		
6.03 Y			I-GA-13-25-3, I-GA-39-13-3	**2**			I-SY-13-3-3	I-GL-19-21-3, I-GL-33-8-3, I-GL-35-2-3	**4**	6	6
6.03 N			I-GA-38-22-3	**1**	I-TC-1-8-3	I-ML-10-20-3	I-SY-1-8-3		**3**	4	4
6.04 Y				**0**					**0**		
6.04 N				**0**					**0**		
6.05 Y		L-148-5-2	I-GA-6-5-2, I-GA-34-9-3	**2**			I-SY-13-9-3	I-GL-36-1	**3**	5	2
6.05 N	D-43-5-1, D-59-1-1		I-GA-39-4-1	**1**	I-TC-5-17-3		I-SY-4-8-2, I-SY-4-17-2		**5**	6	1
6.06 Y	D-13-5-3		I-GA-5-6-3, I-GA-15-15-2, I-GA-21-30-2, I-GA-22-4-3, I-GA-32-6-3, I-GA-41-18-2	**5**				I-GL-31-25-3, I-GL-32-33-3, I-GL-35-20-3	**5**	10	7
6.06 N	D-13-20-3	L-137-25-2, L-177-30-1	I-GA-41-20-3	**0**				I-GL-33-21-3	**5**	5	3
6.07 Y			I-GA-30-18-2	**1**				I-GL-10-32-2	**1**	2	

Principles	Documentation	Log Book	Manager	M TOT	Developer 1	Developer 2	Developer 3	Developer 1	TM TOT	Total	3'S
6.07 N				0					0		
6.08 Y		L-98-10-1		0			I-SY-14-5-3		2	2	1
6.08 N		L-48-1-1		0			I-SY-9-1-3		2	2	1
6.09 Y				0					0		
6.09 N				0					0		
6.10 Y				0					0		
6.10 N				0					0		
6.11 Y				0					0		
6.11 N				0					0		
6.12 Y				0					0		
6.12 N				0					0		
6.13 Y				0					0		
6.13 N				0					0		
Colleagues											
7.01 Y		L-43-28-1, L-102-20-2		0					2	2	
7.01 N				0					0		
7.02 Y			I-GA-33-29-2	1	I-TC-4-25-3, I-TC-12-30-2		I-SY-18-30-3		3	4	2
7.02 N			I-GA-34-4-2	1	I-TC-9-16-3		I-SY-7-28-3		2	3	2
7.03 Y				0					0		
7.03 N				0					0		
7.04 Y		L-99-10-1, L-181-25-2		0					2	2	
7.04 N				0	I-TC-5-17-3				1	1	1
7.05 Y	D-338-10-1	L-81-30-3		0	I-TC-2-18-2				3	3	1
7.05 N	D-7-1-1, D-342-25-1	L-157-10-1, L-165-25-1		0	I-TC-3-8-3				5	5	1
7.06 Y				0					0		
7.06 N				0					0		
7.07 Y				0					0		
7.07 N				0					0		
7.08 Y	D-154-5-1	L-30-20-1	I-GA-6-9-2	1	I-TC-12-3-2		I-SY-10-27-3, I-SY-11-30-3, I-SY-25-16-3	I-GL-23-19-1	7	8	3
7.08 N		L-12-10-2, L-50-10-1		0					2	2	
Self											
8.01 Y		L-53-15-1, L-125-10-1, L-147-30-1	I-GA-1-11-2, I-GA-29-7-3	2	I-YC-13-12-3		I-SY-16-13-3	I-GL-20-29-3	6	8	4
8.01 N		L-12-25-2, L-38-10-1, L-79-30-1		0	I-TC-15-18-3				4	4	1
8.02 Y			I-GA-13-9-2, I-GA-23-17-2	2	I-TC-4-25-3		I-SY-17-15-2	I-GL-14-9-3	3	5	2
8.02 N		L-139-25-1		0		I-ML-10-9-3			2	2	1

Principles	Documentation	Log Book	Manager	M TOT	Developer 1	Developer 2	Developer 3	Developer 1	TM TOT	Total	3'S
8.03 Y		L-98-25-1		0				I-GL-14-9-3, I-GL-14-14	3	3	1
8.03 N				0	I-TC-5-17-3				1	1	1
8.04 Y	D-324-10-2		I-GA-1-11-3	1	I-TC-2-8-3		I-SY-5-29-3, I-SY-24-28-2	I-GL-25-15-3	5	6	4
8.04 N			I-GA-10-9-3	1					0	1	1
8.05 Y	D-70-1-1, D-290-5-2		I-GA-13-26-3, I-GA-23-12-2	2	I-TC-11-15-3		I-SY-1-12-3, I-SY-16-13-3	I-GL-20-2-3, I-GL-24-16-3	7	9	6
8.05 N		L-53-20-1, L-99-20-1	I-GA-39-4-1	1	I-TC-6-22-3	I-ML-7-26-3		I-GL-13-15-3	5	6	3
8.06 Y				0					0		
8.06 N				0					0		
8.07 Y				0					0		
8.07 N				0					0		
8.08 Y			I-GA-38-3-2	1					0	1	
8.08 N				0					0		
8.09 Y				0					0		
8.09 N				0					0		

D.10.4 Non-Compliant Data Collected for Telco

The number of incidents found in the case study for Telco generally - reflecting its working culture and practices - of non-compliance to the IEEE/ACM Code of Ethics was collected in a repertory grid and can be found in Figures 10 and 11 on the following pages. The number of incidents allocated an importance level of 3 are also shown.

(The captions are provided below due to page restrictions)

Figure 10. Details of Non-Compliance for Telco

Figure 11. Details of Non-Compliance for Telco

Principles	Log Book	Documentation	Total	3'S
Public				
1.01 N				
1.02 N	L-82-25-3		1	1
1.03 N				
1.04 N				
1.05 N	L-175-20-1		1	
1.06 N	L-4-10-2, L-73-10-3, L-146-20-2	D-42-10-2	4	1
1.07 N				
1.08 N				
C & E				
2.01 N		D-68-1-1	1	
2.02 N				
2.03 N	L-47-1-1, L-135-5-3, L-145-30-1	D-38-1-1, D-69-1-3, D-141-10-3, D-241-1-1, D-316-30-1	8	3
2.04 N				
2.05 N	L-148-1-2		1	
2.06 N				
2.07 N				
2.08 N				
2.09 N				
Product				
3.01 N		D-326-10-3	1	1
3.02 N				
3.03 N	L-129-20-1		1	
3.04 N				
3.05 N	L-8-10-2, L-16-30-3, L-173-5-2	D-23-15-3, D-28-30-2, D-145-5-2	6	2
3.06 N	L-1-1-2, L-18-30-2, L-46-1-2, L-99-25-3, L-162-8-3, L-163-5-2	D-136-5-2	7	2
3.07 N				
3.08 N	L-39-25-2	D-27-25-2, D-339-20-2	3	
3.09 N		D-64-2-3	1	1
3.10 N	L-163-15-3	D-29-28-2, D-65-1-2, D-151-5-1, D-171-1-3, D-258-5-1	6	2
3.11 N	L-41-20-2, L-92-20-2, L-128-28-1	D-68-5-1, D-140-1-1, D-153-1-2, D-158-10-3, D-238-10-1	8	1
3.12 N				
3.13 N				
3.14 N				
3.15 N	L-46-30-2, L-52-15-2, L-138-20-3, L-146-5-2	D-60-1-3, D-63-1-2, D-172-1-3, D-220-1-3	8	4
Judgment				
4.01 N				
4.02 N				
4.03 N	L-158-10-2		1	
4.04 N	L-3-1-3, L-38-20-1, L-45-32-2, L-165-10-2	D-92-10-2, D-125-1-3, D-316-30-1, D-372-25-1	8	2
4.05 N				
4.06 N				

Principles	Log Book	Documentation	Total	3'S
Managemt				
5.01 N	L-39-30-2, L-109-20-2, L-128-10-2, L-140-30-2, L-163-20-2	D-4-5-2, D-30-30-3, D-68-5-2, D-91-5-1, D-102-30-2, D-106-1-2, D-286-5-2	12	1
5.02 N				
5.03 N				
5.04 N				
5.05 N	L-26-20-2, L-67-15-2	D-81-5-2, D-103-5-3, D-107-5-2, D-299-15-2	6	1
5.06 N	L-1-5-2		1	
5.07 N				
5.08 N				
5.09 N		D-120-10-3, D-326-10-2	2	1
5.10 N				
5.11 N	L-3-1-2		1	
5.12 N				
Profession				
6.01 N				
6.02 N				
6.03 N		D-138-5-1	1	
6.04 N				
6.05 N				
6.06 N	L-67-10-2, L-145-10-2		2	
6.07 N				
6.08 N				
6.09 N				
6.10 N		D-120-5-3, D-326-5-3	2	2
6.11 N				
6.12 N				
6.13 N				
Colleagues				
7.01 N				
7.02 N	L-63-10-2, L-64-30-1, L-111-10-1		3	
7.03 N				
7.04 N		D-150-1-1	1	
7.05 N	L-11-1-2, L-67-10-1	D-16-1-2	3	
7.06 N				
7.07 N				
7.08 N				
Self				
8.01 N	L-162-10-3		1	1
8.02 N		D-61-10-2	1	
8.03 N	L-140-30-1		1	
8.04 N				
8.05 N				
8.06 N				
8.07 N				
8.08 N				
8.09 N				

Appendix E. The Case Study

E.1 Introduction

A case study was conducted for data to be collected which would enable the aim and objectives of the research to be met. Further justification for this approach is provided below, together with procedures considered necessary for rigorous research. The expected outcomes of the case study are then presented, for both the research and the company selected to participate. The company is then described, in terms of its culture and working practices, particularly those pertinent to software development. The redundancy program is then described, followed by the role of the author at the company.

E.2 Additional Justification

A case study was chosen as what is problematic in practice can be defined more readily, stimulating our thinking leading to the generation of new ideas about cause and effect - under the constrains of an organisational context. In addition, the complex social, political and economic context in which new systems are developed becomes more identifiable. Furthermore, knowledge gained can be shared through published reports, providing an additional route to understanding through secondary data, without having to experience unethical practices first hand.

Sauer (1994) identified the concluding value of a systematic and rigorous case study may take the form of - in the context of this research:

- Raising problems about the phenomenon of ethics
- Stimulating theories of the causes of non-adherence to ethical codes
- Stimulating theories of the causes of adherence to ethical codes
- Stimulating theories of the cause-effect chains which lead to non-adherence
- Stimulating the development of ethical adherence mechanisms

Sauer (1993) identified other possible advantages of adopting a case study strategy:

- Theories and models of non/adherence are illustrated
- The application of those theories and models is highlighted
- The processes by which non-adherence occurs is shown

- They are a surrogate for costly experience

The methods used for data collection were questionnaires, interviews, documentation analysis, physical artefacts and participant observation.

E.3 Issues for Consideration

E.3.1 The Case Study Protocol

The protocol is considered necessary to ensure the reliability of a case study by ensuring the researcher(s) knows why the research is being done, what evidence is required, what variations are anticipated - with any necessary corrective action - and what constitutes evidence for supporting or refuting any given proposition. The protocol is a detailed statement describing what the research is aiming to achieve, includes questions to be put to the participants and, typically consists of five sections:

1. Overview
2. Field Procedures
3. Case Study Questions
4. Guide to the Case Study Report
5. Pilot Case Study

These sections will now be described, where considered pertinent to the research.

E.3.2 Overview of the Case Study

The overview of the case study consists of the objectives of the research. These were identified above as to:

1. Identify the role and importance of ethics in development teams
2. Evaluate IEEE/ACM Code of Ethics and Professional Conduct for Software Developers as an ethical framework - as previously defined
3. Identify how working practices, policies and procedures are influenced by the framework
4. Identify how the application of the framework is influenced by working practices, policies and procedures

The meeting of these objectives was achieved by observation, document analysis, a series of semi-structured interviews and a questionnaire. These enabled the participants in the case study the opportunity to give information on a wide range of issues related to situations where ethical standards were compromised or in conflict.

E.3.3 Field Procedures

It is possible for a case study to be carried out in a library and omit the need to leave the researcher's institution. It is doubtful, however, whether such a case study is able to produce the rich insights needed for it to be accepted as rigorous academic research. This is due to many reasons, including (a) the age of the information collected from a library which is likely to be two years old or more, (b) the information would be to general for specific use and, (c) there would be little control over the quality of the information obtained. These problems outweigh the benefits, such as speed and reduced cost (Housden, 1992). A company was selected for use, therefore, to enable the following characteristics of a good case study to be present. It should (Yin 1994):

- Be significant
- Be complete
- Consider alternative perspectives
- Display sufficient evidence
- Be composed in an engaging manner

Telco was encouraged to take part with a list of possiblel benefits. The company was the 'case' being studied, with the individuals relevant to the research questions forming the 'logical sub-units' (software developers and their project managers), which enabled the case study to be identified as embedded. Chapter 2 found that a diverse employee population is necessary for a department to function efficaciously in the networked world in which we all now live and work. This was an additional reason for choosing Telco as the principle project team to be studied was found to be multi-cultural, consisting of English, Scottish, Indian, Irish, Cypriot and Greek nationals.

Access to the participants for interviews and the administration of a questionnaire needed to be determined and agreed from the outset of the case study, and a schedule was created for each participant. Necessary resources needed for the entire duration of the case study were identified as: a desk, PC, printer, telephone, pens and paper. A quiet room at the company was also selected so that interviews could be held and, work requiring concentration could be done, without disturbance. In the likely event of the researcher needing to ask questions, a procedure was agreed upon so that assistance could be sought as appropriate. The accommodation of unanticipated events included the possibility of the author himself becoming lethargic. This was catered for by structuring a typical week into three days dedicated to assisting the project manager and two days a week dedicated to conducting research activities, or to allow time to catch up, meet his supervisors, reflect or just take 'a breather' as necessary.

E.3.4 Case Study Questions

Case studies enable several techniques to be carried out which can collect the evidence required. These techniques include: semi-structured interviews, direct observation, participatory observation, surveys/questionnaires and document analysis. The section entitled 'Case Study Questions' contains questions, however, that are created by the author for himself to answer. Associated with each question is also a list of probable sources of evidence - which where expected to take the form of those described above. The questions formed were also useful prompts when conducting interviews with the case study participants, but the main purpose of these questions was to keep the author focused on his objectives as the case study progressed. The questions raised regarding these research aims and objectives, with appropriate sources of evidence are shown in Table 26 below.

Research Questions	Probable Sources of Evidence
Is the IEEE/ACM Code suitable as an ethical framework?	IEEE documents, literature review
What is needed for successful implementation?	IEEE documents, literature review
How is its value perceived?	Project manager, team members
What ethical issues have arisen?	Project manager, team members
What factors influence your ability to work ethically?	Project manager, team members

Table 26. Case Study Questions for the Author

This section has identified two types of questions that need to be asked in an interview: those put to the author and those put to participants. The researcher has to be careful, though, of being able to distinguish between asking questions about individual interviewees and the case itself (Yin 1994).

A case study question can also be in the form of an array table, which Yin (1994) calls 'table shells'. The row and column headings indicate the data to be collected by the researcher. Such tables can help by facilitating what data needs to be collected, ensure consistency of data categories being collected and finally, aids understanding of what will be done with the data after it has been collected. With the availability of the IEEE/ACM Code and the use of repertory grids as described above, the use of table shells in the research was considered unnecessary.

E.3.5 Guide to the Case Study Report

Although the report's inclusion in a research document is easily justified, it is invariably missing from case study theses. This is invariably due to their non-acceptance into journals at some later date. But by describing the data needed to be collected, the report reduces the chances of the researcher having to return to the company to collect additional data at some point after the initial collection has been completed. (This was particularly important in this instance as many participants in the case study were made redundant in the latter part of the research.) The major headings that were identified are presented below, as the key focal points of the case study reports. They were established early in the research process to act as an aid - in alignment with the eight principles of the IEEE/ACM Code - when the semi-structured interview questions for software developers were constructed:

- Do you consider the public when decision-making?
- Do you use the property of your employer or clients only in ways properly authorized and with their consent?
- Do you strive for high quality in development and maintenance?
- Are you professionally objective when evaluating software or documents?
- Do you ask others to be unethical?
- Do you obey all laws governing your work?
- Do you assist your colleagues in professional development?
- Do you further your knowledge in the area of IS development?

E.3.6 A Pilot Case Study

The case study protocol usually requires a pilot case study to be carried out. On this occasion, however, a pilot case study was not carried out as the study had three attributes which removed the need for one to be undertaken. Firstly, it was an in-depth study, secondly, it was a longitudinal study over a six month period and, lastly but by no means least, it was very much context dependent.

E.4 Case Study Field Procedures

Yin (1984) recommends creating case study procedures to ensure a rigorous research. In the context of this research, the procedures consist of interviewee's schedules, a procedure for seeking assistance, a breakdown of acceptable evidence, a network diagram and a list of desired attributes of a researcher. These are provided in more detail below.

E.4.1 Interviewee's Schedules

This is provided for the VTC team and pertinent employees from other business units only.

Name	Availability
Gordon Anderson	Daily 10-11am
Paul Neubert	Daily 10-11am
Andy West	Daily 10-11am
Shabaz Yousaf	Daily 10-4
Gary Lefman	Daily 10-4
Mark Lewington	Daily 10-4
Ted Curran	Daily 10-4
Bob Cool	Daily 10-4
John Hill	Daily 10-4
Chris Frangoudes	Daily 10-4
Gavin Whitehead	Daily 10-4
Mandeep Rohilla	Daily 10-4
Anthony Hawkins	Daily 10-4
Ikenna Orange	Daily 10-4
Ioannis Tsiolis	Daily 10-4
Georgina Whiteley	Daily 10-4
Aideen Fahy	Daily 10-4

Table 27. Participants' Availability

E.4.2 Procedure for Seeking Assistance

The following procedure was agreed by the researcher, his supervisor and the case study participants for instances when assistance was sought:

- Clarify what assistance is needed
- Identify who is best suited to provide the assistance sought
- Explain what is required
- Record assistance given

E.4.3 Breakdown of Acceptable Evidence

There are six categories of acceptable evidence for use in case study research: interviews, observation (direct and participant), documentation analysis, archival records and artifacts. Each can be broken down into sub-divisions or described as shown in Table 28 below.

Category	Sub-division/description
Documentation	Newspaper articles, project records Administrative documents, e.g. proposals progress reports, other internal documents Letters, memos, etc., meeting minutes, agendas announcements, other written reports
Archival Records	Personal records, e.g. diaries, calendars telephone listings Organisational records, e.g. budgets and charts Service records, e.g. showing number of clients served over a specific duration Maps and charts showing the geographical attributes of the chosen site Lists of names, etc. Survey data, e.g. data previously collected about the site or census records
Interviews	Structured (formal Survey) Semi-structured (Focused) Open Ended
Direct Observation	Incidences such as: Meetings, factory work, classrooms, path activities Can observe the environment such as: Condition of the building and/or the working area to reflect the climate or financial position, the location of furniture/equipment to reflect individual status
Participant Observation	The researcher takes a role within the study such as:

	Employed as a member of staff in an organisation Being a key decision maker within an organisation A resident within a community Having a specific role to play within a community
Physical Artifacts	These can be observed and/or collected and may include: A work of art, a tool, a technological device, a piece of equipment.

Table 28. Breakdown of Acceptable Evidence

The problems associated with creating a database are categorized in terms of: notes, tabular materials, narratives and documents - each of which is described below. Case study notes can take the form of interview transcripts, analysis of documents or observations. They may be recorded on cassette(s), in the form of photographs, hand-written or on typed paper, floppy disks, CDs, etc. The researcher need not concern himself with transferring the data into a standard format - but instead should concentrate on getting the database complete, categorized (probably under the headings identified in a case study protocol), accurate and available for later use - by himself and others, for example, examiners and/or other researchers. Transcribing interview tapes can be expensive and should be considered carefully beforehand. Case study documents need categorization and subsequent filing to be orderly and controlled, so that any document at any time can be retrieved as required. This can be facilitated by annotating the references to identify the storage location of each document, such as in the researcher's office or home, a specific library, another institution or on the internet. Tabular materials (if collected) need to be categorized and filed as with case study documents. These materials may include items such as: survey data, other quantitative data, archival evidence and observational evidence - in this case, observational evidence was recorded in the author's log book. Narratives may be used as evidence but which form part of the database and not the final report. Such a narrative might be of a form currently growing in popularity consisting of questions and answers - both of which are supplied by a researcher. The purpose here being to integrate (specific) evidence collected and converge upon the facts of the issue at that time - followed by the documentation of the connections formed. As the questions and answers are kept in the database and not in the report, the researcher need not use up valuable time on editing and presentation of the data.

E.4.4 Attributes Desired of a Researcher

So far a description and consequent critique of the interpretive paradigm and case study technique have been given to justify their appropriateness in enabling the aims and objectives of the research to be met as identified in Chapter 2. The selection and justification of the interpretive paradigm highlighted the personal and individual nature of the research which cannot exclude consideration of a researcher. Conducting a case study makes demands on his intellect, ego and emotions, all of which need to be managed as their demands are greater here than any other research strategy - as the procedures for data collection are not routinized. In a laboratory experiment, for example, much of the work could be done by the researcher's assistant as a minimum amount, if any, of discretionary behaviour is required, i.e. routine. Case studies have no such routine, thus demanding on the researcher's emotions as described above. A description of a researcher's role facilitates the consensus on envisaged difficulties as outlined below. Walsham's (1993) description is used here, as deemed most appropriate in the context of this research.

The author expected to assume the role of an IS evaluator as his responsibilities during his employ were to include carrying out a formal evaluation exercise, a personal evaluation study, monitoring actions and consequences and, forming his own assessments, demonstrating managerial competence and facilitating evaluation discourse between relevant stakeholders. A researcher should also understand and facilitate the sharing of experiences from previous assessments with the shareholders to encourage an explicit learning environment. This includes the view that the questioning of decisions is acceptable, everyone is a learner, moral issues can be discussed and all assessments carried out are legitimate.

As such, it is most desirable for a researcher to have had some kind of formal training prior to the start of a case study as they are one of the most difficult kinds of research - especially for a novice researcher - argued Yin (1984). Ideally, the researcher should be well-trained and experienced. This is due to three reasons. Firstly, the researcher will feel trapped due to the continuous interaction between the theoretical issues being studied and the data being collected. Secondly, the data collection environment is out of his control - he must integrate the company's day to day events with his data collection activities/plan. For example, his interview schedule must cater to the interviewee's schedule and availability. (This is addressed in the field procedures section of the protocol via the use of a table showing agreed access, e.g. day and time windows, for each participant.) Thirdly, the researcher needs to be, and stay, aware that he is intruding on the case and not vice versa: a possible consequence being the interviewee might not co-operate fully when answering questions being put to him. Careful use of language and behaviour by the researcher - indicating his appreciation of the participant's time and contribution to the research - will prevent or reduce any negative feelings the participant may have towards the research and/or the researcher.

Ideally a research methods course will have been undertaken beforehand at the researcher's institution and possibly even a research retreat attended to obtain more information or just to clarify understanding. The retreat could, for example, be organised by researchers themselves to focus on issues they have identified which could then be taught/explained/discussed by an experienced researcher/academic.) This should enable the creation and use of a protocol resulting in a high quality case study executed with smooth management, i.e. problems dealt with satisfactorily and reliability of the case study results increased. The author has studied research methods having attended such a research methods course and retreat - both organised jointly by research students at the Department of Information Systems and Computing at Brunel University.

On a practical day-to-day level, the researcher should ideally be able to ask good questions, be a good listener, be flexible, have a good understanding of the issues at hand, be honest and unbiased by any preconceived notions. He must be able to operate autonomously within the chosen company and be able to make 'intelligent' decisions as necessary. (This was demonstrated at the company selected to take part at the outset by the positioning of the author high up in a slide showing the hierarchy of the development team presented to all staff - at the penultimate level, across from the technical consultant.) A researcher needs to know why the case is being studied and what data is required, in addition to any anticipated variations and their respective corrective/accommodating action(s). For any given proposition, he needs to be able to identify from the evidence collected whether it is either supportive of, or contrary, to that proposition. These issues are addressed with a professional outlook on the research, i.e. commitment and rational thinking. Both these attributes indicate good preparation, fair execution, honest analysis and conclusions will be done by the researcher. The acronym 'PRO' summarizes the motivation of a researcher to work in this way: Privileged to be able to do this research, a Responsibility to the company taking part/supervisors/viva examiners and lastly, this provides an Opportunity to explain what he has done, found and concluded.

E.5 Expected Outcomes

E.5.1 For the Research

On completion, the case study is expected to have the following outcomes:

1. An understanding of the value and use of ethical frameworks in IS will have been acquired
2. The influence of frameworks on development practices will have been identified
3. The influence of organisational/cultural/social factors on the success of a framework will have been identified
4. An ethical framework may have been created or amended by the author
5. A questionnaire will have been conducted together with semi-structured interviews
6. The author would have acquired some practical software project management experience

E.5.2 For the Company

The company which took part in the case study was encouraged to do so with the list of possible benefits shown below. By taking part they could:

1. Gain an understanding of the value and use of ethical frameworks in software development
2. Identify how ethical frameworks influence development practices in organisations
3. Identify how existing organisational and cultural issues influence the success of an ethical framework
4. Be involved in the design or amendment of a framework
5. Promote the company image to new/existing customers thereby increasing competitive edge, in addition to publicity in academic journals and at conferences
6. Benefit from research findings by management in general in addition to software project management

E.6 The Company Selected

E.6.1 Introduction

Telco Systems was founded in the United States and had doubled in size every year since its debut in 1984. Telco has offices in many countries around the world, including: Switzerland, Spain, Denmark, Germany, Poland, Italy, UK, Sweden, Belgium, Greece, Czech Republic, Netherlands, Finland, Portugal, Ireland, Croatia, Cyprus, Hungary, Romania, Norway, Austria, Bahrain, Egypt, Israel, Turkey, South Africa, Russia and Ukraine (D-108).

Telco Systems Limited was selected for the case study for several reasons, viz:

1. It is a large international IS company
2. Well known in academia and industry
3. The findings were expected to be read with interest by academics and practitioners
4. The project manager was interested in my area of research
5. An informal ethical working environment was already established
6. A framework was already identified and complete to monitor and evaluate adherence
7. A budget available for relevant expenses
8. The project manager accepted a flexible time plan separating research from work
9. The case study could start within the author's initial schedule, i.e. February 2001, just two months after release of latest version of ISO 9000
10. There was a possibility of further research after the award, or otherwise, of ISO 9000

In addition, secondary factors influencing Telco's selection were:

1. They were able to offer practical software project management experience
2. The project management techniques and jargon used were familiar to the author
3. The office was within acceptable commuting distance from both the author's home and Brunel University
4. Internal training courses at Telco were available to the author as felt necessary
5. A comfortable working environment would be provided

6. There was the possibility of training for the author offered at their head office in the US
7. Telco had won an award for being the top E-Business (D-185)

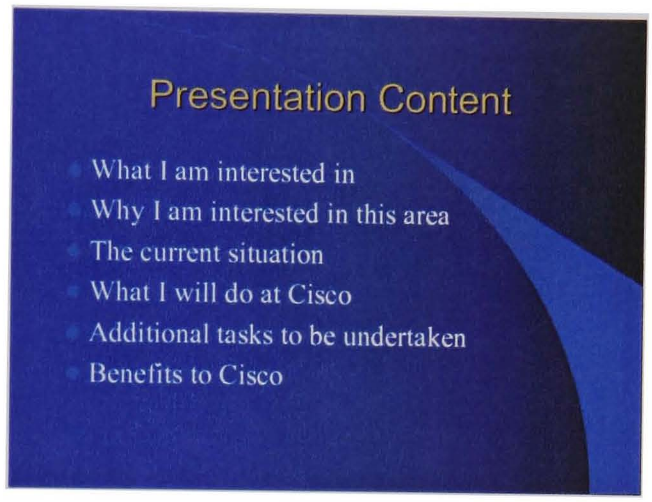
It was difficult initially to get acceptance to carry out the case study at Telco as the people needed to give the go ahead were frequently out of the country or unobtainable by telephone - emails sent to Telco were returned with error messages which could not be resolved until the author received emails from staff within Telco - some weeks later. A great deal of frustration was experienced by the author at the outset, almost to the point of selecting another company. Agreement to continue was eventually obtained, however, which enabled the finer details of the cases study to be agreed.

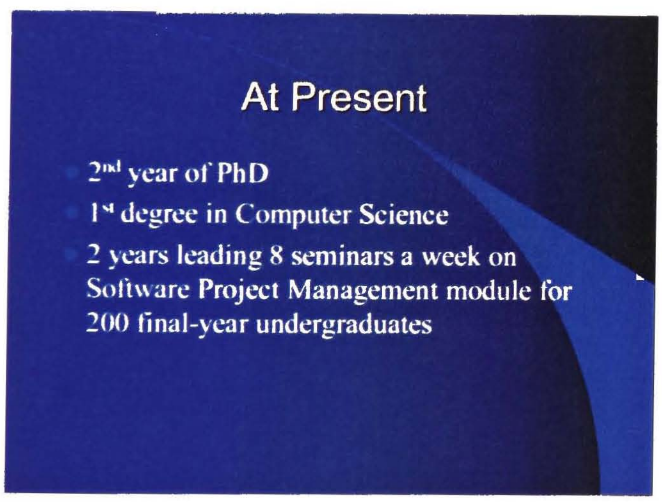
The case study began on 12 February 2001 and finished on 25 August 2001. The author was assigned to a project team called NSSU-UK and which was expected to grow by 50% in the twelve months following the start of the case study. At the outset of the research, the company was frequently advertising on television and, the author was issued with a new lap top computer shortly after the case study began. To understand a little of the environment found, in an initial log book entry the author had entered stated that unlimited fruit and soft drinks were available free, infinite training appeared to be available to all and expenses were paid almost instantly and automatically.
Assisting Telco with their project management activities and helping to determine and roll out the processes necessary for them to be awarded ISO 9000 encouraged Telco to offer payment to the author in return for his contribution. Although this practice is not uncommon, in this case it did possibly add a further ethical perspective to the research. It was necessary for the author to remain loyal to meeting his research objectives and, not be influenced by the financial assistance offered by Telco in a way that might compromise the standard and integrity of his research efforts. Initially the author felt this would not be problematic, although acknowledging the possible difficulties. This was overcome by negotiating the payment to the author on a pro rata basis. This was calculated on a three day week working for Telco and two days a week for the author to conduct his research: either at Telco, the University or at his home. The author's manager at Telco left the management of the research and work for Telco to the author to organise which was welcomed.

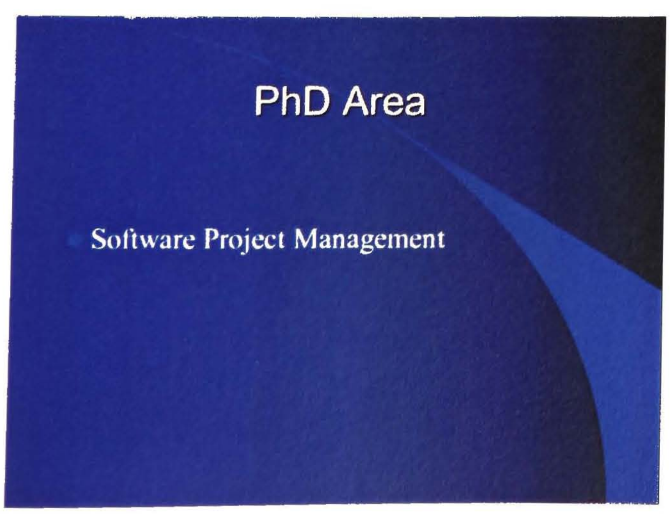
The author was assigned to the VTC project team, consisting of a project manager and almost twenty software developers, team leaders and a technical consultant. Ultimately the project team was reduced to just five developers and, as they were providing support for the whole of Europe, they did not expect to have much time for any software development.

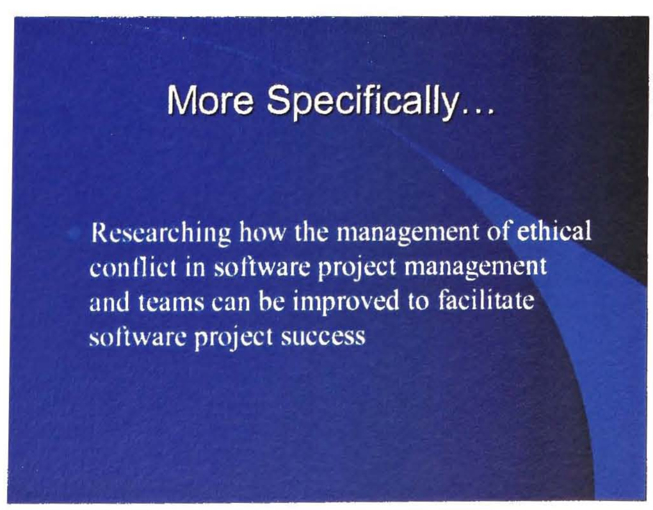
E.6.2 Presentation Given

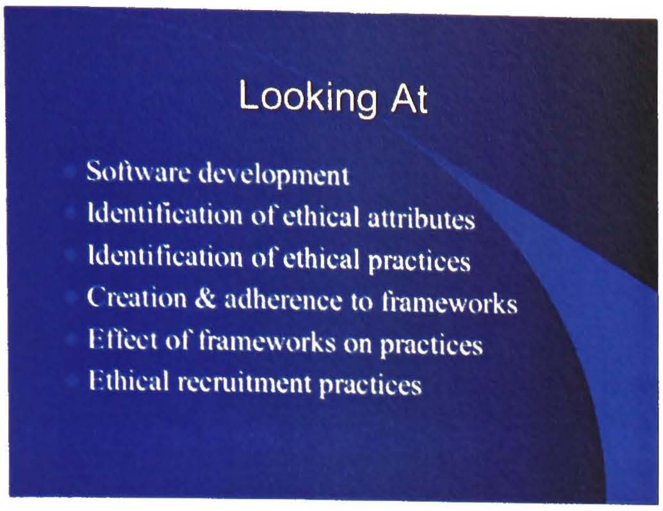
The following slides were presented to the VTC project team manager and members shortly after the case study commenced.

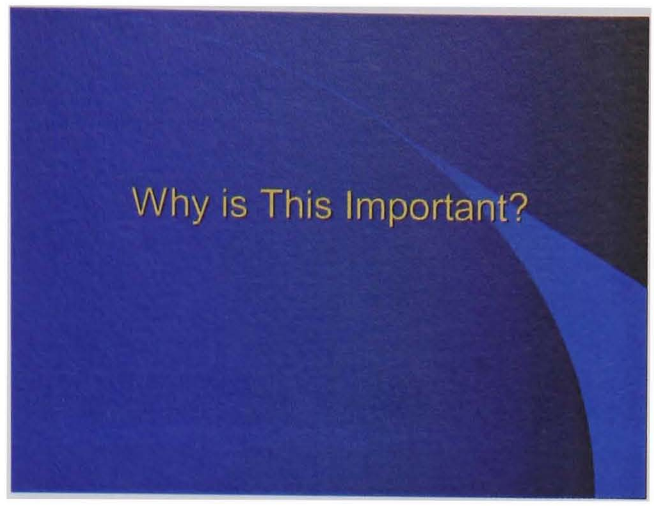


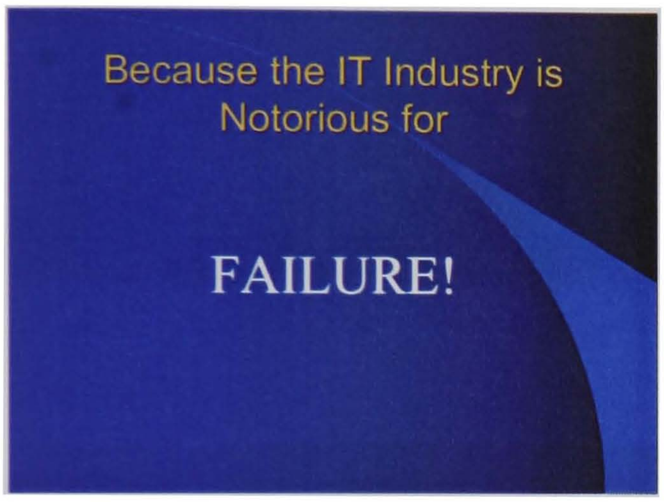


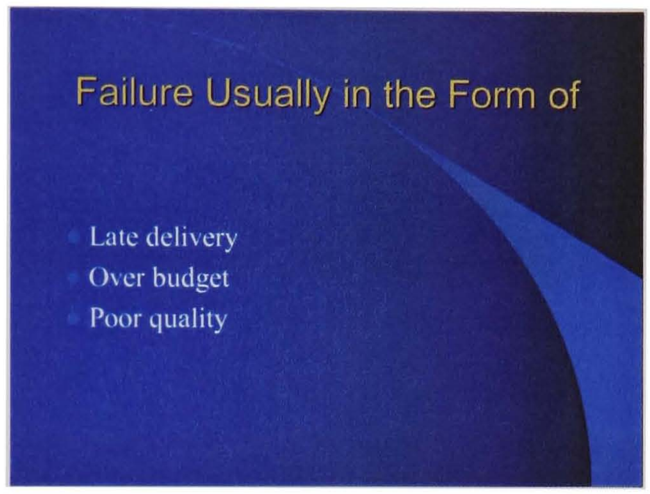


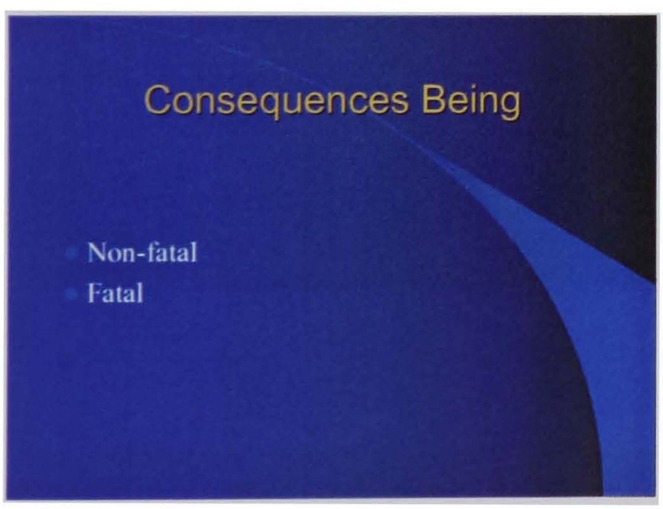



Looking At
Software development
Identification of ethical attributes
Identification of ethical practices
Creation & adherence to frameworks
Effect of frameworks on practices
Ethical recruitment practices

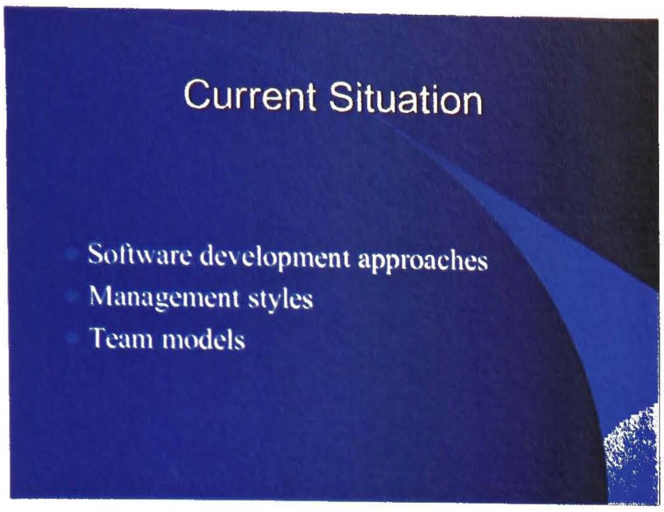

Why is This Important?


Because the IT Industry is Notorious for
FAILURE!

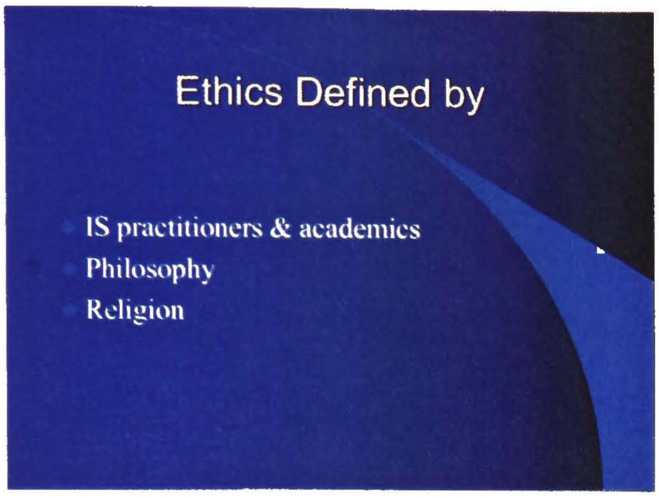

Failure Usually in the Form of
Late delivery
Over budget
Poor quality

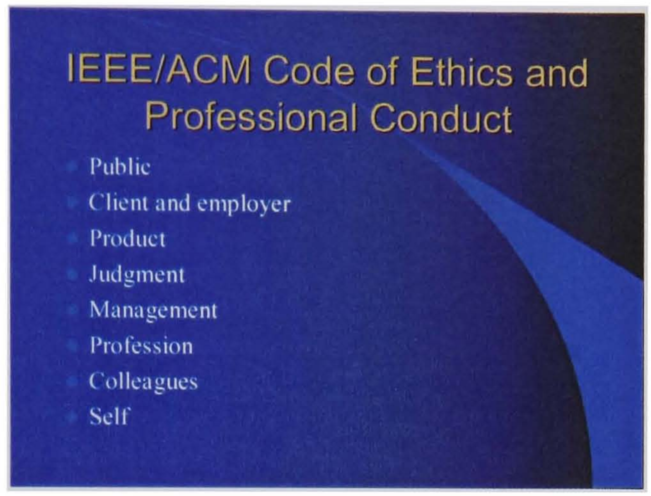

Consequences Being
Non-fatal
Fatal

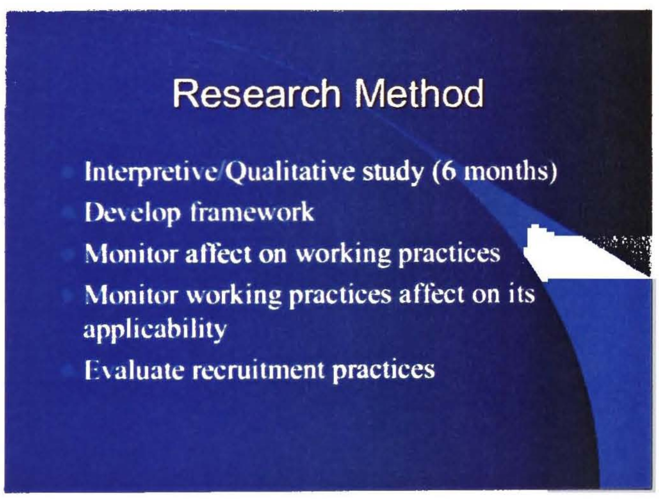

Failure Based on
Conflicting or compromised ethical standards
Language difficulties
Tensions on Time/Cost/Quality constraints
Disregard for social issues

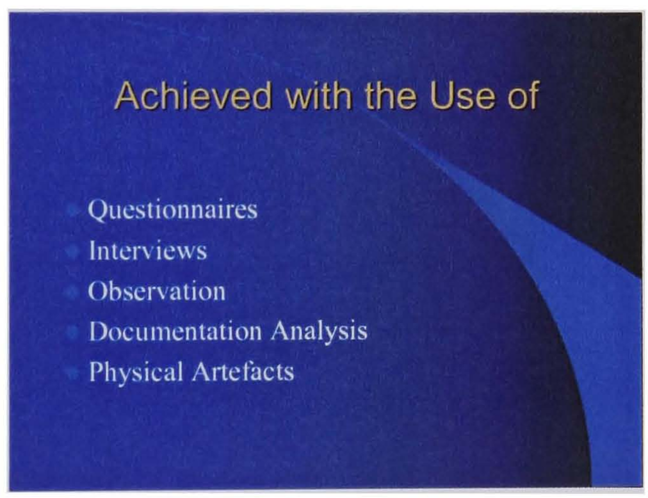

Current Situation
Software development approaches
Management styles
Team models

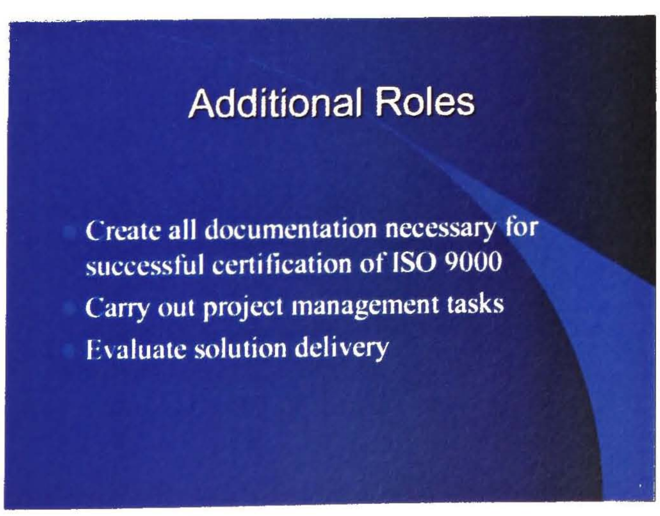

Belbin's Team Roles
Plant
Resource Investigator
Co-ordinator
Shaper
Monitor/Evaluator
Team-worker
Implementer
Specialist
Completer

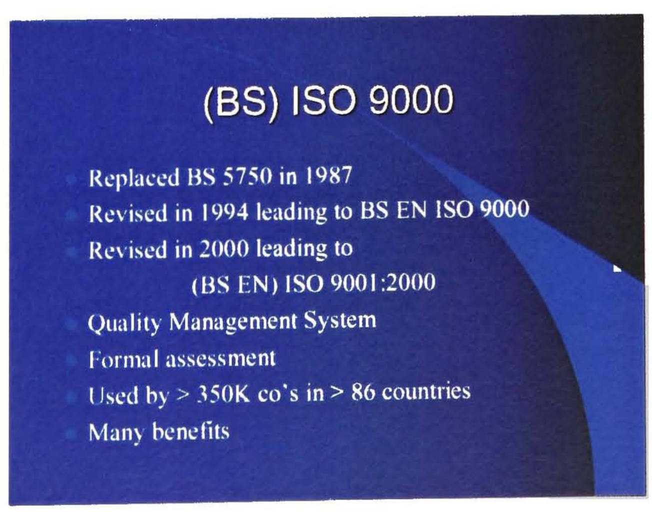

Ethics Defined by
IS practitioners & academics
Philosophy
Religion

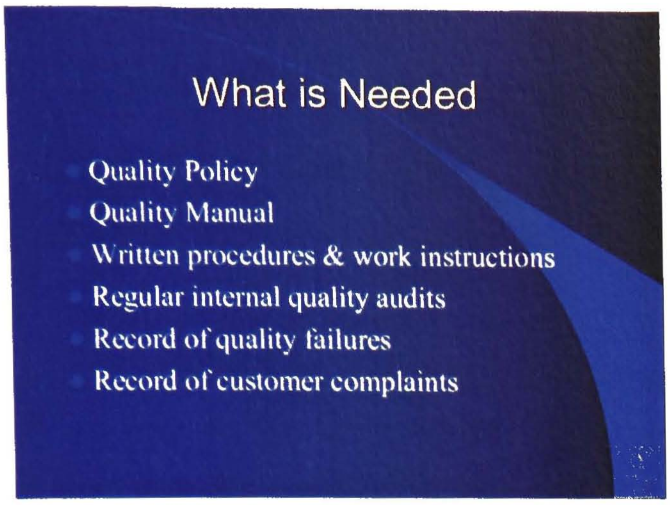

IEEE/ACM Code of Ethics and Professional Conduct
Public
Client and employer
Product
Judgment
Management
Profession
Colleagues
Self

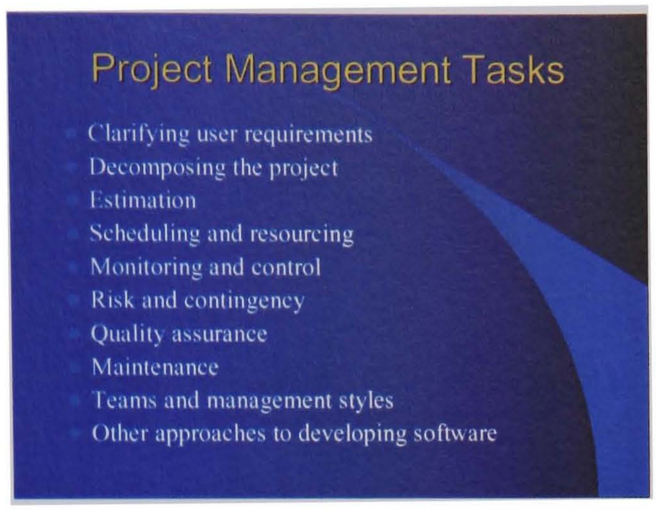

Research Method
Interpretive/Qualitative study (6 months)
Develop framework
Monitor affect on working practices
Monitor working practices affect on its applicability
Evaluate recruitment practices

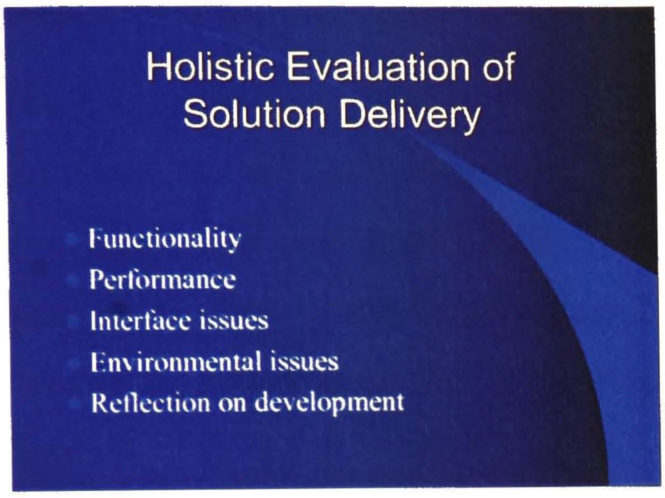

Achieved with the Use of
Questionnaires
Interviews
Observation
Documentation Analysis
Physical Artefacts

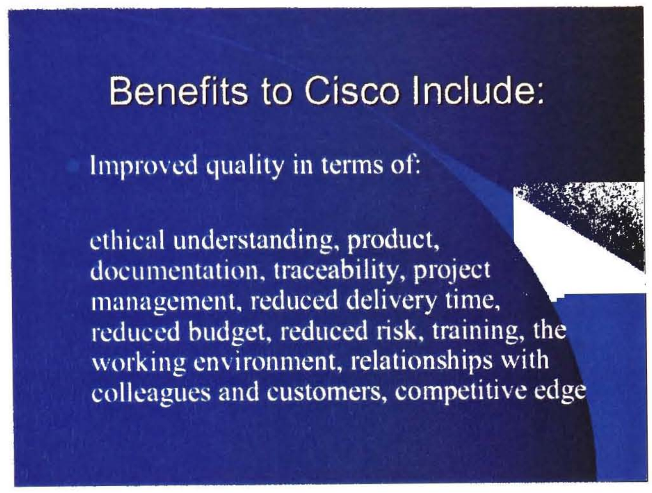

Additional Roles
Create all documentation necessary for successful certification of ISO 9000
Carry out project management tasks
Evaluate solution delivery

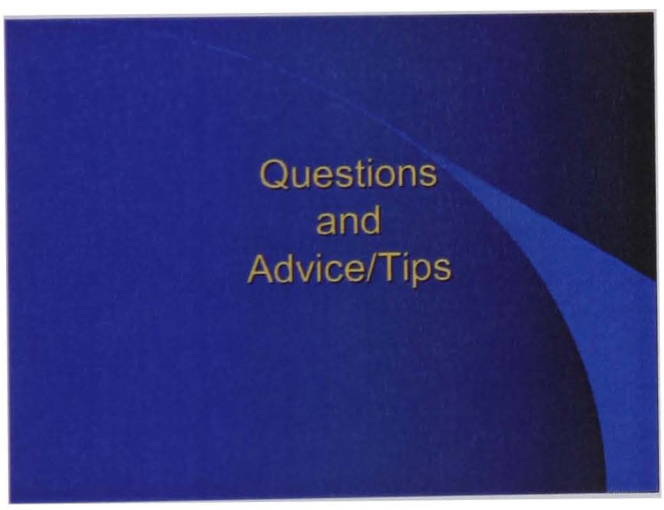

(BS) ISO 9000
Replaced BS 5750 in 1987
Revised in 1994 leading to BS EN ISO 9000
Revised in 2000 leading to
(BS EN) ISO 9001:2000
Quality Management System
Formal assessment
Used by > 350K co's in > 86 countries
Many benefits


What is Needed
Quality Policy
Quality Manual
Written procedures & work instructions
Regular internal quality audits
Record of quality failures
Record of customer complaints


Project Management Tasks
Clarifying user requirements
Decomposing the project
Estimation
Scheduling and resourcing
Monitoring and control
Risk and contingency
Quality assurance
Maintenance
Teams and management styles
Other approaches to developing software


Holistic Evaluation of Solution Delivery
Functionality
Performance
Interface issues
Environmental issues
Reflection on development


Benefits to Cisco Include:
Improved quality in terms of:
ethical understanding, product, documentation, traceability, project management, reduced delivery time, reduced budget, reduced risk, training, the working environment, relationships with colleagues and customers, competitive edge


Questions and Advice/Tips

E.6.3 Culture

Telco culture is described on the reverse side of the ID badge worn by all permanent staff:

> 'Quality team, team work, drive change, no technology religion, empowerment, frugality, market transitions, stretch goals, trust, fairness, integrity, open communication and customer success'

The Telco mission is to create unprecedented value and opportunities for its employees, customers, visitors and eco-system partners. The Telco vision is 'To change the way we live, work, play and learn'. With regard to product quality, a poster in one project manager's office reads: Carrier class means high availability - 99.999% (less than 5 minutes downtime per year). 'Customers demand it, systems require it and Telco is committed to it'. For such a large company, documentation plays a big part in contributing to this vision. The hierarchical document structure is shown in Figure 12 below.

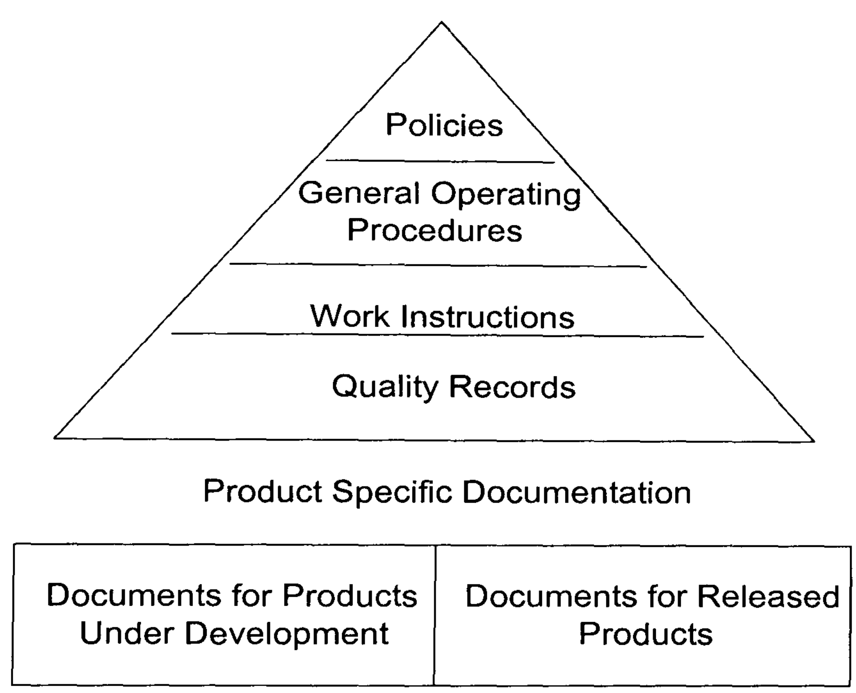


Figure 12. Document Hierarchy

There is a global 'Trust Policy' in place. Staff can enter and leave offices they please, with any piece of company equipment. This was due to cheap implementation and improves staff morale. For example, when a project manager went to a US for a meeting, he arrived late at night and could not find his hotel. He then went to the Telco office and found he had full access to the building with his UK electronic pass. Conversely, an

Australian staff member experienced difficulties when talking to the Finance Department about one expense claim and exasperated,

'I don't have any receipts or credit card statements so you will just have to trust me.'

Processes and paperwork are considered both difficult and expensive to implement. Consequently, time sheets are not a Telco practice, although the option to use them can be exercised when a new employee joins the company. The technical director preferred to use coffee room conversations and email to distribute information gathered. He said that everyone must be proactive in getting information and not just wait to be told.

Project managers typically wore a shirt and tie, with the developers, leaders, technical consultant and technical director wearing smart/casual clothes. Some developers regularly wore Telco issued clothing displaying the company name and logo. The quality manager was American and very frequently wore clothes displaying the company logo. Whilst on holiday last year, the project manager saw two Telco managers at the same hotel - identified by their caps displaying the Telco logo. A developer once wore a shirt displaying a CMM logo, but when asked about it, he did not know what it stood for.

Most developers worked from 10am until late. Lunches usually lasted half an hour and were taken in the company restaurant, with visits to a local pub on most Fridays. The majority of team members worked at home on Fridays, however, and those who came to work tended to start and finish earlier than usual. The working hours in the office were heavily influenced by the rush hour traffic, with most team members using the local motorways which are located close to the Stockley Park office. The office was invariably very quiet, with the telephones ringing infrequently.

A variety of soft drinks, tea, coffee and fresh fruit were available without charge to all staff - albeit reduced when frugality measures were increased. Smoking was not allowed anywhere in the office buildings, resulting in cliques in entrance doorways with conveniently placed ash-trays provided. Meeting rooms, servers, even lifts, and so forth, were named in groups taken from: mountains, composers, scientists, universities, planets, writers, precious stones, rivers and history periods.

Principles				Teams					Total
	DT1	DT2	DT3	DT4	Quality	Recruitment	HR	Cisco	
Public									
1.01 N									
1.02 N								1	**1**
1.03 N			1						**1**
1.04 N									
1.05 N									
1.06 N		1			1		1	1	**4**
1.07 N									
1.08 N		1							**1**
Total		**2**	**1**		**1**		**1**	**2**	**7**
C & E									
2.01 N									
2.02 N									
2.03 N	2							3	**5**
2.04 N									
2.05 N									
2.06 N	1								**1**
2.07 N		1		1					**2**
2.08 N									
2.09 N									
Total	**3**	**1**		**1**				**3**	**8**
Product									
3.01 N	1							1	**2**
3.02 N									
3.03 N	1	1		2	1				**5**
3.04 N			1						**1**
3.05 N	4	3	4	2	1	4		2	**20**
3.06 N	7		1		1	1		2	**12**
3.07 N		1	3	2					**6**
3.08 N	5	1		2					**8**
3.09 N	1			3				1	**5**
3.10 N	2	6	1	1				2	**12**
3.11 N	3	2	1	2				1	**9**
3.12 N									
3.13 N									
3.14 N									
3.15 N			1	1		1		4	**7**
Total	**24**	**14**	**12**	**15**	**3**	**6**		**13**	**87**
Judgment									
4.01 N				1					**1**
4.02 N									
4.03 N	1			1	1	1			**4**
4.04 N	3							2	**5**
4.05 N									
4.06 N									
Total	**4**			**2**	**1**	**1**		**2**	**10**

Figure 5. Summary of Level 3 Non-Compliance Found

The culture extended personally to permitting the advertising of staff houses for sale on the intranet (D-250). Approval was also given for the sale of ladies clothes by outside vendors in meeting rooms (D-263). Collections were organised to raise money for Comic Relief.

E.6.4 Relationship between the UK and the US Parent

The UK-based team leaders referred to project managers in the States as bean counters as they are only interested in ticking off boxes for tasks which have been completed, with no regard for what they are, how they are carried out or what influences them. The VTC project manager's private comments to the author whilst attending a conference call involving American project managers consisted of: 'What a load of nonsense,' 'These project manager meetings are the worst ever meetings,' 'This is a learning ground for wallies,' and, 'Their competence is negligible.' He further stated Americans may work longer hours and take fewer holidays than the British, but their output is less than their European counterparts and of poorer quality. The VTC project manager believes UK developers are much better than US developers as A Levels, etc., are quite specific, whereas US education is more generic.

E.6.5 Organisational Structure

At the outset of the case study Telco consisted of the following departments: Pre-sales Engineers, Salesmen (Account Managers), Software Development, Customer Advocacy, Professional Services, Recruitment, Human Resources and Finance.

E.6.6 Recruitment Practice

Recruitment at Telco is handled by several departments/groups which work almost separately from each other. These groups are: Human Resources (HR), Recruiters, E-Recruiters and Graduate Recruiters. Software project managers can also be included in the list as a separate group. Of the 14,000 graduates interviewed by Telco across Europe in 2000, only 12 were made an offer of employment. The Employee Referral Program (ERP) pays bonuses to staff who recommend people for HR to interview who are subsequently offered (and accept) employment by the company. This accounts for 60% of all recruitment. This scheme is not open to students, vendors or temporary agency staff, although open to fixed term contractors. The author queried this with HR arguing the quality of staff recruited is surely of more importance that the origin of the lead and they replied the scheme is currently being looked into. An HR senior recruiter, who trains junior recruiters, suggested the author 'Do a deal,' with a permanent member of staff and split the bonus. When the author mentioned the 'unethical' rules regarding ERP to a junior recruiter, he made no suggestions but simply stated, 'Those are the rules and I go by them.'

All job applicants with the possibility of an offer have a minimum of six interviews. The VTC project manager had twelve - six in the UK and another six in the US. An Induction Course is mandatory for all new employees, during which freebies are distributed which include: a key-ring, Telco cap, suitcase tag, Amex application form, T shirt, ID badge holder and a WPR (Work Place Resources) business card. One new starter told the author that after being with the company for two months, he still had not been granted access to the labs, even though that is where he carried out most of his work.

E.6.7 Software Development Line Management

The line management structure for all software development projects was structured as follows:

VTC -> Project Manager -> Technical Director -> Vice President -> Vice President -> Senior Vice President -> President/Chief Executive Officer

E.6.8 Software Development Practice

A feature based development approach was used called the Great Engineering Model (GEM) - synonymous with Component Based Design (CBD)/Incremental Design. It cost approximately $34,000 to train each developer, with many developers trained up at each of the four support centres around the world. One developer postponed his holiday for three weeks so that his imminent deadline could be met. Software released was supported globally by a 'follow the sun' approach. Bugs found in software are given a priority code between 1 and 3 by customers and a severity code between 1 and 3 by the assigned developer. Bugs were awarded a status for each stage of its resolution: New, Assigned, Investigation, Health and Resolved. More detail is provided in the main body of the dissertation.

E.6.9 Structure of the VTC Software Project Team

The structure of the VTC project team is shown in Figure 13 below, including the project manager's placement of the role of the author. Although the size of the team diminished considerably as a result of the redundancy program, the roles still existed.

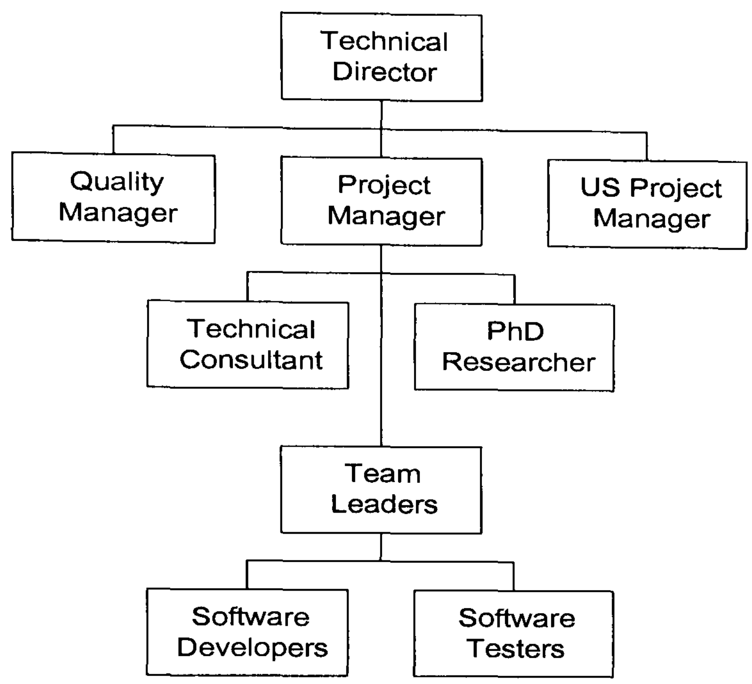


Figure 13. VTC Structure Chart

The team name was changed from NSSU-UK (Network Software Services Unit-UK) initially to VTC (Video Technology Centre). This was to consolidate all the activities between the different groups and enable consistency - resulting in a seamless join from a customer's viewpoint.

Multiple nationalities were present in the VTC team, consisting of English, Scottish, Irish, Indian, Greek, French, Chinese and Cypriot. The names of the participants in the project team are shown in Table 29 below.

Name	Position
Gordon Anderson	Project Manager
Paul Neubert	Quality Manager
Andy West	Technical Consultant
Ted Curran	Team Leader
Bob Cool	Team Leader
Jon Hill	Team Leader
Mark Lewington	Team Leader
Chris Frangoudes	Test Team Leader
Shabaz Yousaf	Developer
Gavin Whitehead	Developer
Mandeep Rohilla	Developer
Anthony Hawkins	Developer/Tester
Gary Lefman	Developer/Tester
Ikenna Orange	Developer
Ioannis Tsiolis	Devloper/Tester
Aideen Fahy	Recruiter

Table 29. VTC Project Team

E.6.10 The Redundancy Program (RiF)

Telco's Chief Executive - John Chambers - announced the restructuring of the company and the redundancies to the workforce on April 23, 2001. Below is an extract from his statement:

> This is going to be a very challenging week for Telco. Our challenging business environment is continuing to impact our worldwide revenue and growth. We are currently experiencing a global macro economic downturn, with resultant significant reductions in capital spending from our customers. Similarly, Telco has undertaken a number of initiatives designed to reduce costs and ongoing expenses. However, given that the largest expense within Telco is employee headcount, this is an area of reduction that we have decided needs to be addressed. As all of you know, we tried everything humanly possible to prevent having to do a restructuring or layoff, but when the market dropped 30% quarter-over-quarter, I think all of us understand why this was needed. There will be about 8,500 people affected in this restructuring and layoff, spanning all lines of business and functional groups. Over 2,500 of those will be contract and temporary workers. The decision on making these changes is the toughest thing I have ever done in my life. I wish I could tell you there was a way we could avoid it. However, the one thing about Telco is we deal with the world the way it is, not the way we wish it could be. Focusing on profit contribution across all key segments of the industry is at the very top of our priorities now. We want profit contribution, market share and technology leadership. Even though what we are going through is very painful, we are positioned better now than ever to break away. I'm very comfortable that we have the culture, the customer focus, and the internal strengths, even during periods of setbacks, to build the company of the future. Our geographic balance, our product balance, our ability to help customers make those transitions, our ability to show how we've done it ourselves, and our ability to take

> advantage of market inflection points are all Telco strengths. I'm counting on us working together to make it happen (D-54, D-55, D-97).

The economic downturn was experienced in many countries concurrently. The Turkish government even halved the value of its currency in an attempt to bolster its economy - with a company previously interested in buying products from Telco then canceling its orders as a consequence. Telco's competitors also had to lay off large numbers of employees: 3M: 500, Marconi: 1500, Cable & Wireless: 2000, Motorola: 3000 and Nortel Networks: 1000. Telco's other competitors include Alcatel, Extreme, Foundry, Cabletron and Siemens. In March 2000 Telco's share price was approximately $80, but due to the decline in the US economy the share price had gradually fallen to $40 at the start of March 2001, and just two weeks later had fallen again to just $20. Consequently the US parent company cut back globally, with every job at risk and, the cancellation of the Goldwing product was decided. The impact of the decline in the US economy on the UK resulted in many companies in the same sector also having to cut back significantly which also included widespread redundancies. With this known, the developers at Telco did not all look for alternative employment with other companies in the same field as their seemed to be little point (L-62-30). All those made redundant received the full package. Some of those let go were offered positions elsewhere in the company. Cutbacks were known to include 8,500 redundancies, but free soft drinks, fresh fruit and the car valet service were still available.

The first meeting was held on 24th April 2001 with the VTC team and HR to formally discuss the planned redundancies, but no numbers were known, only the process to be followed: elect the representatives, conduct a consultation period with management, then those affected directly by the redundancies would be announced. A 90 day period was required, including a 30 day notice period for those affected. Telco paid those made redundant 6 months salary and they kept their company cars and mobile telephones for one month. The UK-based HR department was governed by HR department based in the US. The Goldwing product was expected to be outsourced - but could also be cancelled completely. It was illegal to choose staff for redundancy based on personal skills. The criteria had to be based on technical skills no longer considered necessary. The project manager said he needed everyone as they had very similar skill sets so he had to choose on personal skills. Those made redundant had their details removed from the intranet immediately. Developers made redundant would be rehired as soon as possible, with an amended job description to get around the three month legal restriction if necessary - the HR representative present when this was discussed in a tem meeting half-heartedly agreed as this could be an illegal practice. The VTC project manager expected the economy to recover in the near future and therefore expected to rehire those let go. He told the developers to close down their work professionally as they may have to work on it again later. The project manager stated that everyone in the US office was walking around 'Steaming and angry,' due to the bad way the redundancy process was handled there. As a result, he wanted to do it professionally here. The VTC project manager was not worried about being made redundant himself as the last time it happened to him he found another other employment within a week. He was thinking of getting out of the 'rat race' in four years anyway to possibly drive trucks. He believed it was easier for him to find another similar position than for other members in the team. He expected that if only two developers were to remain in the team after the redundancies have been announced, they too would leave of their own accord shortly afterwards.

The VTC project manager tried to keep the team informed on a daily basis, but rarely had any definite news. The team had a meeting and a team leader, acting as a representative for the others, told him they did not want to know anything other than definite news and, not subjective speculation. The project manager was taught by HR in the week of the redundancy announcements that the developers could be expected to follow a model of emotional upheaval and, could be treated according to where they were perceived to be in the model. He admitted to being at his worst at the end of the week when those made redundant no longer came to the office.

After an email was sent out stating the technical director and a senior project manager had been made redundant, along with many more in the US, the mood in the VTC team office became more serious. Just prior to the redundancy announcements, the project manager told the team to come in every day so as to be present when the news was given out. The team was reluctant as they had finished their work and most had a long journey in. An email was distributed in the US requesting an increase in productivity among those still waiting to hear as it had dropped since the cutbacks had started to take effect. Three VTC graduate developers expected to be made redundant but were not too worried because of the package offered (6 months salary). One had slowed down his development work so he could do some online study before he might have to leave. After the technical director had been let go he told the project manager that was going to be 'severely affected'. The project manager then told every department he could think of around Europe and asked them to contact the US and convince them that his team was needed. This they did, and the decision to scrap his team completely was ultimately abolished in favour of a reduction (L-154, D-33).

When the redundancies were being confirmed to the affected team members, one developer copied files onto two boxes of floppy disks before leaving the office. Each person leaving had to hand over their ID badge, laptop and Amex card. The goodbyes were relatively easy as everyone agreed to meet for lunch in the pub the following Friday. As the day progressed, one developer continued to study online for a qualification, although he had packed all his personal belongings into boxes anticipating his own redundancy. In the late afternoon, the remaining developers were invited into a meeting room together and told they were to be kept on. At the pub lunch following the redundancies, all but two developers turned up. One developer suggested to the author that he keep the laptop issued to him as the company would have loads of spare ones when the redundancies had been implemented in full - he considered doing the same. When the author reminded him he was studying ethics he became uncomfortable and changed the subject.

In December 2000 the project manager was told to increase the team size to 34 by June. By June 2001 it was down to 5 (the project manager, two developers, two supervisors and the technical consultant - plus one supervisor on long term sick leave. The project manager stated than when going for job interviews and asked what you are looking for in terms of salary, the developers should say 'I was valued at Telco at £x per year,' and not, 'I'm looking for £x.'

The redundancy packages offered were known to be larger than (legally) necessary as Telco additionally wanted those developers affected to be open to the idea of possibly returning at some point in the future and, not feel resentful towards the company (I-ET-30-3). Furthermore, the large package offered was for the benefit of those unaffected by RiF. A feeling of insecurity by developers who were unaffected by a redundancy program which offered just one month's salary as compensation might have thought that if there was going to be another round of redundancies, they might as well seek alternative employment immediately and not suffer again the anguish incurred this time around (I-TW-22-8). At this juncture in the redundancy program, developers were told to attend the office daily as the notifications regarding redundancy were imminent - despite there being little or no development left to be done (L-152).

The team members retained were: Gordon, Andy West, Shabaz, Mark Lewington, Ted, Bob Cool and Gary. The team members made redundant were: Ioannis, Anthony, Chris, Melvin, Mandeep, Ikenna, Jon Hill, Fred and Gavin. Those not offered alternative employment: Gavin, Ioannis, Fred and Ike - Ioannis and Fred due to 'first in last out', Ike due to poor performance not improved and, Gavin unknown. Before the redundancies were announced, the project manager had considered setting up a new team (if his was laid off) to work for an outsourcing company. He later found out that the technical director had had the same idea, but the project manager did not think he would have been successful as he had previously messed the company around regarding their contract.

E.6.11 Role of the Author

Initially the author's role entailed assisting the VTC project manager with both project management tasks and, the preparation for ISO 9001:2000 certification. When the case study began, certification was planned for November 2001 but was later postponed due to an economic downturn. The project management tasks included maintaining a Gantt chart using Microsoft Project 2000, showing all the tasks associated with the projects, namely Goldwing, Lotus and Mistral. Goldwing was the largest project and required most of the available resources. The ISO preparation began with an internal audit, but due to the mass redundancies, the results of the internal audit were not presented to the remaining team members or project manager. Shortly after the internal audit, the author administered a questionnaire on the VTC team members. The questionnaire results were later not used but the exercise did facilitate the direction of the research. The initial overall plan, therefore, was as follows:

- Establish research direction
- Assist in project management tasks
- Assist in preparation for ISO assessment
- Collect relevant data to meet research objectives

Appendix F. Graphs of Evidence Collected

The evidence collected of non-compliance to all the principles is now presented, followed by explanations of how they occurred. The charts below show the number of incidents per project team/business unit for each sub-area of each principle. The number of occurrences is shown on the Y (vertical) axis, with the four development teams (1, 2, 3 and 4) and Quality (R), Recruitment (R), Human resources (HR) and Telco (T) shown along the X (horizontal) axis. Multiple colours are used for each principle in the tables to separate the sub-areas, for example, red for Principle 1.01, green for 1.02, blue for 1.03. etc. (The 'N' indicates Non-compliance to the principle). As tables can only present a maximum of six principles areas in any one table - restricted by the software used - multiple tables are presented for principles containing more than six. Principle 1 consists of 8 sub areas and, therefore, has two tables - for Principles 1.01 to 1.06 and 1.07 to 1.08.

F.1 Non-Compliance to Principle 1: Client & Employer

Principle 1: 'Public' is summarised as 'Software developers shall always behave appropriately to public interest (health, safety and welfare)'. A breakdown of the non-compliance found across all the project teams and business units is shown in Tables 30 and 31 below.

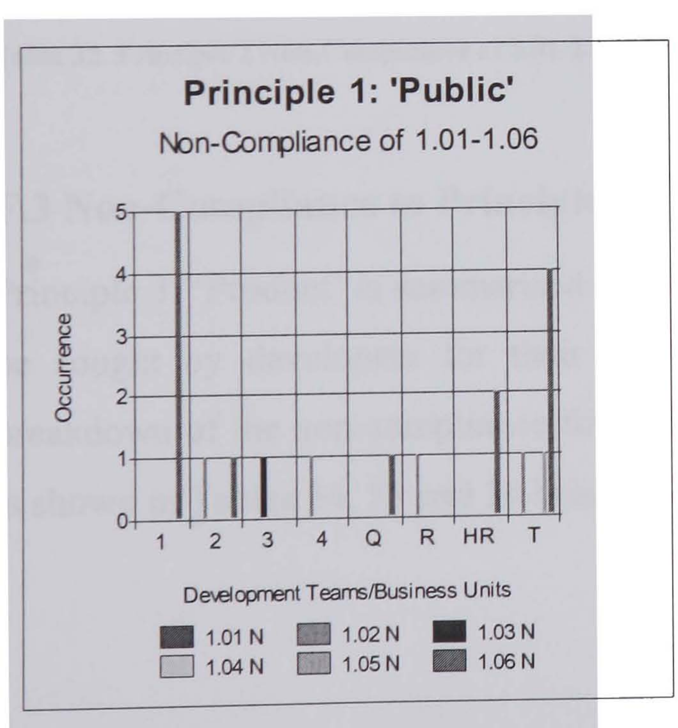


Table 30. Principle 1 Non-Compliance of 1.01-1.06

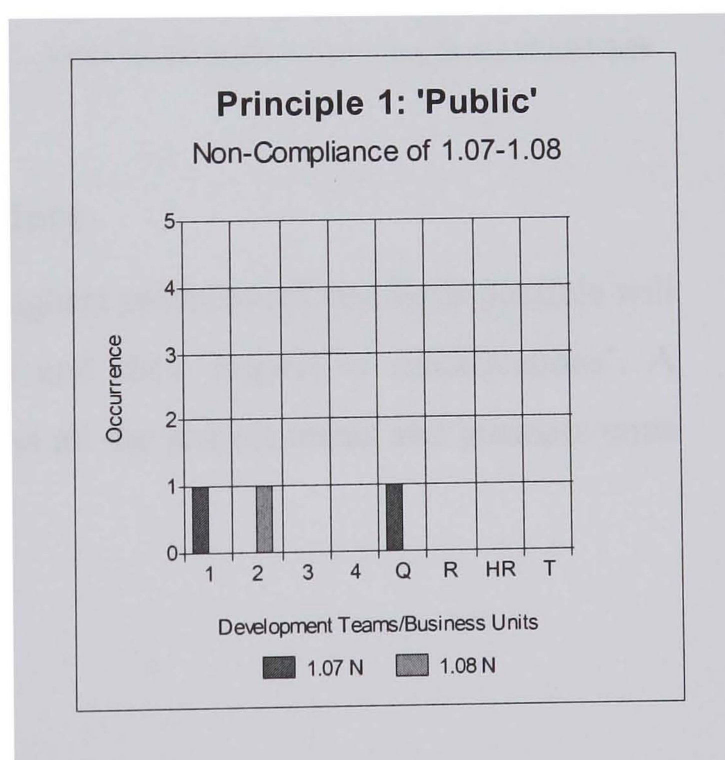


Table 31. Principle 1 Non-Compliance of 1.07-1.08

F.2 Non-Compliance to Principle 2: Client & Employer

Principle 2: 'Client and Employer' is summarised as 'Developers will always behave in the best interest of their employer and client - as long as it is consistent with public interest'. A breakdown of the non-compliance found across all the project teams and business units is shown in Tables 32 and 33 below.

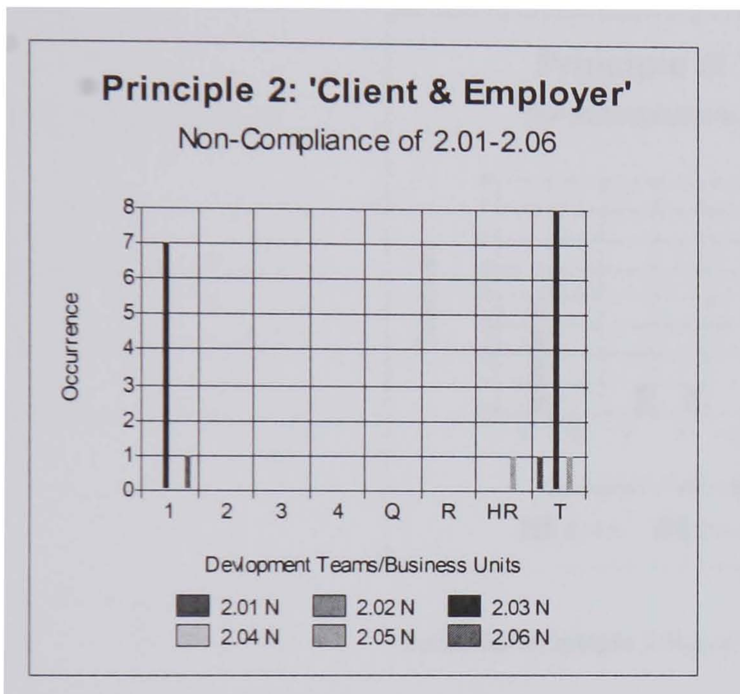


Table 32. Principle 2 Non-Compliance of 2.01-2.06

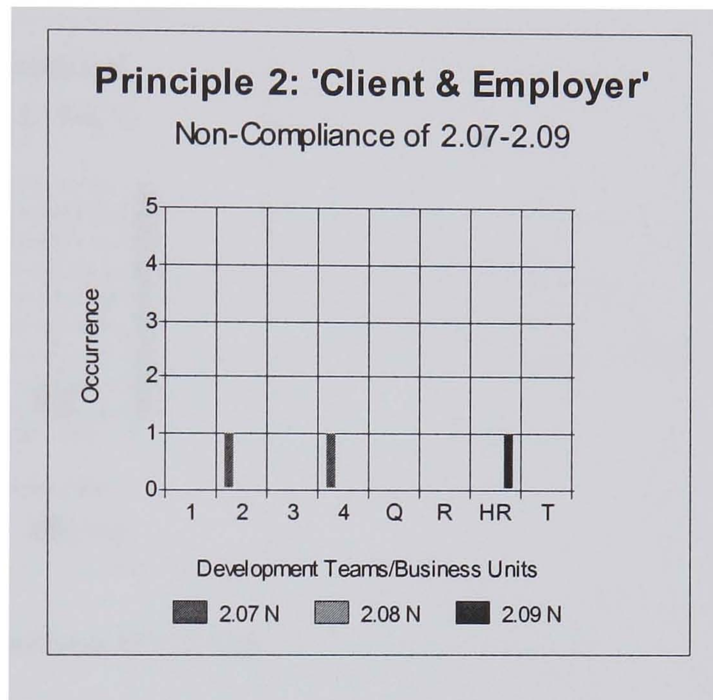


Table 33. Principle 2 Non-Compliance of 2.07-2.09

F.3 Non-Compliance to Principle 3: Product

Principle 3: 'Product' is summarised as 'The highest professional standards possible will be sought by developers for their products and their respective modifications'. A breakdown of the non-compliance found across all the project teams and business units is shown in Tables 34, 35 and 36 below.

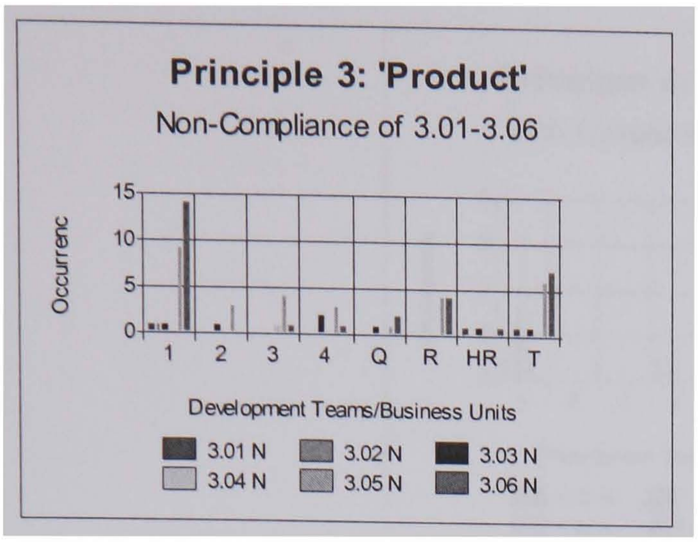


Table 34. Principle 3 Non-Compliance of 3.01-3.06

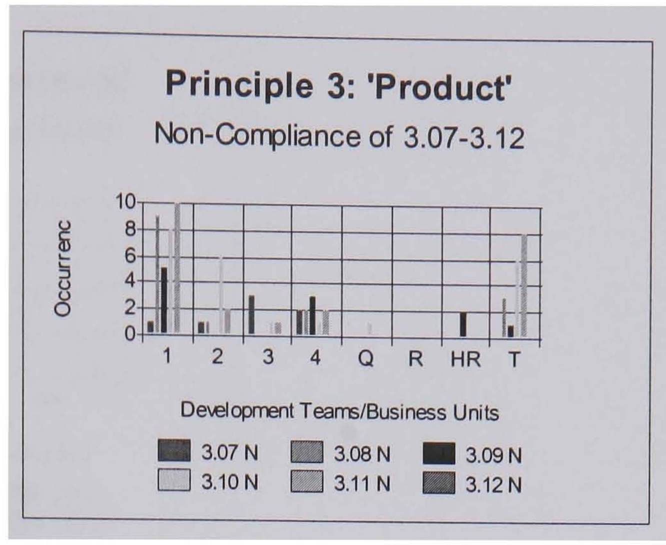


Table 35. Principle 3 Non-Compliance of 3.07-3.12

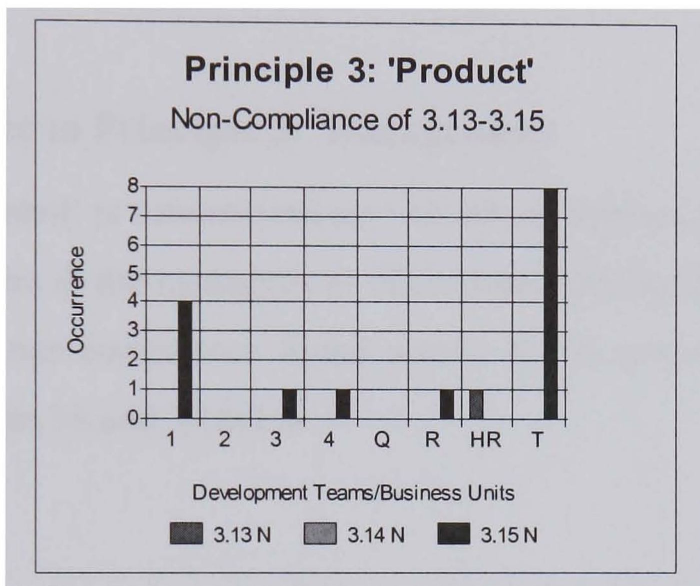


Table 36. Principle 3 Non-Compliance of 3.13-3.15

F.4 Non-Compliance to Principle 4: Judgment

Principle 4: 'Judgment' is summarised as: 'Developers will exercise integrity and independence when making professional judgments'. A breakdown of the non-compliance found across all the project teams and business units is shown in Table 37 below.

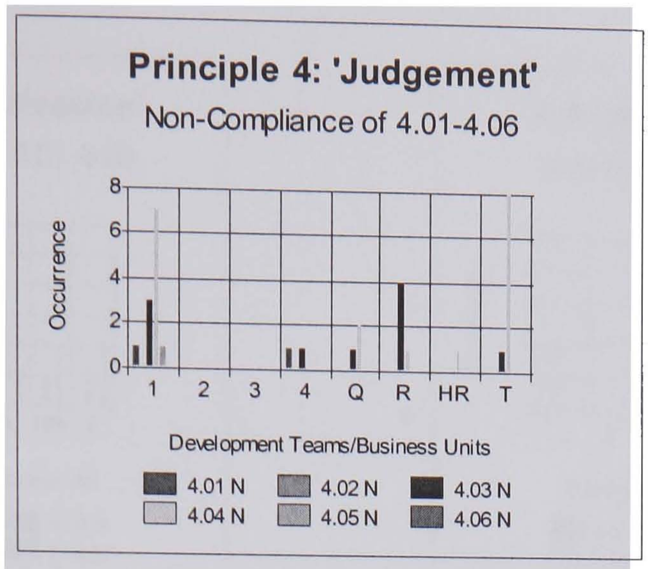


Table 37. Principle 4 Non-Compliance of 4.01-4.06

F.5 Non-Compliance to Principle 5: Management

Principle 5: 'Management' is summarised as: 'An ethical approach will be promoted and exercised by developers in the management of software development and maintenance'. A breakdown of the non-compliance found across all the project teams and business units is shown in Tables 38 and 39 below.

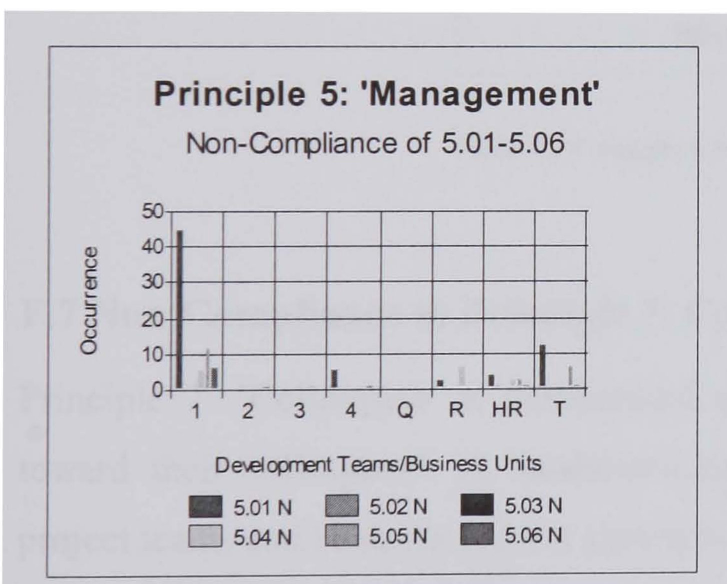


Table 38. Principle 5 Non-Compliance of 5.01-5.06

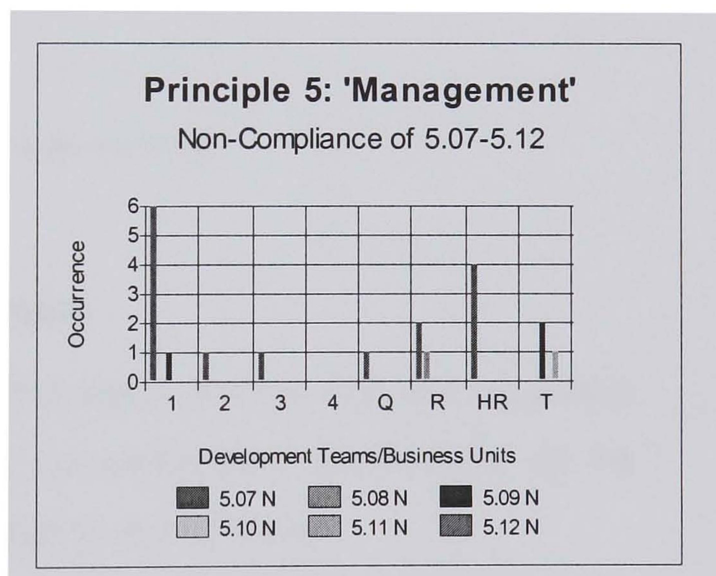


Table 39. Principle 5 Non-Compliance of 5.07-5.12

F.6 Non-Compliance to Principle 6: Profession

Principle 6: 'Profession' is summarised as: 'The integrity and reputation of the profession will be increased - consistent with public interest'. A breakdown of the non-compliance found across all the project teams and business units is shown in Tables 40, 41 and 42 below.

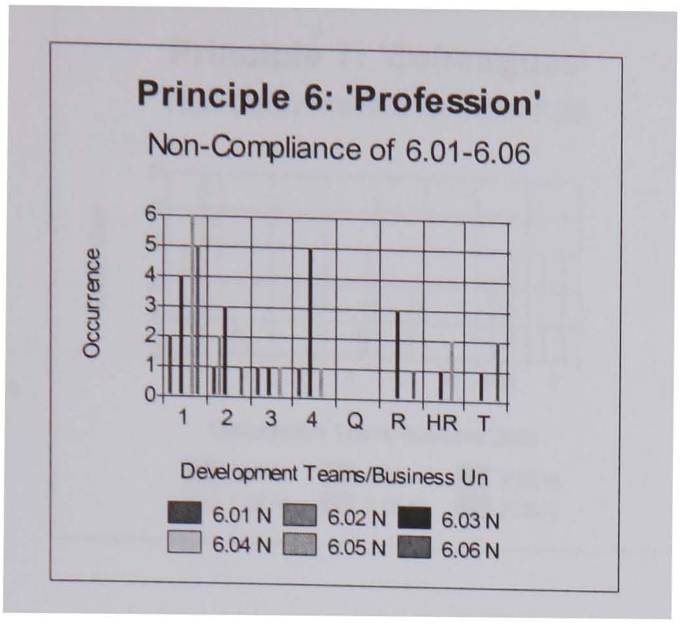


Table 40. Principle 6 Non-Compliance of 6.01-6.06

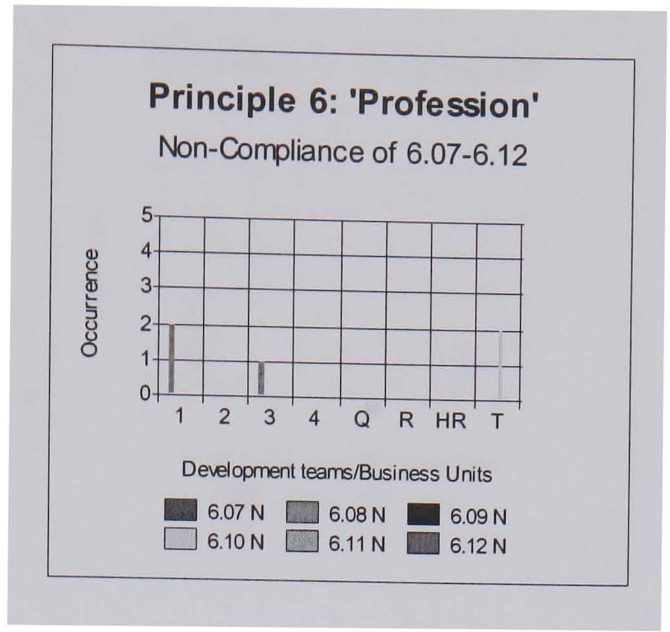


Table 41. Principle 6 Non-Compliance of 6.07-6.012

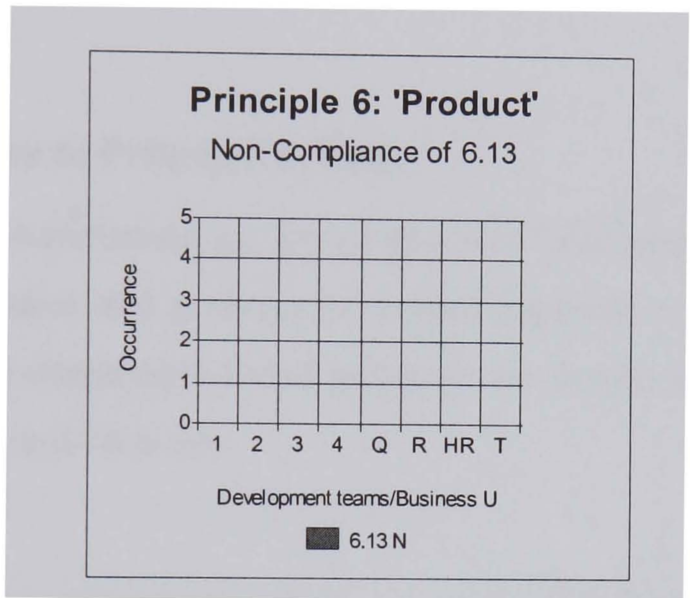


Table 42. Principle 6 Non-Compliance of 6.13

F.7 Non-Compliance to Principle 7: Colleagues

Principle 7: 'Colleagues' is summarised as 'Developers will be fair and supportive toward their colleagues'. A breakdown of the non-compliance found across all the project teams and business units is shown in Tables 43 and 44 below.

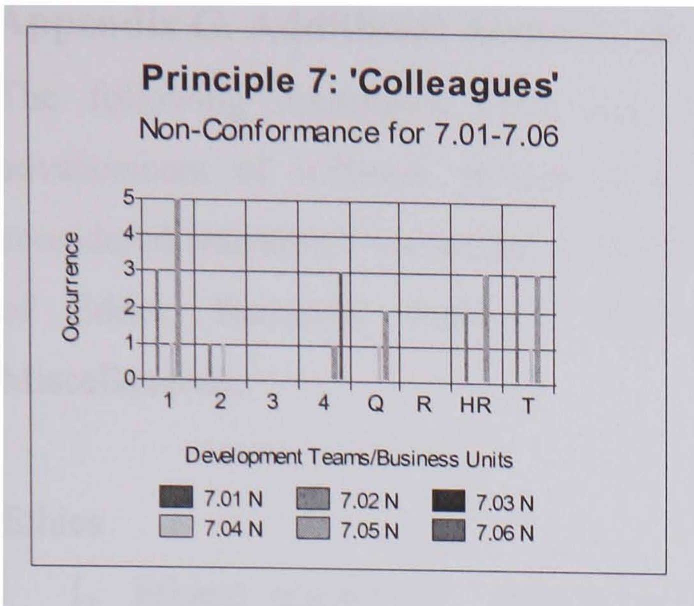


Table 43. Principle 7 Non-Compliance of 7.01-7.06

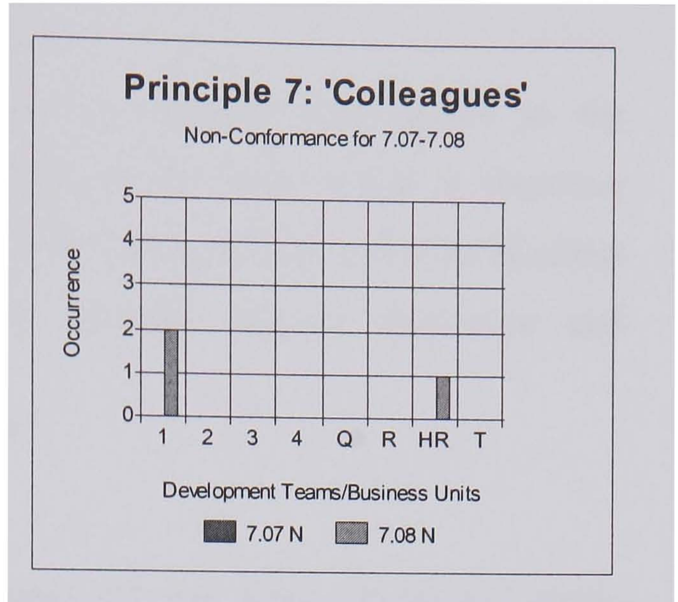


Table 44. Principle 7 Non-Compliance of 7.07-7.08

F.8 Non-Compliance to Principle 8: Self

Principle 8: 'Self' is summarised as 'Developers will take part in life-long learning of the profession's practices and promote an ethical approach to the practices used'. A breakdown of the non-compliance found across all the project teams and business units is shown in Tables 45 and 46 below.

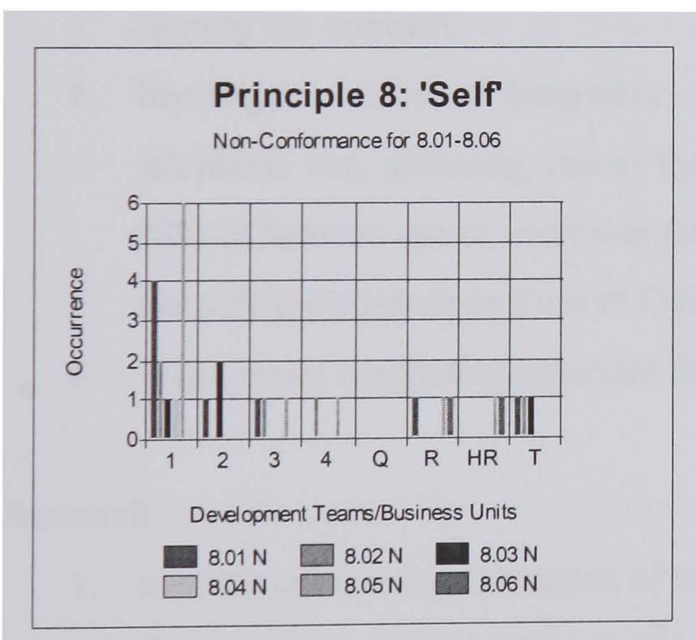


Table 45. Principle 8 Non-Compliance of 8.01-8.06

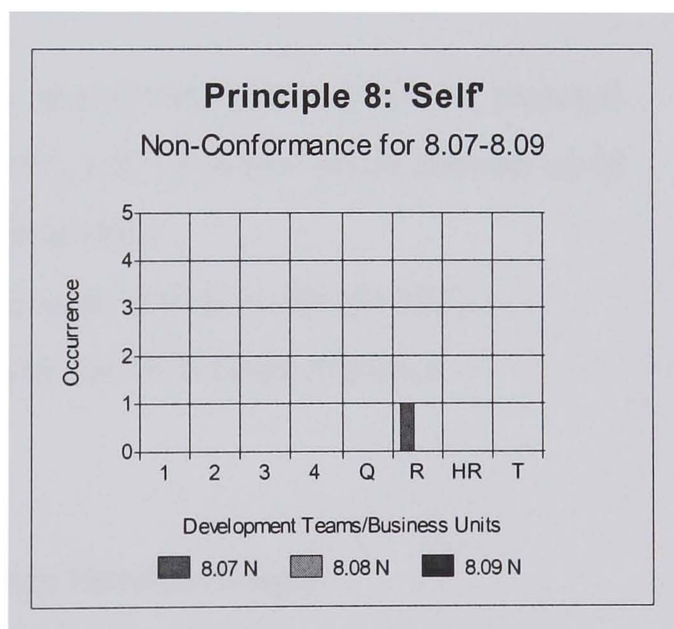


Table 46. Principle 8 Non-Compliance of 8.07-8.09

Appendix G. Additional Areas for Research

The following disciplines have been identified as possible contributors to the advancement of software development and research into these areas is therefore considered warranted. To aid the reader, they have been categorised under the headings of Ethics, Research, Software Development, Quality, Human Resources and Miscellaneous.

Ethics

1. Ethical investment - investment in companies that meet ethical and moral criteria specified by an investor
2. Investigate the Internet and Engineering Task Force (ITF) for adherence/standards/ ethics etc.
3. Identify which professional bodies across Europe require knowledge of ethical issues (e.g. Eur Eng) and how this can be extended in to UK positions
4. Examine the management of ethical conflict between departments in different locations (same building, national, international)
5. Identify the ethics involved in contracts. E.g. at negotiation, it was said explicitly that the contract content and reality of working in Telco would be different. Also, no management was present at the only meeting held prior to signing the contract
6. Investigate ethics of behaviour in the workplace, with regard to personal telephone use, planning, use of the Internet, etc. (A recent study showed up to 48% of Internet use at work was for personal use.)
7. Identify the purpose and use of QuickEthics, an ethical model (D-333)
8. Write paper comparing practices found with the BCS Code of Ethics

Research

1. Identification and codification of case study research designs
2. Clearly define the strategies and techniques for analyzing case study evidence
3. Construction of tests to determine the suitability of a student to carry out a particular research method, e.g. a case study
4. Identify clearly the start and end of programs, decisions, implementation processes and organisational change when identified as the 'case' to be studied
5. Identify how the typical exclusion data collection from other aspects of the research process can be avoided

6. Identify how data collection techniques can address or minimize the design problems of case studies
7. Investigate new methods for testing the reliability of repertory grids
8. Write a paper on transcribing interview cassettes: experiences and techniques, how to record coughs, 'erm', pauses, laughter, questions from interviewee, etc. - if necessary
9. Ascertain how the health industry manages to achieve an average 70% response rate of questionnaires submitted, in order to try and improve the very small percentage achieved in IS
10. Determine the effect of response rate, quality and quantity of electronic questionnaires administered compared to hard copies completed and, time taken for completion
11. Determine the appropriateness of the following statistical packages for IS research: SPlus, SAS, Minitab – and compare with SPSS

Quality

1. Contact Marc Lederman (mledermn@Telco.com) who is responsible for creating user documentation. Establish any standards adhered to, how ambiguity is avoided, documents are complete in meeting their objectives, etc. (D-75)
2. Compare membership numbers, benefits, requirements, etc, between the IEEE/ACM Code, the BSC Code and the (C)IPD
3. Evaluate the work of the American Productivity Quality Center (APQC) regarding best practices and how it might help IS (D-223)
4. Linguistics and Philology - study and use of language

Software Development

1. Ascertain the suitability of project manager decision making tools which accommodate un/certainty - in the scientific disciplines of Operational Research, Statistics and Economics
2. Research the English and American project management professional body 'Body of Knowledge' (D-349)
3. Compare Telco's development flow chart (D-30) with, for example, Yourdon's and discuss
4. Conduct similar research with Telco's competitors: Nortel, Lucent Technologies, Sienna, WorldCom, Extreme, Foundry, Cabletron etc., also at: Racal Radar Defence in Crawley, and GEC Marconi in Harrow, Middx.

5. Conduct research at Miercom with regard to testing as it offers independent lab testing for network solutions
6. Investigate how many assumptions are made by the team over a period and which were correct and which were not, with their respective consequences
7. Establish the processes followed in the airline industry as it is known that a transfer of knowledge and lessons learned at the end of projects - unlike in most IS development
8. Ascertain the value and use of using software called TIMS for testing - an online test management system (L-104-10)
9. Identify the value and possible contribution to software development/management by the following authors: Tummala & Tang (1996), Handfield (1996) and Ridley (1997)
10. Identify the use and value of using MML (Man Machine Language) - used by Telco developers. Also for MDL (Message Definition Language)
11. Compile a list of carefully chosen questions to enable a project manager and customer to determine whether both sides have understood the requirements (completely and correctly) put forward for any given new system, so that a consensual agreement can be realistically reached
12. Law - recent developments in the area of IS

Human Resources

1. Compare Belbin's team roles with those used by Telco and those found in the teams observed (D-197, D-366)
2. Identify the criteria used in selecting the head of Telco as 'Best Boss in America' and whether it accurately reflects the practices found in the case study where appropriate (D-185)
3. Organisational behaviour - formal goals, procedures and administrative arrangements
4. Management theory
5. Penology - study of punishment

Miscellaneous

1. Identify, if any, the codes of practice at the following professional organisations and ascertain how adherence is achieved and maintained, if at all: The Royal Society, The British Academy, The Law Society, The Devonshire Association for the Advancement of Science, Literature and Art, The Royal Institute of

British Architects, The Royal Astronomical Society, The Royal Society of Chemistry, The Royal Economic Society, The Royal Geographical Society, The Geological Society, The Royal Historical Society, The Linnean Society of London, The General Council of the Bar, The Society of Public Teachers of Law, The British Medical Association, The Institute of Physics, The Royal Institute of Public Administration, The Institution of Chemical Engineers, The Institution of Civil Engineers, IEE and The Institution of Mechanical Engineers, the Restaurant Association and all government departments, National Society of Professional Engineers