

4. The uncertainty of ethics in IT¹

Mark Haughey

Fair Work Ombudsman, Australian Government

Introduction

This paper² focuses on ethical issues that are unique to Information Technology (IT) practice, not 'ethics in the workplace' issues or 'sales ethics' issues. These latter areas, while very relevant to IT (for example monitoring the workplace or promoting a product), are generic and existed before IT. As a discipline or profession, IT derives from engineering to a degree (the common job title of hardware engineer or the software engineering concept) and, therefore, some of the ethical issues faced by engineering will translate to IT. This is particularly the case with the development of projects and there will be many an ethical issue (although probably not considered in that light) in getting a project completed on time, on budget and in scope.

Fear, uncertainty, doubt (FUD)

The IT industry, in common with many human endeavours (just think of the recent debate on climate change) is skilled at generating fear, uncertainty and doubt. While most prevalent in IT sales, especially when considering switching suppliers, it also exists in the IT workplace when it comes time to consider extending people's contracts. This is not unique to IT, although the 'lock-in' that exists with a vendor or reliance on specific skill sets (not only product or tool related, but often unique to the particular business utilisation of a product

1 I have no formal training in ethics. My majors were in pure mathematics and computing science as part of my Bachelor of Science (Mathematics) degree from the University of Adelaide in 1979. Since then, I have worked exclusively for the Australian Public Service in the field of IT, mainly in senior executive roles. While I consider myself a public servant first, being an IT professional comes a close second. As a public servant, I have attended ethics training courses, principally in support of procurement activities. While I have a general interest in the subject I want to stress that this paper is not about ethics as a discipline.

2 Disclaimer: This paper represents my own opinions. It does not represent my current or past employers; and it does not represent the views of the Australian Public Service or the Australian Government, past, present or future. This paper is based on my observations of the IT industry: firsthand, observed, read about and speculated on. As such, it has been referred to as 'a view from the trenches'; in which case, be prepared for a bit of mud.

or what has been built with a tool) makes them particularly acute. The reliance on a software engineer easily surpasses that placed on a skilled tradesperson or a medical specialist (you can always get a second medical opinion).

But, what of the behaviour of the software engineer? What ethical challenges do they face? Should they be providing skills transfer? This assumes that there is salary or staffing available to provide an understudy, and that the understudy has the same skill potential. This concept of skill transfer is not common in other professions or trades. The best that is offered is advice on how to avoid the problem in the future (how to conduct necessary maintenance or preventative medicine, perhaps). What the IT professional faces is the belief that they can somehow transfer their skills. This is not true for all recipients of IT services, particularly in the hardware area or packaged software, but, in the development and maintenance of application systems, it is generally accepted that in-house staff can support the resultant system. In the absence of in-house IT staff, the solution is to contract the developer to provide ongoing support. A key component of this transfer of knowledge is through documentation, which of course is one of the first things that is negatively impacted if there are pressures on the project.

Should the IT professional be promoting solutions that have a transition path, which can be supported by a lesser skill base? Often this was a business requirement to start with, but invalidated as complexities increase (and any successful system is sure to be extended in innovative ways and even unsuccessful systems will have a series of fixes applied to remedy shortcomings). As this tends to happen after the initial budget has been consumed, the upkeep of documentation and other maintenance artefacts will often suffer. This might not be a problem if the people supporting were involved in the development and if they have the support as their primary focus.

Permanence

What makes IT workers a permanent fixture in a business? Unlike an engineering project that builds something concrete, there is often a less tangible outcome resulting from an IT project, but a much more flexible product. When a tangible product does not work it can be a project in its own right to rectify the malfunction (for example, product recalls, or even a return to the drawing board). With IT, it is often a simple fix, often as easy as restarting the computer or downloading a patch.

Is this a question of maintenance? Apart from operation and administration, the maintenance of a software system requires at least the same level of skill that was used in developing it (especially given the issues that can exist around

the system documentation). When enhancements are added to the maintenance work, the value of IT as part of the business quickly becomes apparent (in the way that elite sports teams will have their own medical support). The ability of any significant business to function, to enter new markets or deploy new products or services, to meet changes to government regulations or gain insight into its business from the data it has collected in doing business, is directly tied to its IT systems and the ability of its IT workers to safely deliver enhancements and maintenance. This reliance makes the application of fear, uncertainty and doubt so effective in the IT industry.

The move to commercial, off-the-shelf software lessens this, if the customisation trap can be avoided and the vendor and any underpinning technology is sound. Often, however, it is just a deferral of the challenge, which is only alleviated by IT becoming a commodity item. Indeed, this has happened across the IT industry for a variety of things, most obviously those items in the end-user consumer space where scale provides the volumes for commoditisation to thrive. But what happens when the software is unique to a business?

Precision

As with the precision of mathematics, in computing there is the ability to programmatically represent real-life systems. As such, IT is about provable facts: the system did exactly what it was programmed to do. There is no room for ethical dilemmas as there is no choice. Bugs aside, an IT system does what it was programmed to do. Given that premise, there is no ethical issue with an IT system: it is black and white. An IT system treats everyone with the same level of access in the same way. It behaves identically each time that the IT system is used. There are no 'What ought I to do?' conscience challenges. While I have no data on it, I have heard many IT people described as black and white as well. This polarity can lead to the problem of a finished IT system behaving in the way that it was specified to function, not in the way it was necessarily intended to operate.

Ethical problems certainly arise in the creation and the subsequent use of IT systems — this is where the non-IT people get involved. What ethical challenges, however, might be presented to the IT people during the creation of an IT system?

The journey

Once initiated, the creation of an IT system has three aspects:

- A set of specifications
- A timeframe
- A set of resources, funding or treasure

To use the analogy of a journey: destination, time of arrival, and transport. What is really wanted is the IT system, created according to the specification, delivered on time and under budget. This is where the uncertainty starts to creep in.

Heisenberg uncertainty principle

To quote Wikipedia 'the uncertainty is an effect caused not just by observation, but by any entanglement with the environment'. And, in building an IT system, we have three potential moving surfaces: specifications, time and resources. Closely observing one of these three in an IT development — the journey — can cause the other two to distort unpredictably. Just as a constantly whining 'are we there yet' can produce an apparently much longer journey, locking in the delivery date for an IT system can see the specifications not being met and the costs going up. It is also true that the specifications are only final when the system is delivered and that the ultimate documentation of the specifications is the system's code. Of course, a skilled IT workforce is required to be able to read the code.

The dilemma

This uncertainty gives rise to unethical practices:

- 'We don't understand the specifications or the timeframe but will just ask for lots of money in the hope we can work something out.'
- 'We can never meet the timeframe so let's cut some corners to come in on budget.'
- 'They want it now so it's going to cost them dearly.'
- 'For this amount of money, this is as good as it gets.'

There is nothing unique to IT in these practices; they would be faced by most construction projects. In taking short cuts, the risk is not that certain things will

be incomplete (for example, the composition of helpful error messages), but that the absence of such components will be acceptable because of the assumption that they can be easily added later. Indeed, this may not be an issue at all, but something that was negotiated with the business and a joint decision made. But, if the decision process is obscured in the project details, and the programmer chooses to reuse code, for instance, in multiple parts of an application rather than create it as a callable subroutine, is this what they ought to have done? Given these challenges — and with IT projects there are many more beyond these fundamental risks — how do we guard against them?

Educate the observer

This was a strategy that was employed widely and, possibly, successfully early on, before everyone became an expert when they got a PC or their nephew went to college and became a whiz at churning out systems in some small scale development tool. What is needed is a level of IT-maturity in business: experience with similar-sized implementations and requirements; the possibility of having worked with the same development team before, using the same tools and methodologies; and, knowing what tolerances should be built into the funding to cater for reasonable potential delays.

While IT awareness in business has increased, the focus is more on the use of good project management methodologies that allow for informed and reasonable observation. Such an approach provides a more holistic view of the project, the ability to manage multiple developments comprising different teams and tools (project management providing the common language) and to allow for changes in the business people engaged with the project. Failures of course, still occur, for example, when:

- The (absent) project manager is told a story different to that which is unfolding: 'We ought to tell the manager we're running behind, but that will only require us to write reports on why and put us further behind so we'll say everything is okay and work a bit harder to catch up.'
- The methodology employed was unsuitable for the size of the development (sledgehammer to crack a walnut or, more frequently, a bike to take the extended family on an African safari): 'I just need an issues list.'
- Or, just plain misunderstandings or poor communication creates confusion and delays (standard human failings).

Chunking

A good protection against these failures is chunking: breaking a large activity down into more manageable pieces. To use the journey analogy, it is to have frequent rest breaks. Breaking up IT developments into phases is an art form. Often, it is sufficiently tricky simply to identify meaningful milestones. It can also make funding harder in a variety of ways (business might be looking for a big investment or conversely, having made a small investment, may not want to make further ones). It all comes back to the maturity of business. Have we bitten off more than we can chew? Is a system like this appropriate for an organisation of our size? What are other, similar businesses doing? How can I prove the concept without betting the business? IT practitioners can help, but are often at risk of being blinded by business enthusiasm, or being lost withing a big project or, and this is more likely, excited at the prospect of using some new technology.

Value for money

This leads to IT governance — the accounting for the business value of the money spent on IT. It also needs to cover live systems and not just those under development. As an aside, it is much easier to commission a new system than terminate one that has got into trouble; and, for some obscure reason, it often seems that it is far harder to actually turn a system off that is no longer in use.

But how is business to know if they will get value for money? It can be hard getting accurate costs for an IT system. Even when it goes live, there is no certainly that all costs have been identified, especially those associated with maintenance. This represents further uncertainty for businesses, especially given the pace of technology change. Is it possible to relate the IT cost to a business transaction? If the cost of the IT system stays static, wild swings in business volumes can alter this figure dramatically — the dollar per transaction approach.

Expectation management

The business has requested an IT system and has a range of expectations about the system and its delivery. Many of these expectations will be associated with risks, but often hidden by assumptions, some obvious and some unstated.

In order to manage expectations, effective communication between the IT professional and the business is vital: it must be timely, relevant and regular.

There needs to be attention to the medium used for the message — how ought news to be conveyed? Face to face or a lengthy written report with the bad news hiding somewhere near the last page? Or, perhaps, a post-it note left on the door (or on the minute outcomes in the file so it can be removed later to preserve reputations)? Again, should I try email, SMS or a phone call? This all requires careful thought.

The reporting process needs to have been established in advance — a regular communication or governance arrangement put in place. It has to fit the culture of the organisation and the personality of the managers (it is no good sending the boss an email if they never check their inbox). The message needs to be composed in the language of the business, which is not easy if the IT shop is isolated from the business. Along with effective communication, the next two items also help with expectation management.

Engagement

Engagement by IT professionals is vital — they must be part of the business. Most businesses require that individuals perform their tasks without harming anyone in the process. Engagement shouldn't be that hard, although, to be fair, communicating with some people can be a challenge. Where the systems required are central to the business and are the things that make that business unique, the developers need to be associated with the business because, for one reason, the system will encode much of the intellectual property of the business processes and procedures. Heavily outsourced systems create a distance between the developers and the business, where one should not exist.

Being part of the business removes some of the uncertainty from when it is time just to say 'no' to an unreasonable or poorly conceived request, versus just getting on with the job. Being part of the business lessens the 'little knowledge of the business' problem; it at least alerts the developers to potential scalability requirements and it also allows an informed decision on how long to go on with a struggling project or system (unlike building a bridge, there are often simple fixes that can be made to an IT system to make two disparate parts meet).

Packaging

Unique challenges in IT relate to its often-intangible nature. This is why there tends to be a lot of packaging around off-the-shelf software. The old programs that I wrote using punched cards used to look impressive, but even that could be misleading if it was padded out with embedded comments or, more worryingly,

highly embedded IF THEN statements when a single CASE statement would have been more elegant. The other challenge is the rapid creation of the IT industry and profession and the extent to which it now touches everyday essential activities. These facts further cause expectations to overflow and put pressure on people when they are thinking of what they ought to do.

Conclusion

What is to be made of my journey? As a chief information officer, with over 30-years experience in IT, working for what I would say is a very ethically aware employer — the Australian Public Service — I have seen that the demands on IT have become increasingly complex. Complexity breeds ethical challenges and IT is just as much at risk of ethical issues as any other profession or human endeavour. What can be a right or fair answer can often distort under a quest for accuracy — the uncertainty principle. To reply to a question with ‘Yes and No’ might be truthful, but it is generally not very helpful. It might not even be fair.