

6. ICT is not a profession: So what?

John Weckert and Douglas Adeney
Charles Sturt University and University of Melbourne

What is a profession?

Information and communications technology (ICT) is not a profession in any significant sense, but this is not a slight on ICT. It is, of course, a profession in an insignificant sense. If a person develops software for a living as opposed to doing it as a hobby, that person is a professional software developer. Some people sing for a living and others just for fun. The former are professional singers while the latter are amateurs. In this sense any occupation is a profession insofar as those engaged in it are making a living from it, but this is not what is meant in discussions about whether or not ICT is a profession. What is in contention in those discussions is whether or not ICT is *merely* an occupation or whether it has characteristics, like medicine and law are claimed to have, that set it apart as a profession (this is the sense in which ‘profession’ will be used in the rest of this chapter).

What are the characteristics that distinguish a (so-called) profession from other occupations? The following, commonly suggested criteria for a profession, based on Deborah Johnson’s list (2001), are:¹

- (a) The need to master a special, esoteric body of knowledge. This body of knowledge is usually obtained through higher education — for example, a degree in law — and is required in order to practise in the legal profession.
- (b) Some amount of autonomy for both the profession as a whole and the individuals in it. Professionals have the autonomy to make decisions in their daily work, justified by the fact that they are experts and this is

¹ The Australian Council of Professions definition is this:

A profession is a disciplined group of individuals who adhere to ethical standards and who hold themselves out as, and are accepted by the public as possessing special knowledge and skills in a widely recognised body of learning derived from research, education and training at a high level, and who are prepared to apply this knowledge and exercise these skills in the interest of others. It is inherent in the definition of a profession that a code of ethics governs the activities of each profession. Such codes require behaviour and practice beyond the personal moral obligations of an individual. They define and demand high standards of behaviour in respect to the services provided to the public and in dealing with professional colleagues. Further, these codes are enforced by the profession and are acknowledged and accepted by the community.

based on their training. So, for instance, a lawyer will have the autonomy to decide on the best way to defend his or her client, as opposed to taking orders from some higher authority. This autonomy also extends to the profession as a whole. The profession regulates itself, making decisions about who is able to enter it and to what standards they should be held. It will generally be able to do this with little government interference.

(c) Connected to the autonomy of professions as a whole is the existence of a formal organisation. A profession will usually be governed by a unifying body that is recognised by the government — for example, the Australian Medical Association (AMA). This body may have several roles, such as setting standards for certification and having the authority to expel members from the profession.

(d) A profession will also generally have a code of ethics that both sets the standards of behaviour for its members and signals to members of the public what to expect when they engage the services of a professional. For example, medical professionals are usually taken to be governed by the general principles of beneficence, non-maleficence, autonomy and justice.

(e) Finally, professions are typically understood to fulfil a social function. This will often be connected to a social good, such as health or justice, or concerned with more general social benefits, such as providing the technologies and infrastructures to enable society to function effectively. In general, the idea is that professionals serve the interests of the public. So, for instance, the role of the lawyer is to promote justice, the role of police is taken to be the provision of security, and the role of engineers is the provision of technologies that enable things such as communication and transportation (Johnson, 2001: 60–61).

Doubts about ICT as a profession

We have three reasons for doubting that ICT is a profession. The first is the diverse nature of ICT, the second relates to the debate over licensing and certification, and the third is that there are no compelling reasons to believe that there are any professions in any significant sense at all.

The diverse nature of ICT

One difficulty in calling ICT a profession should be noted, and that is the difficulty of knowing exactly what ICT is. ICT covers a wide variety of activities and is, to a large extent, an umbrella term. These activities include those of:

- information technology managers
- system managers, designers and analysts
- programmers
- software designers and engineers
- technical support
- ICT trainers
- database and systems administrators
- multimedia specialists and web developers
- computer network professionals.

(see also ACS, 2011)

Computers play a role in all of these occupations, but that is not enough to justify calling ICT as a whole, a profession. It does not follow from this, however, that some parts of ICT could not be professions; for example, software engineering. If there is to be serious discussion of ICT and professions, it should probably begin with an examination of which parts of ICT most resemble recognised professions (see Holmes in this volume for a similar position). Currently software engineering is probably the most likely candidate and most of the discussion concerning licensing has been in this field. But it is even doubtful whether software engineering can be considered a profession.

Certification and licensing

Software engineering (SE), as was just suggested, has the most potential to be considered a profession. The vigorous discussion regarding licensing and certification in the industry suggests, however, that SE (and by implication ICT in general as well) is not considered a profession by a substantial number of those employed in the industry. Licensing has frequently been advocated, but just as frequently attacked, and implemented only in very few places. The ACM, for example, after giving the idea consideration, came out against it in 1999. Before proceeding, it is worth distinguishing between licensing and certification. A person is certified as being a qualified member of an occupation if, typically, he or she has satisfied certain requirements considered necessary for that occupation, for example, has passed particular examinations. While certification indicates that a person has certain skills, someone without certification can still practise.

Licensing carries legal weight. In occupations where there is licensing, a licence is necessary to practise legally. In the ICT industry, discussion of licensing has generally been limited to software engineering, but possible certification has been raised for ICT more generally.

The most common objection raised against licensing seems to be that, because SE is relatively new, there is as yet no recognised body of standard professional knowledge that all practitioners must have. This is something that is one of the definitive aspects of other professions (condition (a) above). According to John White and Barbara Simons, the ACM council believed that the 'state of knowledge and practice' was too immature to give assurances that 'the licensed engineer is capable of producing software systems of consistent reliability, dependability, and usability' (2002: 91). Tony Healy (2004) and John Knight and Nancy Leveson (2002) also argue that no generally accepted body of knowledge exists. Others, however, see it differently. Danielle Boykin (2007) discusses the introduction of examinations for SE licensing, something that clearly assumes an accepted body of knowledge. Don Gotterbarn, a well-known campaigner for SE professionalism, has suggested (in conversation) that there is something absurd in maintaining the argument that, because knowledge is still developing in some field, that field cannot be a profession. On that basis, not even medicine would qualify as a profession, given that medical knowledge is still advancing. Some background to this debate can be found in Nancy R Mead (2009).

Another concern is that licensing 'would expose software developers to malpractice suits' (Healy, 2004). This, in itself, is no objection to licensing. Most professionals are in this position and software developers are the odd ones out. This objection to licensing is expressed also by Knight and Leveson (2002), who believe that currently (or in 2002) professional standards for SE are not yet clear enough to make 'best practice' easily discernable. If indeed best practice and body of knowledge are so poorly defined in SE then, perhaps, the objection does have teeth but, as has just been seen, not all believe that the state of SE is so parlous.

While certification is weaker than licensing, as we noted, some argue that many of the same problems arise in ascertaining the criteria for certification (Knight and Leveson 2002).

The argument here is not about whether there should be licensing and/or certification in the ICT industry. It is about whether ICT is mature enough yet to be considered a profession. The doubt within the industry does not of course show decisively that ICT is not a profession.

Scepticism regarding professions

A stronger reason for being sceptical of any claim that ICT is a profession is that good reasons exist to doubt that there are professions at all in any significant sense. Two problems become apparent when attempting to give an account of a profession.

First, standard accounts do not pick out only what are currently considered professions; they clearly encompass occupations that are not normally considered professions. On those criteria, plumbing should be a profession. Plumbers clearly have an esoteric body of knowledge not acquired by the rest of us. The skills of a plumber might not be obtained through a higher education degree, but their achievement does require a number of years of training on top of normal secondary education. Plumbers also have a degree of autonomy regarding how they undertake their work and how they solve particular problems; and they must belong to a formal organisation in order to practise, and this formal organisation has a code of ethics. Finally, plumbers clearly fulfil a social function (imagine a city without proper plumbing). They appear, therefore, to satisfy the criteria for a profession.

Consider the following extract from the Master Plumbers' Association of Queensland (2002):

CODE OF ETHICS, TRADE CUSTOMS AND PRACTICES:

1. It is considered that the health and well being of the community as it is affected by all Branches of the Plumbing Trade is of prime importance and shall be protected to the best of the member's ability. The task of raising the living standards of the community is one entrusted to the members of the Association, ...
2. A member shall observe the highest standards of honesty in all of his transactions, ...

It calls for the health and safety of the community and high ethical standards, and assumes special knowledge and skills when it calls for 'the highest quality of service and true value for money'. This is something that could not be provided without that knowledge, and its inclusion acknowledges a clear social function for the profession. It also acknowledges the vulnerability of their clientele. This is true not only regarding the plumber's work, but also because frequently, clients must give plumbers access to their homes at times when they are not present.

The only real difference from members of recognised professions, say doctors and lawyers, seems to be that plumbers get dirty, with real dirt; but adding a

criterion of cleanliness to the definition of a profession seems odd. In any case, some scientists, for example those working in agriculture and biology, often get dirty, with real dirt, too; so, either they are not members of a profession or plumbing is no less a profession if getting dirty is allowed.

An additional reason for denying that plumbing is a profession, from Gotterbarn, is that members of professions have responsibilities to advise customers 'of potential negative impacts that a new system may have on the customer's business, quality of life, or the environment' (2004: 116). Members of professions do have this responsibility, but so do plumbers. When a plumber is hired, more is wanted than a technically good job; advice is normally required on whether what is asked for is, in the plumber's opinion, a good idea. Again, this does not distinguish plumbers from accepted professions.

The second problem is that the criteria offered seem arbitrary. Picking an occupation, say medicine or law, and then constructing a list of its characteristics and using that list as the criteria for a profession is hardly convincing. Why not choose plumbing or fruit picking? Historically a profession was a vocation, a calling from God. The clergy had a vocation, artisans an occupation, but this too has the ring of arbitrariness.

The considerations so far suggest that the notion of a profession is not useful in discussions of ICT, for four reasons: first, ICT is too diverse to be considered a 'profession'; second, doubt exists amongst ITC practitioners that ITC is ready to be called a profession; third, accounts of a 'profession' do not foreshadow only those pursuits that are commonly called 'professions'; and, finally, the criteria seem arbitrary. Before rejecting the notion altogether, though, it should be noted that it may have a use in encouraging good behaviour. Those who consider that they have a profession, rather than merely a job, may have extra motivation to carry out their work in a more conscientious and moral fashion, in order to uphold the good reputation of the so-called profession and because they are bound by a code of ethics. But, as we have just seen, plumbers too are bound by such codes, so it is not clear that the notion of a profession is important for providing motivation for good conduct. In any case, there does not seem to be much empirical evidence that members of recognised professions behave more ethically in their work than others.

A professional attitude

The notion of a profession may not amount to much, but that of a *professional* is important. The view being advocated here is that someone is not a professional

on the basis of being a member of a certain occupation, a 'profession'. Rather, professionalism comes from having a *professional attitude*. The fictional mechanic, Mr JLB Matekoni, has such an attitude:

Mechanics could be the conveyers of the most serious news ... on those occasions a suitably grave expression was required; one should not give bad mechanical news lightly, as Mr JLB Matekoni had felt obliged to tell his apprentices. He had overheard Charlie telling a woman that her car was *finished*, and on another occasion the young man had told a client that his brakes were the worst brakes in Botswana, adding, *And I've seen some pretty bad brakes in my time!* No, that was not the professional way, not that those young men understood what professionalism was all about (McCall Smith, 2010: 32).

For Mr JLB Matekoni, being a motor mechanic involves much more than being able to fix cars; equally important is how the mechanic deals with customers. He has a *professional attitude*. A professional attitude involves, at least, taking pride in one's work, trying to do it as well as possible and actively considering ways in which it might be done better, looking at it in the broader context of society as a whole, and taking responsibility for what does. Admittedly, there is more scope for the exercise of such an attitude in some fields of employment than in others: a worker doing repetitive work on a factory production line may not have much scope for reflection on how he or she might do his or her job better, or for pride in doing it well. But in a great many types of work, for example, law, medicine, plumbing, and ICT, there is ample scope for a *professional attitude*, and such an attitude should be encouraged and appreciated.

A professional attitude is particularly important in occupations where the person is an expert in the sense of having significant knowledge and skills not had by those outside the occupation. Special responsibilities, over and above those that we all have as human beings, are attached to this expertise, and a professional attitude involves taking these responsibilities seriously. Before elaborating on this, we look a little more closely at the concept of responsibility because of its central role in discussions of ICT professionalism.

The word 'responsibility' has three distinct senses that are important for our purposes. First, there is the *causal* sense: if it is said that lightning was responsible for a fire, what is meant is that lightning caused the fire. It is ambiguous however to say that someone was responsible for a fire. This might just mean that the person indicated caused the fire, or it could mean that he or she is blameworthy (or praiseworthy, depending on the circumstances) for lighting it. This second sense is *moral* responsibility. Someone is morally responsible for an action if he or she played a causal role in the action and if it is appropriate to attribute praise or blame for causing it. A person who is responsible in the causal sense

is not always responsible in the moral sense. The third relevant sense is *role* responsibility. Certain responsibilities go with particular roles. The project manager's role carries the responsibility of overseeing the whole project, while the programmer's role has the responsibility to produce good code.

Someone may not be morally responsible for an event even when that person is causally responsible. This is an important point, especially in the context of professional attitudes in the ICT industry. When is a person (P) morally responsible for some event (E)? Three conditions must be satisfied: (1) P must have caused E, or knowingly allowed E to happen when he or she could have prevented it; (2) P must have intended to cause E, or allowed it through neglect or carelessness; (3) P must have been free to choose to, or not to, cause E. Condition (1) says only that P must have done whatever it was that caused E or not done what could have prevented it. If P rather than some other person, Q developed faulty software, then P is responsible, not Q. The second condition is satisfied if P intended to produce faulty software, or if it resulted from neglect or carelessness, for example, through neglecting to test the software thoroughly. Condition (3) shows the importance of autonomy. If P is insane or has no choice but to develop the software, and within an unreasonable amount of time, then moral responsibility is diminished or absent altogether.

In summary, then, P is morally responsible for faulty software if he or she

- developed it
- intended it to have faults or (more likely) was careless or negligent in some way
- was acting autonomously and not under duress.

Moral responsibility is related to *accountability*, but is not identical with it. Accountability can be taken as moral responsibility but it is not always used in this sense. Someone might be morally responsible for an action but not held accountable. This is commonly the situation in the computer software industry, where software developers are commonly not held accountable for malfunctioning software, even though a good case can be made that they ought to be because they are morally responsible. They are morally responsible because they developed the software (as well as meeting the other two conditions above). Frequently of course, where people are not held accountable, it would be argued that this is simply because they are not morally responsible. It is one thing, then, to urge workers in a given field to develop a professional attitude, or a more professional one, for the sake of their clients. It is another thing to seek to protect those clients by making the workers accountable. Some system of certification and/or licensing can support this move.

It might be objected, reasonably, that this picture is overly simplistic because software is not usually developed by just one person. Normally, the software development will be undertaken by a team, possibly a large one. In such cases, the argument goes, nobody can be held morally responsible. This is the problem of collective responsibility, sometimes called the 'many hands' problem.

Because many people are involved in typical software development projects, when something goes wrong, it is not always easy to say who is morally responsible, and who, if anyone, ought to be held accountable and liable for any damages. One solution is just to say that the group, or organisation, is responsible. In everyday talk we do this frequently, for example, when a large company is responsible for environmental damage. Two problems are worth pointing out here. One is a degree of unfairness. Not everyone in a group is equally responsible or even responsible at all. Not all employees will have played a role in the damage that was caused. The second problem is that moral responsibility is something that can only be attributed to autonomous human beings, not organisations.

While it is probably true that, in most cases where there is collective responsibility, no individual will bear the entire blame, it does not follow that moral responsibility cannot be attributed to many individuals. Many can, therefore, be held accountable to varying degrees. Not everyone in the development team will necessarily bear the same level of responsibility for the faulty software and some, perhaps, will have no responsibility at all. A careless programmer can bear responsibility, as can someone who is negligent in the testing of the product. In most cases, ultimate responsibility must be borne by the project manager whose role is to oversee the project. Responsibility cannot be avoided simply because a large team is involved in software development.

A further aspect of collective responsibility must be noted. While not all individuals are at fault, and those that are do not share equal responsibility, there is a sense in which all, with only several exceptions, can plausibly be held accountable. If I do nothing to try to change the situation of my team's or company's carelessness, I am helping to perpetuate a climate in which faults, mistakes or accidents are more likely to occur. So I can and ought to be held accountable to some extent, even though I did not cause the events, or intend them to happen. In Larry May's terms, I can be morally tainted even if I cannot be blamed for the event itself (May and Hoffman, 1991, chpt 15). The exceptions are where I have protested or attempted to change the situation, or am not in a position of influence; for example, if I could not reasonably be expected to know of the situation (See Feinberg 1970, chpt 9 for more discussion of these exceptions). Given this, all, or most, members of a group, team or company can

be held collectively responsible for faulty software. Software developers can no more avoid moral responsibility for their products than developers and builders of aeroplanes, cars or bridges.

A central component of a professional attitude, it was stated earlier, was taking seriously one's responsibilities and acknowledging that one is morally responsible for one's actions. Three senses of responsibility were distinguished: causal, moral and role. The discussion so far has primarily concerned the first two, but role responsibility is also important in professional attitudes. The practitioners of many different occupations have responsibilities to others — because of their roles — over and above the responsibilities that they have as human beings. An electrician has special responsibilities with respect to the wiring of houses, just as a medical practitioner has special responsibilities towards patients, and software developers do towards their clients. Doctors, lawyers, computer scientists and others have professional bodies with codes of ethics to regulate or guide their members. Builders and plumbers have the same, even if they are not called professional bodies. A professional attitude involves a moral stance. This includes the ability to see one's work in the context of the larger society, how it contributes to that society, and taking moral responsibility for one's work. Codes of ethics attempt to clarify and partly formalise this stance.

A professional attitude in ICT

At the beginning of this chapter we made the twin claims that ICT was not a profession in any significant sense and that this did not matter. The argument has been that there are no professions, in the sense that many claim that there are, but that nothing much follows from that because what is important are *professional attitudes*. These professional attitudes are closely related to the special responsibilities that come with particular occupations, and this set of occupations is much broader than the set of the so-called professions. People in many occupations have special responsibilities over and above the responsibilities that we all have. A medical doctor, for example, has the responsibility to provide health care, and 'the role of the lawyer is to promote justice', as we saw in e) above. We all have some responsibility as citizens to assist others with respect to health care and justice, but not to the degree that doctors and lawyers respectively do. ICT practitioners have special responsibilities, too, with respect to society because of their expertise. Not only their direct clients but also the general public are vulnerable to the systems developed and maintained by these workers. It does not matter if ICT is not a profession and is not seen as one, even by its practitioners, but it is vitally important that those in the industry have professional attitudes toward their work, especially those who develop and maintain computer systems, including networks. This is the area

in which professionalism in ICT has been most discussed and with good reason. These systems have a profound impact on modern life and system failures or malfunctions can vary from annoying to catastrophic. While there have been many calls for greater professionalism, it is not clear that these calls have been heeded. A recent report suggested that professional attitudes in the industry are not as prevalent as is desirable, and the industry as a whole has been reticent to recognise moral responsibility for software systems. The industry clearly is not devoid of people with professional attitudes (see Section Two of this book), but such attitudes do not always come to the fore as much as they should. For example, in a recent study, some of the interview comments suggested that there is not a strong commitment from all ICT workers to professionalism. While only one comment states explicitly that 'Most IT people do not want to be professional', other statements, such as 'Lots of money in the industry', 'Computing has been seen as a way to make money and then leave', 'People don't take responsibility', 'Gap between what the customer wants and what they need. Often they are given what they want with full knowledge that this is not what they need', 'No encouragement to be responsible', and 'Outsourcing is a problem because of lack of responsibility' indicate that many practitioners do have professional attitudes and they are concerned about the substantial number of their colleagues who do not share their attitude (Lucas and Weckert, 2008).

Typical licensing agreements for shrink-wrapped software also indicate a lack of professional attitudes in the industry:

8. Disclaimer of Warranties.

A. You expressly acknowledge and agree that use of the Apple software and services is at your sole risk ..., the Apple software and services are provided 'as is', with all faults and without warranty of any kind, and Apple and Apple's licensors ... hereby disclaim all warranties and conditions with respect to the Apple software and services, ..., including ... satisfactory quality, Apple does not warrant ... that the operation of the Apple software or services will be uninterrupted or error-free, or that defects in the Apple software or services will be corrected (the text has been changed from uppercase in the original to make it more readable) (APPLE INC, 2008).

The above disclaimer makes it clear that no responsibility is taken for faults in software. It is common for new cars, vacuum cleaners and so on to be faulty, but we expect that the warranty will cover their replacement or repair. Responsibility is taken for those products. It is true, of course, that thorough software testing is difficult. It is true, too, that where a problem arises it could be caused by the operating system or the hardware on which the purchased

software is running. Nonetheless, these disclaimers, which are the norm rather than the exception, do suggest that the ICT industry as a whole does not take responsibility as seriously as it could, and taking responsibility is, as we have argued, part of professionalism. The commitment to professionalism in the sense of having a professional attitude as defined earlier, does not appear to be as strong as is desirable.

Conclusions

We have argued here that ICT is not a profession, in any significant sense, for a number of reasons, but, primarily, because the notion of a profession itself is not important or useful. What is important is having a professional attitude and all that that entails, particularly a willingness to take responsibility for one's actions. Encouraging such attitudes should take precedence over efforts to make ICT a recognised profession. ICT practitioners play an essential role in many aspects of modern life — their work has major impacts on the economy and, in the case of safety-critical systems, their decisions can have life and death implications. Having a professional attitude in the ICT industry is equally important as in medicine and law.

References

- Apple Inc, 2008, 'iwork software license agreement', <<http://images.apple.com/legal/sla/docs/iwork09.pdf>> 10 November.
- Australian Council of Professions (ACP), 1997, 'Definition of a profession', <<http://www.professions.com.au/defineprofession.html>>
- ACS, 2010, *Australian ICT statistical compendium 2011*. <http://www.acs.org.au/__data/assets/pdf_file/0019/9307/Australian-ICT-Statistical-Compendium-2011.pdf>
- Boykin, D, 2007, 'Is it time to license software engineers?', *PE Magazine*, December, <http://www.nspe.org/PEmagazine/pe_1207_Software_License.html>
- Feinberg, J, 1970, *Doing and deserving: essays in the theory of responsibility*, Princeton University Press.
- Gotterbarn, D, 2004, 'Informatics and professional responsibility', in TW Bynum & S Rogerson (eds), *Computer ethics and professional responsibility*, Wiley-Blackwell, pp 107–18.

- Healy, T, 2004, 'Licensing developers won't work', *TechRepublic*, 24 November, <<http://www.techrepublic.com/article/licensing-developers-wont-work/6310377>>
- Johnson, D, 2001, *Computer ethics*, 3rd edn, Prentice-Hall, Upper Saddle River, NJ.
- Knight, JC, & Leveson, N, 2002, 'Should software engineers be licensed?', *Communications of the ACM*, November, pp 87–90.
- Lucas, R & Weckert, J, 2008, *Ethics and regulation in the ICT industry*, report for the Australian Computer Society, Centre for Applied Philosophy and Public Ethics, Charles Sturt University, Canberra.
- McCall Smith, Alexander, 2010, *The double comfort safari club*, Little, Brown Book Group Limited.
- Master Plumbers' Association of Queensland, 2002, 'Code of ethics, trade customs and practices', <<http://www.thisplace.com.au/sponsors/mpaq.htm>>
- May, L, 1991, 'Metaphysical guilt and moral taint', in L May & S Hoffman (eds), *Collective responsibility: five decades of debate in theoretical and applied ethics*, Rowman & Littlefield Publishers, Savage, Maryland. pp 239–54
- Mead, NR 2009, 'Software engineering education: how far we've come and how far we have to go', *The Journal of Systems and Software*, vol 82, pp 571–75.
- White, J, and Simons, B, 2002, 'ACM's position on licensing of software engineers', *Communications of the ACM*, November 2002, p 91.