

## On Professionalism in IT

Les Hatton, CISM, Kingston University  
<http://www.leshatton.org/>

I should perhaps issue a health warning here. If you are looking for an article in praise of IT and its professionalism, this isn't it. Before beginning to write, I decided to look up this word in the OED. *Professionalism* is 'the collective qualities or stamp of a profession'. The lead definition of *profession* is 'declaration of belief in a religion' or 'vow made on entering a religious order' and a bit later on we get, 'especially one which involves some branch of learning or science'. Let me muse on some elements of this.

I have to say that I am rather ashamed of being a member of the profession of IT as we practice it today. For a start, it certainly isn't a science. It has no measurement basis, little repeatability and is awash with unconstrained creativity and as such cannot distinguish between change and progress. As a result, software isn't soft and has little or no engineering in it. Instead it is mostly a fashion industry with an increasingly fore-shortened opportunity time for new technologies and a relentless drive to replace the old and tested with the new and untested. This has many negative implications. For one thing, it is increasingly difficult to teach because for every project there are any number of different technologies competing for student's attention. We actively encourage students to work on real projects for industry and as a result one of my colleagues this year had students submitting projects in C, C#, C++, Java, PHP, MySQL, XML, HTML, XHTML, VB.Net on XP, Mac OS X, Linux and even Vista with Eclipse, Netbeans, Ant, JWSDP, Glassfish, DreamWeaver, Developer Studio, .Net with maybe even some Etruscan – I may have missed a few but who cares. Then some industrial observers have the cheek to criticise academia for not producing enough students with the 'skills required for today's knowledge economy'. Well being able to read, write and add up would do for a start but we seem to be busily driving such low-level skills out of our education system as witnessed by the latest news that GCSE Physics is being downgraded to something in between Postman Pat and the subject formerly known as General Studies.

This lack of focus and obsession with fashion allied with historically low esteem, indifferent pay, and little career infrastructure other than promotion to management as soon as somebody gets vaguely competent at building systems, means that students are simply choosing not to do IT anymore and I don't blame them. The bottom line is a catastrophic fall in the number of students applying for engineering and science in general and IT in particular – its something around 50% in the last 5 years in the UK and down again this year. At the same time, industrial activity is picking up strongly which means that there is an increasingly wide gulf between supply and demand. The implication will be that the majority of IT people working on systems over the next few years will be very inexperienced indeed. Given that experience is the single enduring factor in high quality systems, this does not bode well. However good your process is and goodness knows we have beaten software process to death in the last twenty years, the quality of the product still depends mostly on the quality of the engineers building it. We look to process for consistency. In other words, we can be as 'professional' as we like about the bureaucracy and management of software projects but if the engineers are in short supply and of generally low experience, the systems they build will not be very good. Is there evidence of this happening ? There most certainly is.

I have spent most of my computing career studying how systems fail, the general idea being that it is not a sin to make a mistake, it is a sin to repeat one. Well its clearly not a sin to repeat mistakes in IT. In fact it seems to me after all these years that if we deliberately tried to sabotage every system we build, we could not do a much more comprehensive job. Witness the recent House of

Commons Select Committee report detailing the negligent squandering of somewhere in excess of 10 billion pounds a year of public money on grandiose schemes clothed in secrecy, managed incompetently and producing nothing of any lasting value. Over three years ago, the Royal Academy of Engineering issued a rather damning report summarising the problems of the time and saying exactly the same thing. Two years before that, the National Institute for Standards and Technology in Washington said exactly the same thing quoting an equivalent figure of around \$60 billion dollars a year in the US. As far as I can see, nothing much has changed in the last twenty five years. Just how many times do we have to say this before we get the idea ? The same mistakes continue to occur again and again and most of the failures I have studied (lots), could have been avoided using techniques we already know how to do.

Many of the devices I use are full of defects. My television set-up box is a worthless piece of junk and crashes every seven hours according to my records. Its predecessor was even worse. It worked for just under a minute until a little dialog box appeared on the television inviting us to 'Download Software Update'. My daughter beat me to the handset and entered "Yes", just as I said "No !". About half way into the update, the box crashed. The download was not double-buffered so it hopelessly corrupted its own memory and back it went. As I walked through the doors with it, the assistant greeted me with "Download Software Update ?". I nodded. "Put it with the others" he replied, pointing to a stack in the corner. My brand-new upgrade to a condensing boiler on the back of Britain's commitment to the Kyoto agreement having to replace a perfectly serviceable existing boiler with a new one of a lifetime shorter than the residual life of my existing boiler, (you couldn't make this stuff up), was delivered with a programmable controller with the following instructions for rebooting: "Remove left hand battery, remove right hand battery, press button A for two seconds and then A and B together then restore batteries in opposite order". Well of course, how else would you do it ? What happens when you lose the instructions ? How on earth would an elderly person cope with this kind of arbitrary nonsense ? In fact how would any vaguely rational person deal with this ? I neglected to say that it was designed to be a wireless connection to the boiler but had only one frequency and a range of around 100m so if anybody else gets the same one, we can happily control each other's boilers.

Modern cars are increasingly subject to recalls for software problems and even made the New York Times in 2005 with the famous shaking Mercedes and the Ford that baked its back-seat passengers. There have been engine management system recalls, ABS recalls, power-train recalls and of course airbag recalls for those of you happy to sit half a metre from a software controlled bomb. I personally have clocked up an ABS failure, an oscillating sun-roof and interestingly obscure and obviously software-related problems with various radios. I recently drove from Hamburg to Braunschweig in a hired BMW and was unable to work out how to turn the radio off until I hit upon a 'Connect to Non-Existent Sound Source' option deep down in the menu system. I thought it was just me until Jeremy Clarkson reported the same thing on "Top Gear". Whatever happened to parsimony ? More and more I am drawn to how accurate the late and sadly missed Douglas Adam's depiction of computer-riddled future society in the Hitch Hiker's Guide to the Galaxy is likely to be - "their fundamental design flaws are completely obscured by their superficial flaws" as I think he said. Right on the button, Douglas.

Its not just devices. On a more serious note, all my published research in the last 15 years suggests that our many scientific simulations contain defects which have an unquantifiable effect on their results.

The lack of an empirical basis to our 'profession' continues to be a major stumbling block. It is all very well standardising process, languages, tools and anything else we can get a committee together

for, but this is nearly always done on 'expert opinion'. In areas like protocol standardisation where you are simply trying to agree, this has generally been very successful but in many other significant areas where you are trying to *improve*, it has been very unsuccessful. Improvement needs measurement. The standardisation of programming languages is a very good example. It is relatively simple to get enough support to add new features of generally unquantifiable behaviour in terms of reliability. It is the devil's own job to get them out again and so languages just grow and grow. In 1978, the iterative construct on real variables was introduced into Fortran. It was a terrible mistake because the resulting loop is indeterminate. The Fortran committee bravely managed to remove it in 1990 - the very year it was added to C ! Today, the latest standards for both C and C++ are so wickedly complex that no compilers to my knowledge implement the full standard after almost 9 years. In embedded control systems we work instead with the previous C standard now 17 years old because it is still possible to understand it.

After the above, you may think me a pessimist but I am not. We are hopefully at the early stages of the most successful movement ever in IT, that of open source, which has Linux as its flagship. Linux is quite simply stunningly reliable. I have examples in my files of large cluster users quoting > 400 execution years between kernel failures. Linux is built by thousands of willing volunteers around the world with no management bureaucracy. It is built in the programming language C which causes connoisseurs in most programming language experts and to cap it all, it is built using a software process officially deemed chaotic. In spite of, and who knows even because of this, the Linux kernel is arguably the most reliable complex application the human race has ever produced. It would really be awfully nice if we understood this just a little bit better, although when you look at it closely, the phrase 'common sense' often springs to mind. Open source may well enable us to do root-cause analysis properly giving a pathway towards serious incremental improvement using highly reliable components like Apache and PHP. I will look forward to this to stabilise our many disastrous efforts of the last twenty years.

Returning to where I came in, if IT is a profession, my money is on 'declaration of belief in a religion'. There certainly isn't much science in it and if learning relates to learning from one's mistakes, there isn't any of that so far.