

The supply of good engineers may dry up

Computing is full of surprises even at its roots and periodically, computer scientists will line up to argue some great point of the day. Conflicting opinions will fly backwards and forwards like naval shells at Jutland and eventually everything settles down and the industry's technology lurches off in yet another unpredictable direction. What is never offered is any experimental evidence because computing is not an engineering industry, it is a fashion industry with pretensions to high technology. A case in point is the humble 'goto' statement, (known as an unconditional jump). In the 1970s, everybody who was anybody rushed to offer an opinion, its detractors claiming that it led to many mistakes and its supporters claiming that some things could not easily be done without it. No evidence was offered, only opinions. This pattern has been repeated on numerous occasions since with OO, 'small is beautiful', software process maturity, formal methods (the use of mathematics instead of just thumping on the keyboard while whistling), CASE tools and lots of other ideas being argued and as often discarded, all of them without any convincing attempt to subject them to experimental test.

The result of all this heat is very little light. Until relatively recently, we still had no idea whether any of these ideas were good or bad. When we did get around to shining some experimental evidence on them, many of them turn out to be deeply flawed or massively oversold. The Linux kernel exemplifies some of these concerns. As I have previously reported, the Linux kernel is one of the most reliable complex applications ever produced by the human race and yet it breaks just about every rule we hold dear. It is a 'chaotic' process in software process terms, it does not use mathematics, it is written in a programming language 'C' which causes conniption fits in critical system designers, its designers are scattered over the globe and it is just about the best thing we have ever produced. Can we really have got things so wrong ?

There are a number of salient factors about which I and others have written at some length but one key factor emerges again and again in high quality systems - the engineers are individually highly competent. Individual engineering quality seems to be the bottom line in successful IT systems development. If its good, you have a chance, if its poor, no amount of new concepts, languages, process models, extreme programming or whatever will help.

This brings me nicely to the point of this column. You may not be aware of it but the industry seems to be picking up significantly. At the same time, applicants for computing courses in Europe, the US and Australia to name a few are once again down by at least 25%, (science and mathematics are equally badly hit). In two or three years time, the shortfall between supply and demand will be such that very inexperienced people will once again be building serious systems because there isn't time to train them properly. We have known for thirty years the havoc this can wreak on the best run IT project since Fred Brook's eponymous "The mythical man month", so I hope all you prospective employers out there have a plan in place to deal with this. You will need it.

lesh@leshatton.org