Greetings from Bob Jolliffe

CONVERSATIONS WITH DERRICK

During my short stint at the fine department of Computer Science at the University of Pretoria I enjoyed many a conversation with Derrick. Mostly these were excuses to smoke a cigarette, but we nevertheless made great progress on solving the problem of getting students to produce quality in program code. Nowadays I don't get to see him so much and I worry that some of our earlier insights may be lost. I'd like to take this opportunity to smoke another cigarette and try and summarize where we are.

Both Derrick and myself start from a sensible common position, which is surprisingly missed by many: the programming of a computer is first and foremost a human endeavour. It may bear some resemblances to a sausage factory or to a machine (where programmers are referred to as "resources") but these are just resemblances. Stripping away the veils which software engineers erect to confuse us, the process of constructing a computer program consists of a myriad acts of authorship. In the tradition of all good academics, it makes good sense to refer to Wikipedia at this point. What is meant by "authorship"?

> "An *author* is defined both as "the person who originates or gives existence to anything" and as "one who sets forth written statements" in the Oxford English Dictionary."

The second definition provides an accurate account of what is being done but doesn't give us much hint as to why it is being done, and certainly not how it could be being done well or badly.

The first definition suggests the author is doing something more than just "setting forth". She is the author as creator. That is, she is doing what human beings do when they are being most human. Now what drives the author to create quality in these creative acts is where Derrick and I begin to diverge.

I think for Derrick, and I look forward to him correcting me for putting words in his mouth, it has always been at least something of a question of ethics and a moral imperative. The programmer *should* strive to produce quality code because it is somehow the *right thing to do*. After all, the production of a sloppy piece of code can have nasty consequences ranging from an ugly user experience through to lives being placed at risk. Or even, God forbid, some capitalist losing pots of money.

I in turn have argued that the programmer, if she is to be a good programmer and a self-conscious creator, should seek to produce elegance and beauty in her creation because she *can* rather than because she must. These very acts of creation are acts of emancipation.

We are probably both a bit off the wall, perhaps overly effected by nicotine craving, but I think our different madnesses have had an interesting consequence to our approach to teaching generations of students. There is a well known (mis-)saying that you can lead the hordes to culture but you can't make them think. After many years of trying I have come to believe the truth of this with increasing conviction. But if you can't make the buggers *think*, you can at least make them *feel*.

Whether driven by a Catholic sense of guilt or a marxist drive for self emancipation, I think we have both recognised that it is sometimes possible to inculcate into these aspiring creators a sense of responsibility for their muddled thinking. Something that makes them *feel* the need or the urge to un-muddle themselves.

In this I think Derrick and I have shared something, which besides the odd cigarette, I have thoroughly appreciated. Of course he's still got a lot to learn about Marx and authorship, but he's showing good potential. I on the other hand am probably something of a lost cause – perhaps if I could just make it back to the confessional I might yet come right.

[via EMail: 17 June 2008]