

Why I'm not a programmer today

Dave Guard

My exposure to computers began at about the age of seven. This was probably mostly thanks to my grandfather being a member of the New Zealand Computing Society and my father always being keen on the latest technology. In the mid eighties, I got my first computer, a Sinclair ZX Spectrum; I recall transcribing code from a book and then recording it to tape so that I could play the games I had produced. I used to love it, and even in those early days it was clear that I had a predisposition to problem solving and an analytical mind. Later on, I had quite an exposure to Amigas too, although that was sadly a fairly short-lived experience. Around '95-'96, when my Amiga 500 finally packed it in, I ended up with my first PC, running Windows 95. Oh joy!

About five years later I decided to study programming. Prior to enrolling, Tony (FSM's Editor In Chief) convinced me to start using GNU/Linux and to learn Unix commands and C, as he said that this knowledge would give me a head-start in the course. Despite his grumbling, he helped me set up a dual boot so I could run both Windows and GNU/Linux on the same machine. After a couple of months, I was picking up the basics of C from a book fairly quickly, and getting used to using the Unix command line.

Our expectations of the course were wrong and all of this good work was to be undone. Shortly after starting the course I started to slide down the slippery slope of proprietary software programming.

The course was a Diploma of Information Technology (programming) and was 2 years full time. It was supposed to be a generic, system-independent "IT and programming course". However, its main thrust was in the direction of Visual Basic, Java and Windows NT. The better lecturers taught the VB and Windows NT subjects, which of course made them easier to learn. The one Unix unit (that's right, there was only one!) was taught by a guy who could out drone a blowfly playing didgeridoo. He spent his time at the front of the class, with one arm of his glasses poised in his mouth as he stared at the ceiling, trying to remember that anecdote we all knew so well and hated so much. There were no units on free software. The lecturers wouldn't even share notes!

I did quite well in the end and gained my Diploma of (proprietary) Information Technology. However, my exposure (apart from to Tony, who wasn't teaching me anyway) had been to a closed community of lecturers who, themselves, would not share or collaborate. I had not been told about the free software community or that I could work on free software projects that would give me exposure, experience and credibility. The existence of free software seems obvious to me now, and would do to most of our readers. But for an IT newcomer, it's amazing what isn't obvious.

At the end of the course, I felt a little empty. I felt like I didn't belong to the software development world. While I enjoyed programming, I didn't feel very comfortable with the secretive, un-sharing, proprietary world around it.

Eventually, my career took me to a completely different place, where my programming skills were welcome but not crucial.

It took about another five years and my continued friendship with Tony, before I became involved with the free software world. And now, through my experience, I fully realise the importance of exposing young programmers to our community. Some major faults in the course became clear: as students, we should have been encouraged to join free software projects. While plagiarism was a concern through some aspects of our training, we should have been encouraged to share code and taught how to re-use and adapt it for use in our own computer systems projects. This is after all how it's done in the real world—you're not supposed to reinvent the wheel all the time!

Today, I am not a programmer; I see myself as a "missed opportunity" for the free software programming world, and I'm probably not the only one. I feel that we (the free software community) should be doing our best to stop this story from being repeated. I believe we need to ensure that potential future programmers are aware that free software is out there, and what it means. They need our support, their understanding of free software needs to be nurtured and they need to be made welcome into our free software community.

If you work in IT education and you believe in free software, make sure you encourage young programmers to become involved in free software and explain the benefits of doing so. If you live near an educational institution, let them and their students know about projects you are involved in and encourage them to volunteer.

Let them know how fantastic the free software world can be for them and how great they can be for it.

Copyright information

© 2005 Dave Guard

Verbatim copying and distribution of this entire article is permitted in any medium without royalty provided this notice is preserved.

About the author

Dave is the Senior Editor at Free Software Magazine.