Why Good Programmers and Good Companies Need Good Personal Project Time

Matthew T. Rupert

12/2010 - Draft

In the 1950s, long before innovative companies like Google and Twitter were doing it, 3M encouraged its employees to use 15% of their work time to engage in person research projects. The result? Masking tape and Post-It Notes, to name two (Kretkowski, 1998). Many of us are already aware of Google's 20% approach. Intel, Hewlett-Packard and Twitter are other companies with various policies that encourage employee research projects.

These projects, while innovative in some way, need not be directly related to the project (or even company) that the employee is working for. The intent is to remain innovative in some way. In the case of Google, the results are astounding (at least as of 2006), "Fifty percent of what Google launched in the second half of 2005 actually got built out of 20% time (Mayer, 2006).

Of course we all know the now famous stories of how Bill Gates began tinkering with software at a very young age and Steve Wozniak, simply an enthusiast (the Captain Crunch story still fascinates me), began producing the earliest Apple computers in a garage. These stories are now legendary. And if the riches of Apple, Microsoft and Google aren't enough, I suggest watching the movie The Social Network to learn more about Mark Zuckerberg, whose brainchild now has us wondering what use the Internet was before there was Facebook.

EMPLOYEE RESEARCH

Every single *good* software engineer that I know became a software engineer for one reason and one reason only: A fascination with the world of software and a love for the endless possibilities. I remember the first time I ever sat in front of an Apple IIe. While most children simply played Oregon Trail I broke out of the game and poked around in the thousands of lines of BASIC code. An entire new world had opened up, and I stared wide-eyed into the green screen wondering what great things I could create. When I talk to other programmers my age we all share such enthusiasm for the early Commodore 64, Apple IIe and Tandy computers. We began our own "research" projects very young, writing line after line of BASIC and messing with commands like POKE and PEEK.

As far as I can tell it was around the mid-1990s that software engineering became known as a good, high-paying career. I'm all for being paid well, but a side effect of this was an abundance of people entering into computer science majors without the same love and fascination for the field as some of the other students in the same classrooms. Because of this truth, I must first start by acknowledging that not every "programmer" on the staff will be wired in such a way that personal research projects bear much fruit (or are even enjoyed). If this is the case, I would suggest one of two things:

1. Don't force it.

2. Hire better programmers (i.e., programmers who, in their 30s, 40s and 50s still have the same fascination and zeal for the world of software).

Bibliography

Mayer, Marissa (speaker). (June 30, 2006). *Marissa Mayer at Stanford University*. [Seminar]. Martin Lafrance. Event occurs at 11:33. http://www.youtube.com/watch?v=soYKFWqVVzg. Retrieved December 23, 2010. "Fifty percent of what Google launched in the second half of 2005 actually got built out of 20% time."

Kretkowski, P. D. (1998). The 15 Percent Solution. *Wired Magazine*.