



Maddog autographed magazines at the Linux Pro booth at LinuxFest Northwest 2009.

Quality software engineering

WANTED

When it comes to software engineering, we need more of it.

BY JON 'MADDOG' HALL

Recently I returned from the fantastic LinuxFest Northwest 2009 conference in Bellingham, Washington, a small city north of Seattle and home to a variety of people, from self-described “ancient hippies” to software people who have fled Redmond for a quieter life. On my return flight, which left from the Bellingham airport, I sat next to a gentleman of “about my age.” When I greeted him, he responded with a hint of a Scottish accent.

Our small talk turned to our occupations: I told him about my job “selling Free Software,” and he told me about his job as a systems engineer for Chevron. As the conversation continued, he discussed all of his efforts to use Microsoft products and the number of times they jammed up on him. His voice grew warm as he talked about how Unix systems and Linux systems were much more stable and how he liked them a lot better.

Then he said something that I had heard a long time ago: “Of course, for mission-critical applications, real mission-critical applications, the type of applications that absolutely have to work, we would never use software-controlled computers. Hydraulics are the way to go. Software is just too unreliable.”

My face turned red – after all, my life revolves around software and digital computers. Systems I have helped create have launched astronauts to the moon, run automated warehouses, and performed other “mission critical” work. But as I sat there and listened to his stories of Chevron losing US\$ 60,000,000 a day because some software person neglected to test their code, I thought back to some of the projects in which testing seemed to be an “after the fact” issue.

Getting Testy

I remember I once received a copy of field test software and tried to install

it on my computer, but it would not install. Thinking that it was because of my particular hardware configuration, I looked at the source code for the installation program, which fortunately was written in a scripting language, and I saw that it was impossible to get through the code via any path. In other words, the engineer who had written it had not tried executing the code even one time.

Immediately, I walked into the engineer’s office and admonished him because he had jeopardized the entire field test of the product and, thus, the entire projected shipping date. People’s businesses and livelihoods depended on us making those dates, and although we did not want to ship a defective product, it was important to meet those dates.

On another occasion, we had determined – through no fault of Digital’s – that 12,000 memory boards had a defective chip, which meant that all 12,000 would have to be recovered and remanufactured. Back then, memory was close to US\$ 1,000 a megabyte, so not only were we looking at a potential US\$ 12,000,000 loss to the company, but a lag time in shipping a new system.

One potential solution was to do a software “strobe” of memory every few milliseconds; however, the software could not tell whether the board in any particular system had this defect or was a normally acting memory board. So these particular modeled systems would have to “strobe” memory as long as they were in use.

A hardware engineer proposed that Ultrix (our Unix system at the time) simply put this “strobe” software into the kernel, thereby “solving the problem.” I pointed out that the problem was with hardware, and there was no guarantee that this hardware would continue to run Ultrix. Someday it might run VAX/Eln, a real-time operating system used for various mission-critical operations.

I said that perhaps when the control rods for the nuclear reactor need to be lowered, VAX/Eln will pause for a few milliseconds to strobe memory, but when it goes back to lowering the rods, the nuclear reactor will be a pile of ash.

The hardware group remanufactured the memory boards.

Quality software engineering is serious work. We need more of it. ■