FAIRCHILD LABS

I recently had the experience of going full circle in my career, going back to a similar role to where I started. In engineering it is an enlightening experience – though I imagine that would be true in any career. One of the first things you learn is you weren't nearly as smart as you thought you were way back when.

I was lucky to start my engineering career at Compaq in Houston in the 1990s. In those days Compaq had a mix of electronics industry veterans and young and hard charging engineers, and though I had little experience, I was able to fit in well.

The engineer who most influenced my career was Steve Fairchild. As somebody who was on the front lines of a rather competitive battlefield, it is my opinion that Steve was the person most responsible for the success of both SAS and SFF HDDs.

Back in those days, I didn't think so fondly of Steve. He was big, loud and could be intimidating. My relationship with Steve was all business, I never got to know him outside of work, but I imagine in real life he was giant teddy bear. While he was unmerciful in his engineering criticism, it was all about the engineering, never personal or vindictive. I have watched Steve cut down both vendors and peers in exactly the fewest words possible and without any room for rebuttable. You did not act like you knew what you were talking about around Steve unless you knew what you were talking about.

Our team was responsible for qualification of SCSI HDDs, and my role was programming the Big-Real Mode DOS utility we used for testing, Psuite. Sometime around 1998, I ported Psuite from Compaq's internal SCSI driver library to Adaptec's CHIM library. In a rush to get their chip into production, I had been required to spend weeks in California helping Adaptec debug their chip with my software. It wasn't a lot of fun for me or for Adaptec, as it consisted mostly of me waiting on them to deliver some code to test and debug.

We had a plan to port Psuite to some modern 32-bit OS, and my idea was for Windows. My key selling point was vendors had to have Windows SCSI drivers, so we wouldn't need to do any babysitting, we could just give them the Windows Psuite application and they could debug their own drivers without my direct help. My own motivation was: there wasn't much career potential programming on DOS and the other alternatives were too specialized, but there was plenty of work programming on Windows. Linux was too immature and had even bigger technical challenges to overcome than Windows.

I never had the nerve to ask Steve or anybody else what Steve's actual job was, he was a member of the technical staff and knew everything. His role on our team was performance. His role for our software development team was to greenlight the performance of our software.

My approach is always code first, research later. In the early days of our discussion, I had already ported a basic version of Psuite and proven that the performance would be within 1% of our DOS tool. These were spinning HDDs, and even though we were testing the fastest available on the market, they were terribly slow compared to most other subsystems on a server, and I had the general attitude that performance methodology was not a big deal. My biggest challenge was on the functional side, where

we did not have the same visibility to SCSI protocol compliance that we did with our DOS built-in driver. I had the support of my manager with the relative performance, with the caveat, “As long as Steve is okay with it.”

Steve wasn’t okay with it. After presenting my 1% data, Steve tore my plan apart, coming up with a list of around ten performance items that I would have to address. I seem to recall that at least half of the list were things that weren’t even measurable with our DOS tools, so not only did I have to code these new measurement techniques on Windows, with its rich libraries I could leverage, but then go figure out how to back-port the same techniques to DOS so that I could prove to Steve there was no difference.

This went on for months. We’d have these weekly meetings, where I’d want to talk about all the really clever coding I done to move forward on the functional side, and Steve would tell me I’m wasting my time until I’d finished the methodology tasks he’d created for me. It bugged me, because 90% of the problems we worked on were functional error recovery problems, and Psuite was clearly the best tool in the industry for that. Hard Drives are and were always slow, and it did not matter what software tool you used to measure them, they’d still be slow.

I remember after one particularly brutal meeting, I’d gone back to my office to reconsider what Steve had said about my data, and thought I’d caught him in a mistake. I took my printout to his office, and within about 30 seconds, he’d completely turned me around. Not only did I misinterpret my own data, I hadn’t even been asking the right questions. Before I could escape, Steve had me sit down, where he proceeded to draw a diagram on paper and lecture me for an hour, when all I wanted to do was go back to my office and cry. I know the subject was system level performance, and I know what Steve was trying to convey was that methodology was the most important part of measurement, the numbers themselves didn’t matter. I think he was trying to make me feel better about doing all this work that had absolutely no relevance to the current technology but might one day help him with whatever he was working on. I didn’t feel better about it. Boy, do I wish I’d paid attention, would have saved me a lot of trouble down the road.

When I completed the last item on Steve’s list, and I presented the data to Steve in our team meeting. He said, “What are you going to do about aborts?” No, “Good job, man.” or “You were right all along, I should have never doubted you,” but what are you going to do now about the real problem I was keeping you from working on all along? By this point, I’d had to complete so much code to validate Steve’s requests, the project was pretty much done, except for the hardest part. Steve was no help for me there. Apparently, he’d taught himself the Windows Storage Stack fast-path by having a young engineer do a bunch of seemingly useless experiments for him. He had no idea how the error path worked – that was for mortal engineers.

I left Compaq not long after finishing that project to work for Adaptec. During the early days of SAS, I worked very closely with Steve as a customer and he treated me with respect and continued to mentor me, especially when it came to how to deal with my own bosses. After I made a particularly embarrassing and potentially career ending mistake, Steve was gracious and gave me the single best piece of career advice anybody ever gave me. I’ll keep that gem to myself for now, but one thing Steve knew was that in computer business, the engineers and not management or marketing held the real power. For Steve, it was more than just an obligation to point out engineering weaknesses and come up with clever solutions. I believe he felt responsible for making well engineered products, and he knew with the right methodology and data, he could always hold marketing and management to account.

Sadly, Steve was a mortal engineer. I don't remember the dates, but it seems like he got sick right around the time HP shipped SAS. I didn't see much of Steve after that, we went out to lunch a few times a year. In the beginning Steve initiated these lunches. Steve always paid for his own lunch, long before it was HP policy not to accept paid meals from vendors. Somewhere along the way, I got the reputation as one of the few engineers Steve would actually go out to lunch with, so every few months marketing or sales of whatever company I was working for would get the idea that we could make an inside run at some kind of influence or get some information from Steve. I didn't bother with subterfuge with Steve, he would have seen right through it, I would just come right out and tell him what my people were after, and he'd usually say something like, "I'm not going to tell you that." Then we'd have a nice lunch and talk about engineering.

In 2010 I took a job in China. We corresponded a few times via email, Steve was very curious about both life and engineering in China. I came back to the states at the end of 2012, but only saw Steve once, walking very slowly down the hallways of HP. He looked exhausted but took the time to remind me that we needed to catch up and tell him all about China. This never happened. Steve died in 2013.

I've had a very successful career, and for the most part, I've been able to do the kind of engineering I think Steve would be proud of. I never got to tell him what he meant to me when he was alive, but as I've grown older I think about him more and more. Every time I smell some weak engineering from marketing, or management, or other engineers, I ask myself, "What would Steve say?"

Steve engrained in me the notion that engineers are responsible for the tools they use. And Steve said to me, "The methodology is the most important thing." The industry lost a legendary engineer when we lost Steve. He was the best engineer I've ever worked with.

This work is dedicated to Steve Fairchild.

Contact Information:

Brandon Awbrey
scuzzydude@hotmail.com
(713) 305-5620