

Engineer's Character

Contributed by Andreas Haberstroh

Saturday, 19 May 2001

Last Updated Friday, 13 November 2009

I've been to many interviews in my lifetime, as the employer and employee. One of the recurring questions in almost all the interviews I've participated in has been: "What makes a good software engineer?" This is an interesting question. I've listened to many engineers tell me what they think is important, and the answers are remarkably consistent. It's almost as if they've read the same book, and are just spewing out what they've read. Good communication skills, good problem solving skills, able to write and debug code.

I, of course, take a completely different approach. What I believe makes a good software engineer also defines my m.o. I list these in no particular order: Consideration No Prejudices Why Consideration is a very important word in my book. Merriam-Webster's Collegiate Dictionary (<http://www.m-w.com>) defines it: "continuous and careful thought." The actions that I perform today have an impact on tomorrow. Consideration is the action of making sure that you think before you do something terribly stupid. As a programmer, my consideration is directed towards one target; the user. Now, the user has many definitions. There is the direct user of your code. This is the fellow that uses your code to compile with his code. Your code should be as flawless as possible before handing it off. Nothing is worse than receiving code that doesn't work as advertised. It shows lack of forethought, not to mention, lack of professionalism. Checking in broken code ranks right up there. And, of course the ultimate user is the fellow who uses the end product. This is the guy that runs your program. If it crashes, then you didn't take into consideration some aspect of the design of the product. It may also be that your product does physical harm to some person. Now, there is something to think about! Sure, there are product defects in every product you find on the market. The difference is in the number of defects, the level of severity, and the amount of consideration you put into designing the product from the start. Prejudice is a more elusive problem. People have a tendency to rely on their experience to get through problems. Experience is a tool, one of many that an engineer has in his arsenal. However, experience, if used without judgment, becomes a problem in itself-it leads to prejudice. Engineers who see a problem, and then immediately jump onto a solution path react in this way because their experience tells them, "I have seen this before and know how to solve it." They have prejudged the problem, and immediately try to solve it. This happens at all levels of engineering: design, implementation and debugging. Time is wasted and nerves are wracked. My last ingredient is for an engineer to ask the question: Why? Curiosity is an important precursor to innovation. For instance, asking, "Why do it this way?" can trigger thought that might lead to a better way to build a product. Curiosity is also an important component to debugging. The ability to ask yourself why until a solution is found is important. Engineering is all about problem solving. Every problem in life is solvable. Every problem shares something in common with another problem. However, every problem is unique in its own way. How an engineer solves the problem is what's important.