

Maturity

Or the Lack of It!

It is extremely odd that I was asked to write the BackTalk for this issue given that the theme has to do with maturity. For those who have never met me, you can rest assured that while I might grow older, I steadfastly refuse to grow up. Maturity? It must apply to my coding skills and certainly not to my behavior.

I have been a coder since the 1960s. I always thought I was pretty good. Luckily for me, back in 1997, I became an instructor for Personal Software Process (PSP), and later also qualified as an instructor for the Team Software Process (TSP). My coding skills truly matured. And, while I never taught any CMM® classes, I certainly knew the value of the CMM. It allowed me to assess the maturity of my organization, just as the PSP and TSP allowed me to assess the maturity of myself and my team, respectively.

But, how do I assess the maturity of my code itself? Is it the formatting? The documentation? The self-evident identifier names I use? Is there a way to look at a piece of code, and say, "Yes, this is very mature code?" Well if it is COBOL, that certainly makes it mature, right? And—if it IS mature code—do other developers appreciate the value and worth of the code? As a matter of fact, what about code that is not just "mature" in terms of high quality, but "mature" in terms of age?

One of my students recently pointed me to an excellent article by Joel Spolsky regarding code maturity. In it he discusses the value of "old code" and makes the observation that newer programmers always want to throw away old code figuring that it is bad. They are most often always wrong! He points out that new programmers treat old code as if it somehow rusted; the mere fact of it getting old makes it somehow less correct or useful. As Joel accurately points out, there is absolutely no reason to suspect that you are going to do a better job than whoever wrote the code "back in the day." And, of course, as all of us mature coders know, old code represents an accumulation of knowledge that took potentially years and years to accumulate. Mature code has been refined, sifted through the sieve of testing, and then strengthened by multiple updates and improvements. The mature code, if thrown away, will require yet another generation of programmers to find out that the old code is complex and hard to read only because the actual problem it solves is equally complex and hard to understand.

Having pointed this out, how do you know if you are one of "those coders" who write old code? Well, having the Internet as a guide, I have compiled a short list of items that might help tell you if you are one of those developers who write "old code." Do any of these apply to you?

- 6 a.m. is when you get up, not when you go to bed.
- You hear your favorite song on an elevator.
- You watch the Weather Channel. And you carry an umbrella.
- Jeans and a sweater no longer qualify as "dressed up."
- You do not know what time Taco Bell closes anymore.
- Your car insurance goes down instead of up.
- You feed your dog Science Diet instead of McDonalds leftovers.
- Sleeping on the couch makes your back hurt. And, if you do fall asleep on the couch, it takes two tries to get up.
- You no longer take naps from noon to 6 p.m. You want to, but you just can not find the time.
- Movies are not loud enough.
- Friends that call after 9 p.m. start the conversation out with "Did I wake you?"
- You know your pharmacists name.
- You start getting the "senior discount" without asking for it.
- Dinner and a movie is the whole date instead of the beginning of one.
- You actually eat breakfast food at breakfast time. Anytime you buy cereal, it's for fiber content, not the cute animal that advertises it.
- 90% or more of the time you spend in front of a computer is for real work.
- Grocery lists are longer than macaroni & cheese, diet Pepsi and Ho-Hos.
- You have read this list, and realized that several items apply to you. And it got less and less funny as you kept reading.

Have no fear, my friends. We are all getting "mature."

So is your code. And it is valuable. Resist the urge to simply rewrite something because it is not new. Not everything you compile has to be in Ruby, C#, or Javascript.

Cars grow old, and might become valuable antiques. Mostly, however, they simply wear out and are replaced. Luckily, your code is not an automobile. If it survived the first few years of use it only becomes more and more valuable. It has been debugged. Not only does it work, it represents accumulated knowledge. It works so well, you probably cannot afford to get rid of it.

Much like us mature developers.

David A. Cook, Ph.D.
Stephen F. Austin State University

CMM® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.