

## Eric P. Bloom: Your technology skills have a two year half-Life - Part 2



**By Eric P. Bloom**  
**Your IT Career**

*Copyright © Eric P. Bloom, All Rights Reserved*

*Last week's blog "Your technology skills have a two year half-life and 6 ways to stay current" was heavily read and hotly debated. Thank you to all and here is my reply.*

Thanks for all your emails and blog comments on last week's column. I love people's feedback to my columns as it helps me look at things from different points of view. Quite frankly, I agreed with almost all that you have written.

To begin, the original question asked by the reader was related to the customization of a software package, rather than programming in a base technology, like C and PHP, which has a different set of factors.

As a software package example, in the late 1980s I was a DBA working on Oracle Version 3. Whereas Oracle is obviously very alive and well, none of the database tools or processes I worked on then still exist. This was pre PL/SQL and query optimization had to be done by hand because the SQL engine didn't work all that well. That said, Oracle as a software package then, is totally different than Oracle the software package now. Sure, data normalization is the same, but virtually all of my exact Oracle 3.0 skill set at that time is now outdated.

That said, is the knowledge I had then, transferable to the latest version of Oracle now? Yes, I believe it is. But to the original point, my Oracle Version 3 skills were much more marketable in 1988 than they are today. Potential employers looking for a DBA would consider my knowledge to be old, and thus less desirable than someone with more current knowledge.

There were also a number of comments stating that C was very marketable 20 years ago and is still very marketable today. These comments are 100% correct. I love programming in C. I was also a C programmer. In fact I wrote a (marginally coherent) book based on the original Brian Kernighan and Dennis Ritchie C. It's a great language. Knowing C made it easier for me to learn Pascal, JavaScript, PHP, and a number of other languages.

Even C, however, evolved into C++. Imagine if today you only know K&R C and didn't understand function overloading, structured programming, base classes, and other related advances such as ODBC, XML, and multi-threading.

Also, C now has more competition than it did then, same as COBOL which I also programmed professionally. There are less companies today programming in C and COBOL because of Java. I'm not saying that Java is better or worse than C or COBOL, I'm just saying that they were used more widely in 1990 than they are today, thus further reducing the marketability of older technologies. In fact, I loved Borland Turbo-C, which of course today is far less marketable than the Microsoft .NET tool set.

I would also like to specifically address a number of comments regarding the faulty thinking of HR professionals and hiring managers that think knowledge of a language like C provides no basis for understanding like Java. I also think this logic is faulty. Here are two examples. First, I was a Microsoft ASP programmer. I personally found that being an expert in Microsoft ASP made learning PHP extremely easy. Basically, all I had to learn was the new syntax. Second, I was a C programmer who had to learn Java. I must admit that when trying to learn Java I had to learn to use Eclipse, it had some funky new inheritance rules I had to learn, and the packaging part took me a little getting used to. But, like ASP to PHP, once I got the basics down, my programming skills were extremely transferable. Hearing these comments, I wonder if these managers were ever a C programmer themselves. My thought is they were not.

Lastly, I would like to return to my earlier comment regarding base technology, like C and PHP, having a different set of factors. I think with these base technologies, their marketability half-life can be much more variable than just simply two years, but I think the basic concept is sound. This half-life could be dramatically shorter or longer based on the technology and your timing. For example, if you were an ASP programmer the day Microsoft announced .NET, then this half-life was almost immediate. I know because I lived through it. On the other hand, if you are primarily a PHP or COBOL programmer at a time of relatively low technological enhancement, the half-life would be longer, assuming of course that the general demand for your skill set remained constant.

In closing, I would like to specifically thank one of my readers, S. T., who sent me a couple of great emails on this topic. Thanks :-)

Oh, one more thing. A few of the blog comments referred to me as a marketing person. I hope from today's blog you now know I'm a techie.

Best wishes to all and thanks again for your great commentary. I really appreciate your input.

Eric

Until next time, work hard, work smart, and continue to grow.

*A former CIO, today Eric P. Bloom is the President and CTO of Manager Mechanics LLC, a company specializing in **Information Technology (IT) leadership development and IT soft skills***

**training.** *Eric is also a keynote speaker, nationally syndicated columnist, certified professional coach and a weekly columnist/blogger for ITworld.com*

*Eric can be found on Twitter at @EricPBloom and at [www.ManagerMechanics.com](http://www.ManagerMechanics.com).*