



TenStep Supplemental Paper

4 February 2004

Don't be Afraid to Train Your Experienced Developers in Newer Technologies

The IT industry has seen a lot of changes over the past thirty years. Application development has certainly changed as well; that is, the tools and technology have progressed. Low level assembler programming gave way to COBOL, which gave way to client server development, web development, PDAs, etc.

On the other hand, has the development process itself changed radically over the years? Yes, there are new languages and databases, but when it is all said and done, doesn't the coding still involve assignment statements, loops and conditional logic (If-then-else). Does the age and experience level of the developer play a role in whether or not he/she can become proficient in the newer technologies?

These questions are especially relevant in today's marketplace. Development positions are not as hot as they were a few years ago, and many experienced developers are struggling to find replacement positions as a result of layoffs and company closures.

Technology stereotyping

One of the problems encountered by older developers is technology stereotyping. As an example, let's say a company is looking to re-train two developers. One is 45 years old and the other is 29. You also have two openings to fill. One is for a mainframe COBOL position in the Application Support area. The other opening is for a web developer using Java. Neither candidate has the required skills for either position, but your company will invest in training for both people. Now, which one do you slot for each position?

Given this very arbitrary and unfair scenario, these are the first thoughts that come to most peoples minds. The "kid" is offered the Java web development position. The "old-timer" gets offered the mainframe support position.

Such is the fate of older developers. Just because a person is older, does that mean that he or she cannot learn new technologies and new tools?

Do older workers have a hard time learning new technologies?

One of the fundamental questions asked of older developers is whether they have the ability to pick up the newer technologies. There is no one answer that fits everyone. On one side, some older people are extremely savvy about new technology and can run circles around most people in their understanding of how technology works and interacts. On the other hand, some older workers who have invested years in understanding the technology they know are not necessarily interested in investing the time picking up the newer development languages and tools.

So, if you know one development language, can you easily pick up another? Are they really all just different syntax for "Do Loops" and "If-Then-Else" statements? In fact, there are big differences.



TenStep Supplemental Paper

When learning PL/1 and COBOL, to a certain degree they are just different ways of doing the same thing. However, the newer technology does require a paradigm change. Event triggered programming and the visual development environments are definitely different than the PL/1 days. Web development is different as well. You cannot take your mental understanding of mainframe development and apply it to development in client-server or the web. It does take a more fundamental change in thinking.

So, when it comes to the question of whether people of any age can pick up and become proficient in this way of developing, the answer comes down to a matter of will and skills.

Do they have the will?

The first question to ask is whether the developer has the will to learn. Some older mainframe and mid-range programmers are interested in learning newer technology. However, as was mentioned previously, some older developers want to ride out the mainframe / midrange wave until they retire. In a sense, this is not a bad thing. These older applications will be with us for many more years and many younger workers are not interested in learning them or working in the technology. So, although sticking to the older technology gets riskier every year, many older developers don't have many years left until retirement, so it might work out all right. On the other hand, they are accepting the risk that if they get laid off, they may find it harder to find a new position given their more limited skill set.

Can they learn the skill?

Now the question is, if they want to learn, can they? Of course, everyone is still different. Many people are amazingly flexible in their mental outlook. They may not have the current skills today, but if you invest in training and coaching, they can get through the learning curve to become very productive.

Not everyone will be able to make the mental transition to the new development paradigms. You have already screened out the ones that do not want to learn; however, some that want to learn will still not be successful. In a sense, they will not be able to "get it."

This is to be expected as well, and again, it is not all about age. Many younger developers never make it as well. In fact, in some development organizations, the individuals that are considered the weakest developers are the younger employees that just do not have the aptitude for development.

Treat everyone individually, give everyone a chance

It is extremely helpful to have a multi-skilled development staff so that you have the flexibility to assign people to whatever work is of the highest priority. This implies teaching new skills to everyone on the team, not just the youngest members. Of course, this does not apply if team members do not want to learn the new technology, but generally if workers have the will, they can be taught the skill, regardless of their age and prior experience level.



TenStep Supplemental Paper

Summary

The IT workforce is aging. Many of the older programmers have a broad base of experience in the development environment. They know how to develop and support applications. As companies move forward into more advanced and cutting edge technologies, they need to invest in many of these older workers as well. Look at your organization today. If the mainframe, people are all in their 40's and 50's and the web developers are all in their 20's and 30's, you have some problems. First, you are stereotyping workers based on age, and this says you are more willing to hire younger people than retrain current staff in the newer technologies. Second, you are not taking full advantage of the experience of these experienced workers in your vital new technologies.

If older workers have the will, they can make the transition to the newer technology. Give them a chance and you will see reap the benefits for many years to come.