

Your Best Leadership Talent May Be Right Under Your Nose – in Your IT Department

By Jeffrey S. Shaw, CLU, ChFC, Executive Director, [Life Insurers Council](#)
From the LIC Bullet, September 2016

Historically, the most successful and innovative life insurance companies were driven by very strong sales and marketing influences. These areas were acutely aware of where they lagged the competition and were the most motivated to differentiate through innovation. There was a constant challenge to maintain balance between the edgy, entrepreneurial, “seat of the pants/back of the napkin” attitude that pushed the company forward, and the less exciting voices of reason that rightfully advised risk aversion, financial restraint, and pragmatism.

Although this struggle for balance continues, the driving force behind the most successful companies today no longer resides in sales and marketing. Anything a company wants to do regardless of whether it’s groundbreaking and transformational -- or perfunctory and fundamental -- more than likely relies upon a foundation of technology.

Try to think of a department in a life insurance company that hasn’t been dramatically transformed by recent innovations in technology. For Marketing and Sales, it’s robo-advisers and agent portals that make “easy to do business with” a quantifiable and marketable differentiating factor.

Underwriting/Issue is benefiting from e-apps, rules engines, and consumer data that has dramatically improved the efficiency and effectiveness of the underwriting process. Even Actuarial has become familiar with “economic scenario generators” that have raised the range of projected outcomes – and complexity -- by a factor of ten or more.

Most companies have no trouble coming up with ideas about how to become more efficient, more profitable, and more competitive. In fact, a common challenge frequently discussed at LIC Technology committee meetings is prioritization of projects. It’s no longer the marketers who are pushing the envelope of innovation. Every department is being bombarded daily with information about new technology that will make their jobs – and their lives – immediately and infinitely improved. And although not every idea is ideal, the cost/benefit of many are easy to compute and make a strong case for implementation.

Some companies have their prioritization process down to a science. Every business unit manager has to quantify the potential benefits from their projects and make a very specific capital commitment in return for IT support. But even the best thought-out plans can go awry. Squeaky wheels, CEO pet priorities, corporate politics, and the very real challenge of putting an accurate dollar value on every proposal adds an element of chaos and unfairness to the most detailed plans – and render less-deliberate prioritization processes practically useless.

In the past, IT viewed the rest of the company as their customers and abdicated prioritization responsibility to the users. In fact, a common complaint among IT leaders has been the lack of ownership that their users often have for their own projects. Requests would be made and quantified, only to find out months later that the expected results were rarely forthcoming.

This is very similar to the frustration actuaries express when marketers propose new products with grandiose sales projections that never materialized. But today, a good product actuary has to have a reasonable working knowledge of the market place. It's no longer valid to hide behind the excuse that "marketing told us to do this" without having the ability to challenge and contribute meaningfully to the conversation.

The same can be said for programmers.

In fact, this has defined the most successful and innovative companies we work with today. No longer are their IT staff victimized by the faults of their users – they are taking the lead and bringing innovative ideas and new technology to users who didn't even realize these opportunities existed.

This subtle cultural shift that turns IT from order takers to thought leaders is a huge challenge. Many companies haven't ever operated this way and many CTO's haven't viewed their role – or their staff's role – from this perspective. Furthermore, many programmers themselves shy from the spotlight and prefer to keep their heads down and focus on what they do best – writing code.

Further compounding these issues is the challenge of attracting skillful programming talent to a life insurance company. General Electric developed an entire ad campaign called [What's the Matter with Owen](#) that hilariously featured a young programmer trying to explain to his parents and friends that he was actually excited about developing software for GE. The FBI has also lamented their challenges in hiring coders because of their struggle to make the FBI "look cool". Of course, life insurance companies don't have this problem at all!

Perhaps a better solution would be to leverage the talent we already have.

A recent [article](#) by Mitzi Reaugh titled "Learning to Code Helped Me Become a Better CEO" offers some insightful observations about some of the unique skills possessed by *any* successful programmer. Although the article is written from the perspective of how the author taught herself to write code and applied those skills to improve her abilities as the CEO of a small start-up, the observations about what she learned from the coding process highlights the inherent abilities already resident in most insurance companies.

For example, most of us don't think about the level of detail programmers have to subscribe to. One errant key stroke and the entire system crashes. Coding may not always be structured as efficiently and thoughtfully as possible, but it has to be absolutely accurate or it won't work.

This leads to another coding attribute – a "stick-to-it" attitude that requires focus on the final outcome. Imagine the frustration of wading through hundreds -- or even thousands -- of lines of code in order to find that single errant key stroke that crashed the system? Detail orientation and perseverance are highly valued traits we aspire for in all of our employees, yet they are a prerequisite for programmers.

Another aspect of programming that may seem counter-intuitive is how collaborative they are. We've all heard the jokes about difficult it is to communicate with IT staff but the fact is that they are very effective at communicating with each other. In fact, the foundation of their job requires collaboration. Programming requires fluency in multiple coding languages and there are always people with a broader "vocabulary", more experience, or specific expertise. There are numerous online resources that document how products work and provide forums for Q&A. In addition, it's rare for a programmer to

only work on their own code. More often than not, they are fixing, adding to, or combining programs that have already been written – often by people who are no longer with the company (and often for good reason!).

And this leads to yet another valuable skill that programmers possess – they are problem solvers. Whether it's figuring out why a current program isn't working, or trying to come up with an elegant solution for a new program, the essence of coding is to figure out how to solve a problem. This isn't something they do on occasion – this is something they do all day, every day.

One has to look no further than Google and Facebook for examples of wildly successful new companies that were started by – and continue to be run by – programmers. One could argue that the unique technical and behavioral abilities of programmers makes them ideally suited to tackle the key challenges facing our industry in terms of disruption, aging distribution, and disinterested consumers.

For sure, not every programmer is CEO material, but do you even have a process in place to determine if any of them might be?

In Jim Collins' iconic book "Good to Great" he emphasized the importance of "getting the right people on the bus and in the right seats". Most of us would probably agree that the "right" people would include detail oriented problem solvers who persevere through difficult challenges and are good at collaboration. More than likely, your company already has a few of these folks, but they've been stashed away in some dark corner so they can write code without any interruption.

It may be time to consider putting them in a different seat.