



5 Areas That Attract and Retain Great Software Development Talent (from a Manager's Perspective)

PAGE 1

Overselling Yourself Could Come at a Cost

Don't set yourself up for
failure in the job interview.

PAGE 9



STOUT®

Fueled by the most powerful
technology available: Human Intelligence.®

Copyright © 2015 Stout Systems Development, Inc. All Rights Reserved. PRINTED IN USA.
For complete copyright and trademark notice see stoutsystems.com/legal

5 Areas That Attract and Retain Great Software Development Talent

(from a Manager's Perspective)

by Peg Bogema

Lately, when we talk with our clients, we hear the same story. Attracting great development talent has become quite a chore. And because of that, companies have a heightened interest in making sure that they are doing the right things to retain the talent they already have.

As a software development manager myself, I'm keenly aware that my developers are barraged with come-ons from companies looking for people with their skills. So like many managers, I'm always asking myself, what can we do to retain our people? It isn't just doling out raises. There is a practical limit to that, for sure, but it also isn't the only factor in why developers stay or leave.

Similarly, when I am hiring for my own team, I have to do some soul searching and make sure that my company really is a great place to work. And when I find that we are lacking, I am forced to figure out how to close the gap.

So, this is what I think software development managers should consider improving—at least the main things. But before I dive in, let me call to your attention that no matter what I think, and no matter what you think, developers have their own opinions on this topic. With that in mind, I've asked my colleague Dave Sweeton to pull together his thoughts on what companies can do to differentiate themselves from their competitors. His thoughts will be published in an upcoming edition of *The Informatizer*, so stay tuned.

#1 The Stack

The biggest turn-off I hear from developers is that they don't want to work someplace where the development stack isn't modern. They realize that there are cases where legacy should stay legacy. But there are also limits to how much they can tolerate. Companies in general, and software development managers in particular, need to find a strategy

that introduces new technologies into the ecosystem. Or they need to make the time to do wholesale rewrites. No developer wants to turn into a dinosaur.

If this isn't a problem for your shop, then give yourself one star and move on.

#2 The Career Path

If you've got a modern stack, then the next question you have to ask yourself is whether or not developers have some kind of a career path. Technical personnel are hired in for a specific role. Are they stuck in that role forever? For some, that's perfectly fine. Once a developer, always a developer—as long as their technical skills aren't stagnating. But for many, they are looking for some kind of an evolution, professionally.

Developer moves to Senior Developer moves to Technical Lead or Architect. With commensurate increase in pay along the way.

That fresh-out who hired in for QA—just to get a job as a computer professional—really wants to be a developer.

Developer has slung code for 20 years and is ready for a change. Is there an option to move into project management? Or maybe pre-sales engineering?

Rest assured, if you aren't facilitating growth for your team members, they will go someplace else as part of their evolution. You should know what each team member's goals are so that you can assist them in achieving them within your company.

#3 Upskilling

If you cannot offer a career path for your team members, you might be able to retain them by taking measures to upskill them.

And if you do have career paths available, upskilling team members and then promoting

them has zero downside for your company or department.

In either case, paying attention to upskilling is important. And the approaches to this are limited only by your imagination. A few ideas:

- Provide mentoring so that developers learn new skills from other team members.
- Cross-train team members. A developer can learn QA skills. The reporting specialist can learn UI development. The developer working on device drivers can learn real time programming. And vice versa in each case.
- Pay for training—conferences, classes, online, whatever. Or at a minimum don't make them take paid time off to attend.
- Give developers a fixed number of hours per week or month to learn a new technology on company time. A developer

can learn Angular or Knockout or Android development by following online tutorials and developing sample projects.

Your team will most definitely have ideas about what would be beneficial and interesting to them. While it's true that a developer could learn a new skill or get a certification and then go elsewhere, there is also the possibility that your employees will grow professionally to the benefit of your own company. A culture that encourages personal growth doesn't grow on trees. Many employees will consider it such a great perk that they wouldn't dream of working elsewhere.

#4 The Process

Nothing defeats developer morale more than feeling adrift.

If there is a feeling of constantly shifting sands. If there is a feeling that nothing ever gets done. If there is a feeling that

requirements are never, ever right. If... Well, there are a thousand ifs.

Having and following a process cures these ills. So find out that works for your company and your team—and then stick with it.

This may seem like a small point, but it is, in fact, one of the number one complaints I hear when talking with developers about why they are seeking new work. The work environment is nuts. Too chaotic. Too unpredictable. Nothing ever gets done. No job satisfaction. This all boils down to one thing—process. So get one.

#5 The Goods

Okay, you have to have the goods. This comes down to a handful of things. Wages should be competitive, so knowing what other companies are paying for similar roles is important. It's definitely not the sole element in a compensation plan, so don't obsess

over a few thousand dollars. But if you are paying \$65,000 for a developer when other companies are paying \$85,000, you may have problem on your hands.

Excellent healthcare benefits, disability, retirement match, bonus structure, great paid time off—savvy employees understand that these are significant components to the overall compensation plan. Hiring people out of university settings can be difficult—even though their salaries are notoriously low. Who else offers six weeks of PTO?

But if your company doesn't have the resources to beef up the compensation plan to the level that would make it genuinely competitive in the local marketplace, other things can be used to attract and retain great people.

Flex time, where people can set their own hours within reason, is very attractive. That's especially true for workers with kids, where

running out to a parent-teacher conference or getting a kid to a soccer game can be a major problem on a set schedule. But it's also true for the occasional developer who is far more productive on a quirky schedule. Requiring that developer to arrive at 9:00 AM bright eyed and bushy tailed is not exactly attractive if his or her best hours are more like 12:00 PM to 9:00 PM.

Where possible, providing some remote/telecommute days is another very attractive option—for some developers. Others would die if made to stay home all day, pressed from all sides by distractions or annoyed by the lack of interaction with coworkers.

Take a look at any of these five areas and see if there are one or two of them that you can work over so that your team is THE team in their own eyes.

Peg Bogema is Stout's Vice President of Operations. With Stout since 1997, her duties include the supervision of development projects and personnel, recruiting, and accounts.

A Yearend Message from our President, John W. Stout

We have an amazing profession. Software professionals often have the opportunity to create culture-changing applications that make a difference in people's lives.

While I do advocate degrees and training, this field is different in that licenses and degrees are not always required. And, if you do good work, the pay can be pretty good. One 18-year old I know made a lawyer's income doing software development. He had no college degree, and was completely self-taught.

I heard about a doctor quitting his practice to become a software developer. Better hours, less liability, more fun.

When making your list of things to be thankful for this holiday season, despite the challenges, you may be thankful you chose this profession!



Tight software budget?

At Stout, we'll work *with* you to find the software solutions that fit you best—both technically and financially.

The unknowns, the
undefineds, and the risks.

No, it's not the name of
a band. It's the day-to-day
of software development.

And Stout has done it
for over 20 years.
We can handle it.

Call Now

877.663.0877

StoutSystems.com

Overselling Yourself Could Come at a Cost

by Brian Skory

When it comes to preparing your resume, your 30 second “elevator speech,” a LinkedIn Profile, preparing for an interview and, ultimately, pitching yourself to a prospective employer, the vast majority of topics I’ve covered concerned selling yourself. As I’ve discussed over and over, these are your opportunities to present yourself in a way that persuades a prospective employer that you are, indeed, the person they’ve been looking for.

Something that has recently come to my attention, however, has to do with the topic of overselling yourself in the interview. I am not talking about situations where someone blatantly lies about their capabilities (something, by the way, which I strongly recommend against). Claiming to know C# when you have only casually dabbled in it would be an example of that type of fabrication. Rather, I’m talking about making ambitious promises—which, while well intentioned, in reality have a good chance of bouncing back on you if undelivered.



One company we know had a need for business development in a very specific technical space. One of the candidates they were interviewing had plenty of business development experience in several different sectors but not specifically in theirs. He didn't see it as an issue, however. Business development was business development in his mind. So he worked very hard to convince the hiring manager of his capability to transfer his skills and knowledge base to their particular space. Eventually, the hiring manager was sold and decided to bring him on. Unexpectedly to this candidate, coming up to speed on their product line proved far more difficult than he had anticipated, and eventually he was let go for "weak performance."


Another company was doing scientific development in C#. One of the developers interviewing for the role was well versed in that arena but as a C developer. He, too, convinced the hiring

manager that it was a small task for him to come up to speed on C#, but after months of frustration on many fronts, both he and his manager decided it was not a good fit and they parted ways.

The biggest problem in both of these cases is that each candidate must now account for the fact that they were only at their new position for a handful of months. They could choose to simply not list the position on their resume, but now they have to account for that period of time.

It is much better to avoid these complications by being honest with yourself about your capabilities and not setting yourself up for these types of failures by overselling yourself in your job interview. Promise to deliver what you know you can and your new job experience should be much more pleasurable.

Brian Skory is Stout's Technical Talent Manager. He began his technical career 20 years ago and the key to his success has been his consistent ability to find and hire the best talent.



**Does your
software "team"
need some relief?**

We have the tech talent you need.

Contact Stout Today

StoutSystems.com

877.663.0877