Breadth and Depth

**WE ALL WEAR
MANY HATS,
BUT MAKE SURE
YOU HAVE ONE
THAT FITS WELL.**

KATE MATSUDAIRA

I often get asked for career advice. One of the things software engineers always want to know is if they should learn some new tool or language. In fact, I can't think of a performance review I have read for a software developer that didn't include something about growing their skills around a particular technology.

That is the nature of our work: it is constantly changing. You have to keep learning, or you will become obsolete.

But for your career, is it better to go wide and learn a lot of different things, or is it better to go deep and learn a few things really well?

MAKING THE CASE FOR GOING DEEP

Recently I was talking to another engineering leader about hiring and staffing. I asked which technologies he wanted people to know, and he responded that it didn't matter—a good software engineer can work on anything.

This has been the thinking for many large software companies in the past, and there are definitely merits to it—especially when you are hiring inexperienced candidates straight out of school. As I have worked longer in the industry, however, I have started changing my thinking.

I would argue that it is

important to go deep in at least one area, and it is almost always better to hire people who have a solid depth of experience in the tools and technology they are using.

Why do you need to have deep knowledge?

Really good software engineers for a particular language or technology will exhibit qualities such as these:

- ➔ *They are productive.* They produce an amount of work that is above average, and they are able to get things done. This means they know how to use the tools of their trade well and aren't slowed down by not understanding something. They use their brain power for harder problems, not learning how to do the basics.
- ➔ *They make smart tradeoffs.* They are able to understand the risks of their decisions. They have failed before and so can avoid mistakes. If there is a library or prebuilt code somewhere, they are probably aware of it—they may even have contributed to it or used it in past projects.
- ➔ *They help others.* Teammates go to them with questions because they have done this before and know how to do it right. This expertise makes them natural leaders or mentors.
- ➔ *They hit their deadlines.* Their estimates are almost always accurate. They know how long something should take them because they have experience working on similar projects.
- ➔ *They are still growing personally and professionally.* Since they are comfortable with the technology stack, they can focus their time learning skills in other areas such as leadership, communication, or even new technologies.
- ➔ *They have a deep understanding.* When engineers work with a particular technology for a long time, they learn

As a technology executive, you can't just understand the technology—you also have to understand the business.

the nooks and crannies of how it works—the good parts and the ugly parts. This can force them to think about the very constructs of how the technology was built. For example, discovering a bug in a foundational library teaches them to write better software. This depth also allows them to pick up other technologies faster.

When you hire engineers who have experience in a particular technology and can operate at this level, they are able to get up to speed quickly. They will have to learn the way your particular systems and software work, but they won't have to learn the tools and technology ~~it was~~ written in.

While there is definitely value in gaining that deep experience—and it should be part of your career plan—it is also important to have breadth.

MAKING THE CASE FOR GOING WIDE

When I look at my own career path, I can say it is my breadth that has helped me the most. As a developer, the fact that I understood operations, lower-level operating systems, and compilers helped me write better code. And as a manager, having experience with other disciplines helps me work with those people better.

For example, we have all written software tests, but when you work with really great testers and learn the way they think and approach problems, this helps you not only work better alongside them, but also write more robust code.

As a technology executive, you can't just understand the technology—you also have to understand the business. You have to focus on customers, think about product, and be able to understand the financial implications of your

decisions. Often, you are managing people doing a job you have never done yourself—but you still need to be able to measure their impact and mentor them to success.

In these cases, it isn't enough to know just one technology well; in fact, if that is the case, you won't be successful. You have to have breadth and be able to think through other aspects of the problems you are solving, and you must have a broader understanding to work with people in different roles.

SO SHOULD YOU GO WIDE OR DEEP?

I once had the privilege of having a mentoring session with a VP at one of the great software companies. I asked him about his background and what he felt got him to his position. He told me that he started his career as a software engineer and soon realized that being the best software engineer would be hard; software testing, on the other hand, was much less competitive, and a lot of people didn't have the passion for it that he did.

He changed his career path to pursue software testing and went on to write several books. He became one of the best in the world at testing and was asked to speak at conferences all over the world. The VP title and responsibility came later but was a natural progression.

His advice to me: *pick something you can be the best in the world at*. Specialize in it. Pursue it above everything else. Success will come along with it.

In many ways this is true. When you are truly exceptional at something, you build career capital, and you can trade that capital for bigger paychecks, more flexibility, or even fancy job titles.¹

When people ask me the question of where they should focus their time—should I keep learning one technology or spend time learning a new one?—I ask them that very question: What is the one thing you could be the best in the world at?

The answer might be going deep or going wide—the important thing is to spend your time on building the skills that will move you to where you want to go.

References

1. Shin, L. 2017. 7 steps to developing career capital—and achieving success. Forbes; <http://www.forbes.com/sites/laurashin/2013/05/22/7-steps-to-developing-career-capital-and-achieving-success/>.

Kate Matsudaira is an experienced technology leader. She worked in big companies such as Microsoft and Amazon and three successful startups (Decide acquired by eBay, Moz, and Delve Networks acquired by Limelight) before starting her own company, Popforms (<https://popforms.com/>), which was acquired by Safari Books. Having spent her early career as a software engineer, she is deeply technical and has done leading work on distributed systems, cloud computing, and mobile. She has experience managing entire product teams and research scientists, and has built her own profitable business. She is a published author, keynote speaker, and has been honored with awards such as Seattle's Top 40 under 40. She sits on the board of acmqueue and maintains a personal blog at katemats.com.

Copyright © 2017 held by owner/author. Publication rights licensed to ACM.