

Full Stack Developer vs Software Developer

Full Stack Developer vs. Software Developer, which is right for you? Let's reframe the question.

To specialize or not to specialize, that is the question.

The software industry is not the only industry that faces this question. No matter what industry you're in, it's a loaded question. Like the answer to most queries, it depends. If you're in a corporate setting, specialization is typically the norm, and within start-ups, it's the opposite. In a small group, the specialty will impede a person's ability to create value. In a large group, the value specialization brings has the potential to be off the charts.

I'm a generalist and have always had a propensity to chase the knowledge dragon.



Look! It's a Knowledge Dragon in the wild.

This approach has worked for me, but it's not without its

challenges. I'm a jack of all trades, master of none. As a result, my imposter syndrome symptoms rage on. My specialist colleagues leave me in the dust on several topics. Fortunately, many people recognize the value of a multi-faceted perspective and will accept my lack of more in-depth knowledge of subjects. I can't know everything.

Sometimes I consider deep diving with one technology and sticking with it. I then sweep that consideration aside when another shiny paradigm comes along. I've always loved learning big concepts that apply to all technologies. I've also enjoyed playing with new tools and breaking them ever since I was a child. I would disassemble radios, inspecting them, then put them back together again. I would try to fix them and sometimes with success.

Anyway, that's enough about me, this is about you.

Should you be a full stack developer or a software developer?

Many businesses start with a specific product and specialize in that product. It's usually a successful strategy until it's not. It's 2020, and [Zoom](#) is trending well over the past six months.

Market Summary > Zoom Video Communications Inc NASDAQ: ZM

+ Follow

144.82 USD **+1.38 (0.96%)** ↑

Closed: May 6, 8:12 AM EDT · Disclaimer
Pre-market 145.80 **+0.98 (0.68%)**

1 day 5 days 1 month **6 months** YTD 1 year 5 years Max



Specialization appears to have fueled most of their success. Zoom invested in a specific product with a limited set of applications. In the beginning, their product's quality was low, but they've improved over the years.

Meanwhile, Microsoft is bundling Microsoft Teams into Microsoft Office, and it now has video conferencing. Microsoft is not a company that specializes. It has always focused on creating general solutions for broad applications. It remains to be seen if teams' video conferencing will gain a foothold.

So why am I talking about companies and not you?

Companies are an excellent way to observe the advantages of generalization vs. specialization. We can correlate the

outcomes that companies realize with either approach then choose the right path based on our goals. Start asking yourself some questions.

- Do I want to start my own business in the future?
- Does mastery bring me joy?
- What will I do with a broad set of knowledge?
- What about a deep set of knowledge?

This question is loaded and requires some introspection to answer.

If you're building a team, finding a balance of generalists and specialists is likely the right approach. Without specialists, the hard to solve problems that require a deep understanding of software technology are impossible to solve. Without a generalist, a team might miss out on a perspective that will make a challenging problem go away altogether. The [law of the instrument](#) comes into play when we have tipped the balance too far towards specialization.

I suppose it is tempting, if the only tool you have is a hammer, to treat everything as if it were a nail.

[Abraham Maslow](#) – 1966

I once worked with a guy that used the SQL Server database for everything. He succeeded in tightly coupling the whole enterprise to SQL Server. This guy stored the entire user interface definition in the database. He was enthusiastic about the power of this paradigm. I wasn't sure if he was

joking, but he talked about creating a SQL MVC book.

On the other hand, knowing a small amount of everything isn't useful. It's important to deep dive into key concepts like design patterns, clean coding techniques, etc.

In the end, deciding between being or hiring a full stack developer vs software developer is not cut and dry. This decision depends on your goals, your organization's goals, and the environment. Here is my take on the subject.

Full Stack Developer vs Software Developer

Which is the right choice?

Software development is incredibly complex with lots of nooks and crannies.

The Software Developer is already spread thin and must bear with a large [cognitive load](#).

The Full Stack Developer bears an even larger cognitive load and may become overwhelmed.

Full Stack Developer

Pros

- Can move from team to team
- Variety is the spice of life
- Learns to quickly learn new things
- Brings a diverse perspective to problems

Cons

- [Cognitive load](#) can get out of hand
- Squirrel?!
- Focusing can be a challenge
- Might make the wrong decisions when using complex tools

Software Developer

Pros

- He had one job, and he did it well
- Can become the lynchpin that makes the plan come together
- It's easy to find a job if your specialization is in demand
- Lower workloads bring better focus

Cons

- Might miss the forest from the trees
- Opportunities to save time and effort could be missed
- If the specialization becomes obsolete, what now?
- [Overfocus](#) is a thing. Are you missing an opportunity?

Conclusion

Like any comparison or solution, take mine with a grain of salt. The differences between these two disciplines can be vast or just a few frameworks away from full-stack.

It's not easy to see or pick the right path. Whether you're a developer or a company deciding between the two you have your work cut out for you. Do some soul searching and try to find the right balance.

Death by 1000 Cuts



Death by 1000 cuts, it's an apt expression in today's software world. I'm a user, and I create software. In a single day, I encounter an excessive number of bugs and annoyances. It's not fair to subject anyone as a user or a user that creates software to frustration and suffering.

The Apocalypse

[Some people say](#) that it is a sign of the impending apocalypse. Once I convinced myself that they were right, [I wanted to save and share the message](#), but even that wasn't possible. It's not just bugs that are causing issues but annoyances like having your cursor stolen from you. Unfortunately, someone thought their app was more important than what you were doing, and the operating system allows a developer to hijack your experience.

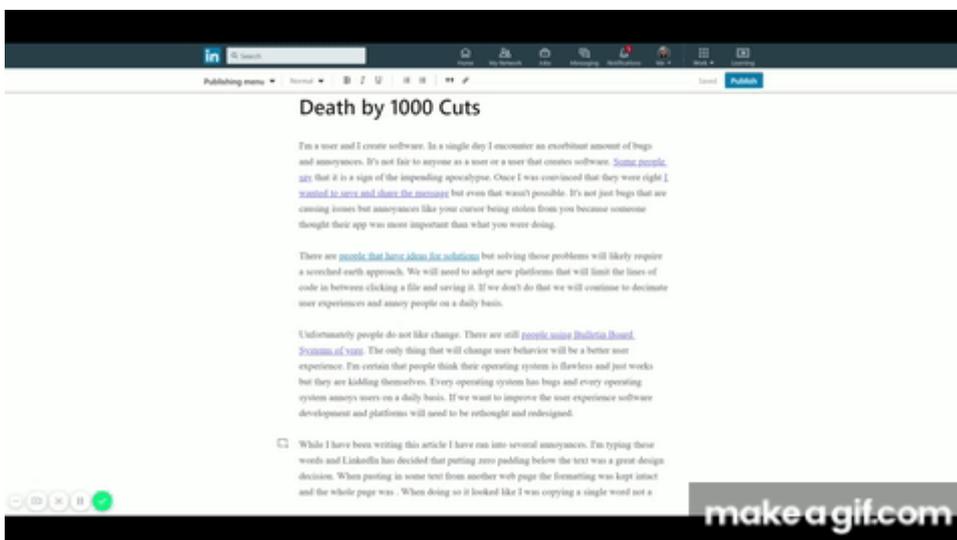
Some people have [ideas for solutions](#), but solving those problems will likely require a scorched earth approach. We will need to adopt new platforms that will limit the lines of code in between clicking a file and saving it. If we don't do that, we will continue to decimate user experiences and annoy people.

Change is Hard

Unfortunately, people do not like change. There are still [people using the Bulletin Board Systems of yore](#). The only thing that will improve user behavior is better user experiences. I'm confident that people think their operating system is flawless and just works, but they are kidding themselves. Every operating system has bugs, and every operating system annoys users daily. If we want to improve the user experience, software development and platforms will need to be rethought and redesigned.

It's a Battlefield

While I have been writing this article, I have run into several annoyances and possibly bugs. I'm typing these words, and LinkedIn has decided that putting zero-padding below the text was a great design decision. When pasting in some text from another web page, the formatting was kept intact, and the whole paragraph went bold.



I just pressed enter after inserting that image, and my cursor was underneath the download dialog. It doesn't take long to realize you are in hell, and there is no way out.

Love & Technology

I love my wife; she does not love technology. I love using software to improve productivity, and we just started using [Evernote](#). Evernote has been around since 2008. I would have hoped that 11 years would be enough time to iron out the bugs, but my hopes were shattered. I was sharing a notebook with my wife, so I grabbed her phone and went through the sharing process. I spent well over ten minutes doing this because someone decided that they would not push an invite to devices but instead require synchronization to occur. Why?!

More Empathy?

I do not know what the answer to this madness is. I do know that I don't particularly appreciate living in it. Humans tend to have blind spots when it comes to understanding the experience of people that interact with them. If we spend more time thinking about how we can improve each other's experiences, maybe we can solve the technology problem as a human problem.