



Walt Schlender [Follow](#)
twitter.com/wschlender
Oct 1 · 5 min read

Why you learn the most when you feel like you're struggling as a developer



The times when I have made the greatest leaps in my development skills have been when I struggled to learn something new and didn't give up. This has not always been easy for me. I have come across a few ideas that have helped me push through my personal blocks.

Perhaps you can relate.

. . .

It's all too easy to feel frustrated when you are faced with software engineering problems that are hard. I sometimes experience a feeling of wanting to give up. In situations like this I've observed my rational mind coming up with reasons why I should go spend time on something I'm more familiar with or complaining that the problem is beyond me.

I know I'm not alone.

I see people on Reddit talking about the challenges they're having with their classes and I have had many conversations with other professional engineers who are facing doubts about their ability to grasp complicated concepts that their colleagues seem to "just get".

Although it's not easy, being able to persevere in the face of these challenges is so valuable.

Over the years I have developed a few mental strategies that have helped me get through these hard times.

Here are a few viewpoints that are helpful for me.

1. A person becomes a better developer through their hard work and effort.

Is it talent or hard work that makes a great developer?

What do you think? Are some people just born great developers or do people become great developers through effort?

I believe that it is impossible to know the true answer to this question so I have chosen to believe that a person *becomes* a good developer through their efforts.

It's a more useful viewpoint. It means that if I am struggling to learn something, I *can* learn it if I work hard.

It also means that I have to accept that things won't always be easy or fun. Hard work is required.

2. When you try to do something for the first time, it might be hard—and you're not supposed to be good at it immediately

As a programmer, I sometimes find myself feeling frustrated because I run into something I don't understand and I feel like I *should*

understand it.

There was the time when I joined a company that was using git and was surrounded by git experts. There was the time I finally had to face the fact that my SQL skills weren't very good.

In each of these instances a part of me felt that I *should* be good at these skills. After all, wasn't I a senior full-stack engineer with years of experience?

The fact was that although I was an engineer with a lot of experience, no amount of experience could change my reality—I was learning the details of these skills for the first time, and at first I wasn't that good at either of them.

Although sometimes new things are easy, sometimes they are not. I have found that a useful viewpoint is that when I do something for the first time I am not *supposed* to be good at it.

It goes like this...

"I've never programmed in Java before—I'm not supposed to be good at it. That's why I'm taking this class"

"I've never committed code to a git repo before—I'm not supposed to know how to do it. That's why I'm asking my colleague for help"

This kind of thinking defuses that voice in my head that's saying I'm no good or I'm going to fail. I very well may fail and I'm probably not good yet. I'm not supposed to be good—that's why I'm giving it a try—to get better.

3. Working on code is not always supposed to be fun, and even though a task is not fun, it can still be done

Sometimes I run into tasks in software development that are not fun.

Some piece of data processing code in my Spark cluster is causing nodes to randomly fail, or some library that should work will not work no matter what I do.

These moments are not fun, and sometimes I honestly would rather do anything other than spend my time trying to figure out what's going on.

The thing is, I know that coding is not always *supposed* to be fun and sometimes I just have to roll up my sleeves and 'do the work'.

Even though times like this can be hard, I've found that they usually have a silver lining—it turns out that the hard stuff is where we learn the most.

4. You learn the most when you face the greatest challenges—and failure at times like this is normal

When I challenge myself by trying to do something that is too hard for me, I find that those are the moments when I truly learn.

For me, dozens of examples stand out.

There was the first time that I picked up a programming book—I struggled until suddenly I had learned to code.

There was the time I was trying to architect larger applications. I struggled until I discovered architectural frameworks and learned about design patterns—at which point we threw away months of work on previous code, started over, and rebuilt our app in a few short weeks using the new design patterns.

There was the time that I finally buckled down and really learned SQL. After that I had the skills to work with my company's data science team.

There was the time I tried to architect our company's data processing infrastructure using web-app technologies. I struggled until I learned about data lakes, ETL pipelines, and distributed computing solutions.

The list goes on.

Each of these experiences had its painful moments. I went down a lot of dead end alleys and threw out a lot of code. I tried a lot of things and I failed, a lot.

But that pressure taught me new skills and made me a better developer. Had it not been for that pressure I never would have learned my lessons.

These days I choose to believe that when I am feeling stressed out, struggling, and a bit nervous it is a good thing. I am *struggling* so I am *learning*.

And...

I have learned to welcome those nervous feelings. They're not comfortable but I think the learning is worth it.

The mind is a powerful tool

You can choose what you believe and your beliefs can give you the grit to push through the challenges you will inevitably face on your path to becoming a better developer.

I hope these mental frameworks (or some of your own) are useful for helping you get past your own struggles.

Don't give up. Keep going.

When you don't understand something, keep working hard and keep trying to understand it. You will be surprised what you are capable of if you just try.

. . .

Thank you for taking the time to read my article.

If you found this article helpful, let me know 🙌🙌🙌.

If you want to read more click the "follow" button below and [follow me on twitter](#).

