

We have used too many levels of abstractions and now the future looks bleak

Published on 2023-10-21. Modified on 2023-11-06.

A big percentage of so-called experts today only know how to configure tools, but they understand nothing about how things work at the deeper level. This is a real challenge and a big problem for the future.

A steering wheel is an abstraction that makes it easier for me to drive my car. Power steering is yet another level of abstraction that further improves the driving experience. Abstractions are nice, they generally improve the quality of life. However, in Denmark we have a proverb that says:

Too little and too much spoils everything.

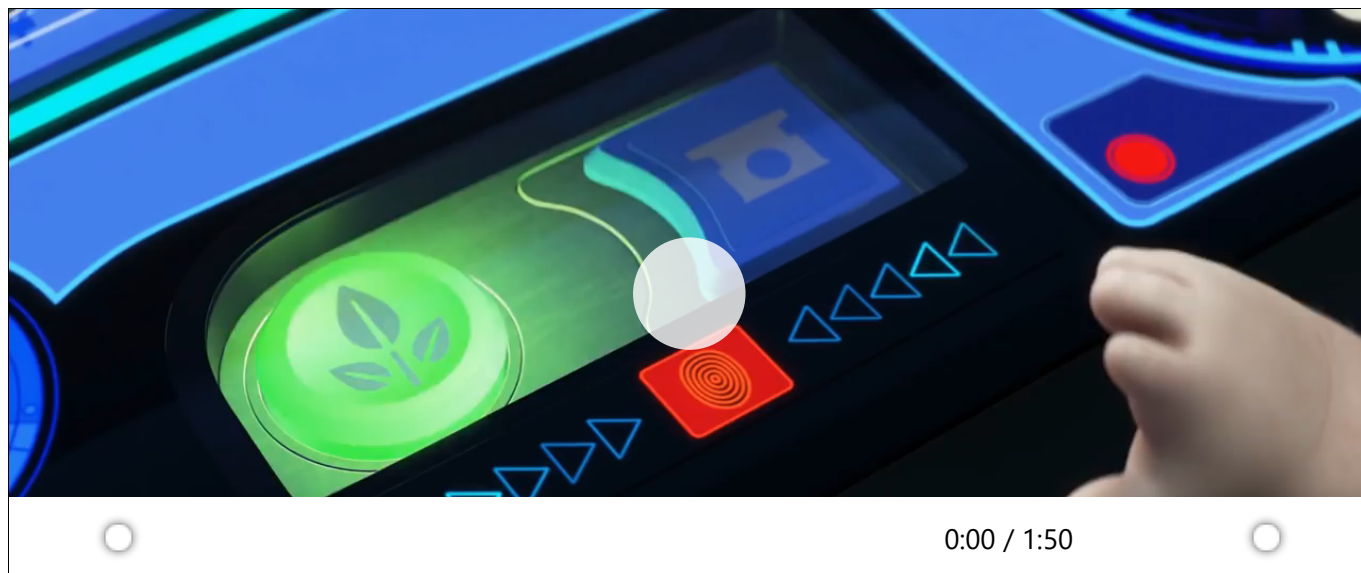
What good does an abstraction do when it breaks and nobody any longer understand how the technology works under the hood?

Everything in the tech industry is driven with a very hardcore eye for profit and very little interest in anything else. So you need to be able to push out new products or new services as fast as possible. This means more abstraction and more automation, less and less people, and less deeper understanding.

Today programmers and system administrators no longer exist, instead we have [DevOps](#) and even [DevSecOps](#), in which the industry is trying very hard to stuff every single task into the job description of a single individual. The tech guys needs to do development (Dev), security (Sec) and operations (Ops), i.e. system administration, but since no single individual can truly master all that, we need to automate as much as possible in order to save money and avoid the complexities of human social interaction between different tech

departments. As a result, the modern tech person is only taught about how to use specific tools, he or she then knows very little about the technology under the hood.

It doesn't help that technology has become increasingly difficult to understand, but more and more of modern life depend heavily upon the tech we're using. So what is going to happen when the level of understanding in the tech industry reaches such a low point in which the majority of people don't even know how to fix the tools they are using?



"Manual scene" from the WALL-E movie.

People have become accustomed to the state of abstraction and they think it's the correct approach and they happily contribute to the mess by adding even more abstraction.

Yes, let's all go back to coding in assembly!

– Sarcastic comment by arrogant developer

We need abstractions, no doubt about it, but every level of abstraction comes with a heavy price which, ironically enough, eventually can cause a massive loss in profit.

Modern programming scares me in many respects, where they will just build layer after layer after layer that does nothing except translate.

– Ken Thompson

Already now a majority of "security people" know very little about security and only about how to use some kind of pre-made penetration testing tool. The penetration testing tool shows a bunch of green lights in its web GUI board and all is assumed well. Yet, a real security expert with evil intentions has broken the system long ago and keeps selling valuable data on the [darknet](#). Nothing is leaked and nothing is discovered. This can go on for years without anyone finding out because, well, the GUI board says that all is **OK**.

Some students today apparently [don't even know what files and folders are](#).

Advice to people studying technology

- Never just follow hype or trends.
- Be curious. Don't just learn tools, try to understand how the underlying technology works.
- If possible, try at least once to manually do what e.g. a configuration tool does for you.
- If possible, try to look at the code for the tool. Even a basic understanding of the code can be very valuable.
- Stay curious. Keep learning. Experiment. Dive deeper into the technology that interests you. If possible, set up a homelab and use it as a playground for learning and breaking things.
- Question everything. Especially things that don't make any sense to you. Don't just assume that someone else knows better - that's how you quickly turn into a blind follower. Sometimes someone else truly knows better, but don't just assume that to be the case by default. And be brave! Stand by the truth and your convictions even if that makes you feel like you stand alone.





The point I am making with this post is not that everything needs to be understood from first principles by everyone, or that you shouldn't use any tools. As I said, we need abstractions. Furthermore, we have people who specialize in different fields such that e.g. the mechanic fixes the truck while the driver drives the truck.

Rather, what I am addressing is the important value of an engineer's attitude toward technology by people working with technology.

In e.g. software development, too many specialists have been abstracted away and replaced by tools and automation and fewer and fewer people understand anything even one layer directly beneath the layer they are working on.

This is a major problem because we will eventually reach a point in which very few people can fix anything in the layers below. And the fact of the matter is that we have already partly reached that point!

About half a year ago I stumbled upon some front-end web developers who did not know that you can create a website without a deployment tool and that you don't need any JavaScript at all, even when the website takes payment. I asked a friend of mine about it, who at the time was teaching a programming class in Python, and he said:

Don't be surprised by this. This is the level today. The industry wants us to mass produce people who know how to "push buttons", not people who understand anything at a deeper level.

I know that there will always be some people who take an interest in the deeper levels, that is not the point. The point is that in software development specifically, we have long reached the point where we have added too many layers of abstraction and too few people understand what they are

doing. The industry is shooting itself in the foot.

If e.g. I am a web developer, whether front-end or back-end, or are doing so-called "integration work", and I am producing websites without much coding or any knowledge of TCP/IP, DNS, HTTP, TLS, security, etc., using only pre-made tools or frameworks, then that will make me about as useful as a monkey with a torque wrench when something goes wrong.

*If you have any comments or corrections please feel free to email them to me.
Also, if you found any of the content on this website useful consider
supporting me on [Patreon](#) ;)*