

Introduction - Knox Lively

We live in a time of instant, or at least near-instant gratification with information at our fingertips within 200 milliseconds or less. After reading a book such as this one, or any book for that matter, the reader often assumes that the tightly woven narrative that is portrayed has always been such. It couldn't be further from the truth. A book, as with a career, or anything worthwhile doing is the culmination of many months, and even years of concentrated effort, often by multiple parties. This book was such. Dave and I worked countless hours while managing relationships, families and our day jobs to follow a goal that we thought worthwhile of sacrificing the better part of a year to.

My career, just as with this book, was the product of daily effort multiplied by time. It didn't appear overnight, there were no fireworks, nor did I even realize I had a career until years down the road.

As I suspect how many of the readers careers began did, mine began in the trenches of IT. Helpdesk Tier 1 to be exact. It was my first salaried position right out of college, and I couldn't have been happier. "I made it!" I thought to myself. I couldn't believe they were going to pay me \$35,000 no matter *how* many hours I worked! Let's just say it didn't take long until I began to strive for more.

From helpdesk I began to teach myself programming and worked my way into more of a Junior Sysadmin-type role at the same company. After learning a couple of programming languages and seeing the power it had to transform my career I was hooked. From there I continued to work my way up to my first traditional Systems Administrator position. In my career path, and my limited understanding of the field, Senior Systems Administrator was the culmination and ceiling for my career path. After a couple of more years I'd landed at just that, a Senior Systems Administrator role. "This is it, I made it." Once again, I thought. But the truth was it didn't really feel any different from my first position working on the Helpdesk. Instead solving one-off problems with people and their desktops, I'd simply migrated to solving one-off problems with servers. My workflow was *still* entirely push-based. Work came to me, and typically at the worst time possible. I was a technical firefighter for all tense and purposes.

Enter my "there's got to be a better way" moment. Sick and tired of legacy systems, legacy thinking, and legacy people I pulled up my stakes and set my sight on new career horizons. I spent months of looking for and researching my next career move.

Luckily not too long into my research I stumbled my way into a DevOps role in Los Angeles. I'd barely heard of the term DevOps, what did it mean? Was it the beta-max of development methodologies, or was there some staying power to this new concept? I wasn't sure, but I sure as hell wasn't about to complain. The money was good enough that I figured I could make myself excited about it for the foreseeable future.

My second stroke of luck was being able to work under a brilliant DevOps engineer. The company was small so it was just us two. I had landed basically what seemed a paid mentorship. A mentorship is something not easy to find anywhere these days, much less getting paid for it. I learned the ins and outs of configuration management tools, proactive monitoring suites, as well as deploying infrastructure as code. This was everything my career had been missing up to this point, a way for me to proactively design my way around future potential problems. Problems that all legacy systems were plagued by. Problems like, snow-flake systems, or rather systems that needed special

hands on maintenance due to drifts in configuration practices across the environment. Other problems like only finding out systems were down only because customers called, or simply trying to keep up with all of the systems we owned seemed to be a thing of the past.

It seemed too good to be true, there had to be a catch. And there was a catch. It required a shift in mindset, specifically the mindset of an engineer or an architect to properly implement and build such systems. The cowboy coder and the typical grump legacy systems administrator mindset was 100% incompatible with this new methodology. I fell into those camps. I hacked my way through every cookbook, recipe, task, you name it. I did so with, not surprisingly, limited success.

I wish I could say I read a couple of books and BAM! I saw the DevOps light and everything was happily ever after. That couldn't have been further from the truth. It took me the better part of my DevOps career to shift my perspective. Only through diligently reading, educating myself, and working with greater engineers on a daily basis was I able to reshape the way I thought about IT, DevOps, and Software engineering as a whole.

My education and understanding of DevOps is now quite simple. Ask me a few years ago if you came to me looking for a definition of DevOps and what it is I do I would have told you to "google it". Not because I'm a jerk, well maybe, but rather because the concept was so elusive to me. It means something different to each person you ask. So, I'll give you *my* definition of DevOps.

Firstly, and foremost, DevOps is a *mindset*. It's not some new-fangled tool made by some trendy software company of which everyone sports their t-shirt. It's a way in which you approach a problem. The tools are, well just that, tools. Pick whichever ones suit you and your organization best. The mindset however, is the real asset. DevOps, when applied correctly, will help you and your organization architecturally plan around problems before they happen. DevOps, its accompanying methodologies and tools *should* help you architect infrastructure that is iterable, scalable and version-able. No different than software development methodologies such as Agile. Agile for infrastructure if you will.

Secondly DevOps *should* empower you and your organization to take on any project no matter how large through the process of breaking large tasks into smaller tasks, delegation, automation, and lastly by using the multiplier of time. Any successful company, no matter the sector, will tell you that this above all is how you build empires.

Thirdly, and perhaps the most important concept DevOps has to offer, is the idea of closing or integrating feedback loops within a tech department, or even across an organization. For far too long departments have been siloed, forced into throwing work over the fence to another, each operating in independence. Even within the tech department processes and technologies are siloed in a way that didn't offer an easy way to see the whole picture. DevOps offers a way of integrating all the various systems and closing the feedback loop for rapid learning, iterating, and re-integration of new knowledge to systems and processes. DevOps is an organic approach to systems design that has been missing in the tech landscape until now.

My hopes and wishes for the reader are that they understand that DevOps is not only a set of tools and practices, but rather a mindset. A mindset that informs each and every decision for their work and the direction of the organization as a whole. However, in reality, we are dealing with concepts larger than DevOps itself. On a more global scale, whether it be creating a family, growing a

garden, or writing a book, each can be achieved by using the same fundamental principles. The principle that through small, incremental, yet measurable changes applied daily can create long and lasting change.

My other greatest wish for the reader is that this book, although loaded with information, quells their fears of mobilizes them to take action on what seems an Everest of tech debt. We all have stared down this mountain and wondered “How can I ever hope to achieve half of what is being asked of me?”. Not enough resources, not enough time, lack of knowledge, etc. The excuses are endless. However, they are not unique to any one individual or organization. There is no DevOps “Never-Never Land” where everything works as it should, no practice ages, nor has any consequences upon its failure. Not even the “big guys” you hear about from your favorite Tech blogger operate in a sandbox. They each got to where they were by adopting a certain set of tools and practices, as well as a mindset that worked for them and their organization. They took these resources and simply started. Each and every day chipping away at the mountain. It’s the only way anyone could ever hope to achieve what is being asked of them and their department.