# [The four pillars of DevOps](https://www.red-gate.com/blog/database-devops/the-four-pillars-of-devops" \o "The four pillars of DevOps)

This is a guest post from Hamish Watson. Hamish Watson is the Operations Manager for Jade Software, an innovative software company focused on solving complex problems for businesses around the globe.  
  
As well as leading a team of consultants who build, standardize and manage enterprise applications built on JADE, .NET and SQL Server, he is the Chapter Leader for the PASS Christchurch SQL Server User Group.  
  
A Microsoft Data Platform MVP, he regularly [**blogs**](https://hybriddbablog.com/) about DevOps and software development.

Firstly, let’s get something out of the way. DevOps is something that polarizes people because there are various definitions of what DevOps is. I actually never use the word with technical people because they hate it.

Instead, I describe DevOps in such a way that they can’t argue with it, by using a slight variation of what Continuous Delivery is:

“Let’s use our tools and create a process that allows us to automate  
the reliable deployment of quality software for our clients.”

This defines the culture of what the tools and processes will be used for, and focuses on the delivery of value to clients. You know – the people who pay us.

With quality software and applications, the experience of clients is enhanced. They’ll love us more. They’ll want to pay us to do more. That’s what you get when you do DevOps correctly.

But first you need to get developers, DBAs and infrastructure people on board by introducing the four pillars of DevOps:



https://www.red-gate.com/wp-content/uploads/2017/01/10px-spacer.gif  
I’ve ordered them this way because I think it’s important to acknowledge the order in which most people approach DevOps.

## Tools

Let’s start with the tools. These are technical people – ergo, they love tools. Tools allow us to make things, write code, and manage stuff.

Thus when technical people think about what they might do with this DevOps ‘thing’, they go to the place where they’re most comfortable.

Having the world’s greatest tools, however, doesn’t actually make things better in terms of delivering value to clients. Not by themselves. You also need to look at the next pillar.

## Process

Without DevOps, the development process is almost invariably quite manual when it comes to deploying new code, whether for the application or the database. It’s often also quite disjointed. Development groups might be practicing Agile, yet have a waterfall approach when it comes to deployment. There could also be communication silos and issues around personnel availability.

So while people might be committed to doing a great job and want to deliver business value to clients, they’re hindered by the processes in place. Step back and take an honest, objective look at those processes, and you can identify what’s wrong – and what you need to fix in order to meet the ultimate goal:

“Repeatable, reliable and automated  
deployment of value to our clients.”

## People

This is where people and, in particular, collaboration comes in. Everyone needs to work together and look at how they’re using their tools, define a better process to use those tools, and automate wherever possible.

I realized when I was introducing DevOps, for example, that having just a Dev and Test environment before deploying changes to Production wasn’t the best process. So the whole team, including infrastructure and operations, DBAs and developers, worked together to implement very small but Production-like environments in our development pipeline. This allowed us to identify problems much earlier in the development process and thus release better code, faster.

I also looked amongst the various teams and talked to like-minded developers and infrastructure people who got what it meant to be able to script the implementation of our infrastructure. This meant that we could work together and define the process of deploying the resources that we needed.

The end result? Gone were the days of sending an email and waiting a month to get infrastructure that was not configured how we wanted. We now had our infrastructure defined as code. We put that code in source control, we versioned that code so that for a particular build we knew exactly what we needed to be built.

So we had the right kinds of tools, the process was being refined and improved, and this was all being undertaken by people. People who finally understood the culture we wanted.

## Culture

Culture. The thing technical people hate. We’re not the kind of demographic who like to talk or think of culture. We liken culture as something where we sit in a room holding hands with the person next to us singing motivational songs.

But to succeed, we have to have culture. It underpins everything we do – it is the partnership of our respective teams. So to introduce the concept of a DevOps culture, start small. I talked about tools and how we as a group needed to agree and work together on tool choice. Our tool choice was borne from a culture of collaborative experimentation.

The process of using those tools then lived or died on whether our culture was one that ensured we understood what each of us needed to do. By understanding developers needed X, Y, Z, and DBAs needed A,B,C, it meant we had an informal peer review process for our… process.

What happened over time was less of a blame culture and more of a helpful culture. Culture will always trump process and by starting small it meant that we had some quick wins and that is a great thing for people to experience. If we’d started too big we would have failed. But starting small and then continually increasing what we could automate meant we had a culture of energetic process improvement.

## And finally

A few words of caution. DevOps isn’t just about tools. DevOps isn’t just about process. DevOps isn’t just about the people. DevOps isn’t just about culture.

If you try and focus on one thing only, you’ll probably do it wrong and fail. You need to embrace all of the pillars. You wouldn’t build a house with only two walls and expect to have a wonderful experience the next time it rained.

I have a saying that I use a lot around the various teams I work with getting DevOps working:

The **Tools** are brilliant but the **Process** fails  
because the **People** don’t get the **Culture**.

DevOps is an approach to culture, process, and tools via people. The measurement of DevOps is to deliver increased business value and responsiveness through rapid, iterative, and high-quality delivery. By embracing the four pillars I’ve listed in this article, you can achieve this.