

## What is the 'Ops' in **DevOps**



### **Acknowledging DevOps' Roots in Infrastructure and Operations**

The term DevOps was coined – by accident, according to Patrick Debois, system administrator and first DevOpsDays conference organizer – as part of the name of the first DevOpsDays conference in Belgium in 2009. Debois chose DevOps because Agile System Administration was too unwieldy. Incidentally, during the emergence of the Manifesto for Agile Software Development in 2001, the term 'conversational' was discussed as an alternative for 'agile'. So, DevOps could have ended up being called Conversational System Administration.

Debois had previously spoken about Agile Infrastructure. Inspired by what Agile had done for software development, he wanted to explore the possibilities of Agile in the Infrastructure and Operations domain. There was never a grand plan behind the word, so we should not try to read too much into it but instead, focus on what experience is available to help us with our challenges.

Ten years further down the road, DevOps has become decidedly more oriented to development and deployment rather than operations, so it's time to revisit its roots.

### **Going Quickly Enough, and in the Right Direction**

So, what are our challenges? Well, the Agile and DevOps movements have certainly increased the velocity of development and deployment. Although, come to think of it, 'velocity' might not be the right word. Velocity comprises speed and direction. Agile and DevOps have speeded things up, but have they contributed to improving the direction? In other words, have we made better investment decisions?

DevOps started from IT operations rather than application development. Yet it now seems to be Dev-heavy and in need of rebalancing. Which IT operations and service management practices are useful additions?

Download hundreds of FREE DEVOPS resources at [Orbus Software](https://www.orbussoftware.com/resources) today!

[www.orbussoftware.com/resources](https://www.orbussoftware.com/resources)



Agile maybe. DevOps not. At least, not in my book. Looking at the DevOps Handbook, that is about as good as DevOps guidance gets, the 67 technical practices – the most concrete guidance – are mostly about doing whatever has to be done as quickly and as reliably as possible. The functionality is a given that has been decided long before DevOps practices come into play. There are DevOps principles that emphasize the benefits of product and customer centricity, but that’s about it. There’s a bit more guidance in the Agile community but it’s most about doing things quickly and deciding on priorities – not coming up with them in the first place.

In summary, more speed than direction.

## Getting Enough Return on Investment

Another challenge is getting return on our digital and IT investments. Value is only realized when somebody acts on a decision that has been improved by correctly interpreted information that has been derived from an automated information system.

Contrary to how they usually think about themselves, IT departments don’t deliver value. At best they co-create it, together with the users. It takes two to tango and it’s up to the users to understand and use the functionality effectively, interpret the information correctly, take the right decisions and act upon them.

As for the IT function, it has to provide access to the information system and to support its use when needed. This support can be automated (e.g. password reset) or manual (e.g. discussing an issue with an expert). This where most DevOps teams start to fall short on expertise. They’ve elevated the process of deploying software to an artform, and proactively monitor systems and take quick action when needed. But then the Ops part of DevOps starts to run out of steam. And more is needed to ensure that the customers and users can realize value.

## Organizing the Ops in DevOps

There’s an organizational issue to consider. DevOps teams are usually organized around products or platforms. Either developing and running an application. Or providing the infrastructure or platform that other teams need to develop and run applications. The teams are usually small, relatively autonomous and multidisciplinary. Multidisciplinary means that the team comprises most of the expertise needed, and individuals can often perform several tasks. The organizational question is whether some tasks should be centralized. The service desk, for instance.

Does it make sense to get each team to fulfil a service desk function? You’d probably agree that first line support should be centralized. But not third line support. What about second line? It depends.





**Phone:** +44 (0)203 824 2907

• **Email:** [enquiries@orbussoftware.com](mailto:enquiries@orbussoftware.com)

• **Website:** [www.orbussoftware.com](http://www.orbussoftware.com)