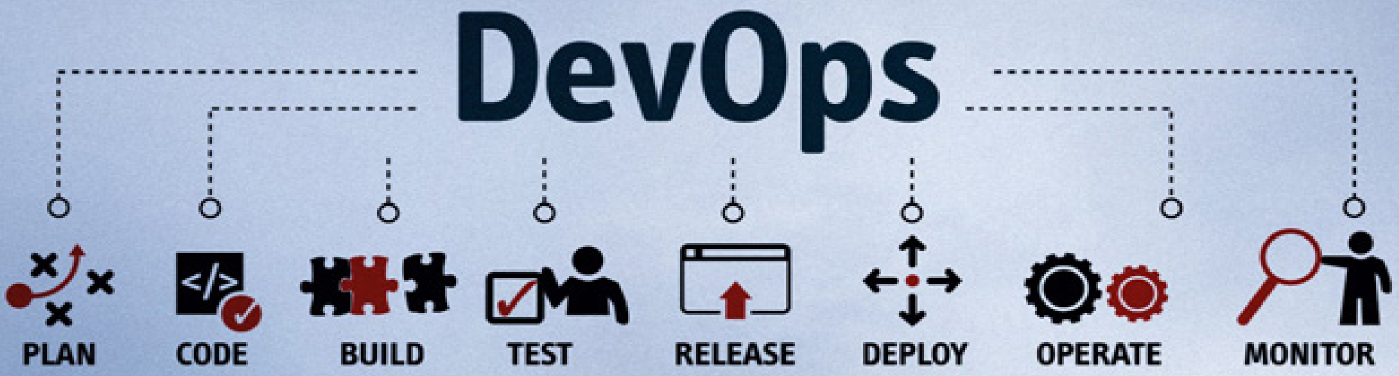
DEVOPS SUCCESSFULLY COMBINING DEVELOPMENT AND IT OPERATIONS

BASICS AND TOOLS FOR A SUCCESSFUL DEVOPS IMPLEMENTATION



CURT W. MEISTER

Introduction

Unlike other methods and frameworks such as Scrum, Kanban, ITIL or similar, there is no "DevOps" with a generally accepted definition. Thus, there is neither a uniform - or at least leading - organization that represents the topic of DevOps nor defines what exactly is part of DevOps and what is not. In the German-speaking world alone, there are almost half a dozen certifiers, some of which award the same or similar DevOps titles and in turn require different examinations, which ask for different, sometimes even contradictory requirements or statements.

Nevertheless, there are some basic themes or principles that most DevOps statements contain - such as efficiency, collaboration, and improved communication. These are not things that many other frameworks and methodologies would not claim for themselves.

Now if we look at a few definitions of how authoritative institutions define DevOps, we find:

Peoplecert: *"DevOps is a term of art that promotes the evolution of existing IT best practices from ITIL, Lean and Agile into an approach to development and operations that supports automation and continuous delivery and fosters a culture of collaboration and learning to help IT deliver business value better, faster and cheaper than ever before."*

Wikipedia: *"DevOps is a process of software development and delivery that emphasizes communication and collaboration between product management, software development, and operations professionals. DevOps supports this by automating and monitoring the processes of integrating testing and deployment through software. Further, it also automates and monitors infrastructure changes by creating a culture and environment where building, testing and releasing software can happen quickly, frequently and more reliably."*

Gartner Group: *"DevOps represents a shift in IT culture, a focus on rapid delivery of IT services by adopting agile and lean practices in the context of a systems-oriented approach. DevOps emphasizes people (and culture) and seeks to leverage collaboration between teams, from Operations and Development. DevOps implementation applies technology - specifically automation tools that can leverage increasingly programmable and dynamic infrastructure from a lifecycle perspective."*

The problem is probably rooted in history. Unlike some methods or frameworks, even at the beginning of development there was not one organization or person who took the lead in the direction of standardization or developed a specific method. Rather, DevOps emerged from agile principles within software development, which sought greater collaboration and communication and the elimination of silo thinking.

Often cited as the beginning of the DevOps movement is a talk Flickr employees John Allspaw and Paul Hammond gave at the 2009 O'Reilly Velocity conference in San Jose, California, titled " 10 Deploys a Tag: Dev and Ops Collaboration at Flickr ".

The topic was clearly an issue of the day for people in different organizations around the world, and after a relatively short period of time, the Devopsdays conference was held in Ghent, Belgium, from where the term "DevOps" began to be perceived and used more widely.

At first glance, the absence of a uniform path and uniform procedural principles may be somewhat off-putting. At second glance, however, this actually also offers a strength for implementation. Instead of a standard set somewhere that is emulated, it is necessary to include one's own needs and the people involved. This can significantly increase the probability of choosing a suitable approach instead of a standard.

The concept of DevOps is broad, but almost everyone will agree that increased efficiency, collaboration and communication are positive changes. However, it's important to discuss the details with your team before implementation. Ideas like increased communication are pointless if each employee has a different approach to achieving them.

Companies that have successfully implemented DevOps include a myriad of small and medium-sized companies as well as global corporations such as Amazon, Netflix, Target, Walmart, Nordstrom, Facebook, Etsy, Adobe, NASA, Starbucks and Sony Picture Entertainment.

For the purposes of this book, we will characterize DevOps as follows: DevOps is a philosophy that promotes a set of principles, procedures or practices, and values with a focus on collaboration between development and operations. The prerequisite for this is a cultural change of the entire organization with the goal of promoting communication and collaboration. This involves balancing the demands of change with those of stability and predictability.

Reasons for using DevOps

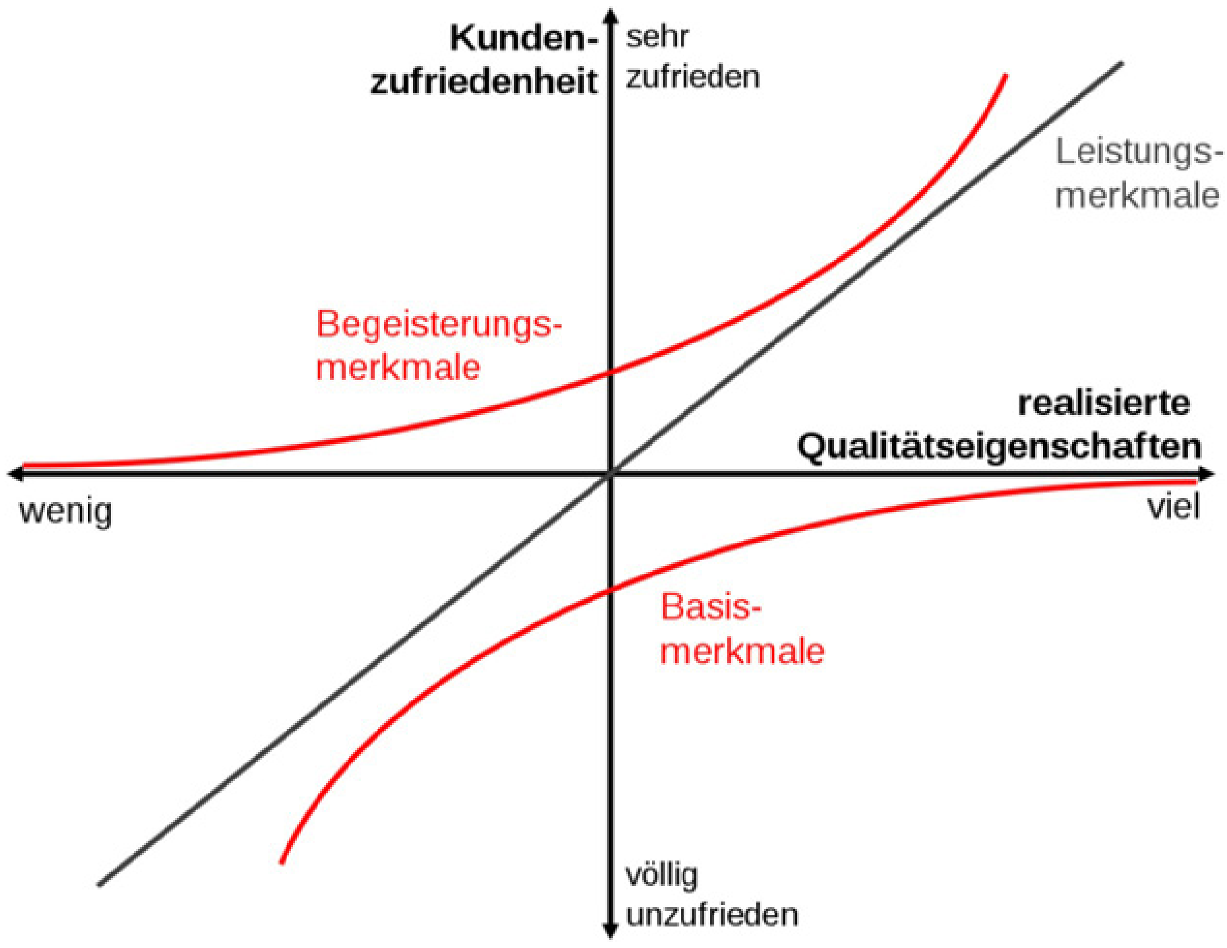
Realization of the business benefit (business value)

Even if in reality some organizations live it differently: Business value is always defined by the customer and not by the supplier. It corresponds to the level at which a service or product meets - or exceeds - the customer's expectations. So the context in which DevOps is typically used would be defined as *"business value is the level at which a service meets or exceeds the customer's expectations."* [1]With this, it is understood that different customers (groups) have different perceptions of business value based on their demands.

When we move in the context of services, a common view is to see business value as a combination of (service) quality, i.e. the extent to which the customer's requirements are met, costs, and the time in which a service is provided. In this context, business value is generally understood to be positive if the three dimensions are perceived to be balanced. It will quickly be seen that different customers apply very different standards in this respect. While some customers tend to be very price-sensitive and are prepared to make compromises in terms of quality or speed of service provision, others are very focused on maximum quality and are prepared to dig deeper into their pockets to achieve this. Others, on the other hand, are in a predicament and are prepared to pay virtually any price as long as their problem is solved.

The Kano model

The Kano model is an approach that helps visualize and better understand different customer preferences.



2

The Kano model distinguishes between five different types of customer perception of products or services:

*"**Basic characteristics** that are so fundamental and self-evident that customers only become aware of them when they are not met (implicit expectations). If the basic requirements are not met, dissatisfaction results; if they are met, however, satisfaction does not result. The increase in benefit compared to differentiation from competitors is very small.*

***Performance characteristics** are aware of the customer, they eliminate dissatisfaction or create satisfaction depending on the extent of fulfillment.*

***Enthusiasm features,** on the other hand, are benefits that the customer does not necessarily expect. They distinguish the product from the competition and generate enthusiasm. A small increase in performance can lead to a disproportionate benefit. The differentiation from the competition can be small, but the benefits enormous.*

***Irrelevant features** are irrelevant to the customer, both when they are present and when they are absent. Therefore, they cannot cause satisfaction, but they also do not lead to dissatisfaction.*

***Rejection Features**: Lead to customer dissatisfaction when present, but satisfaction when absent."* [3]

It may be that a feature is assumed to be a basic feature by one customer / one customer group, but is completely irrelevant for others, or that it is able to inspire. It is also to be expected that a former enthusiasm feature will, over time, become a performance feature or even a basic feature in the customer's perception. For example, several years ago it was an absolute wow feature to have a cell phone with a touch screen that no longer required separate dial