// Don’t waste the value of Devops

BlinkLane’s vision on how to improve software development

Viktor de Keijzer
Frank Tetteroo
Alex Podkanski

To enable corporate agility in the digital age, corporate IT must have practices and tools that allow fast, frequent and reliable deployment of features to customers and end-users[1]. In this article, we reinforce the belief that DevOps is a broad set of practices, principles and tools to remove every last bit of waste from an end-to-end IT delivery chain. Understanding how to identify this end-to-end delivery chain is key for understanding DevOps. We think there is a lack of clarity on what DevOps entails and how it connects to other concepts of agile or scrum. With this article, based on the variety of experiences that we collected, we want to provide clarity on the background of DevOps, why it *can* make sense, what benefits it offers and how an organization could implement it to improve its end-to-end delivery chain.

// Introduction

It was Patrick Dubois who, in 2007, analyzed a Belgium data center migration project and singled out a huge bottleneck in the delivery chain[1]. The handover of work from the Development team to the Operations team proved greatly inefficient. In order to solve this delay in delivery, he proposed to integrate both teams into what we now call a DevOps team. What Patrick in fact did was use the very basics of Lean Management to identify and eliminate waste. It was a clever solution for the problem in the project at the time, and he was not aware that he started a global movement with an unforeseen impact on the way that IT is managed.

The DevOps movement started to grow, being seen as a way to deal with issues of speed that IT departments have been struggling with for a long time. With the absence of a strong and visible thought leader on the matter, the way DevOps currently is explained varies from person to person. Inter alia, Blinklane Consulting recorded explanations of DevOps such as:

- DevOps as a simple integration of Development and Operations in IT
- DevOps as a cultural shift within an organization
- DevOps as a mindset to automate everything

Even though every one of these explanations touches (in one way or another) the basics of DevOps, they all neglect to explain the *core* of what DevOps really entails. Authors often explain *how* DevOps can be used whereas we believe it creates more clarity when the *core* of DevOps is explained. We want to shed light into this grey area of definitions and explanations by focusing on key points. Hence, the following article will answer the following question: What does DevOps mean according to BlinkLane Consulting?

// A broader perspective on agility

As recognized by many, the main benefits of becoming agile include faster time to market, better quality products and services and earlier Return on Investment (ROI)[2]. Recently, agile working methods such as Scrum have been implemented to increase effectiveness and efficiency of software development. Even though this has proven great success in many cases, we experience that many organizations and their customers still do not yet experience the full benefits of being Agile. This is due to the fact that most agile methods and frameworks (e.g. Scrum), as they are introduced in many companies focus on the development stage of software manufacturing. However, the true benefits, in our perspective, are only consumed in production.

According to BlinkLane Consulting, *real* agile software delivery can only be accomplished when looked at from an end-to-end perspective and not only when our eyes are locked at the development stage. This comes from the belief that software only has value when it's available to the customer, or, in 'technical' terms: being put in production.

BlinkLane Consulting approaches DevOps as potential next step for organizations in their agile transformation. Firmly put, the true key to success lies in the end-to-end focus in order to delivery more value continuously and faster. With this end-to-end focus, the business, development, operations, quality assurance, compliancy

[1] Kim, G., Debois, P., Willis, J., & Humble, J. (2016). *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*.
[2] 12th annual state of Agile Survey, VersionOne.

regulators and architecture *all* collaborate to deliver quality products and services to the end-customer in the shortest sustainable lead-time.

The 'why' of DevOps is therefore answered: to become truly agile, one must improve the *entire* delivery chain from the early start of development to the product in use at the end. DevOps helps in this by providing a set of practices which help to improve on the whole chain.

// Understanding the end-to-end value chain

Today's organizations want to have a fast, steady flow from input of the customer straight to being in production[3]. This is desired to be able to respond to the customer's needs as fast as possible. To do so, we want to maximize value delivery in the shortest sustainable lead time. When measuring the lead time, we start the 'clock' at the moment the business hands over the requirements and stop measuring when the software can be used by the customer. In order to get to this final stage, various actions such as testing, quality checks, knowledge transfers, handovers and other steps and procedures are needed which inevitably take time. Yet, when the goal is to have a fast and steady flow of usable software the futile time (i.e. 'waste') should be reduced to a bare minimum.

The well-known Lean approach has been used for decades to reduce waste in a delivery pipeline. Following this line of reasoning, it would be rightful to use the same core principles of Lean on IT. It is exactly what Dubois did back in 2007 and what we believe are the core basics of DevOps: lean helps to identify the waste within the delivery pipeline. With an important remark: traditionally, Lean focuses on the reduction of waste in an operational environment. DevOps stretches out over the entire delivery chain and thus just using the terminology of Lean-IT would be too easy.

As such, figure 1 shows a fictional end-to-end delivery chain of software development. In this fictional example, the 8 wastes, taken from Lean[4], are allocated throughout the whole delivery chain, both development and operations. While these wastes are located individually among the value chain, they could also occur next to each other in operations, development or in between.

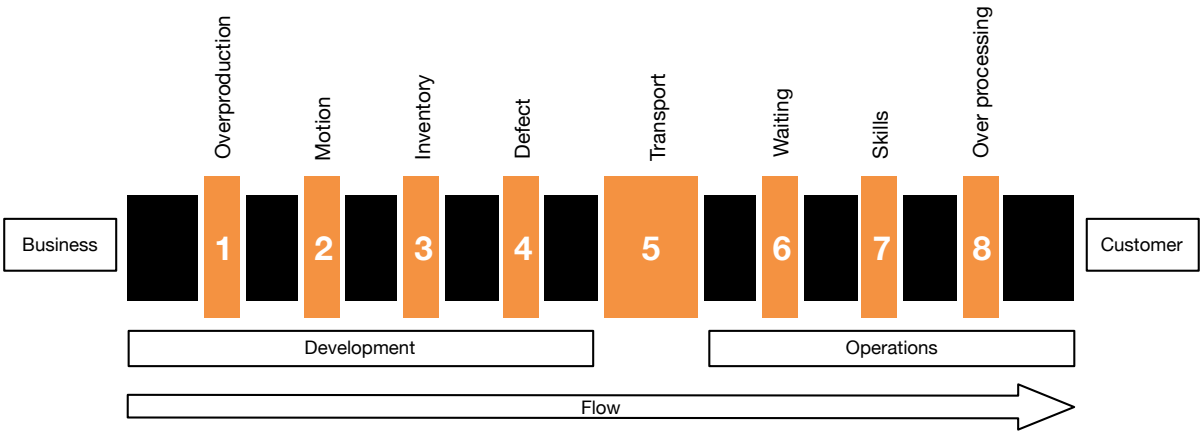

Figure 1. An example of at what stages waste could be found within the value chain

[3] Forsgren, N., Kim, G., Humble, J., & Brown, A. (2017). State of DevOps Report: 2017.
[4] https://goleansixsigma.com/8-wastes/

As it identifies the pieces of waste with number ranging from 1 to 8, table 1 defines these pieces of waste and gives a practical example in the end-to-end value chain of software.

	Waste	Definition	Example
1	Overproduction	Producing more than customer demands for	Creating solutions before or after the demand is present.
2	Motion	Motion of people that does not add value	Time of travel between Developers and Operational engineers to agree on the how, what and when of a release
3	Inventory	Excess material at hand that is currently not in use	Code that has been developed but hasn't been committed to the main branch.
4	Defect	Work that contains errors and requires rework	Defects not found because test is outside of development control
5	Transport	Movement of information or material that doesn't add value	Handover from Development to Operations is done via formal processes
6	Waiting	Idle time created by people, material, information not being readily available	Waiting for approvals (checks), availability of others
7	Non-utilized talent	Not, or underutilizing, the talent of employees	Developer only coding as defined, no creativity, no responsibility
8	Over processing	Overly complex processes delaying progress	Almost impossible to get access to a certain environment. Extensive documents need to be filled before a release

Table 1. An overview of forms and waste and according examples

// How to start with DevOps

Now that we have created a clear understanding of what DevOps really entails we believe it is time to explain how you can start with DevOps. We've seen many companies and spoken with numerous C-level executives, (senior) management and IT-engineers in the past struggle with how to start with DevOps. (Basic) questions that arise include, but are not limited to, "How can I reorganize my organization to establish DevOps teams?", "What automation tools can I use?" and "What type of culture do I have to advocate to support a DevOps mindset?". Every single time we've experienced that an answer is only relevant when the individual situation is clear and the solution is tailor-made. In other words: For BlinkLane Consulting, DevOps is much more a mindset, a way of working than a clear cut and one-size-fits-all package. We believe that in order to ensure a successful implementation of a DevOps way of working, an understanding on the core basics of DevOps, followed by a Value Stream Mapping is needed.

As the traditional Lean states that improvement projects start with a decent Value Stream Mapping we propose the same for software development. In our consulting practice, we've encountered various situations in which identifying the waste during a Value Stream Mapping workshop proved extremely valuable. In addition to this, quantifying the waste in terms of 'time' or even 'money' often proved to be jaw dropping for both development team ("Am I waiting for the QA to complete the manual functional test *this* long?") as leadership ("Now I understand why our initial investment doesn't show it's promised ROI"). Thus, the first step to take is to visualize the Delivery

Pipeline from start (business requirements) to end (product *(i.e. code)* in production) and create tangible insight in the waste. In other words: quantify the waste in terms of – inter alia – time and/or money.

After the identification and visualization of the Value Stream (i.e. Delivery Pipeline) the next step is the most obvious, to eliminate it. This is the step where the true creativity of all involved in the delivery pipeline comes together. We have considerable experience finding sustainable, workable but mostly effective ways of removing the waste. They can be as easy as bringing Development and Operations together and creating the right mindset to more technical approaches, such as introducing automated testing throughout the whole chain. It cannot be stretched enough that either of these solutions are team dependent and only relevant when the bottlenecks of this team have been identified during the Value Stream Mapping.

We provide several DevOps practices that are more widely acknowledged[5]:

1. **Collaborative Culture:** End-to-end goal alignment and cooperative workflows are essential. No amount of technology will suffice if Dev, QA and Ops leaders and teams do not cooperate.
2. **Design for DevOps:** DevOps prefers products that are designed in modular fashion (using microservices, 12-factor apps and containers) - those are easily integrated into high-performance DevOps. That is not to say legacy products cannot benefit from DevOps, but modular, immutable architectures are most readily suited to take full advantage of DevOps.
3. **Continuous Integration (CI):** Merging forward quickly with competent integration practices that minimize rollbacks, interruptions and costly delays in the pipeline.
4. **Continuous Testing (CT):** Relevant quality testing at speed requires multiple strategies that ensure efficient coverage throughout the pipeline with minimal failures delivered to production, yet quick enough to not bottleneck the pipeline.
5. **Continuous Monitoring (CM):** Real-time lifecycle intelligent active monitoring and analytics are essential to accelerate real-time decisions at each stage of the pipeline and avoid bottlenecks.
6. **Elastic Infrastructure:** Infrastructures that are resilient and elastic perform much better. DevOps applies to most types of infrastructures including private data centers, special purpose bare-metal systems, virtualized functions, containerized packaged applications, private and public clouds and hybrid cloud environments.
7. **Continuous Delivery (CD):** Modular delivery packaging preparation and deployment solutions with orchestration of virtualized and containerized applications are preferred over monolithic mutable applications.

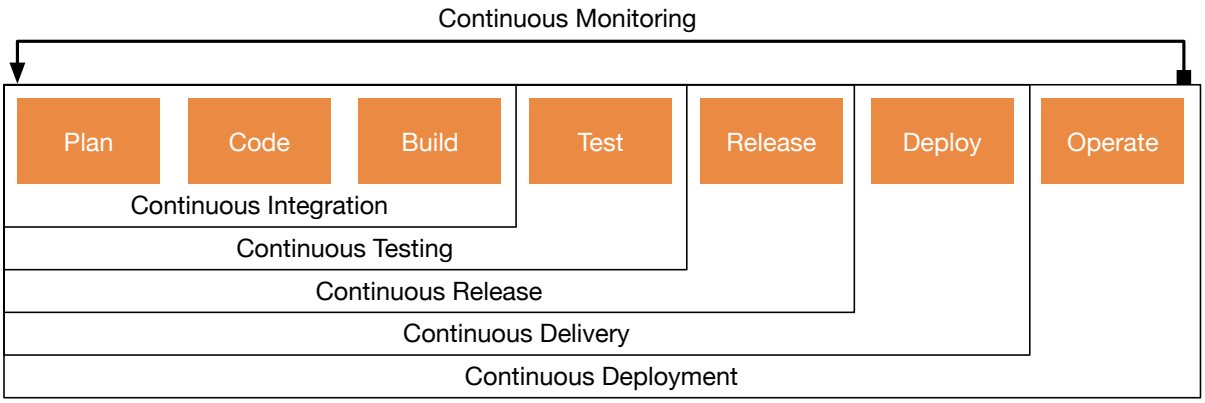

Figure 2. An explanation of where several practices apply

[5] https://devops.com/7-pillars-of-devops-essential-foundations-for-enterprise-success/

// Conclusion

We don't consider DevOps to be the silver bullet of software development. A silver bullet as an easy to use, immediate solution to all the problems, under all circumstances, that arise along the value chain. We *do* believe that DevOps can be drilled down to three easy to understand steps:

1) Define the end-to-end value chain
2) Create insight in the flow of the chain through a Value Stream Mapping; identify bottlenecks
3) Use the best-practices and tools to eliminate the bottlenecks

Thus, we believe DevOps consists of several best practices that help to eliminate all waste from the end-to-end delivery pipeline to enable this fast, frequent and reliable delivery of value. Thus, by creating insight in the entire software delivery chain and identifying waste, we can use best practices, procedures and tools to eliminate that waste.

BlinkLane Consulting

BlinkLane Consulting is an advisory firm founded in 2007. In our 10 year lifespan, we have evolved together with our clients. We continuously innovate our services to keep delivering the value our clients need in order to deal with today's and future challenges. We help our clients increasing business value from IT investments, act more agile and innovative and transform their organizations for the future. For 2018, we focus on the following four themes:

- Digital Transformation
- Strategic Sourcing
- Scaling Agile
- Innovation & Growth

Spaces Zuidas – Barbara Strozzilaan 201 – 1083 HN
Amsterdam – The Netherlands
T +31 204 080 860 – info@blinklane.com –
www.blinklane.com