DevOps - Challenges and Opportunities

Sharing Key Principles, Techniques and Success Factors

Insights from diverse experience improving DevOps

The market is ever more competitive, and products are ever more complex. Small changes can take a long time to reach the customer. As such the focus on DevOps is becoming ever more important. The subject is large, varied and challenging. In this article we want to share our experiences and perspectives from a developer, as well as from a leadership and innovation point of view.

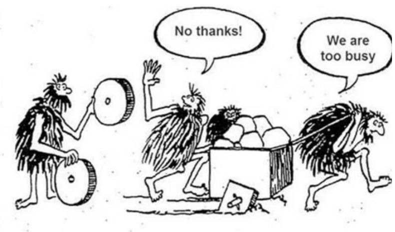

Fig. 1 – The challenge of changes

The goal of DevOps is to deploy quality releases faster, more often or in other words bringing the changes (whatever they are, such as Design Change Requests, Enhancement Change Requests or just Problem resolutions) to the end customers faster, better, and easier. But the path is not an easy and obvious one. In addition to technical challenges, making large scale changes to the development process is not simple and may also require a change in culture which is never an easy task. However, the opportunities are important.

Bringing in transformational change to teams requires high energy, emotional intelligence, and deep commitment to overcome objections and inertia, as illustrated in a humoristic, but a very clear way in Figure 1.

As such first you need to make sure that the desired outcome is understood by all and that there is a real quest and understanding of the need to achieve those goals.

Following we want to elaborate on some principles, techniques, and success factors which we believe can help your DevOps endeavor and act as a baseline to build sustainable initiatives with successful outcomes.

Guiding Principles

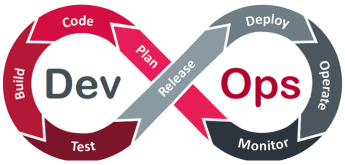

Fig. 2 - The DevOps Cycle

Over the last years, we have seen that following guiding principles can help to take charge of the DevOps improvement goals and make the efforts more effective.

Fast Feedback

A key factor for developer productivity is focus on a single task (use case). A developer hates to wait and loves to see his contributions being deployed and used – but the longer it takes to see the results the more disconnected the developer feels and who doesn't feel more prouder than when seeing their work in action. But the path from planning to coding to deployment can be a very long and tedious one and the developer is very much dependent on the given DevOps infrastructure and initiatives. Ideally, he can commit his changes and get immediate feedback from the automated unit, integration, systems tests but this is as yet a dream wish for many.

However, speeding up that feedback will allow to focus on getting the task done on the environment and context of the product you may have further needs with respect to the level of tests required to allow deployment. We at EVOCEAN are much involved in safety critical products which have an additional certification strain on the testing of the requirements. Fast Feedback is the most important principle, but we need to differ which fast feedback we want to focus on as we cannot get everything at the same time. We need to first focus on what gives the most value. The value will need to be defined having the complete context of the DevOps initiative in mind – it is not always the most visible which is the most valuable and having the largest leverage or impact. Lean principles to find the real bottleneck come here to help. We see a lot of effort being wasted on the wrong focus or value, as many just react to the most visible or the loudest shout. For leadership and management roles it is very important to listen, but also to analyze the complete picture. Being humble and promoting an open and transparent culture can very much help here.

Fail Fast

The second principle "Fail Fast" may be even more important. The question here is how can we fail the fastest, as our code will break, and we need to make sure we break as fast as possible. This is not always easy as you may even be testing against "wrong" requirements making the initial test pass but finding issues down the road. Still, making sure you always think of how to fail fast, you will save not only time but also valuable machine resources and as a result get faster feedback to improve and iterate.

Failure is endemic in many software projects. Many developers report a highly emotional journey and release cycle, full of manual steps which create huge opportunities and risk for errors, as well as inefficiencies, long delivery cycles, and chaos in project organization. The diagram "Emotional cycle of manual delivery" shows this very illustratively.

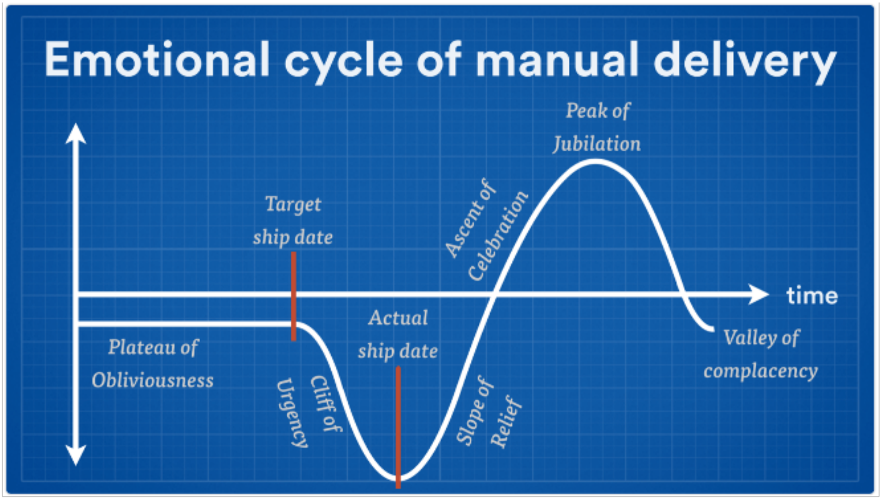

Fig. 3 The other perspective.
(from Atlassian Inc.)

Again, culture has a huge impact here as no one likes to fail. But seeing failure as a learning process, allowing to gain insights will help to make the product more robust and finally the user experience better.

Resource Optimization

Our third principle “Resource Optimization” sounds logical, and many will ask why this should even be mentioned. We however have seen in our work a lot that the global view and context can be missing, and the potential of resource optimization is not used. Further, much has evolved in this field in last years through the impact of for example

virtualization and artificial intelligence. Let us iterate on what we mean by “Resource Optimization.”

First, we need to define the various resources. From a global point of view, we have differentiated as follows - all with a perspective of time and money:

1. Human resources
2. Internal Machine resources
3. Cloud resources (External)

We have differentiated between internal machine and external cloud resources, as there are very important different cost factors on both, which we need to take into consideration. Many organizations are banking on cloud resources for the future, mainly due the flexibility of the availability and scalability. But this also comes with a cost, which many operations are having issues of keeping under control. The developers are happy with allocating abundant cloud resources and the financial control will try to limit this. But the real challenge is to make sure that the allocated resources are fully needed and also utilized. The question is who is taking the lead for this. Who is in charge? Also here tools can help.

Underlying Techniques

Let us now iterate on the underlying main techniques to achieve DevOps.

Automation

Automation is key in all DevOps operations. Here again the most important question is how to prioritize which automation, as it is a long way to a fully automated DevOps pipeline. A large effort will go into the test automation on a unit, integration, system, deployment level but also into the integration of static analysis tools to help the developer as early as possible in creating reliable, functional and correct code.

Integration

With Automation Integration goes hand in hand. The question here is how to build the integration with a view on sustainability as updates will happen on all sides and we want to be as much as possible agnostic to updates, such that the pipeline does not break with every change. Further integration needs experience in the detail, but also having the global perspective in mind in order to focus on what is really important.

"The most powerful tool we have as developers is automation."

Scott Hanselman

Leverage

Further to the visible tool integrations which are needed to meet the primary automation goals there is a huge potential for more leverage, such as applying DevOps platforms giving a completely new perspective including data insights for better decision making, or in various detailed aspects of the DevOps pipeline such as build acceleration for faster feedback or resource optimization. Especially when using cloud resources, you want to ensure the optimal usage, ease of administration, assure correct provisioning, timely activation and deactivation and scaling depending on the real DevOps needs. All this may be supported by specialist tools, and these could take the DevOps efforts to a next level. Having a wide experience in many different projects our consultants have been able to trigger improvement, which were not always seen by our customers. It is however important that there is a real measurable and sustainable benefit of additional or a change in tools.

Important success factors

Following we will iterate on what we believe are the important success factors.

Understand Context

"Improving daily work is even more important than doing daily work."

Gene Kim

At the beginning of every DevOps journey and thus improvement project it is key that you understand the DevOps context and the goals which you and your management want to reach. Initially there are no shortcuts. Having a full picture and if possible, a Value Stream Mapping will help a lot to find out where to start and focus on. DevOps is also about communication across different disciplines and breaking down information silos for better customer experience. It also helps to build up transparency and for better monitoring and data insights, which will help in making more founded decisions – always with the goal to improve the business outcome.

Focus on customer value

Always focus on customer value. But be aware. There is more than one customer. Not just your end customer – which of course is the most important and your macro goal, but you may need to first focus on other customers (internal stakeholders) and thus various micro-goals which should be a result and clearly visible from your value stream analysis. Also, always make sure you communicate your current goals and expectations, and wherever possible define success criteria and metrics.

> *"DevOps represents a change in IT culture, focusing on rapid IT service delivery through the adoption of agile, lean practices in the context of a system-oriented approach. DevOps emphasizes people (and culture), and seeks to improve collaboration between operations and development teams. DevOps implementations utilize technology— especially automation tools that can leverage an increasingly programmable and dynamic infrastructure from a life cycle perspective."*
>
> Gartner

Tool experience

A further key success factor is the tool experience and insights. Understanding the tool and its DNA is very valuable for making sound decisions. Of course, the tool choice may also depend on other decisions and these need to be respected. The tool experience andInsights are very important, especially as the tools are getting ever more complex and as such applying them with the respective Tool DNA in mind is a key success factor for sustainability. However, as the tools evolve and as others appear on the market the challenge will always remain to avoid technical dept and to keep monitoring the landscape to make well-formed decisions on introducing new tools or replacing tools to help in reaching the goals to deliver products and services faster and better. Further, the emergence of tools supplying real-time data insights of engineering processes and product quality will in future help here even more to support to make better decisions and help the continuous improvement moving towards the DevOps goals.

Network Leverage

The complexity of DevOps is large, and the expertise required very fragmented, finding a single partner who can do everything will be a challenge and may be available at the time of decision but may likely not withstand. As such finding a partner whom you can trust and who knows when and where to reach out to others when needed will be key. At EVOCEAN we build on a valuable network and combined with our deep expertise we focus on following areas to advance DevOps:

- *In safety critical product development*
- *In test automation and integration*
- *In complex DevOps environment with many tools*
- *In leveraging GitLab for DevOps*
- *In supplying engineering insight for better decision making*
- *In leveraging build acceleration*
- *In ensuring cloud optimization*

Conclusion

"DevOps is not a Goal, but a never-ending process of continual improvement"

Jez Humble

As a major player in the field of DevOps says: "DevOps is like following a diet or exercise." Diet doesn't say much to us, but exercise does. However, easier said than done. Most DevOps tasks are important but not urgent and a lot of management focuses on the urgent tasks or just on deploying faster and as such fail in meeting their long-term DevOps goals.

DevOps however has a huge potential to save costs and bring products to market faster. Those companies which are successful are reaping the benefits, but also, they had to sweat a lot to get there. So, start sweating and have fun. We will be glad to have a conversation to see how we can help you with our experience to reach your goals. Do not hesitate to challenge these thoughts. You can contact us at info@evocean.com