

MIX AND MATCH – HOW TO BUILD SOLUTION TEAMS

The increasing digitalization turns software engineering processes upside down, as customers no longer order a readymade piece of software, but rather the continuous digitalization of their value chain. AdNovum unites developers and operators under one roof, flexibly putting together solution teams to adapt to the changing demands.

Interview with Christian Siffert and Alexander Nolting.

Looking at your order book, how have customers' needs and wants changed?

CS: They have changed fundamentally. In the past, customers ordered software that was delivered on CD or per download, and then they would run it themselves. Nowadays, customers order a full service. The value chain has been extended, the release cycle shortened. While we used to deliver a new software release twice a year, we now do so weekly.

AN: The value chain also starts earlier. The digitalization begins even before we meet the customer for the first time. Certain digital processes are expected to be in place already, so that customer and AdNovum are able to communicate via platforms.

Does this mean that, nowadays, the customers already send their requirement specifications in digital form, for example via collaboration tools?

AN: Our customers always submit their requirements digitally, but not necessarily via collaboration tools. We often still have the good old Word document. And depending on the order, the Word doc might even prove more efficient, since collaboration is not yet standardized beyond one's own company walls, and also because businesses use collaboration tools differently.

CS: In the past, a comprehensive document listing all requirements would form the starting point for our efforts. Of course we would keep exchanging thoughts with the customer in order to adjust requirements and to clarify the finer details. But it would take a long time for the customer to receive the result in form of an actually running piece of software. Today, all of this happens much faster. We are able to deliver individual features on the fly as the project progresses.

The subprocesses are much more interlocked; we produce "just in time", as they say in the industry. Many of the changes industry went through during the postwar period are the ones IT is going

through in our current times. We used to be an artisanal manufactory, delivering a new release every now and then. Now we are designing complete production lines together with our customers, continually delivering features in small batches.

So there's no longer a start and an end to a project?

CS: A project has, by definition, a start and an end. One puts together a team and at the end dissolves it again. Nowadays, we no longer think in projects, but in products or solutions. Those do not have an end.

WE DESIGN COMPLETE PRODUCTION LINES WITH OUR CUSTOMERS.

What exactly do we sell, then?

CS: Our customers no longer order a software project, but the digitalization of their value chain. They want a continuous solution.

AN: What's more, the solution is frequently no longer focused on the goal that was initially set, as it can already have changed by the end of the day. Customers expect us to be able to quickly adapt to changes. We have to be able to get into their heads and their work environment and think the way they do, in order to find out where they want to lead their business and whether the solution may affect other aspects of their business model as well.

CS: Our customers are aware of the short cycles of consumer devices: See it, order it, receive it – all in less than a week. So they think: Why shouldn't this also be possible for a digitalized business process?



Alexander Nolting and Christian Siffert: working on the coordination of operations and development.

Speed has increased tremendously, we are constantly improving the software. What does this mean for the collaboration between development and operations?

AN: Developers have to be able to develop on a system that is close to production, tailored to the specific requirements of the customer, without distinguishing between development and operations, a system which has been tested and which provides the same support and monitoring processes. Ultimately, the software has to be functioning not just on the laptop of the developer, but also when deployed in the customer's environment.

CS: There's a saying that goes: "The developer just throws the software over the fence." In earlier times, this used to be a manual and explicit handover of both the software and the installation documents. A member of the operations team had to install the software on the production system and had to ensure that everything was working fine. A time-consuming task, but nonetheless acceptable, since it only happened every few months. Having to continuously do the same thing on the other hand is a real pain. The deployment process must therefore be fully automated. This results in a shift in the value chain. In the past, the two work steps were done sequentially, now they run in parallel. Operations sets up and maintains the deployment process as a service, used by development.

In other words: Operations and development now run concurrently, and there is no handing over of the baton?

CS: Exactly. I recently used an image for a blog post: For a new software release, a truck is loaded and sent onto the road.

There is a pace car on the road which you need to follow to deploy the software. This model is far too slow. Operations needs to provide a smooth highway with guardrails. But once the highway is in place, the solution team must be able to drive on it on their own. This requires major adjustments on both sides.

AN: A lot depends on the know-how of the individual. For example, a developer who writes great algorithms, but pays little attention to the infrastructure in which the solution needs to operate, is going to fail. For this reason, it's imperative for both developers and operators to be able to look beyond their own noses.

CS: Yes, absolutely, that's a must these days. The team has to keep an eye on the customer value as a whole, including operations. This means having mixed teams with T-shaped skill sets, and, as mentioned earlier, enhanced support by underlying self-service platforms.

What exactly does the operations team provide?

AN: Operations ensures the stability of both process and solution. It is not necessarily responsible for the process, but knows what can be operated with the physical and personal resources available.

CS: Operations knows the ins and outs when it comes to implementing processes. This know-how has to flow back into the development projects. So nowadays, I think we should talk about solution rather than development teams, as they have both development and operational responsibility.

AN: However, it has to be noted that development and opera-



tional teams often still act separately today and are both focused on their specific tasks. Interdisciplinary working and thinking is not yet the norm.

CS: Naturally, this has massive organizational consequences. Before, development teams reported to the head of development and operational teams to the head of operations. Now, in the wake of the new methodology, new solution teams are suddenly being created that are responsible for both areas. The slogan behind it is: "You build it – you run it."

Let's take a look at it from the customer's point of view: A customer comes to us with a certain requirement. What he wants from us is not software, but a certain service or "value". Whether the business is about leasing cars or transporting cement, he wants his business organized. Which teams within AdNovum are involved in delivering the service does not matter.

Does this mean integration of development and IT operations from a single source is almost imperative?

AN: For a business that offers off-the-shelf solutions, it is not relevant whether development and operations are done by one and the same or two different companies. AdNovum, however, builds custom-made solutions. If we want to offer our customers a complete range of solutions, from consulting to concept development right down to operation and optimization, it is crucial for us to integrate operational processes as part of the development process.

CS: Integrating the two areas is also a question of efficiency – we are not the only software developers on the market. Maybe one day, we will not even call ourselves a software engineering company anymore, but rather an information automation company. Even though software engineering remains a core part of our offering, our performance rating depends mainly on how smoothly the software runs in production and on how long it takes for us to deliver.

AN: The customer's expectations regarding aspects such as reliability and speed are also of great importance. For public-sector companies, long service life, stability as well as low downtime and error rates are primary requirements. By contrast, a company that depends on being able to respond dynamically may be more tolerant regarding operational reliability, if less reliability means increased speed.

How can such flexibility be achieved?

AN: We have reached a maturity level in IT operations that five years ago seemed unthinkable. IT tools are being created with such speed that it's almost impossible to keep track of them all. And it is even more difficult to decide which tool is the right one. Here we need to standardize yet again, or we are going to get bogged down by too many options.

CS: In today's IT, the need for flexibility is pushing standardization through containerization. In the shipping industry containers were introduced in the 1960s. If shipping a ton of goods used to cost 6 dollars in 1956, the same load would merely fetch 20 cents after containerization. Flexibility rises slightly thanks to shorter throughput times, and costs are drastically reduced.

Transferred to software this means ...

CS: Standardizing components releases funds that enable higher flexibility tied in with better quality. If one looks, for example, at the higher expenditure for recurring manual tasks in development, delivery, integration, and costs for putting into operation, the potential for improvement becomes obvious.

AN: I have to be able to answer several other questions, such as: How do I integrate with the customer's infrastructure? Does he already have a foot in our IT door, so that I can simply send the container over? Or do I need to provide the container by different means? In other words, from an operational point of view, containers are only the first step, just a different way of delivering software. If there is no direct integration with the customer,



we have tools that move things into the public cloud, which is basically just a supersized infrastructure operated by someone. If the customer does not permit the use of a public cloud, AdNovum may act as an interface and operate the customer solution within its own private cloud. Or, we develop in the traditional sense and operate a whole business operation stack via a virtual machine. That, too, would be a form of handover. Containerization alone is not the solution. However, it opens up new possibilities and thereby enables us to address some of the challenges we are facing much more efficiently.

**NOWADAYS, SERVERS
CAN ONLY BE
MAINTAINED IN
AN AUTOMATED WAY.**

[Let's come back to the importance of containerization for the collaboration between development and IT operations.](#)

AN: For the developer, the container paradigm brings considerable changes. He has to formally describe operational processes, and thereby sort of becomes responsible for the operation of his own software. The operator, on the other hand, is progressively morphing into a (performance) analyst, optimizing his infrastructure with regard to stability and performance.

CS: To reiterate our earlier point, infrastructure today is increasingly offered as a service, with the individual order being handled automatically instead of individually by operations staff. This makes both suppliers and users more self-sufficient. What's more, it frees up time – hopefully – for operations staff and developers to actually collaborate, i.e. jointly work out measures to improve availability, reliability, monitoring and concrete measures to prevent incidents, rather than just meeting each other sporadically.

[It was long believed that separating development and operations would make processes more efficient and secure. Does this no longer apply?](#)

CS: This principle, also referred to as “separation of concerns”, is often ascribed to the Information Technology Infrastructure Library (ITIL). But actually, you will not find it in ITIL. In view of today's conditions and requirements, I think that a separation is no longer the most efficient way of providing IT services.

AN: ITIL supports by way of “blueprinting” – that is, it specifies how IT services can be set up to achieve good results. However, the separation doesn't need to be too stringent.

CS: In many cases, this separation no longer makes sense. Interdisciplinary teams that combine all the skills are better suited to coping with today's challenges. Developers know how to code, operations staff knows how to operate a system. For example: In the past, there were no more than a dozen servers deployed in production. Today, there may be up to a hundred – or more. In the old days, individual servers were treated like pets, while nowadays server maintenance can only be handled in an automated way. As a result, there is a need for skills to be transferred between development and operations – in both directions! Developers, for instance, have always used versioning. Today, infrastructure components also need to be versioned. As a matter of fact, we are increasingly pressured to version all available resources, from firewall to storage, to be able to restore them quickly and reliably if required.

[How does this impact on the work of developers and operations staff?](#)

AN: The professional skill sets will change. Personally, I expect system administrators and operations staff to have some developer skills. They have to be able to understand how developers work in order to solve problems efficiently. At the same time, developers can no longer just focus on the coding, but must also spare a thought on how the solution will perform in practice



afterwards. A very good example for this is the handling of databases. In the past, a developer would order a database and its setup from the database team. Now that databases are operated within the container, developers often have to set up the databases themselves, hence they must also possess some DBA know-how.

CS: Operations is no longer alone at the end of the value chain and sets up databases manually on request. More and more work is done in project mode. And the vast amount of hardware, from servers to firewalls and load balancers, is described in code or configuration files to enable automated maintenance.

AN: I really like the way you phrased this: "infrastructure as code". We have to describe infrastructure in such a formalized way that it is possible to deduce code from it. Which formal language we use in the process is irrelevant.

That sounds challenging ...

AN: It all depends on how we deal with errors and downtime. It goes without saying that mistakes happen. One aspect in the container model says: "If I fail, just start another copy of me. The copy will take up work at the exact point where I failed." This means, we live safely in the knowledge that mistakes do happen, they have been taken care of and are integrated in the plan.

CS: That's a giant shift. Operations is currently optimized with the expectation that no mistakes occur. But we know that in IT, as well as in globalization, everything is interconnected. Due to the growing interdependency, incidents happen more frequently – and are occasionally caused by services that you didn't even realize had an impact on your solution. As a result, we have to increasingly optimize toward being back in operation as soon as possible. Development and operations have to join forces and set up solution teams that can ensure stability.

How is this done at AdNovum?

AN: We invest in a culture that accepts that mistakes happen, and that seeks ways to get the software up and running again quickly. This necessitates a radical change in thinking. The traditional approach in software development was based on reducing complexity to get a handle on the process – that no longer works in this day and age.

CS: This aspect also has organizational consequences, as development and operations are separate. However, as we operate in a matrix organization, we do not have to turn the organization upside down, but can decide on a case-by-case basis whether it makes sense to deploy a mixed solutions team for a specific project or not. This way we can gather new experiences and find out which is the best solution for us and our customers. ■

Christian Siffert

Christian Siffert, MSc ETH in Computer Science, joined AdNovum in 1999 and has been in charge of Platform Infrastructure Engineering since spring 2016. He gained experience on numerous projects as developer and technical project manager. Since the advent of lean production in IT, he is happy to integrate ideas from his minor subject logistics, or attends lean kanban conferences. In his spare time, he clears his head jogging in the Jura mountains or by spending quality time with his family.

Alexander Nolting

Alexander Nolting, Computer Specialist and Project Manager, joined AdNovum in 2016 as Head of System Integration. He has almost 20 years of experience in managing high-availability solutions for the telecommunications industry. When he is not working, he enjoys riding his motorbike.