Case Study - Self Service Migration to the Cloud

How Eurotours manages to migrate their legacy landscape to the cloud with Codesphere - without any DevOps experts.

codesphere

Introduction

Today we are talking with Patrick Kürzer, enterprise architect and head of the software development department at Eurotours International about how they migrate a 40 year old landscape of on-prem business applications to the cloud and how Codesphere turns this into a developer self-service exercise. We will also discuss what the biggest challenges are and how new technology innovations can help address these challenges.

Simon:
Hey Patrick, thanks for joining us. Large migrations of legacy applications like the one you're currently tackling sound like a giant challenge. Can you provide some background like where you are starting from, which types of applications need to be migrated and where the journey is headed to?

Patrick:
Sure! Eurotours International was founded in 1980 and has had a strong software development track ever since. As a multi-channel-retailer for travel, we have secured a leading position in Germany, Switzerland, Austria and neighboring countries. In over 40 years of software development you can build up quite a portfolio of applications, additionally the travel industry as whole still employs a wide array of legacy applications. It is therefore fair to say we have quite a collection of our own legacy software and many third party dependencies are also based on older technology.
This creates many challenges, these systems run on-prem, require extensive administration and expensive licenses and most importantly lots of expertise from specialized experts. This expertise is increasingly rare and extremely hard to hire for or train. Outsourcing is not even something you can consider in these cases. On top of that we noticed that making changes to these legacy applications is simply a nightmare, making it increasingly hard to keep up with the speed of business requirements in today's day and age.
Therefore we decided a couple of years ago to replace the entire system. Parts of the existing self-built system will move to a third party software vendor in a managed standardized product. Surrounding this new core product we need a wide array of additional business applications in order to meet all our business requirements and remain compatible with all third party systems. These business applications should no longer run on-prem but move to the cloud.
The entire project is scheduled to take 3-5 years and touches all areas of the company group. We are working with around 15 full time developers to make this happen, making sure our applications comprising everything from Gupta, PLSQL, Oracle, Java, Weblogic and everything in between are cloud-ready and migrated.

As for the targets of this project, among other things we expect increased agility. It's much easier to change components of modular microservice architecture compared to a big monolith. Another advantage is that we are able to solve our operational bottlenecks, thanks to Codesphere we can turn the entire orchestration into a self-service workflow for our developers. With this distributed responsibility we are set up much better when team members are on vacation or on sick leave and thanks to the latest technical infrastructure innovations we are set up scalable and resource efficient out of the box.

Simon:
What are the biggest challenges, how does one plan such a project and what are the most important questions to be answered beforehand?

Patrick: The biggest challenge, I think, comes from the sheer size of the transformation. There is no business area that is not affected by the changes. You don't change habits and processes that grew over the span of 40 years in days. The goal has to be to use this fresh energy to separate things that work well from those that do not and establish lean processes going forward without leaving employees behind.
Technically the main challenge is aligning the multitude of dependencies between components, business processes and customer requirements on a common timeline.

About Patrick Kürzer
Patrick Kürzer, has been with Eurotours International in Kitzbühel, Austria for over 10 years. As Department lead for the software development team and enterprise architect he is responsible for the migration of critical business applications.

Project progress in many different areas needs to align ultimately to make sure the lifecycle of old and new products is covered both for b2b and b2c clients. Otherwise the migration leads to problems damaging our customer relationships.

Questions that should be considered carefully before starting such a progress firstly include the thorough understanding of make-or-buy trade-offs, understanding considered solutions end2end while not wasting too much time on solutions ultimately not selected. Secondly the timeline of the project should be questioned critically. In retrospect a shift left with more thorough planning and a more permissive timeline would have helped us a lot.

Simon:
You also realized early on that lift & shift is often not an option. Which parts of software need to be changed before moving them to the cloud, does new technology like AI code generation help with this?

Patrick:
Applications need to be changed because the entire architecture is different, starting with the databases, additional security requirements for cloud operation like VPNs and networking rules all the way to third party systems that might not be compatible. Luckily such a migration also offers an opportunity, technical debt that accumulated can be eliminated and software patched to the latest versions.
Artificial intelligence does help when it comes to understanding snippets of old code, unfortunately we have the experience that it is not sophisticated enough to understand the full context of bigger legacy systems. Hopefully this changes in the future but until then we will do a lot of the migration manually.

Simon:
You chose to tackle this migration with Codesphere's PaaS infrastructure platform, what were the deciding factors? How does Codesphere differ from other providers?

Patrick:
As mentioned before for us it was essential to enable developers to develop and ship software end to end as a self-service. Bottlenecks in DevOps simply slow us down too much. The amazing thing about Codesphere is that our developers can provision all the infrastructure they need without month long training or Kubernetes know-how. The ease of use & flexibility of Codesphere is market leading, anyone can deploy their first application within minutes. At the same time Codesphere doesn't restrict us, we can deploy all types of applications we could ever need.
This creates real independence and accountability in our development teams which allows us to develop applications with modern tech stacks and ship them faster. Additionally we can leverage Codesphere's super fast starting off-when-unused environments to spin up countless development environments without driving up costs. Overall Codesphere was a perfect fit to eliminate the bottlenecks we had with the previous legacy system.

Simon:
One Aspect, I wanted to dive a bit deeper into is the self-service aspect. What does it mean for your business if developers can really independently take care of the infrastructure of their application without having to learn Kubernetes & Co.?

Patrick:
Building up know-how is always risky. You introduce new bottlenecks and dependencies because teaching all skills to everyone on the teams is hardly feasible.
The self-service aspect eliminates this risk. Our onboarding can now happen in real-time instead of months. Codesphere's templates offer a super fast way for anyone to get started. Even classical operations tasks like scaling productive apps up or down, setting up domains & routing are trivial and can be done by any team member. Our developers regain end 2 end ownership over the workflow, making them more flexible and self dependent. This is a big factor unlocking very positive team dynamics.
At the same time Codesphere introduces some standardization across all teams for example by simplifying the CI/CD definition which helps during debugging and keeps everything nicely organized.

Simon:
Awesome thanks! Do you have a final piece of advice for other teams facing similar challenges or projects?

Patrick:
Take three minutes to deploy your first application live on Codesphere! You won't regret trying it.