

A full-page background image showing two hikers in red shirts and dark shorts climbing a steep, rocky mountain ridge. One hiker is lower on the rock, reaching up to assist the other who is higher up. The background features a vast mountain range under a cloudy sky.

# Nobody is left behind.

Stephan Stapel  
Hermes Germany GmbH

- 01** Intro.
- 02** Technical patterns.
- 03** Developing teams.
- 04** Sharing Knowledge.
- 05** Take aways.



A wide-angle photograph of a large, modern industrial warehouse. The ceiling is high, with a complex network of white metal beams, blue structural supports, and red-painted pipes. Large blue curved structures, likely part of a conveyor system, are visible on the left and right. The floor is a smooth, light-colored concrete with two prominent yellow lines running down the center. In the background, more industrial equipment and structural elements are visible.

Hermes Europe is the largest Post-independent parcel company in Europe.

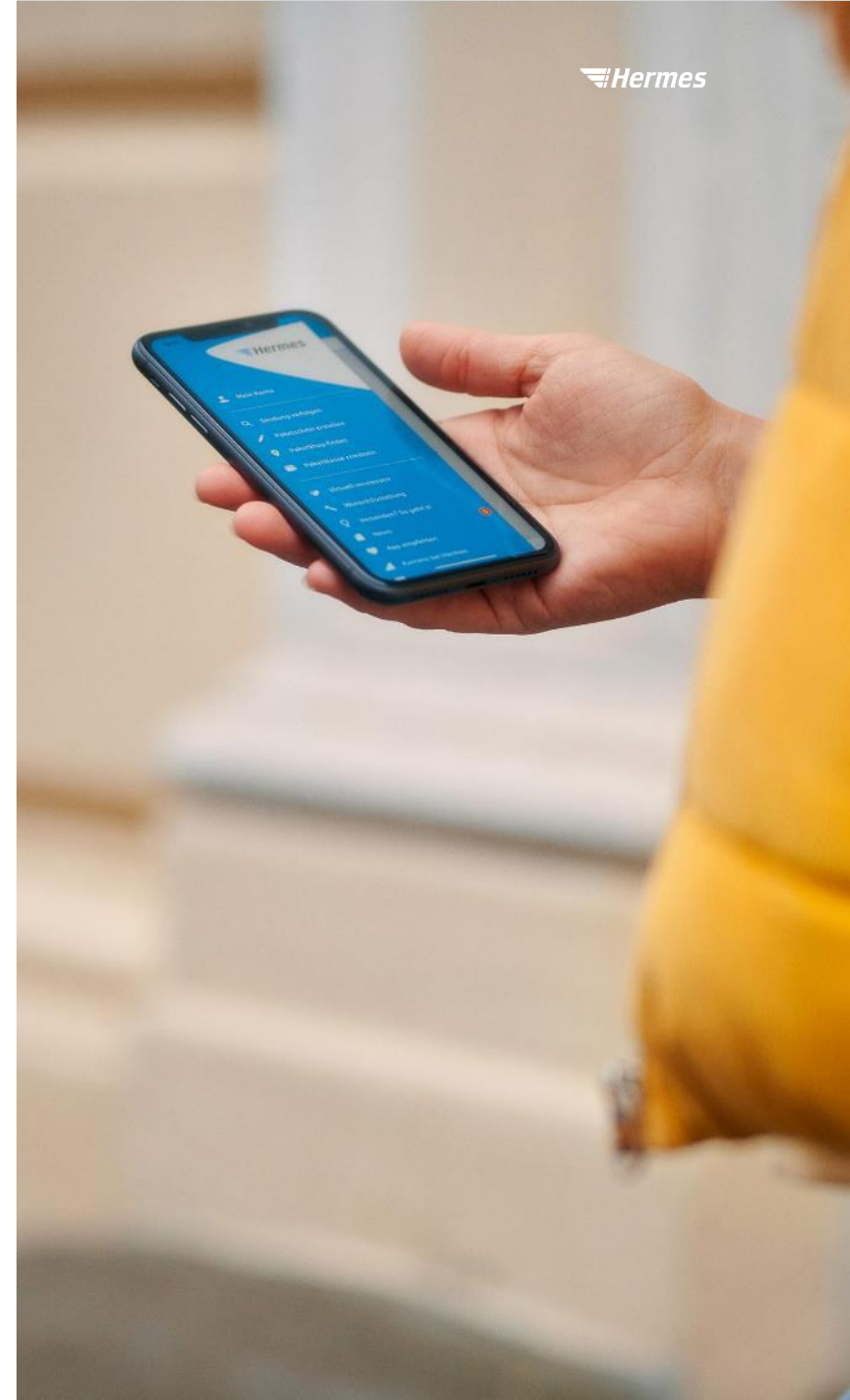
Hermes Germany takes care of parcel delivery in Germany to both doorstep and more than 16,000 Hermes parcel shops.

# Our approach.

DevOps as our foundation.

Continuous Delivery as our operating system.

Set as strategic goals in 2017.





# The results so far.

Tremendous increases in speed, quality, safety.  
Pervasive adoption of cloud technology.



We found that teams are  
running at **different  
speeds.**

We are convinced that  
two speed IT is a blocker  
for a healthy tech  
culture.

Complicates development of IT organization.

**We don't want any losers our the transformation. Yet, it  
takes dedication to keep all teams on track.**



I will use the terms  
**slow system/ slow-system  
team and fast system/  
fast-system team.**



We found **two main reasons** for slowness.

# #1

## Overly complex logic.

Too many domains addressed in the application.  
Dependencies that are hard to manage.

Test scope too large. Test scenario too hard to setup.

# #2

## Technical setup.

Technology that was not meant to deliver continuously.  
In our case: mainly JEE containers.  
One trick pony architecture, not paying attention to sustainability.

# No reason: **Team setup.**

We found no examples that team capacity was an issue.

Nor did we find any example where education or skills were an issue.



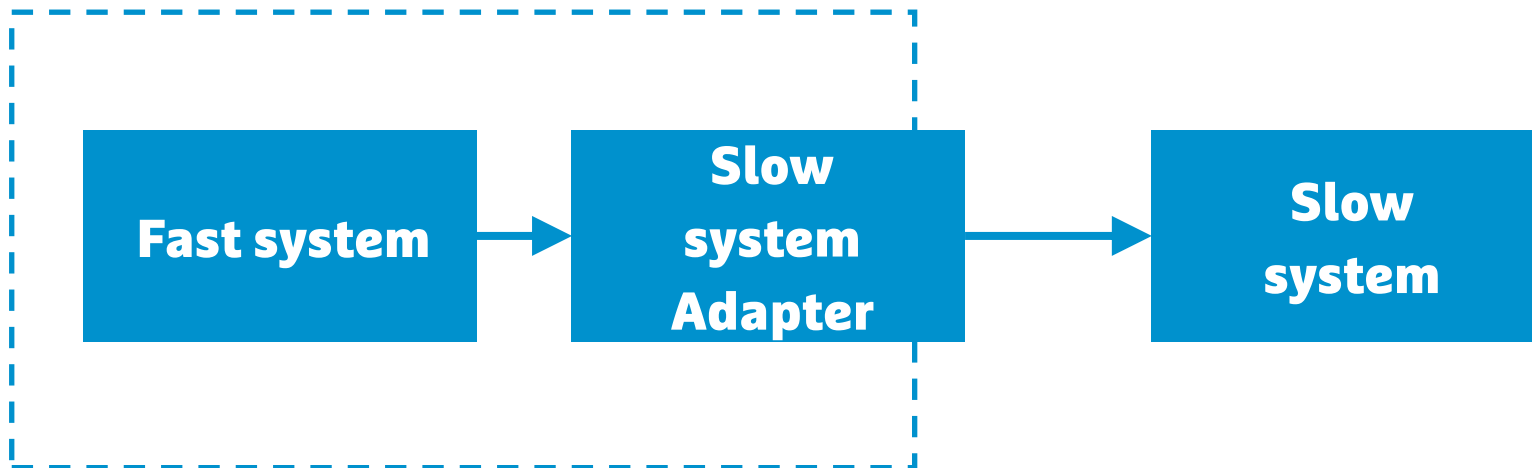
None of the reasons can be  
**repaired easily.**

# 02

We use technical patterns to ease cooperation between slow systems and fast systems.

# Pattern #1

## Isolate dependencies.



Lots of teams make dependencies explicit by implementing a dedicated adapter to the slow system.

This allows to isolate both changes to the own (fast) system and to the interface to the slow system.

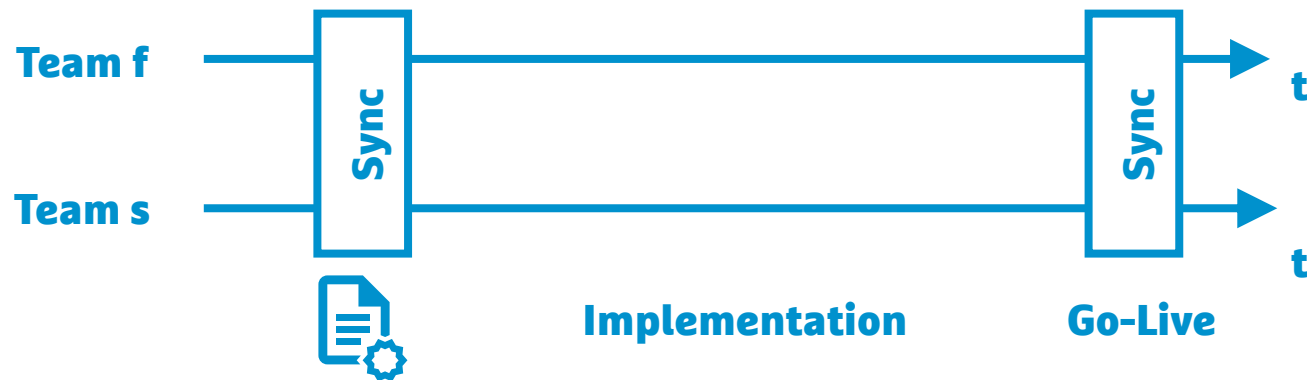


## Pattern #2

# Use contracts to decouple implementation.

Teams agree on formalized interface contract.

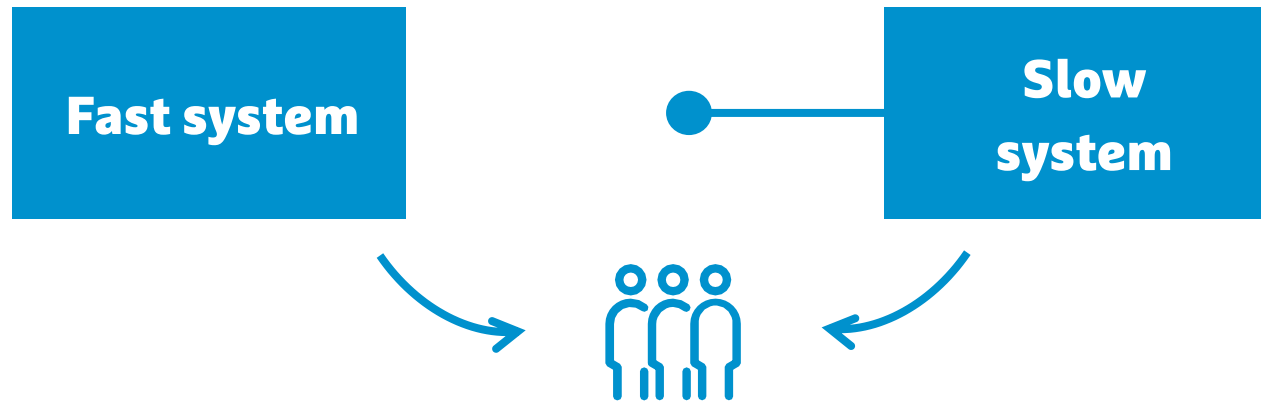
As a consequence, both teams can implement in their own speeds and test the interface in an isolated manner.



Coordination only necessary for interface contract and go-live.

## Pattern #3

### Joint development.



Teams join to implement the necessary changes to the slow system and/ or interface.

Variant: members of the fast system team implement the change, create a pull request for approval by the slow system team.

# 03

Patterns for organizational development.



**Don't give up** improving  
the slow system teams.

# Slow system teams are a big asset for an organization.

They usually have long-living team setups.

Experienced colleagues, experts in their business domains.

## Important:

Slowness shall not be used as a weapon.

Awareness of the own role in the landscape: innovations in other teams might get hindered due to dependencies to slow system team.

Pay attention to the atmosphere between teams.

# The new new.

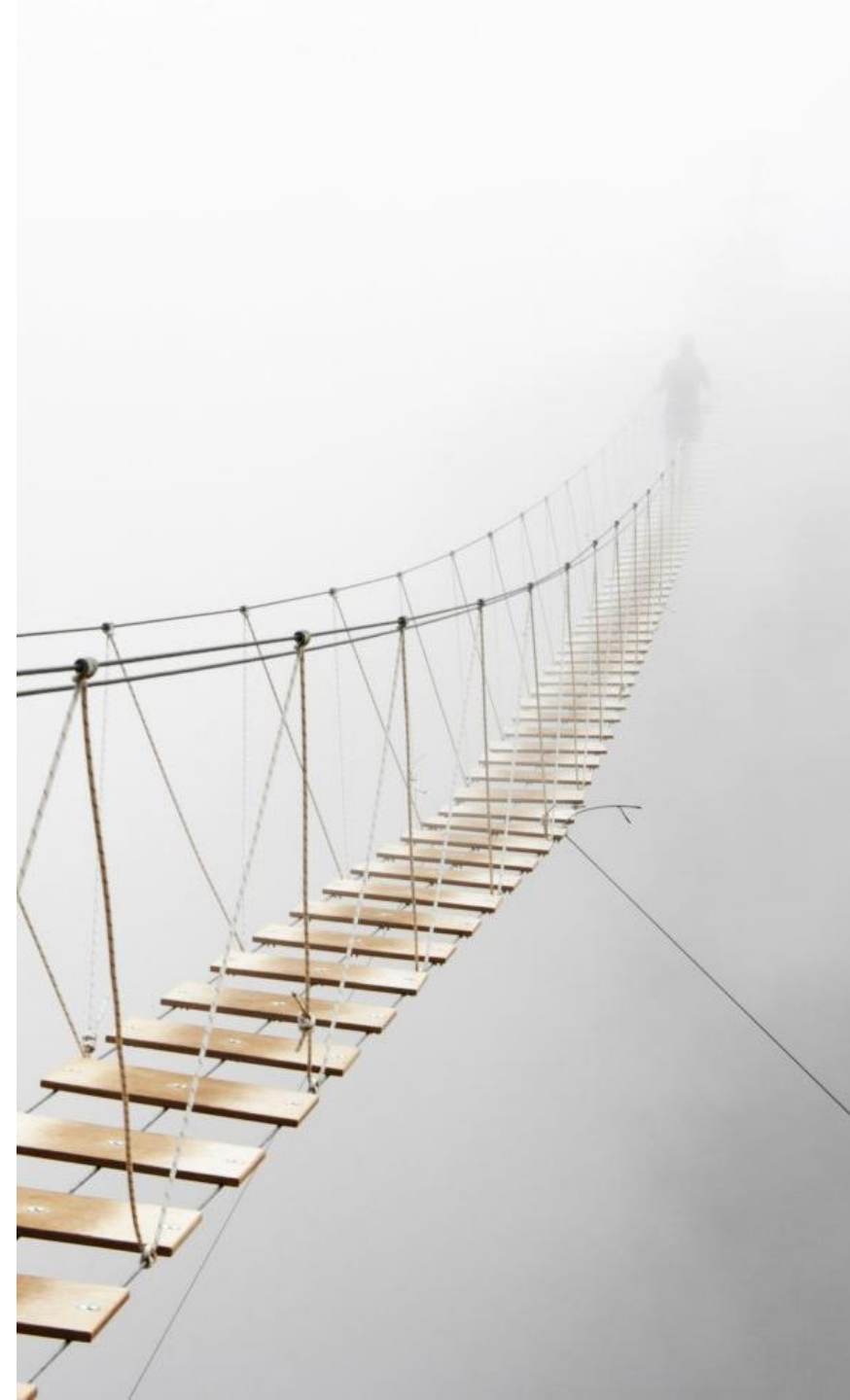
“Everyday Life Continuous Delivery” might still be mad magic for slow-system teams.

Working with of the slow system teams should not be **breaking changes**.

Instead, it should be **step-by-step** improvements.

## **Find your own beat:**

While real Continuous Delivery is hard to achieve, deploying on a cadence of a sprint is often possible.  
They feel to be part of overall organizational development.



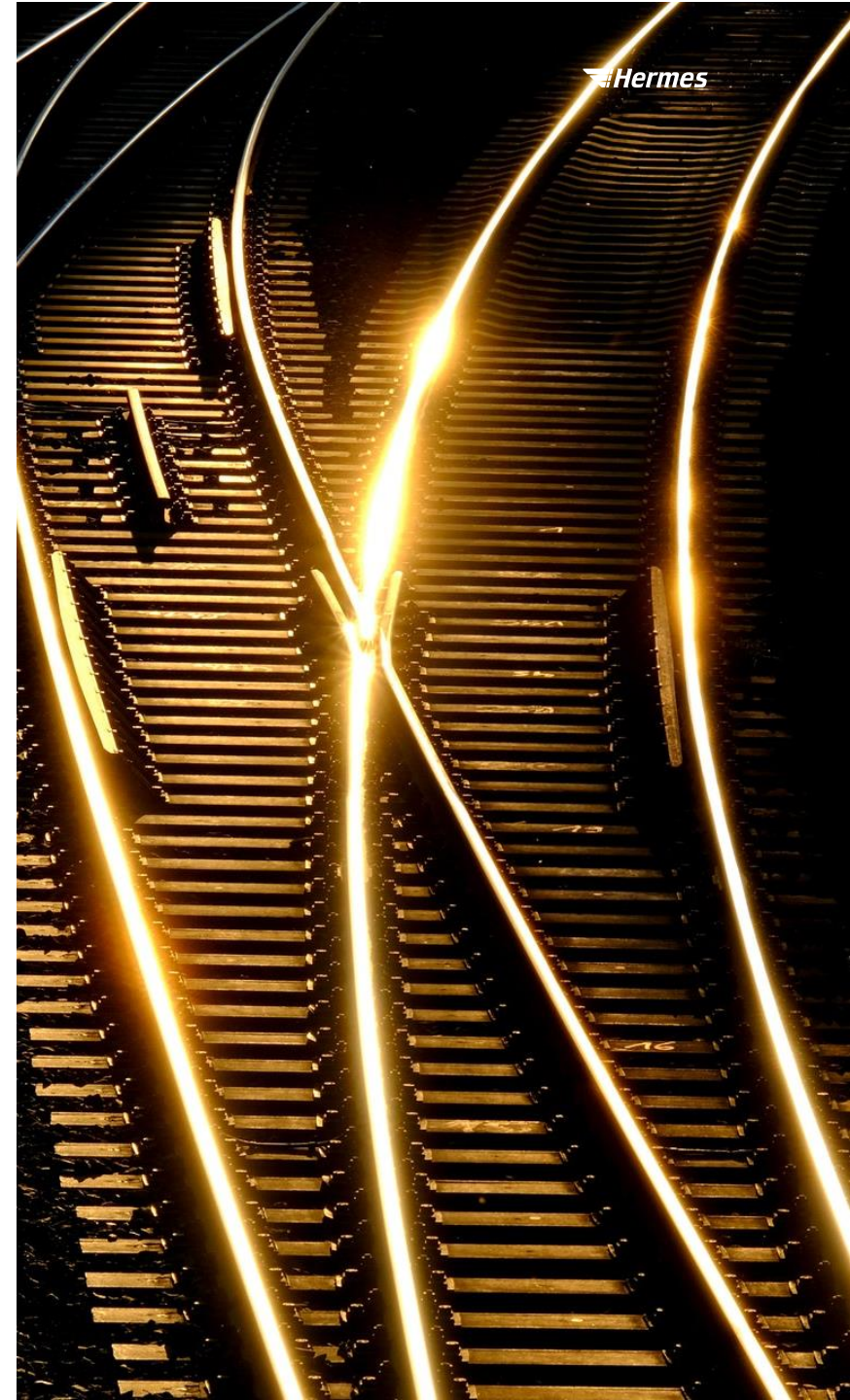
# Pattern #1

## Keep track.

Keep track of development of all teams, especially the slow system teams.

It is easy to hide (or get lost) in larger organizations, make sure to have a proper leadership/team ratio.

Bend-and-wait is no option. All-in means all in.



## Pattern #2

# Ensure sustainability.

The slow systems that these teams take care of are some of the more important applications in our estate.

Steady work on team, processes and eventually technology are necessary – and possible.

Promote the advantages of innovations. Help to translate and adopt these innovations.

### Example:

Good observability is a good investment, even if everything was ok without it for the last years.



# 04

Sharing knowledge.

# Sharing knowledge.

Sharing knowledge is no one-way street. We can learn a lot from the slow system teams!

IT-wide fairs to allow all teams to learn from one another. Both from a functional and a technological perspective.

Open reviews → well-invested time.

Communities of practise to discuss and develop .





# 05

Take aways.

# What is the business impact of our effort?

“We used to intensively train our call center agents with each release. Thus, we were reluctant to introduce new releases of our application on a cadence of 3 weeks. But we found that the changes rolling in every 3 weeks were quite small and easy to explain which allowed us to often skip those extra release trainings.”

– Christine, Trainer, Customer Service

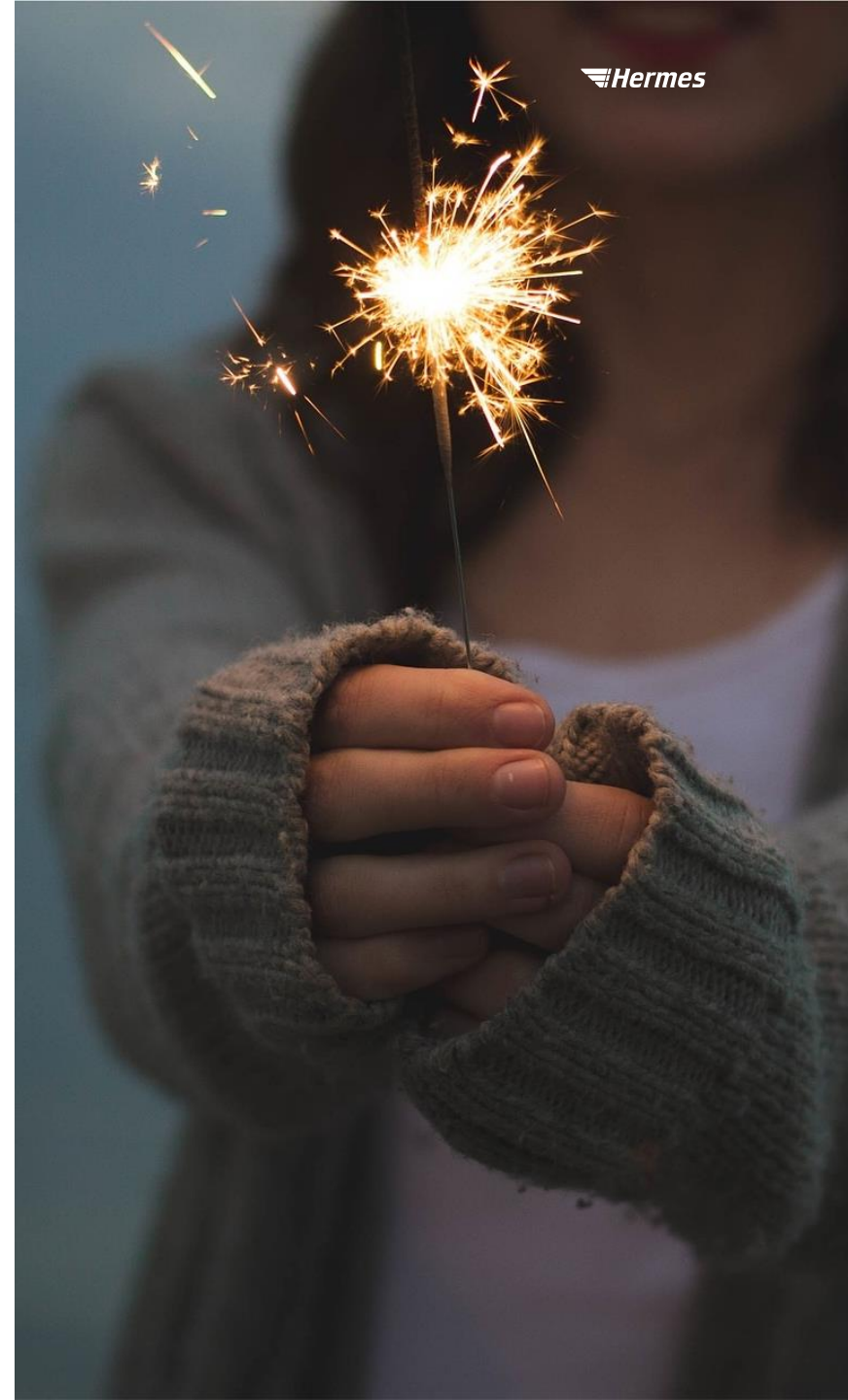


# Take aways.

There is no one size fits all approach in DevOps transformation.  
Different teams are running at different speeds.

It takes dedication for teams which can't absorb modern  
technology quickly and easily.

The effort is worth it for happy people and a healthy tech  
culture!





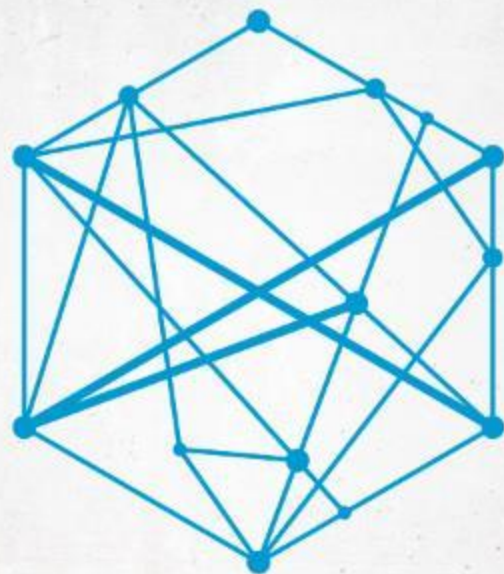
**Stephan Stapel**  
Head of Development

[stephan.stapel@hermesworld.com](mailto:stephan.stapel@hermesworld.com)

 [@StephanStapel](https://twitter.com/StephanStapel)

 <https://www.linkedin.com/in/stephan-stapel>





DELIVERING.  
**TECHNOLOGY.**

 ***Hermes***