



# Software architecture in a DevOps world

Bert Jan Schrijver  
bertjan@openvalue.eu



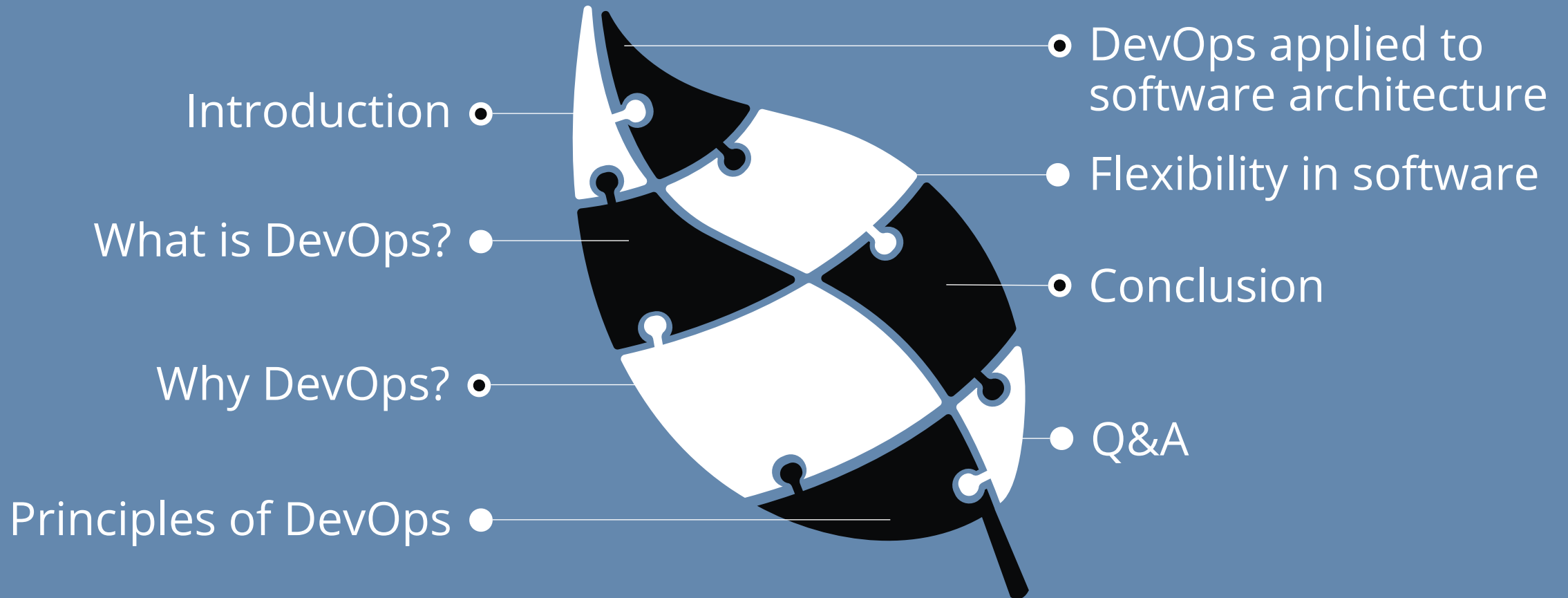
Let's meet  
Bert Jan Schrijver



# Software architecture in a DevOps world

## Outline

---



# What is software architecture?

- Sum of all decisions taken during development
- Making design decisions about important stuff
- Enabling a team to build software that provides solutions for business problems
- Allows you to build systems that are beyond the scale that you can hold in your head

# Definitions

## Who's who in CD & DevOps

### Continuous Integration

Team members integrate their work frequently. Commits are verified by automated builds and tests.

### Continuous Delivery

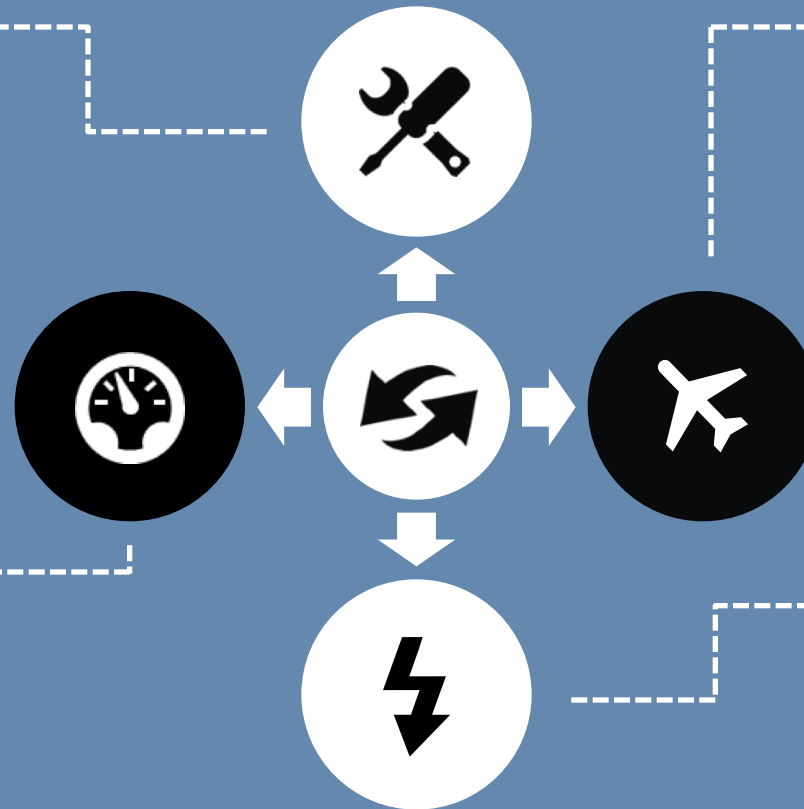
Building and testing software in such a way that the software can be released to production at any time.

### Continuous Deployment

Every change goes through the build/test pipeline and automatically gets put into production.

### DevOps


?!



*"Ship early, ship often, sacrificing features, never quality" - Kyle Neath*

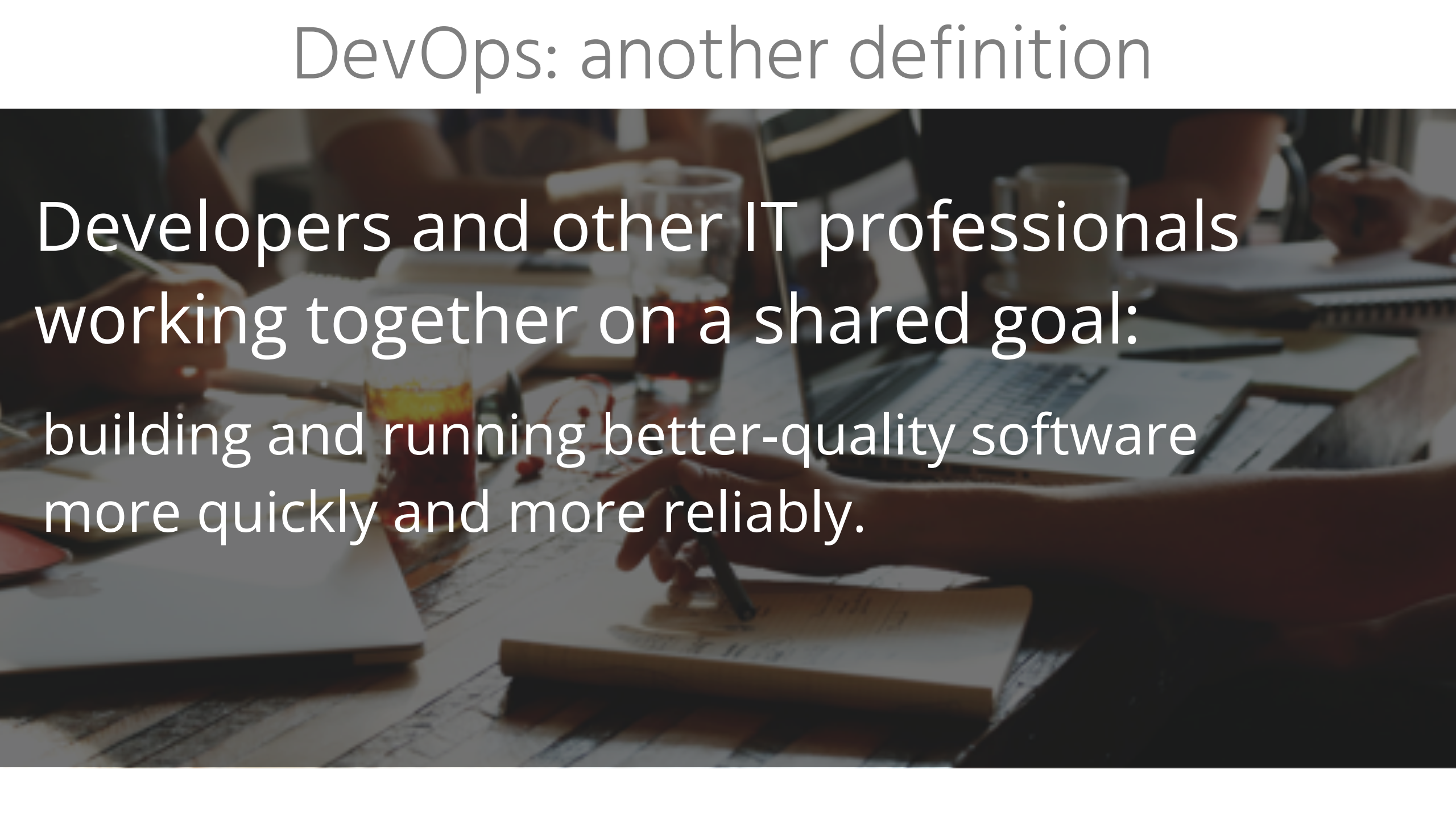
# DevOps: a definition

Development and operations engineers  
being responsible together  
for the entire lifecycle of a product.

The background image is a blurred photograph of a modern office or co-working space. It shows several people sitting at a long wooden table, engaged in work. There are laptops, notebooks, pens, and glasses of water on the table. The lighting is warm and the overall atmosphere is one of professional collaboration.



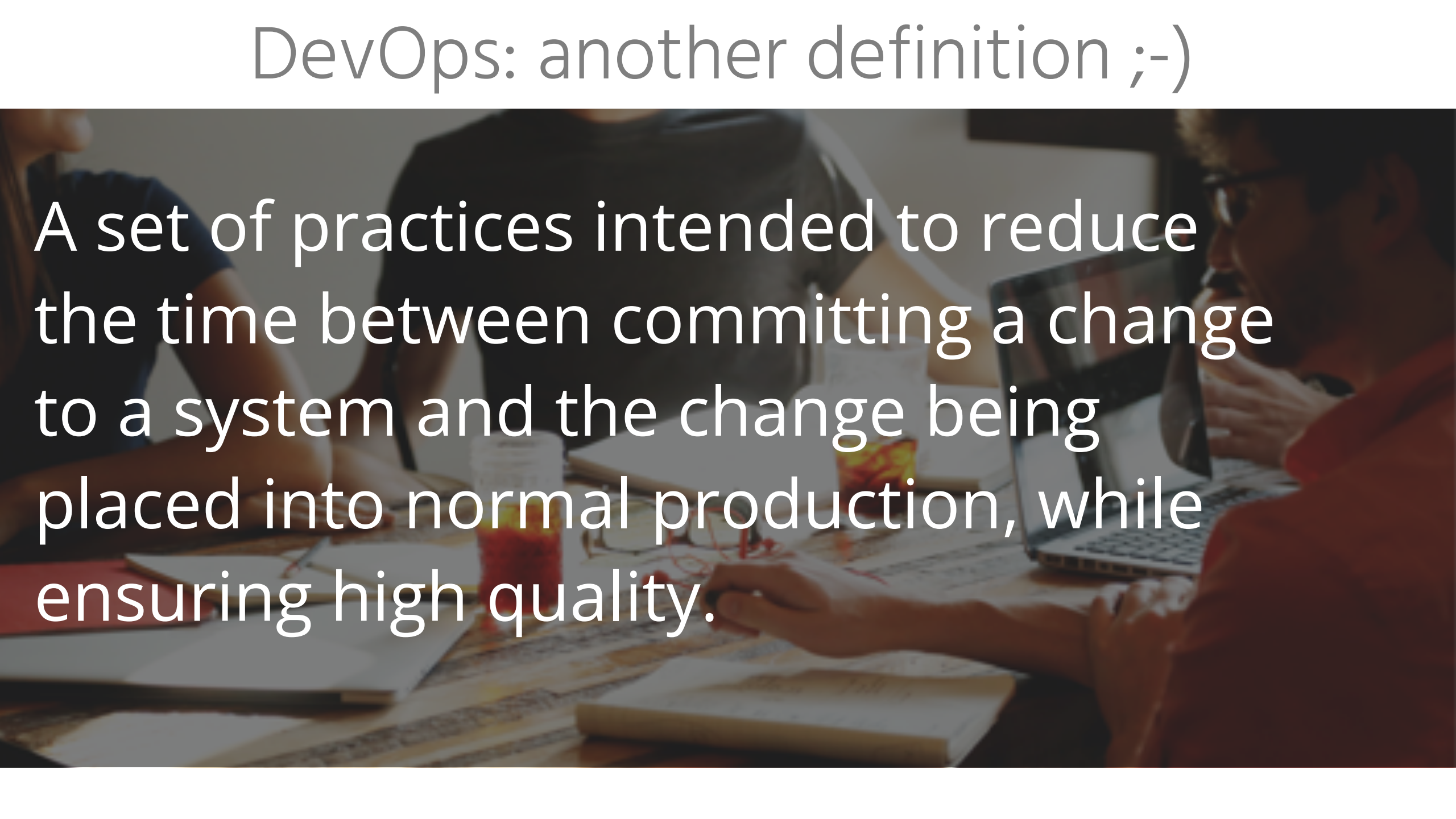
# DevOps: another definition



Developers and other IT professionals  
working together on a shared goal:  
building and running better-quality software  
more quickly and more reliably.

# DevOps: another definition ;-)

A set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality.

The background image is a blurred photograph of a modern office or collaborative workspace. Several people are visible, working at a large table. There are laptops, papers, and other office supplies on the table. The image is intentionally blurred to serve as a background for the text.



# Why DevOps?

DevOps makes your business move faster as an enabler for Continuous Delivery, faster problem resolution and focus on value instead of problems.



A photograph of a basketball game in progress. In the foreground, a player in a white Spurs jersey with the name 'BLAIR' and number '45' is crouching low to the floor, reaching out with his right arm. Behind him, three other Spurs players in white jerseys are visible: one with number '15', one with number '14', and one with number '4' who is looking upwards. The background shows a large arena filled with spectators. The text 'What is DevOps really about?' is overlaid in white, sans-serif font across the center of the image.

What is DevOps really about?




A photograph of two young children, a boy and a girl, in traditional Mexican clothing. The boy, on the left, has dark curly hair and is wearing a white shirt with a red bandana around his neck. The girl, on the right, has dark hair in pigtails with white and blue flower ornaments and is wearing a blue dress with a white lace collar and a colorful sash. They are both smiling and appear to be dancing or moving together. The background is a blurred crowd of people.

DevOps is about culture

A man in a red and blue plaid shirt is walking away from the camera on a sandy beach, carrying a young child on his back. They are walking towards the ocean under a clear sky. The text "DevOps is about freedom and responsibility" is overlaid in white on a semi-transparent dark band across the middle of the image.

DevOps is about freedom and responsibility





DevOps is about empathy

# Principles of DevOps



Gradual changes



Customer oriented



Automation



Ownership



Collaboration



Experimentation



Continuous improvement



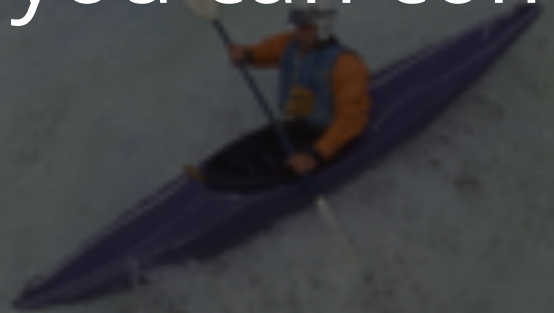
@bjschrijver

A collection of various hand tools is arranged on a dark wooden surface. The tools include a hammer with a wooden handle, several screwdrivers with different handle designs (some orange and grey, some black), a pair of pliers, a pair of needle-nose pliers, a set of hex keys (Allen keys), a set of sockets, a wrench, a small screwdriver, and a small metal tool. The tools are scattered across the frame, with some overlapping. The text "Software architecture and DevOps" is overlaid in the center in a white, sans-serif font.

# Software architecture and DevOps



# DevOps principle: gradual changes

- Applied to software architecture:
    - Start with the simplest solution you can come up with
    - Identify high level components
    - Improve, refine and evolve along the way
    - Focus on dealing with uncertainty
    - Do *just enough* architecture to get through the next sprint ;-)
- 
- A kayaker in a blue kayak is navigating white water rapids. The kayaker is wearing a blue jacket and a helmet, and is using a paddle to maneuver the kayak. The water is turbulent with white foam, and the background shows a rocky riverbank.



# DevOps principle: customer oriented

- Applied to software architecture:
  - Focus on the business' needs
  - Don't design for yourself!
  - Help out with clarifying non-functional requirements
  - Architecture is about tradeoffs

# DevOps principle: automation

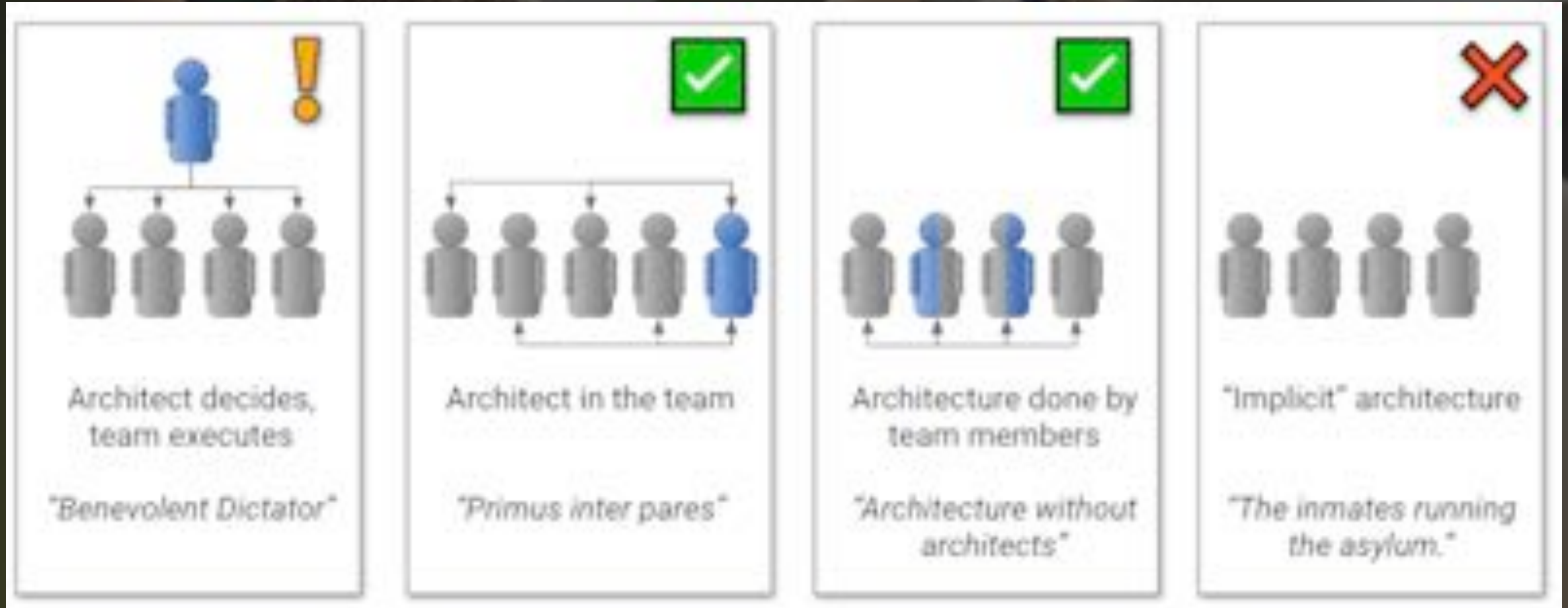
- Applied to software architecture:
  - Diagram generation
  - Automated architecture checks
  - Infrastructure as code, automated provisioning of cloud architecture, autoscaling

# DevOps principle: ownership

- Applied to software architecture:
  - Team owns the architecture
  - Architect needs to be involved with the team
  - Architect is accountable for the end result too!
  - Minimise dependencies to things outside the team



# DevOps principle: ownership



Source: <https://architectlevator.com/architecture/organizing-architecture>

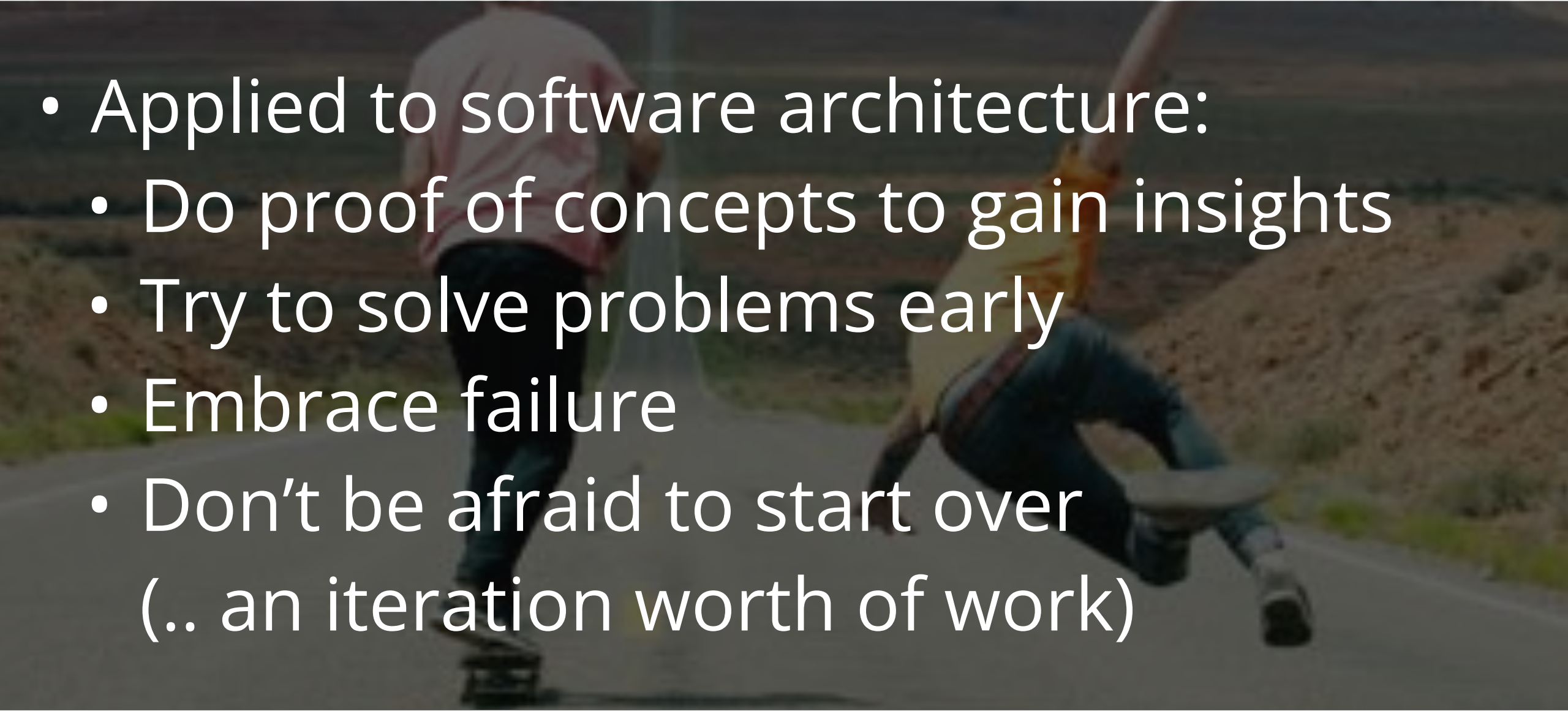


# DevOps principle: collaboration

- Applied to software architecture:
  - Talk to all stakeholders
  - Developers are stakeholders as well!
  - Be transparent, explain decisions, be open for feedback
  - *No room for ivory towers*

# DevOps principle: experimentation

- Applied to software architecture:
  - Do proof of concepts to gain insights
  - Try to solve problems early
  - Embrace failure
  - Don't be afraid to start over  
(.. an iteration worth of work)





# DevOps principle: experimentation

Hi Gergely, a question on your mention of using RFC's to get feedback on designs - I pitched this to our chief Architect yesterday and his take was that it sounds wildly inefficient sending to more than a small closed group. I'd like to know in practice how it worked for you?



08:45

Well, for one we did not have a Chief Architect who shot down ideas without trying.

08:48 ✓

Source: [https://www.linkedin.com/posts/gergelyorosz\\_architecture-software-design-bigtech-activity-6851431869691523072-2wbf/](https://www.linkedin.com/posts/gergelyorosz_architecture-software-design-bigtech-activity-6851431869691523072-2wbf/)

# DevOps principle: continuous improvement

- Applied to software architecture:
  - Focus on feedback loops
  - Ask developers for feedback!
  - Measure & monitor
  - Evolving architecture instead of up-front
  - Software architecture is never finished





A photograph of two skateboarders on a paved road in a desert landscape. The skateboarder in the foreground, wearing a pink shirt and dark pants, is riding away from the camera. The skateboarder in the background, wearing a yellow shirt and blue jeans, is in mid-air, having lost their board, which is on the ground. The text "Continuous Delivery & DevOps sound great, but it won't work here..." is overlaid in white.

“Continuous Delivery & DevOps  
sound great, but it won't work here...”

# “CD & DevOps won’t work here”

- 
- A photograph of a person in a yellow shirt and blue jeans falling off a skateboard on a paved road. Another person in a pink shirt and dark pants is standing nearby. The background is a dry, hilly landscape.
- Stated reasons:
    - we’re regulated
    - we’re not building websites
    - too much legacy
    - our people are too stupid
  - Actual reasons:
    - our culture stinks
    - our architecture stinks

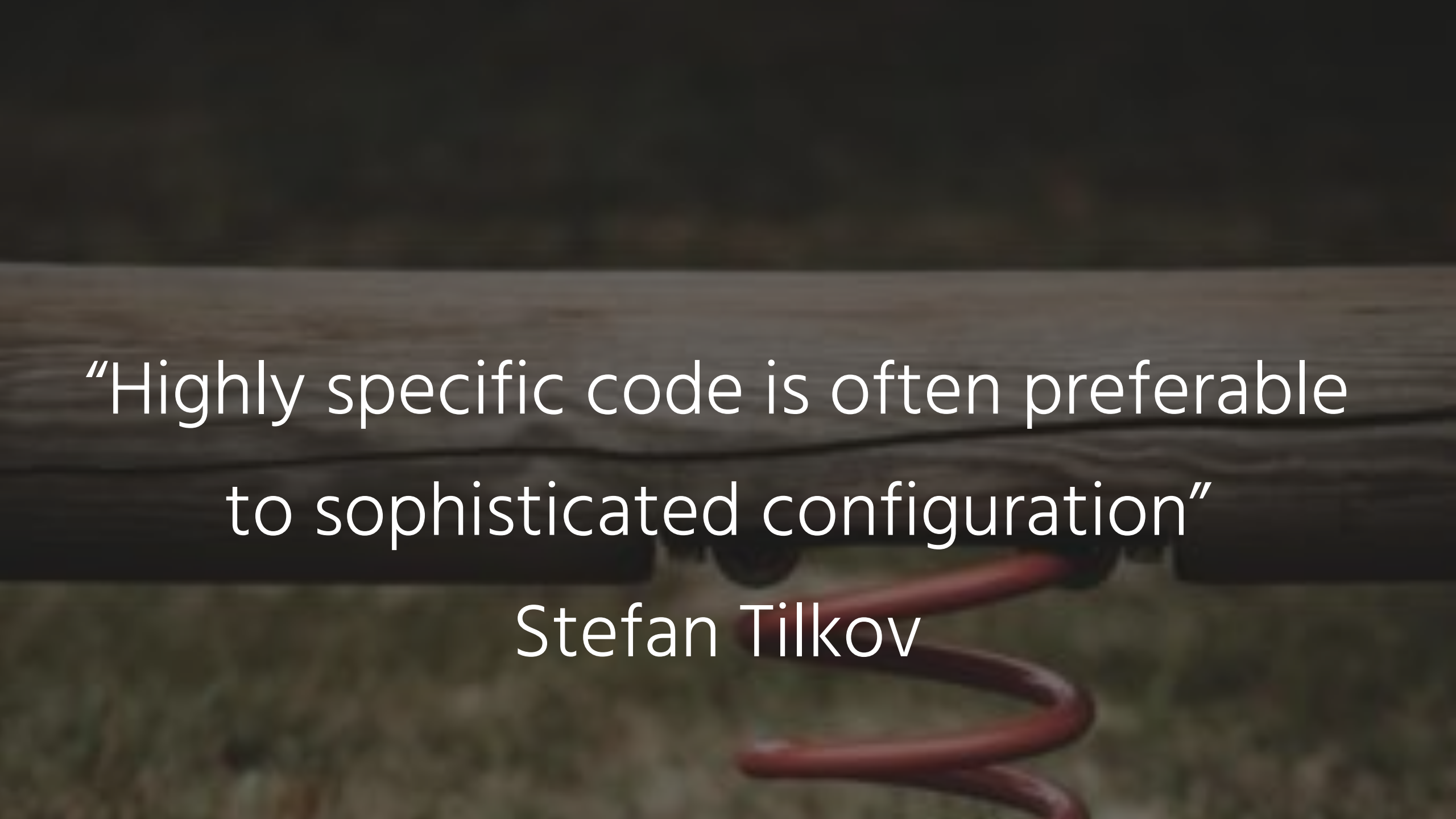
Source: talk by Jez Humble  
[https://www.youtube.com/watch?v=837Z\\_oehhRg](https://www.youtube.com/watch?v=837Z_oehhRg)



A horizontal wooden plank is shown against a dark, textured background. A red metal spring is coiled underneath the plank, extending from the center towards the right side. The text "About flexibility in software..." is written in white, sans-serif font across the middle of the plank.

About flexibility in software...





“Highly specific code is often preferable  
to sophisticated configuration”

Stefan Tilkov

success Ln


Summary

Failure Dr

The background features a dark blue gradient with two large, semi-transparent, light blue rectangular shapes. The top shape contains the word "Success" in a large, bold, sans-serif font, followed by "Ln" in a smaller font. The bottom shape contains the word "Failure" in a large, bold, sans-serif font, followed by "Dr" in a smaller font. The text is white and centered.

How do you work as an architect  
with(in) a DevOps team?

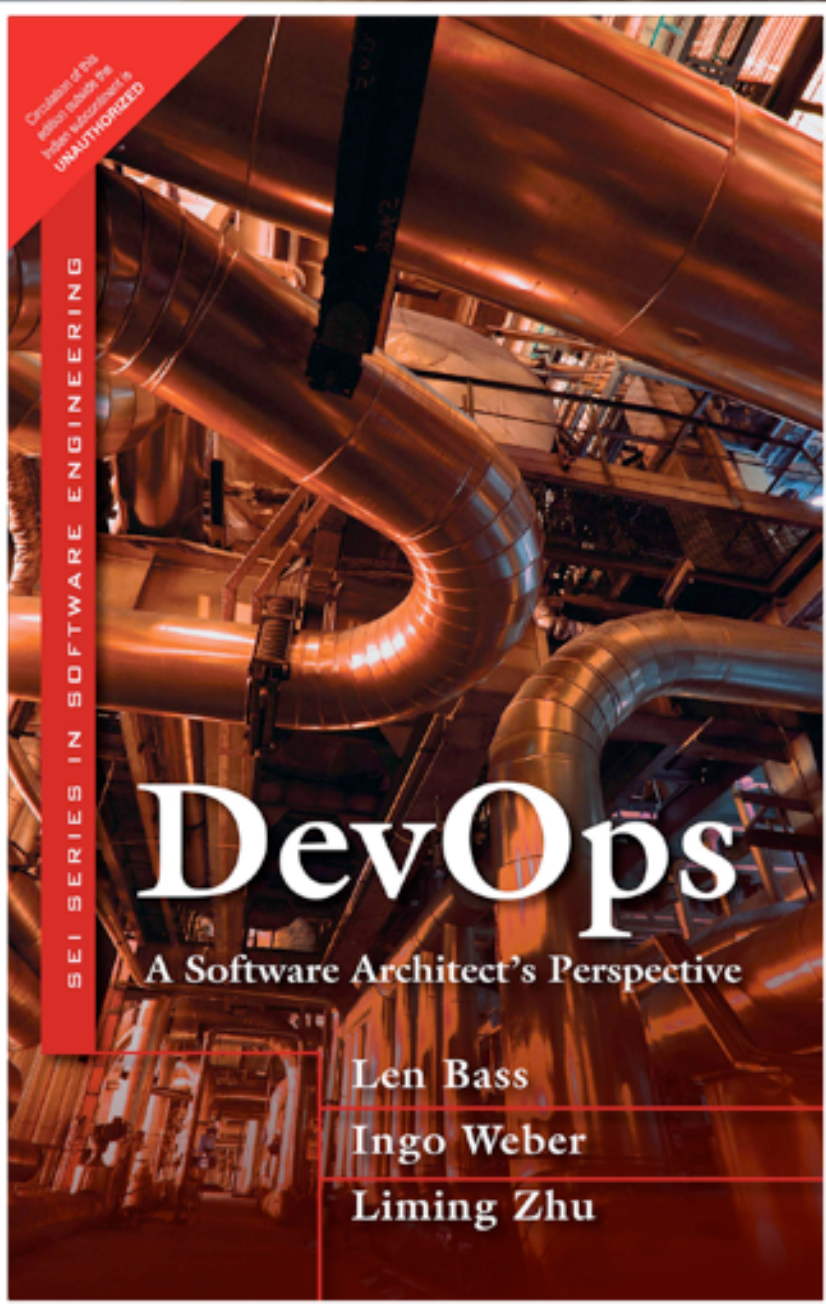


The background features a dark blue gradient with two large, semi-transparent, light blue rectangular shapes. The top shape contains the word "Success" in a large, bold, sans-serif font, followed by "Ln" in a smaller font. The bottom shape contains the word "Failure" in a large, bold, sans-serif font, followed by "Dr" in a smaller font. The text is white and centered horizontally.

By applying the same DevOps principles  
to the domain of software architecture.



**THAT'S IT.  
NOW GO KICK SOME ASS!**



**THAT'S IT.**  
**NOW GO KICK SOME ASS!**

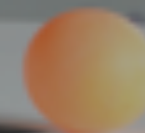




Questions?

Thanks for your time.

*Got feedback? Tweet it!*



All pictures belong  
to their respective  
authors



@bjschrijver