

The background features Dora the Explorer and Boots the monkey climbing a green vine in a jungle setting. Dora is in the foreground, smiling and holding onto the vine with one hand and a purple backpack with a yellow arrow on it with the other. Boots is behind her, also holding onto the vine and looking happy. The title "DORA the EXPLORER" is overlaid on the bottom left in a colorful, stylized font.

# DORA the EXPLORER

of team potential, discovering the Path to High-Performing Teams

# Jacob Duijzer

- IT Consultant @ Team Rockstars IT
- Engineering Manager @ BIMcollab
- Team Topologies Advocate
- Sooner Safer Happier Advocate



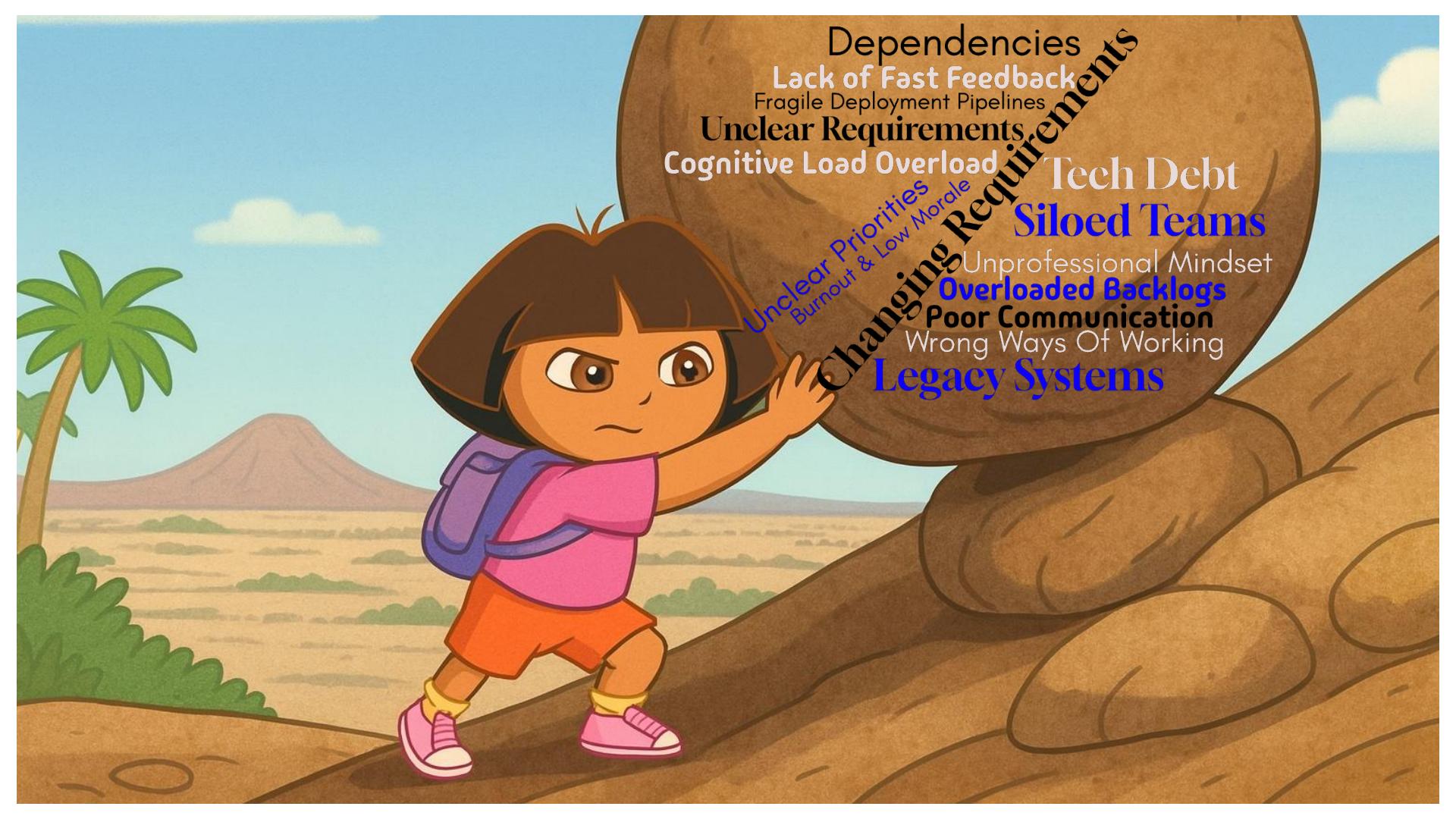
<https://www.linkedin.com/in/jacobduijzer/>



<https://github.com/jacobduijzer>





A cartoon illustration of Dora the Explorer, a young girl with brown hair and a pink backpack, pushing a large, textured log across a dirt path. She is looking back over her shoulder with a determined expression. The background shows a tropical landscape with palm trees and distant hills under a blue sky with white clouds.

Dependencies  
Lack of Fast Feedback  
Fragile Deployment Pipelines  
Unclear Requirements  
Cognitive Load Overload  
Tech Debt  
Siloed Teams  
Unprofessional Mindset  
**Overloaded Backlogs**  
Poor Communication  
Wrong Ways Of Working  
**Legacy Systems**

**Changing Requirements**

Unclear Priorities  
Burnout & Low Morale

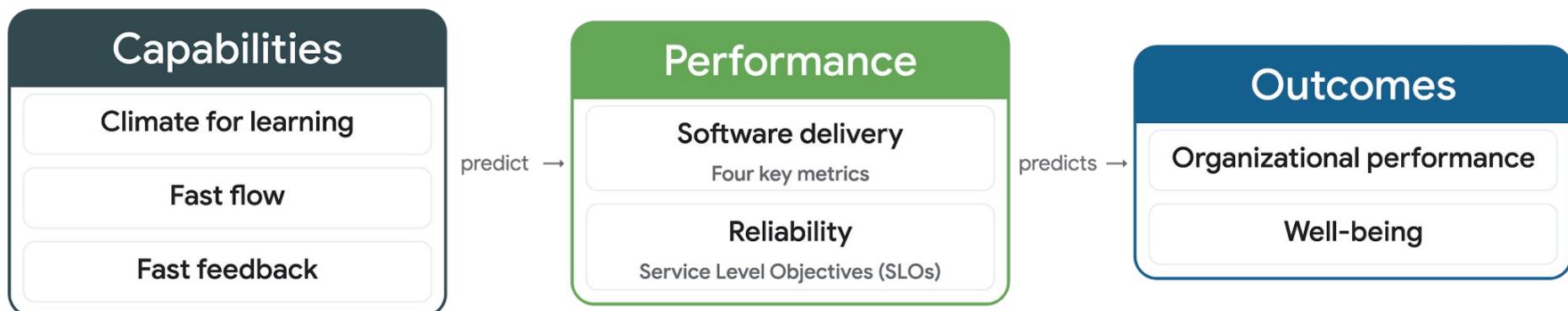
Improvements in software delivery are possible for every team and in every company, as long as **leadership provides consistent support** — including time, actions, and resources — demonstrating a true commitment to improvement, as long as **team members commit themselves to the work.**

*Accelerate: Building and Scaling High Performing Technology Organizations*



DEVOPS RESEARCH & ASSESSMENT





DORA Core model v2.0.0

# Capabilities that enable a Climate for Learning

## Code maintainability core

Make it easy for developers to find, reuse, and change code, and keep dependencies up-to-date.

[Learn more →](#)

## Documentation quality core

Maintain accurate, well-organized, user-centric internal documentation to empower teams throughout the software development process.

[Learn more →](#)

## Empowering teams to choose tools core

Empower teams to make informed decisions on tools and technologies. Learn how these decisions drive more effective software delivery.

[Learn more →](#)

## Generative organizational culture core

Discover how growing a generative, high-trust culture drives better organizational and software delivery performance.

[Learn more →](#)

## Job satisfaction core

Find out about the importance of ensuring your people have the tools and resources to do their job, and of making good use of their skills and abilities.

[Learn more →](#)

## Learning culture optional

Grow a learning culture and understand

the relationship between learning culture and delivery outcomes

## Team experimentation

Innovate faster by building empowered teams that can try out new ideas without approval from people outside the team.

[Learn more →](#)

## Transformational leadership

Learn how effective leaders influence software delivery performance by driving the adoption of technical and product management capabilities.

[Learn more →](#)

## Monitoring systems to inform business decisions

Improve monitoring across infrastructure platforms, middleware, and the application tier, so you can provide fast feedback to developers.

[Learn more →](#)

## Test automation core

Improve software quality by building reliable automated test suites and performing all kinds of testing throughout the software delivery lifecycle.

[Learn more →](#)

## Empowering teams to choose tools core

Empower teams to make informed decisions on tools and technologies. Learn how these decisions drive more effective software delivery.

[Learn more →](#)

## Customer feedback

Drive better organizational outcomes by gathering customer feedback and incorporating it into product and feature design.

[Learn more →](#)

## Continuous integration core

Learn about common mistakes, ways to measure, and how to improve on your continuous integration efforts.

[Learn more →](#)

## Monitoring and observability core

Learn how to build tooling to help you understand and debug your production systems.

[Learn more →](#)

## Pervasive security

Build security into the software development lifecycle without compromising delivery speed.

[Learn more →](#)

## Test data management core

Understand the right strategies for managing test data effectively along with approaches to provide fast, secure data access for testing.

[Learn more →](#)

# Capabilities that enable Fast Flow

## Continuous delivery core

Make deploying software a reliable, low-risk process that can be performed on demand at any time.

[Learn more →](#)

## Database change management

Make sure database changes don't cause problems or slow you down.

[Learn more →](#)

## Flexible infrastructure

Find out how to manage cloud infrastructure effectively so you can achieve higher levels of agility, availability, and cost visibility.

[Learn more →](#)

## Loosely coupled teams

Learn about moving from a tightly coupled architecture to service-oriented and microservice architectures without re-architecting everything at once.

[Learn more →](#)

## Deployment automation

Best practices and approaches for deployment automation and reducing manual intervention in the release process.

[Learn more →](#)

## Streamlining change approval

Replace heavyweight change-approval processes with peer review, to get the benefits of a more reliable, compliant release process without sacrificing speed.

[Learn more →](#)

## Visual management

Learn about the principles of visual management to promote information sharing, get a common understanding of where the team is, and how to improve.

[Learn more →](#)

## Version control

A guide to implementing the right version control practices for reproducibility and traceability.

[Learn more →](#)

## Working in small batches

Create shorter lead times and faster feedback loops by working in small batches. Learn common obstacles to this critical capability and how to overcome them.

[Learn more →](#)



### **Change lead time:**

the time it takes for a code commit or change to be successfully deployed to production.



### **Deployment frequency:**

how often application changes are deployed to production.



### **Change fail rate:**

the percentage of deployments that cause failures in production,<sup>1</sup> requiring hotfixes or rollbacks.



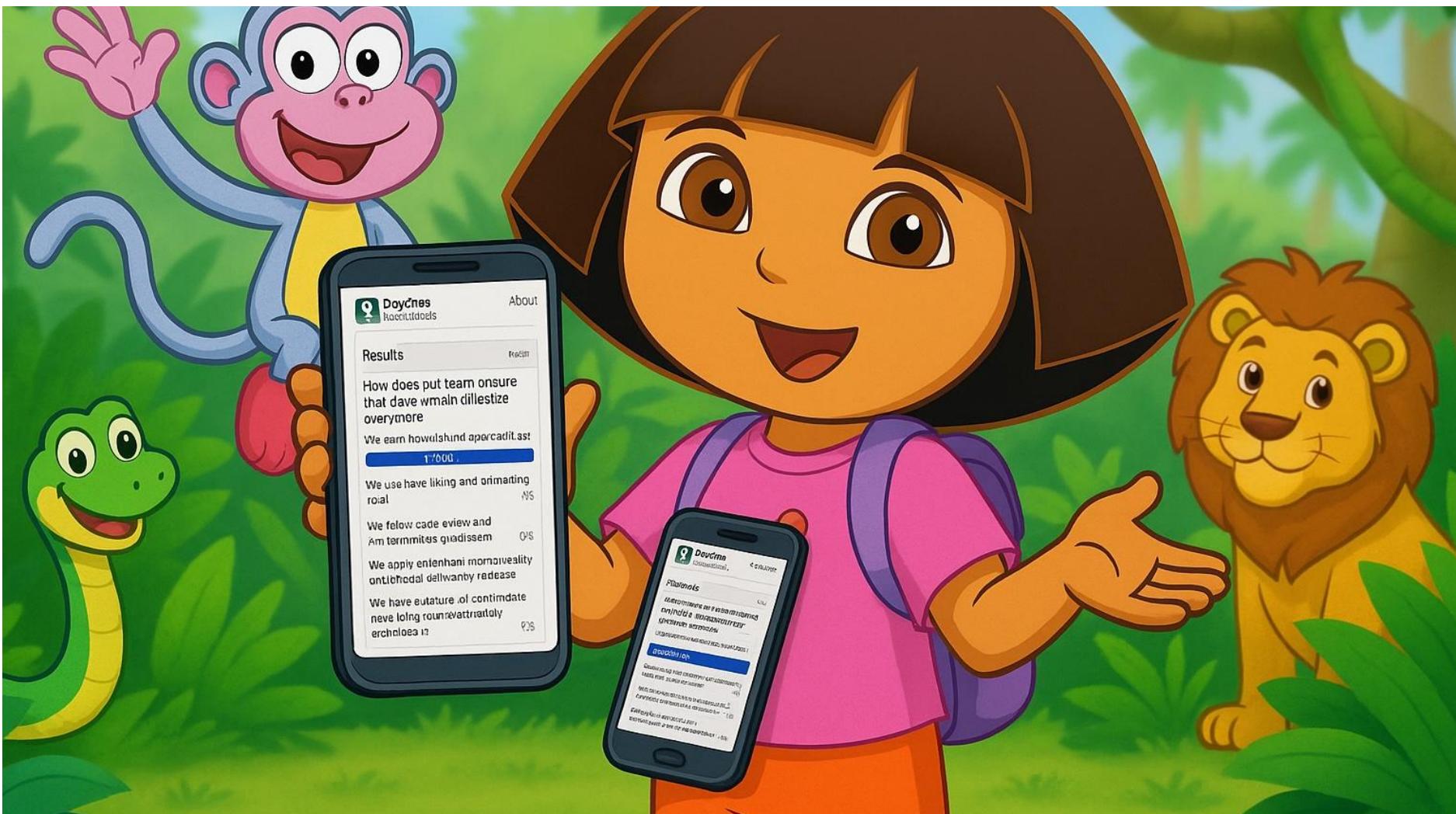
### **Failed deployment recovery time:**

the time it takes to recover from a failed deployment.

## When compared to low performers, elite performers realize



Performance level	Change lead time	Deployment frequency	Change fail rate	Failed deployment recovery time	Percentage of respondents*
Elite	Less than one day	On demand (multiple deploys per day)	5%	Less than one hour	19% (18-20%)
High	Between one day and one week	Between once per day and once per week	20%	Less than one day	22% (21-23%)
Medium	Between one week and one month	Between once per week and once per month	10%	Less than one day	35% (33-36%)
Low	Between one month and six months	Between once per month and once every six months	40%	Between one week and one month	25% (23-26%)



this page is intentionally left blank



**CREATE  
CHANGE**

# How to implement transformation

- ❑ Set goals and enable team experimentation
- ❑ Build community structures to spread knowledge

# Principles of effective organizational change management

- ❑ Improvement work is never done
- ❑ Leaders and teams agree on and communicate measurable outcomes, and teams determine how to achieve them
- ❑ Large-scale change is achieved iteratively and incrementally

# 15% Solutions



Think about everything you've heard—stories, experiments, questions. Ask yourself:

What's one thing you can do this week to help your team move faster, focus better, or learn quicker—without needing permission?

Pair up with the person next to you, and share your plans



# Thanks!



# Thanks!

Feel free to contact me, if you have questions, or want to know more!



<https://www.linkedin.com/in/jacobduijzer/>



<https://github.com/jacobduijzer>



# Sources & Inspiration

- [Accelerate - The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations](#)
- [DORA - Get better at getting better](#)
- [How to transform your organization](#)
- [The Key to High Performance: What the Data Says - Dr. Nicole Forsgren](#)
- [Sooner Safer Happier: Antipatterns and Patterns for Business Agility- Jon Smart](#)
- [DevOps Quickscan Git Repository](#)