



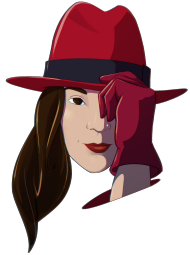
23rd - 27th June

2025

Platform Engineering for Architects

Crafting Platforms as a Product

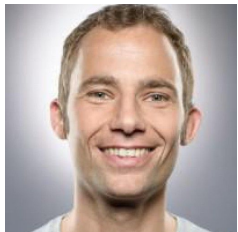
Hilliary Lipsig, RedHat
Max Körbächer, Liquid Reply
Andreas Grabner, Dynatrace



Hillary Lipsig
Senior Principal SRE
RedHat



Max Körbächer
Cloud Native Advisor & Managing Director
Liquid Reply



Andreas Grabner
DevRel & CNCF Ambassador
Dynatrace



Internal Development Platform “Projects” tend to fail.

Adoption

Covering the needs and demands of its end users

Management expectations

Missing to meet promised value



Those failure is often caused by people, culture and processes

- No defined purpose
- Missing a vision
- Technical debts
- Over engineering
- No listening/measurement of acceptance & effectiveness
- “Randomeering”

AND a infrastructure-first thinking.



How infrastructure-first thinking kills your platform ambitions.

Infrastructure-First Thinking

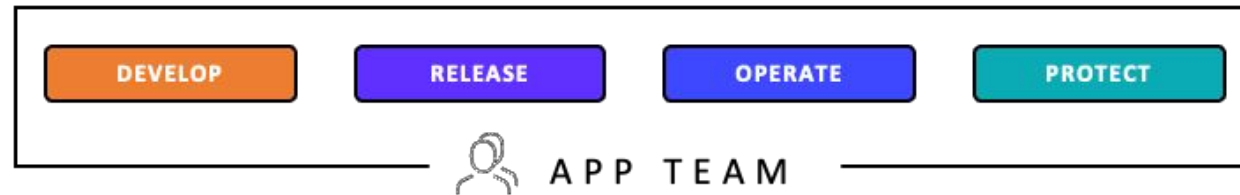
Infrastructure-first thinking is an approach to platform development that focuses primarily on the underlying technology stack and infrastructure components before considering developer experiences or business outcomes. This mindset prioritizes technical architecture decisions, implementation details as the starting point

Characteristics of Infrastructure-First Thinking:

- **Technology-Centered:** Places primary focus on selecting technologies, tools, and infrastructure components
- **Architecture-Driven:** Begins with architectural decisions rather than user needs
- **Tool-Oriented:** Emphasizes building a collection of tools rather than cohesive developer experiences
- **Bottom-Up Approach:** Designs from the infrastructure layer upward rather than from user needs downward
- **Solution Before Problem:** Often selects technologies before fully understanding the problems to solve



New Tech Stacks requires “Shifting Left”!
Teams taking on end-2-end responsibility



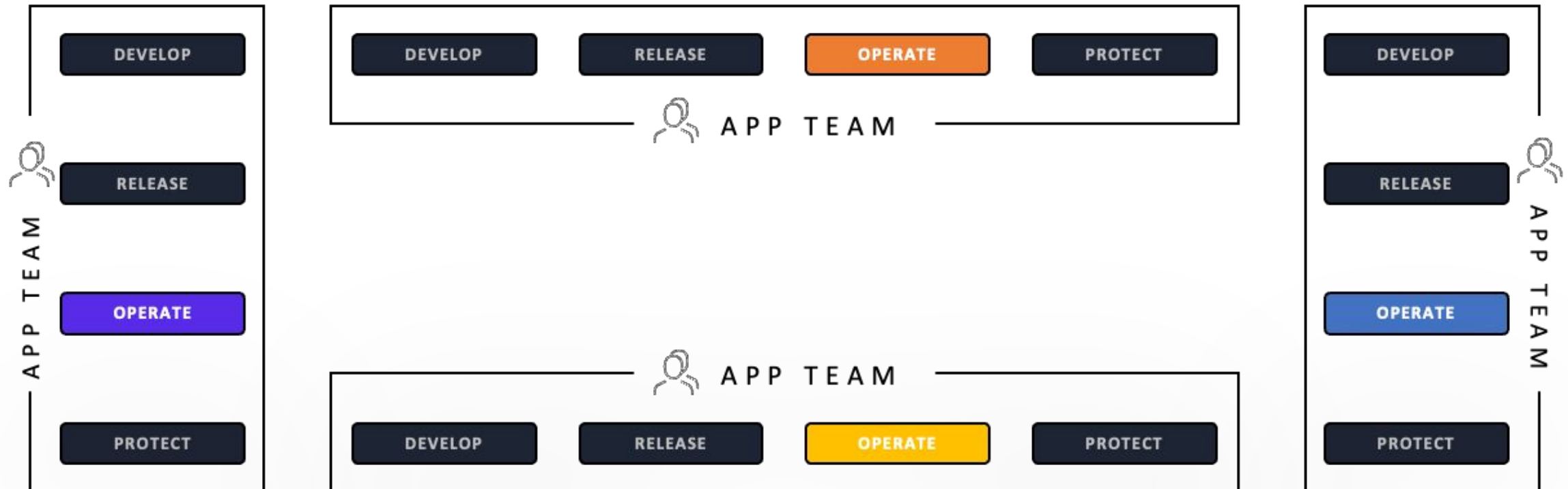


When new tech stack adoption grows -
Teams start do things “their way”!



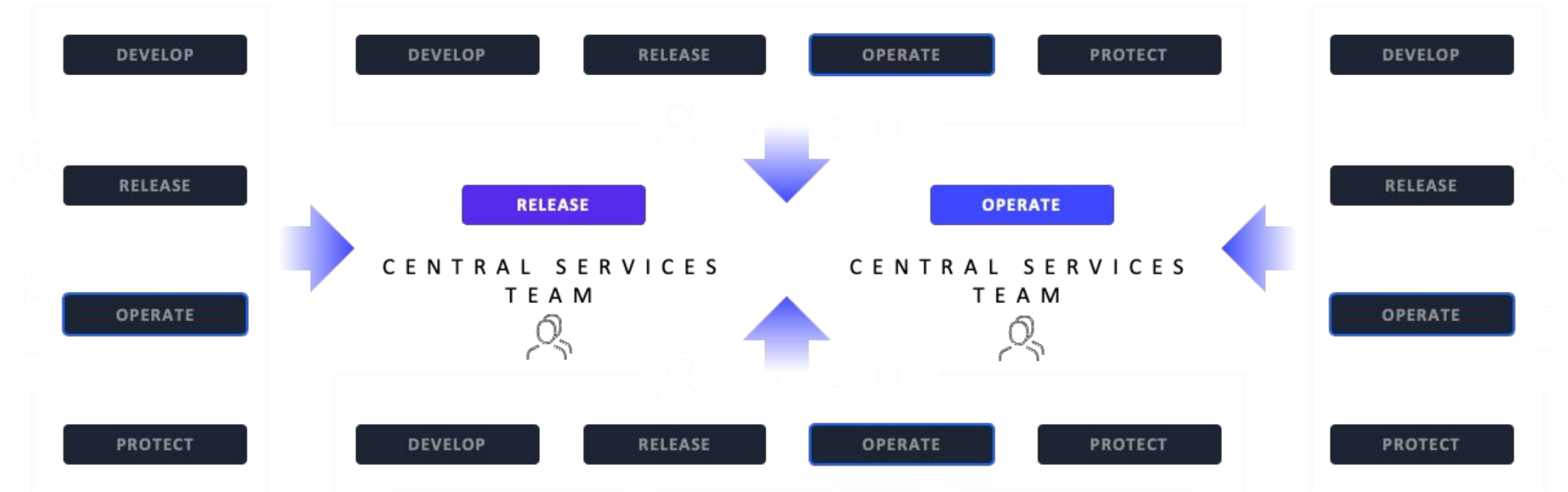


When new tech stack meets old stacks -
“their way” becomes “duplicated effort!”



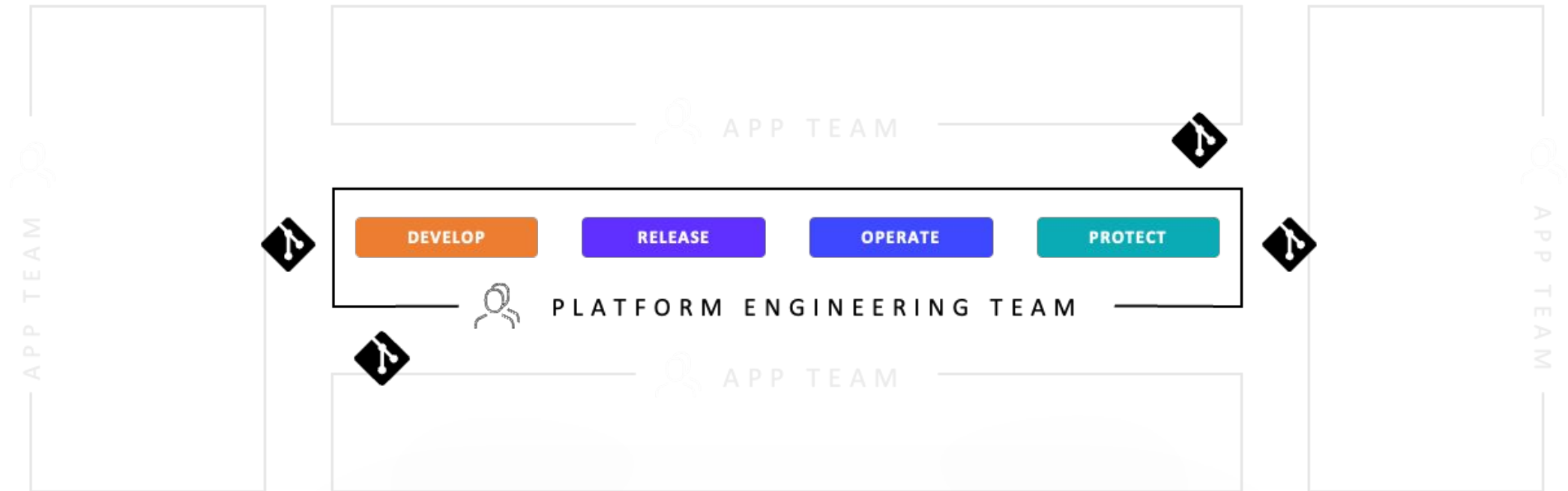


Central Services became the “Patch Solution”





Leading to Platform Engineering: Git* & IDPs (Internal Development Platform) for “Golden Path Self-Service”



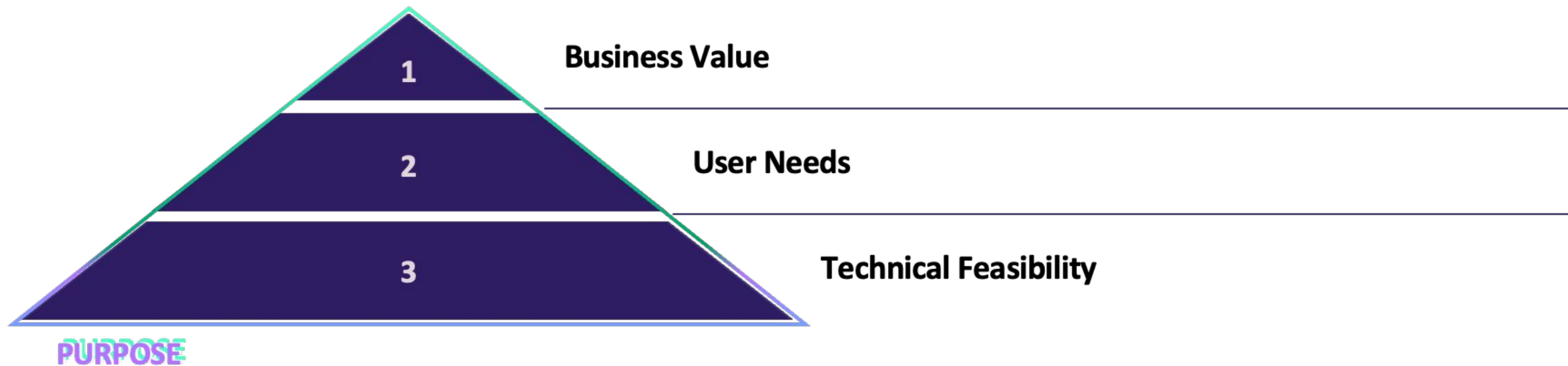


“ A platform without adoption is just infrastructure. A product mindset is what turns platforms into indispensable tools that developers want to use, not tools they're forced to use. ”



The Importance of Purpose

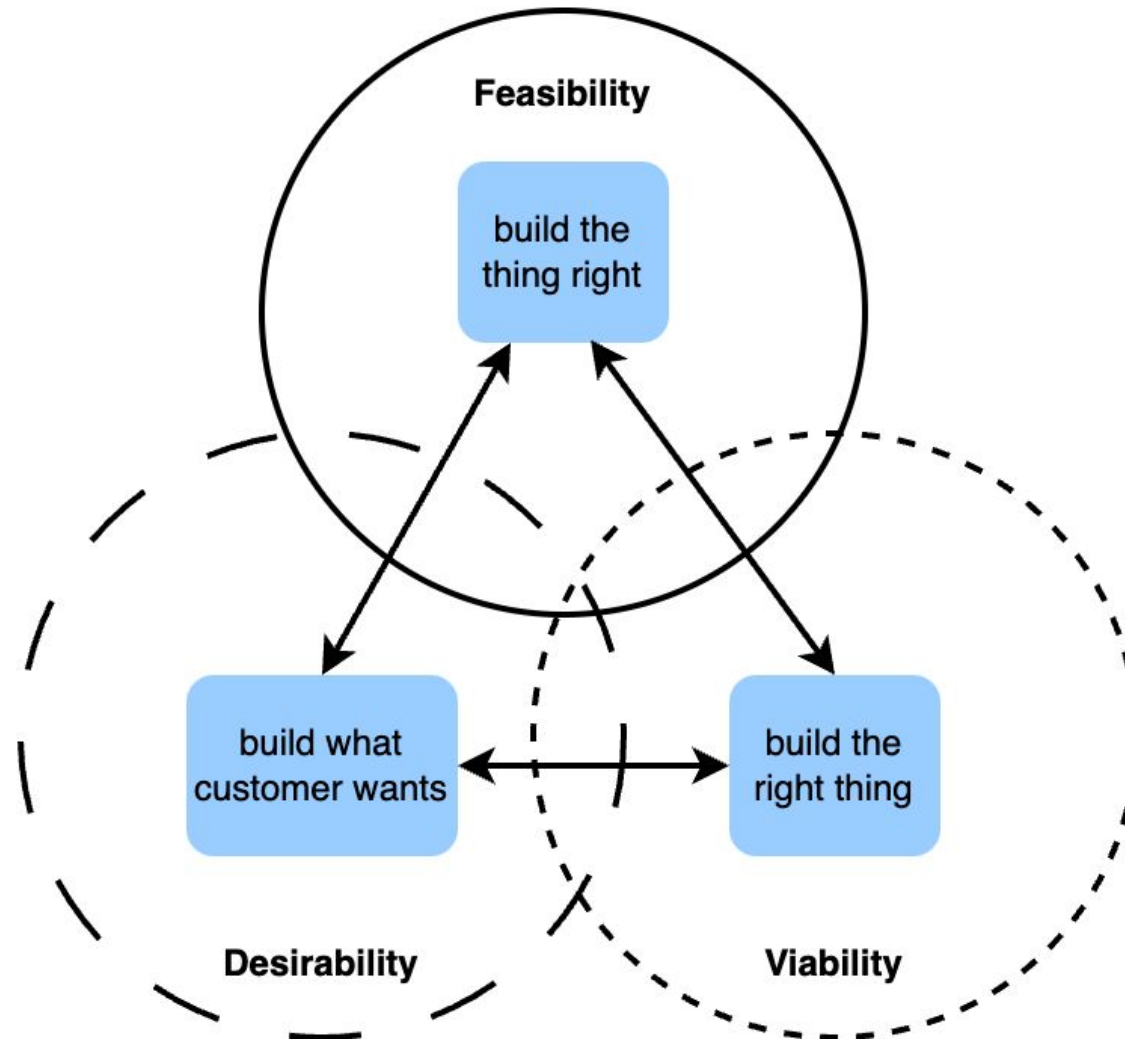
A well defined purpose builds the bracket around business and technical demands giving to it sense and direction.



- Defining purpose ensures that the platform's realization stays on track.
- All features must ladder up to a clear purpose.
- It guides decision-making, as well as resource allocation.

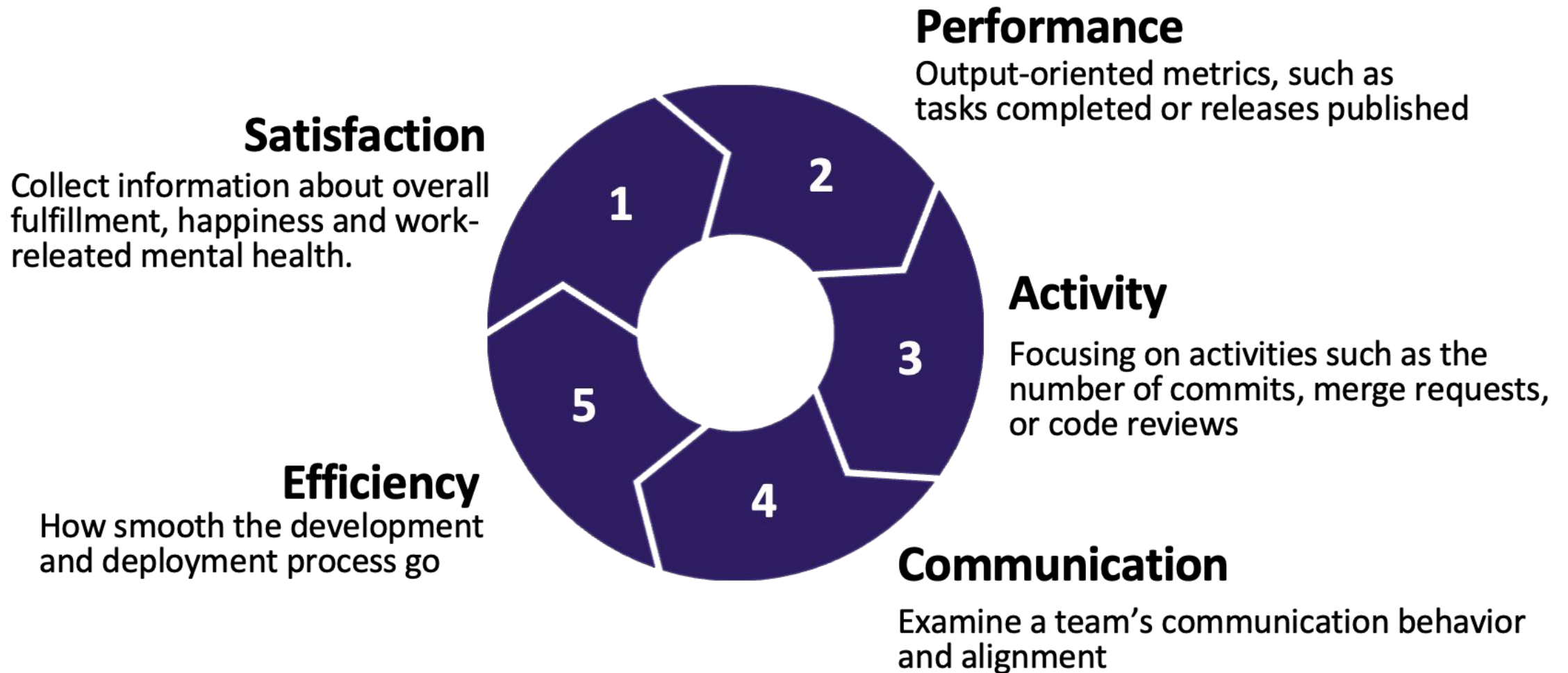


Creating a platform for everyone is impossible





From DORA to SPACE Metrics: Gauging Platform Adoption





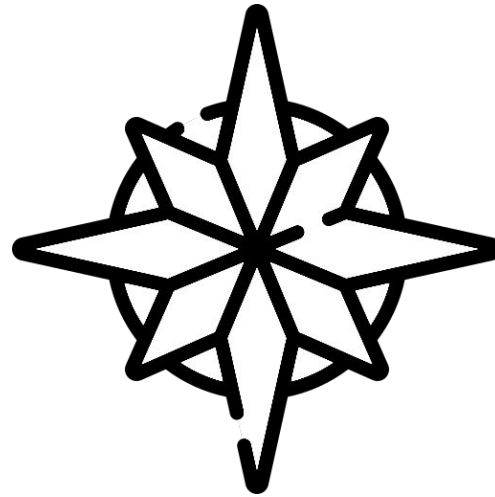
Deliver a clear purpose

User Research

Measure and adept to the efficiency & adoption

Tech Debt Management

Establish Principles



The imperative of change

Challenge Conway's Law

Build a Community



Manage Technical debt by building what matters

1

User Research

Conduct thorough user research. Understand their pain points.

2

Feedback Loops

Establish feedback loops. Continuously improve the platform.

3

Iterative Design

Use an iterative design approach. Adapt to changing needs.



... and then there are the Depreciation Criterias





The latest K8s & tech

Backstage

GitOps

Rebranded DevOps

Infra departments believing to know what someone else needs



Platform Engineering and its success is defined by

People

Culture

Purpose

And anything else between the tech layers

Thank you for joining!



<https://dt-url.net/devrel-pe-book-amzn>

