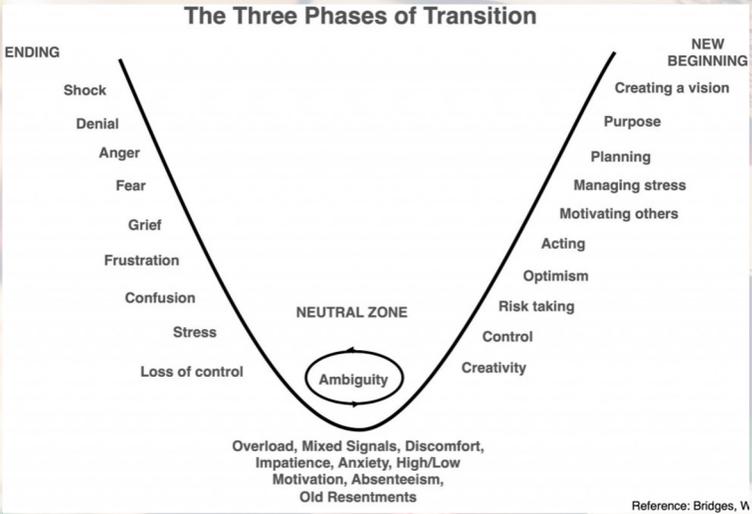


Day Two Digital Transformation: We Fear Change

Building a Path to Production: A Guide for Managers and Leaders in Platform Engineering, Episode 03

Bryan Ross & Coté

Fall, 2023



Bridges Transition Model



Kotter



Prosci ADKAR® Model

Coté



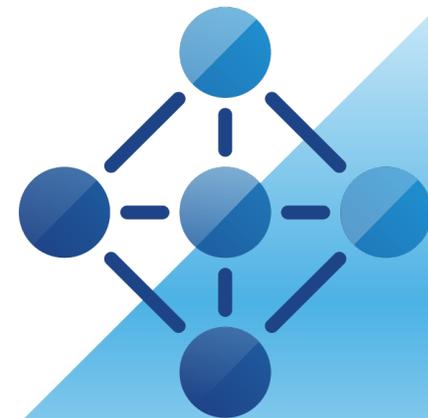
<https://newsletter.cote.io>

Bryan Ross



<https://bryanross.me>

Empathy for the unchanging



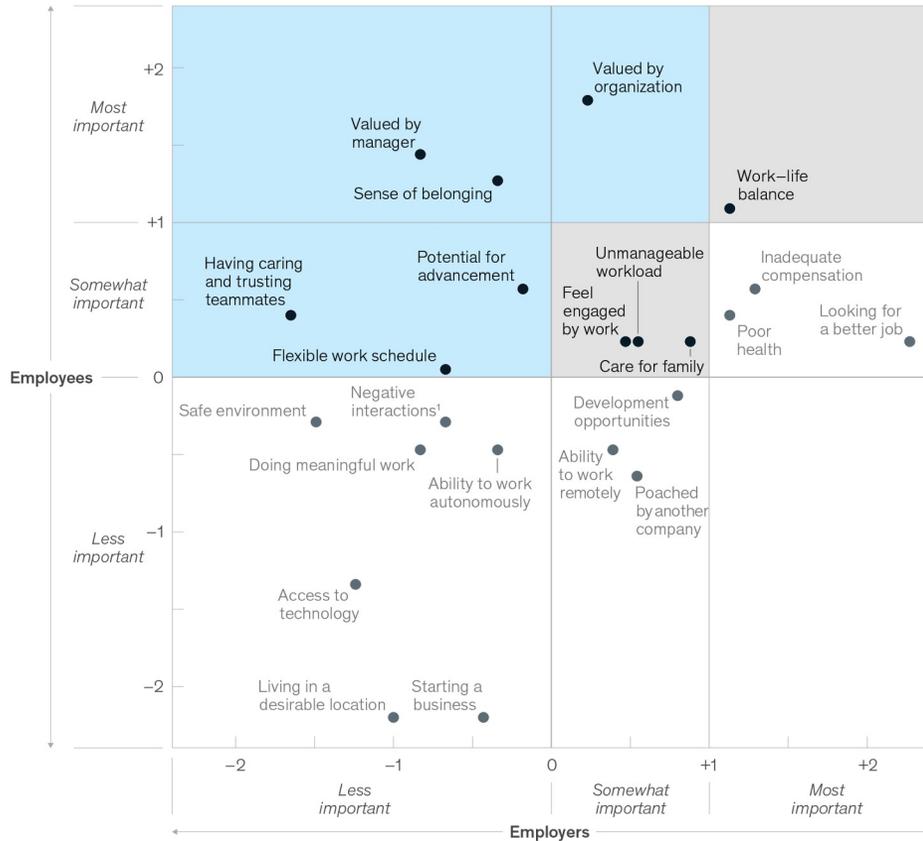


Factors that are important to employees versus what employers think is important

Employer views

► Employers seem to overlook the relational elements that are key drivers for why employees are leaving, such as lack of belonging or feeling valued at work.

More important to employees than employers appreciate | As important to employees as employers think



Note: Standardized scores are reported for both employee and employer perspectives. Employees were asked to respond to the following question: To what extent did the following factors impact your decision to leave your last job? (Not at all, slightly, moderately, very much, extremely); employers were asked to respond to the following question: Why do you think employees are choosing to leave your organization now? (select all that apply)
¹Includes clients, customers, patients, and students.

McKinsey & Company

Adding a focus on opportunities to software developer productivity metrics can offer clearer paths to improvement.

Focus areas by level

● DORA¹ metrics ● SPACE² metrics ● Opportunity-focused metrics

	Outcomes focus <i>Are you delivering products satisfactorily?</i>	Optimization focus ³ <i>Are you delivering products in an optimized way?</i>	Opportunities focus <i>Are there specific opportunities to improve how you deliver products, and what are they worth?</i>
System level	<ul style="list-style-type: none"> ● Deployment frequency ● Customer satisfaction ● Reliability (uptime) 	<ul style="list-style-type: none"> ● Code-review timing ● Velocity/flow through the system 	<ul style="list-style-type: none"> ● Satisfaction with engineering system ● Inner/outer loop time spent
Team level	<ul style="list-style-type: none"> ● Lead time for changes ● Change failure rate ● Time to restore service ● Code-review velocity 	<ul style="list-style-type: none"> ● Story points completed ● Handoffs 	<ul style="list-style-type: none"> ● Quality of documentation ● Developer Velocity Index benchmark⁴ ● Contribution analysis
Individual level	<ul style="list-style-type: none"> ● Developer satisfaction ● Retention 	<ul style="list-style-type: none"> ● Interruptions 	<ul style="list-style-type: none"> ● Contribution analysis ● Talent capability score

¹Google's DevOps research and assessment team, which developed these outcome metrics.

²Satisfaction and well-being, performance, activity, communication and collaboration, and efficiency and flow; GitHub and Microsoft Research developed these metrics, which aim to look at developer well-being as a measurement at the individual level.

³Nonexhaustive.

⁴Benchmarks an organization's technology, working practices, and organizational enablement; see Shivam Srivastava, Kartik Trehan, Dilip Wagle, and Jane Wang, "Developer Velocity: How software excellence fuels business performance," McKinsey, Apr 20, 2020.

McKinsey & Company

A thriving organization focuses on satisfaction, not productivity

Causes of thriving	Because a developer is...
Agency	<ol style="list-style-type: none"> 1) able to voice disagreement with team definitions of success 2) has a voice in how their contributions are measured
Motivation & Self-Efficacy	<ol style="list-style-type: none"> 1) motivated when working on code at work 2) can see tangible progress most of the time 3) is working on the type of code work they want to work on 4) is confident that even when working in code is unexpectedly difficult, they will solve their problems
Learning Culture	<ol style="list-style-type: none"> 1) learning new skills as a developer 2) able to share the things they learn at work
Support & Belonging	<ol style="list-style-type: none"> 1) supported to grow, learn, and make mistakes by their team 2) agrees they are accepted for who they are by their team

FIGURE 1: EXAMPLE METRICS

LEVEL	SATISFACTION & WELL-BEING How fulfilled, happy, and healthy one is	PERFORMANCE An outcome of a process	ACTIVITY The count of actions or outputs	COMMUNICATION & COLLABORATION How people talk and work together	EFFICIENCY & FLOW Doing work with minimal delays or interruptions
INDIVIDUAL One person	<ul style="list-style-type: none"> *Developer satisfaction *Retention[†] *Satisfaction with code reviews assigned *Perception of code reviews 	<ul style="list-style-type: none"> *Code review velocity 	<ul style="list-style-type: none"> *Number of code reviews completed *Coding time *# Commits *Lines of code[†] 	<ul style="list-style-type: none"> *Code review score (quality or thoughtfulness) *PR merge times *Quality of meetings[†] *Knowledge sharing, discoverability (quality of documentation) 	<ul style="list-style-type: none"> *Code review timing *Productivity perception *Lack of interruptions
TEAM OR GROUP People that work together	<ul style="list-style-type: none"> *Developer satisfaction *Retention[†] 	<ul style="list-style-type: none"> *Code review velocity *Story points shipped[†] 	<ul style="list-style-type: none"> *# Story points completed[†] 	<ul style="list-style-type: none"> *PR merge times *Quality of meetings[†] *Knowledge sharing or discoverability (quality of documentation) 	<ul style="list-style-type: none"> *Code review timing *Handoffs
SYSTEM End-to-end work through a system (like a development pipeline)	<ul style="list-style-type: none"> *Satisfaction with engineering system (e.g., CI/CD pipeline) 	<ul style="list-style-type: none"> *Code review velocity (acceptance rate) *Customer satisfaction *Reliability (uptime) 	<ul style="list-style-type: none"> *Frequency of deployments 	<ul style="list-style-type: none"> *Knowledge sharing, discoverability (quality of documentation) 	<ul style="list-style-type: none"> *Code review timing *Velocity/flow through the system

[†] Use these metrics with [even more] caution – they can proxy more things.

“This is a 1 ½ CIO job.”

keep it small

Exec View Work the Same Transform!

Compensation \$ \$\$\$\$

Risk HIGH HIGH

Outcome  

Staff's View Work the Same Transform!

Compensation \$ \$

Risk LOW HIGH

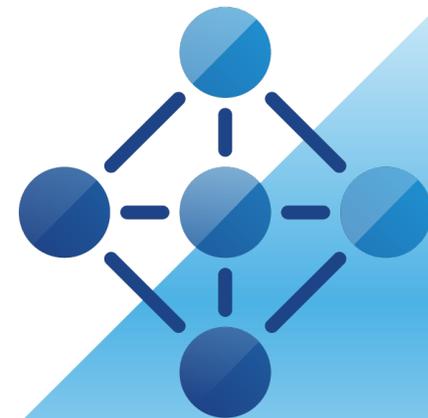
Outcome  

Exec View	Work the Same	Transform!
Compensation	\$	\$\$\$\$
Risk	HIGH	HIGH
Outcome		

Staff's View	Work the Same	Transform!
Compensation	\$	
Risk		HIGH
Outcome		

Diagram: Two green curved arrows connect the 'LOW' box to the '\$' box and the '\$' box to the 'HIGH' box, forming a cycle.

Tools to start with



Make their work-life better, and show progress

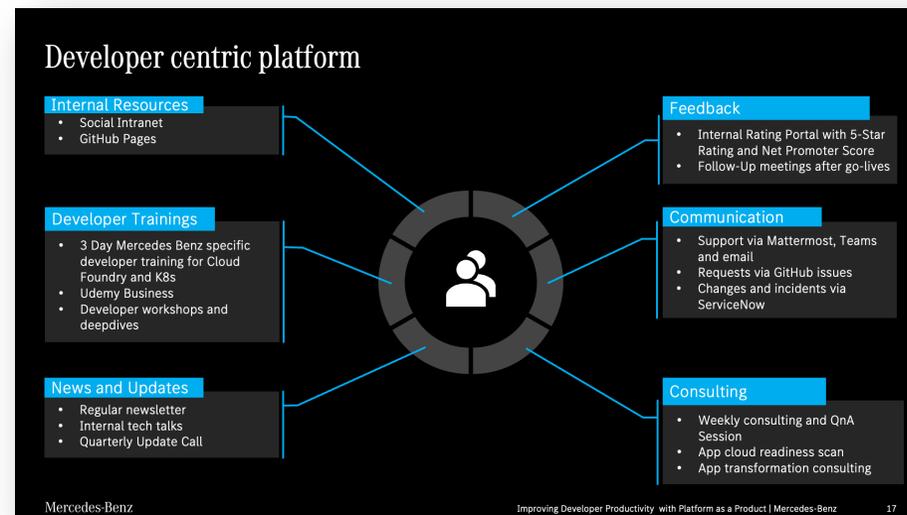
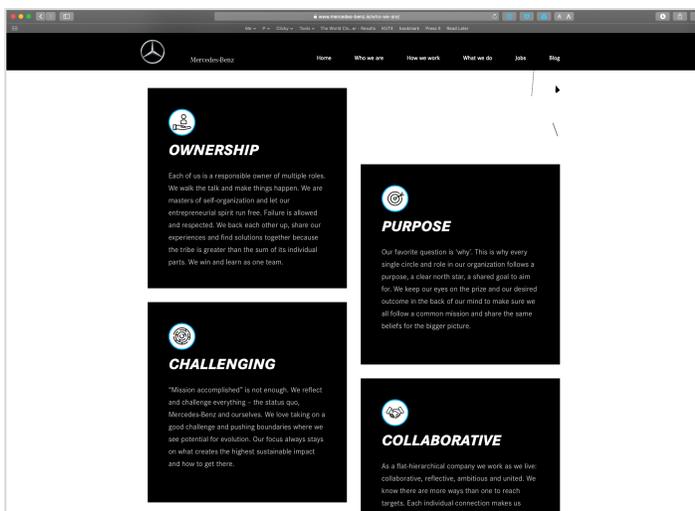
Find the Developer Toil, Confusion, Blockers

- What are we making?
- We have a strong vision for our product, and we're doing important work together every day to fulfill that vision.
- I have the context I need to confidently make changes while I'm working.
- I am proud of the work I have delivered so far for our product.
- I am learning things that I look forward to applying to future products.
- My workstation seems to disappear out from under me while I'm working.
- It's easy to get my workstation into the state I need to develop our product.
- What aspect of our workstation setup is painful?
- It's easy to run our software on my workstation while I'm developing it.
- I can boot our software up into the state I need with minimal effort.
- What aspect of running our software locally is painful? What could we do to make it less painful?
- It's easy to run our test suites and to author new ones.
- Tests are a stable, reliable, seamless part of my workflow.
- Test failures give me the feedback I need on the code I am writing.
- What aspect of production support is painful?
- We collaborate well with the teams whose software we integrate with.
- When necessary, it is within my power to request timely changes from other teams.
- I have the resources I need to test and code confidently against other teams' integration points.
- What aspect of integrating with other teams is painful?
- I'm rarely impacted by breaking changes from other tracks of work.
- We almost always catch broken tests and code before they're merged in.
- What aspect of committing changes is painful?
- Our release process (CI/CD) from source control to our story acceptance environment is fully automated.
- If the release process (CI/CD) fails, I'm confident something is truly wrong, and I know I'll be able to track down the problem.
- What aspect of our release process (CI/CD) is painful?
- Our team releases new versions of our software as often as the business needs us to.
- We are meeting our service-level agreements with a minimum of unplanned work.
- When something is wrong in production, we reproduce and solve the problem in a lower environment.

Organizational Learning



Focus on ways of working.....



Change Yourself!



VMware Tanzu Labs

Own your modernization journey

VMware Tanzu Labs (formerly Pivotal Labs) partners with organizations worldwide to accelerate the delivery of software and modernize legacy apps, while reducing operating costs and risk.

WHY TANZU LABS

WAYS TO ENGAGE

- Build apps
- Modernize apps
- Build a platform
- Transform data

Technical Account Management
Services for nonprofits



Let's start with one project.

Contact VMware Tanzu Labs

[View global locations](#)

First Name

Last Name

Company Name

Business Email

Job Title

Country

SUBMIT

Some problems require more than tools.

Engage with VMware Tanzu Labs to start building the right thing for your users—and get it into production quickly.

