



# Measure for **CHANGE**

Laura Tacho, CTO @ DX // [getdx.com](https://getdx.com)

Is developer productivity  
actually

**IMPROVING?**

Measuring and  
**IMPROVING**  
are two different things

Too many organisations spend effort building beautiful DORA dashboards that nobody looks at.



*Nathen Harvey*

*DORA*

Measurement isn't  
getting us where we

**NEED TO GO**



If we want to  
**IMPROVE**



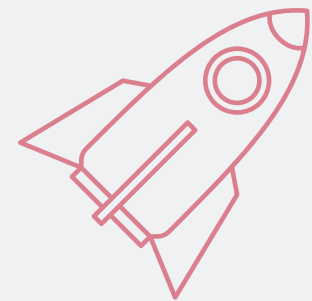
We must change  
the way we

**MEASURE**

Measure for  
**CHANGE**

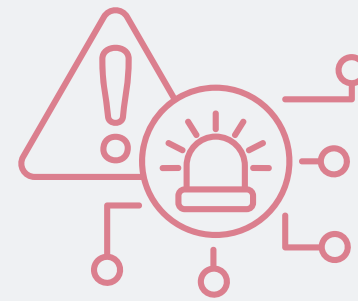


# Developer Experience at Pfizer



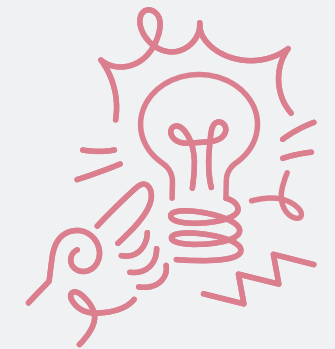
**22%**

more devs can  
deploy in <1 hour



**33%**

more devs can  
resolve an incident in  
<1 hour



**6.6%**

more devs feel their  
applications are  
high quality

There's  
**TOO MUCH  
AT STAKE**  
to move slowly

- Involve your team
- Tie data to decisions
- Always follow up

Qualitative  
**OR**  
Quantitative

## PROS

## CONS

### **QUANTITATIVE**

(DORA, Git, etc.)

- Objective
- Data readily available


- Incomplete
- Doesn't tell you how to improve

### **QUALITATIVE**

(Surveys, interviews)

- Holistic
- Provides actionable context

- Difficult to do well (design, analysis, poor participation)



Dashboards and  
reporting weren't  
helping their teams

**IMPROVE**

# Developer Productivity

## **MYTH:**

Quantitative data is more important than qualitative data.

By humans,

**FOR HUMANS**



Disregarding developer  
feedback only gives you  
half the picture.

Quantitative data

**LIES TO US**

all the time



**PagerDuty** BOT 11:08 PM

Triggered [#3727](#): 500 count is elevated on multiple nodes

Assigned: [Bruce Soord](#)

Service: [Payments API](#)

Dial-in Number: [202-555-0179,,834561#](#)

Meeting URL: [conferencecall.biz](#)

 Reassigned to Bruce Soord | Today at 11:12 PM

Acknowledge

Resolve

Acknowledged [#3727](#): 500 count is elevated on multiple... [Jump](#) | by [Jon Sykes](#)



**jon** 11:10 PM

hmm, looks to be related to a deploy that just went out on the auth stack

[@brucey](#): looks like your deploy, do you know what the problem is?

*pagerduty.com*

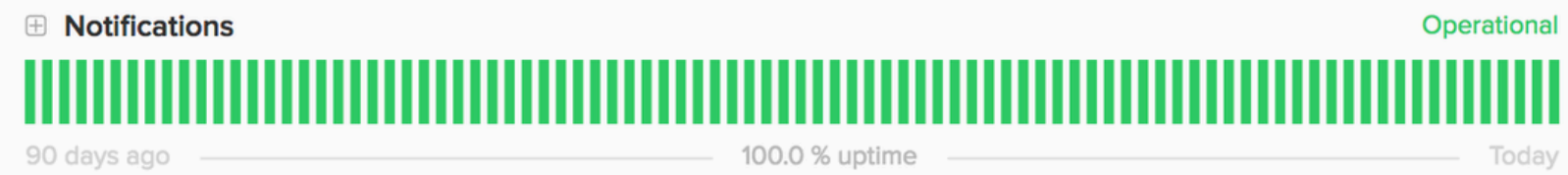
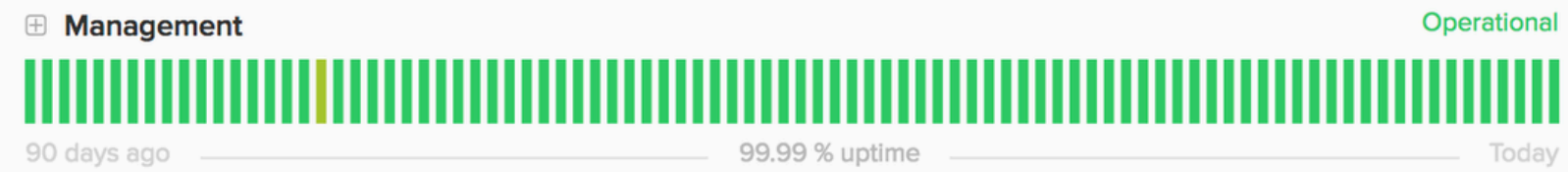
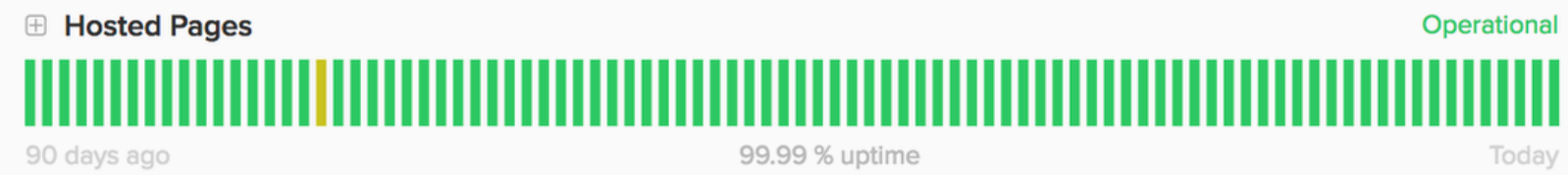


You can only measure  
what you think to  
measure

False sense of  
**SECURITY**  
from lack of data

All Systems Operational

Uptime over the past 90 days. [View historical uptime.](#)



Without these intermediary metrics, stakeholders may think that a year with no outages is evidence that the system is high quality because they lack visibility into how developers are experiencing the code base.

C. Green, C. Jaspan, M. Hodges and J. Lin,  
"Developer Productivity for Humans, Part 7: Software Quality,"  
in IEEE Software, vol. 41, no. 1, pp. 25-30, Jan.-Feb. 2024,



Developer productivity  
is **COMPLEX**



Quantitative-only  
systems of  
measurement are  
**RIGID**

We need something  
**PERMEABLE**  
that co-adapts with  
our teams

Your  
**TEAMS**

know where the  
problems are

Qualitative data is a  
**KEY FACTOR**  
of improving  
developer productivity

Sentiment ▾ All teams ▾ compared to Industry 50th ▾

### Developer Experience Index

Composite score measuring overall developer experience.



Driver	Priority	Sentiment	vs Industry 50th
Deep work	15	33	-27
Managing tech debt	9	38	+2
Local development	7	54	-5
Cross-team collaboration	6	60	+9
<b>Build processes</b>	<b>5</b>	<b>38</b>	<b>-11</b>
Incident response	5	58	-5
Test efficiency	4	50	+14
Requirements quality	4	77	+23
Documentation	3	42	+5
Codebase experience	3	50	-6
Ease of release	3	83	+33
Code review	1	83	+13
Batch size	1	85	+11
Test coverage	0	69	+33

#### Workflow metrics for Build Processes

##### Build stability

% in Target

**10%**

-23 vs Inds. 50th

Target Range

**Less than once every two days**

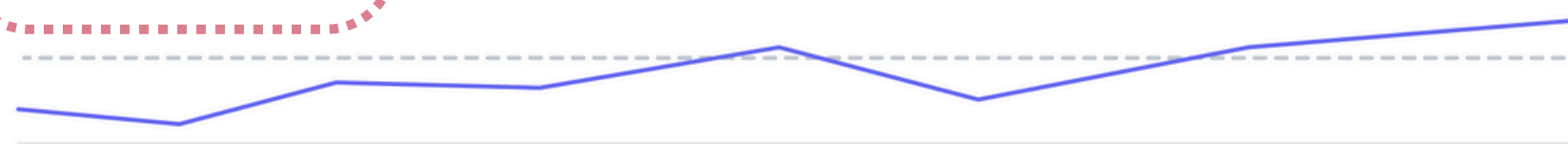
Distribution



##### Build wait-time

**2.8 mins**

-1.1 vs Inds. 50th



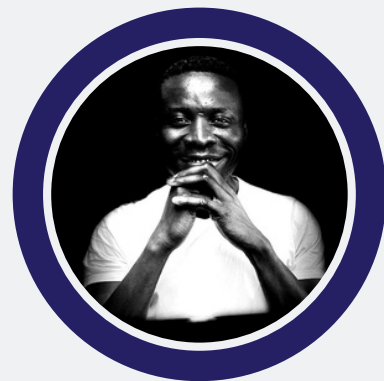
Jan 2024

Mar 2024

**Cara Nolan**  
Data Science  
These have been improving as of late, but it still feels like our hands are very tied using the new CI/CD processes.

**Joseph Mitchell**  
Product Core  
Locally, the build is just long enough for people to get distracted. We need to have near-instantaneous build times so that folks can maintain their mental flow and be as efficient. (every mental flow interruption costs at least 15 minutes, according to some research)

Developers know where the problems are... We needed to understand the full experience of our engineers, which our current tools and processes were not built for.



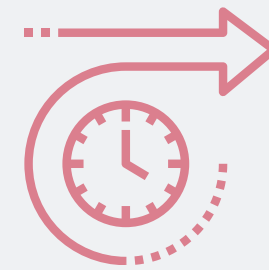
*Moses Mansaray*  
*GfK*

# Developer Experience at GfK



**400**

hours per week of time saved due to better processes, tooling, and communication



**25%**

more devs have more time to focus on work

Measuring and  
**IMPROVING**  
are two different things



## PROS

## CONS

### **QUANTITATIVE**

(DORA, Git, etc.)

- Objective
- Data readily available

- Incomplete
- Doesn't tell you how to improve

### **QUALITATIVE**

(Surveys, interviews)

- Holistic
- Provides actionable context

- Difficult to do well (design, analysis, poor participation)

## **NICOLE FORSGREN, PHD**

Founder of DORA, Partner at Microsoft Research, Author of Accelerate and SPACE Framework



## **MARGARET-ANNE STOREY, PHD**

Co-author of SPACE framework. Computer Science professor at University of Victoria.



## **MICHAELA GREILER, PHD**

Researcher at University of Zurich, formerly Microsoft Research



## **ALBERTO BACCELLI, PHD**

Professor of empirical software engineering at the University of Zurich.



## **LIZ PAVESE, PHD**

Organizational psychologist at Workday. Former head of research at Qualtrics.



## **ABI NODA**

Founder and CEO of DX, Co-author of the DevEx Framework.



# Sentiment and Workflows

How easy or difficult is it for you to do work as a developer/technical contributor at GfK?

- Very difficult
- Somewhat difficult
- Neither easy nor difficult
- Somewhat easy
- Very easy
- I don't know / Not applicable

In the past month, what's the longest it has taken for you to get an approved PR deployed to production?

- Less than 10 minutes
- Less than one hour
- Less than one day
- Less than one week
- More than one week
- I don't know / Not applicable

Sentiment ▾ All teams ▾ compared to Industry 50th ▾

### Developer Experience Index

Composite score measuring overall developer experience.



Driver	Priority	Sentiment	vs Industry 50th
Deep work	15	33	-27
Managing tech debt	9	38	+2
Local development	7	54	-5
Cross-team collaboration	6	60	+9
<b>Build processes</b>	<b>5</b>	<b>38</b>	<b>-11</b>
Incident response	5	58	-5
Test efficiency	4	50	+14
Requirements quality	4	77	+23
Documentation	3	42	+5
Codebase experience	3	50	-6
Ease of release	3	83	+33
Code review	1	83	+13
Batch size	1	85	+11
Test coverage	0	69	+33

#### Workflow metrics for Build Processes

##### Build stability

% in Target

**10%**

-23 vs Inds. 50th

Target Range

**Less than once every two days**

Distribution



##### Build wait-time

**2.8 mins**

-1.1 vs Inds. 50th



**Cara Nolan**  
Data Science  
These have been improving as of late, but it still feels like our hands are very tied using the new CI/CD processes.

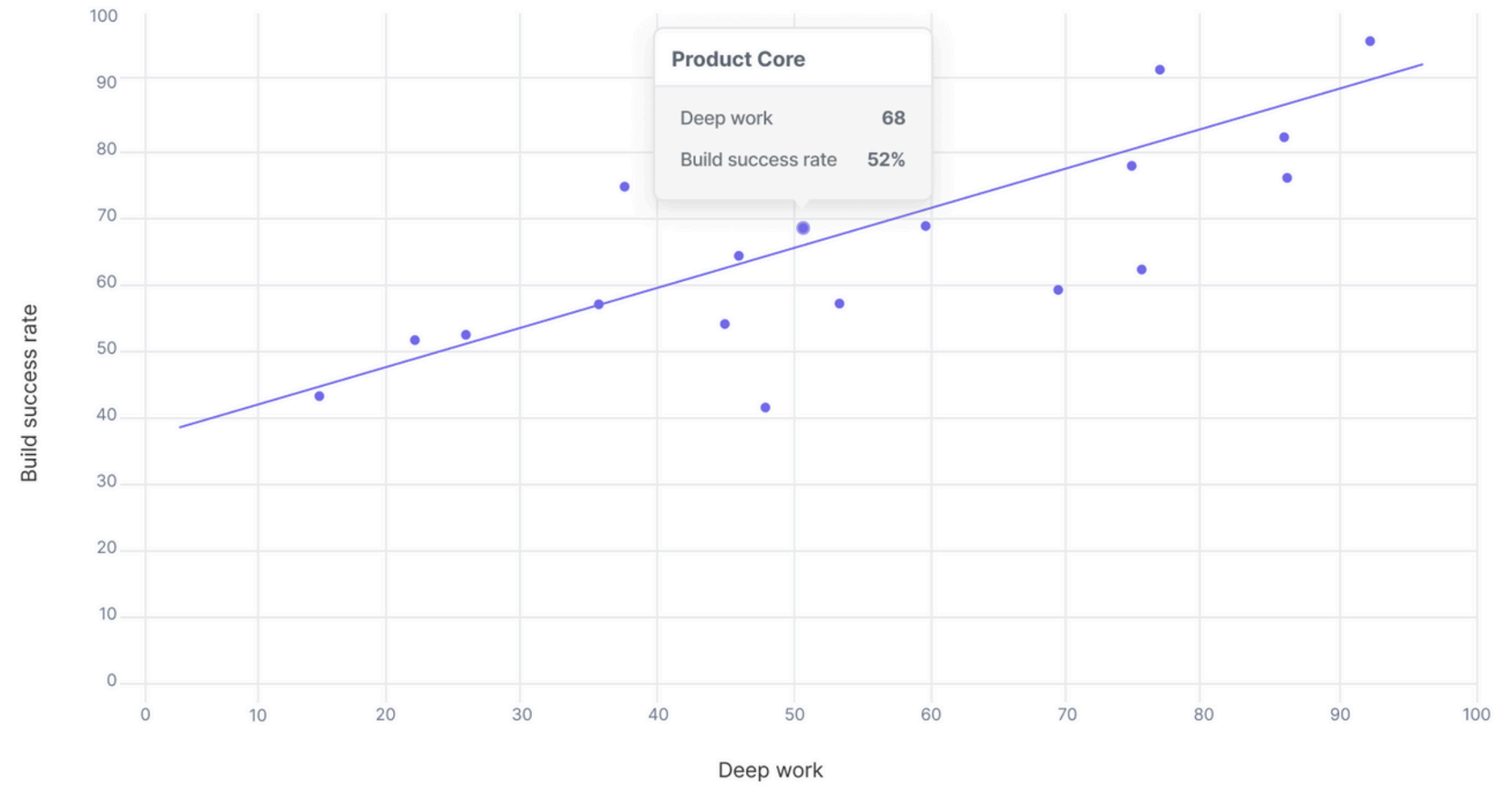
**Joseph Mitchell**  
Product Core  
Locally, the build is just long enough for people to get distracted. We need to have near-instantaneous build times so that folks can maintain their mental flow and be as efficient. (every mental flow interruption costs at least 15 minutes, according to some research)

## Correlations

Reports ▾

Plot DX driver sentiment scores against system metrics

Build success rate ▾ plotted against Deep work ▾





Qualitative  
**AND**  
Quantitative

# Sustained Participation Rates



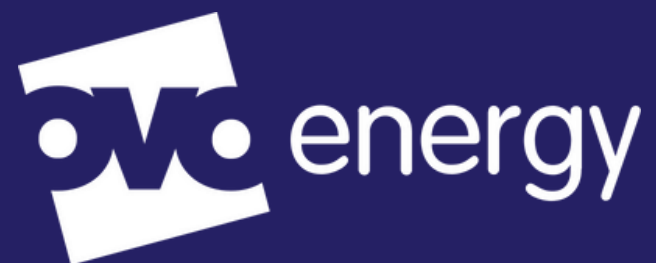
91%

JOHN LEWIS  
PARTNERSHIP

97%



94%



89%



97%

# ▲ Vercel

- 95%+ sustained participation
- Surveys on quarterly cadence
- Teams set focus areas





80%+ of teams

**TAKE ACTION**

based on insights from

DX surveys

We want teams to  
**USE DATA,**  
not just view it



Developer experience  
**SURVEYS**

What if you could capture  
developer feedback  
**IN THE MOMENT?**

# Birds-eye report

Real-time intelligence into usage and satisfaction across your developer platform.

Product	MAUs	Users	CSAT
Backstage	228 ↑	321	89
Airflow repositories	247 ↓	304	67
Mobile CI	201 ↑	212	81
GitHub merge que	177 ↑	198	56
Deploy CLI	143 ↓	160	78

+ Filters v

### How satisfied are you with Backstage service creation?



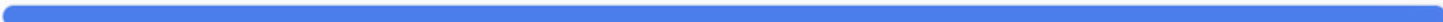
Not satisfied



Somewhat satisfied



Very satisfied



### Comments

 Sajni Patel

The experience was okay. There were a few bumps along the way, but I didn't encounter some errors while creating the service, but was able to resolve them. I would like there could be improvements in error handling and more clarity in the documentation.

 Peter Knudsen

Though the experience was generally positive, there were some hiccups. I found that the auto-generated PRs often failed, requiring manual intervention. Despite this, I appreciate the potential of this tool and look forward to future improvements.

8d ago

Thursday, June 13th



Hi Sarah, how was your experience creating a service through Backstage?

Share response

Message Shawn Williams >

# GitHub Copilot

Projects ▾

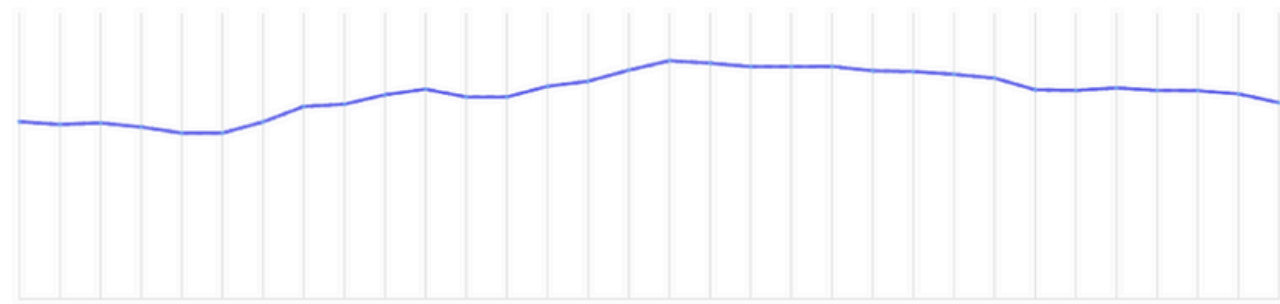
[Dashboard](#) [Users](#) [Events](#) [Surveys](#) [Settings](#)

⚠️ Enable Slack notifications to make sure you don't miss survey responses.

[Setup notifications](#)

Last 30 days ▾

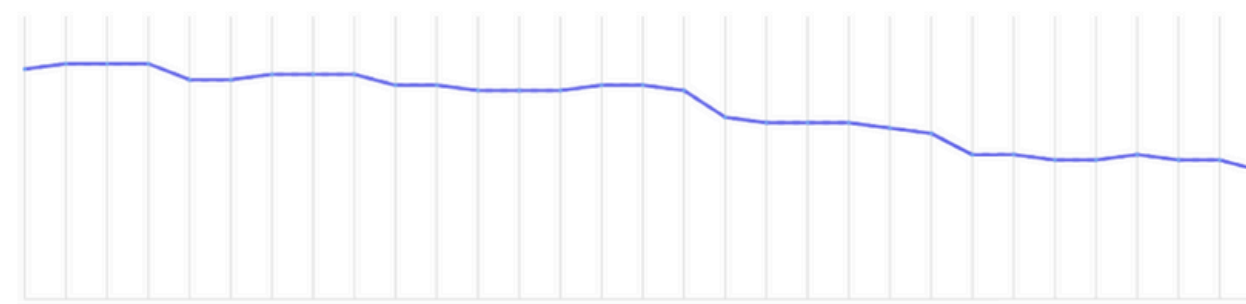
Active users ⓘ  
26 users



May 2024

Jun 2024

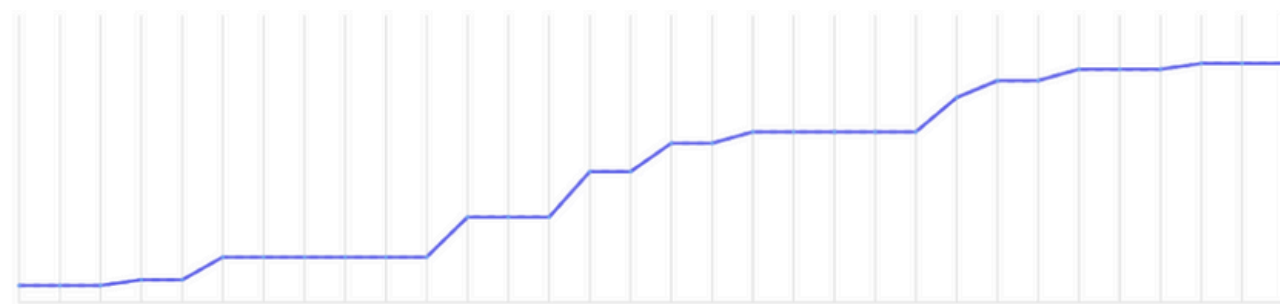
Total users ⓘ  
37 users



May 2024

Jun 2024

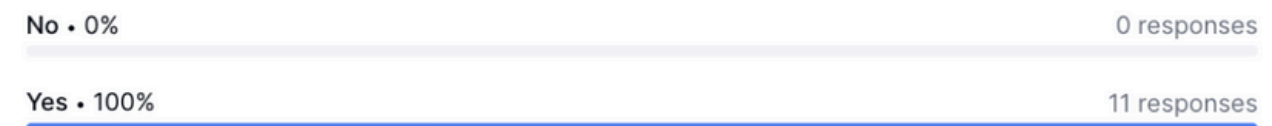
Events ⓘ  
42 events



May 2024

Jun 2024

Did you use Copilot in developing this PR?  
11 responses



Measure for  
**CHANGE**  
by changing how  
you measure



Qualitative data is a  
**KEY FACTOR**  
of improving  
developer productivity



Plaid takes a  
platform-as-a-product  
approach.

Plaid uses data to deeply

**UNDERSTAND**

**AND SOLVE**

developers' problems

I honestly don't know how you would operate as a platform team without survey data...  
It's changed how we think about operating the platform, prioritising different investments, and defining success.  
It's a bit of a cultural shift.



*Russ Nealis*  
*Plaid*

We want teams to

**USE DATA,**

not just view it

Context-free

Artifacts

Throughput

View data

Measurement

Context-full

Knowledge work

Impact

Use data

Improvement



If we want to  
**IMPROVE**



We must change  
the way we

**MEASURE**

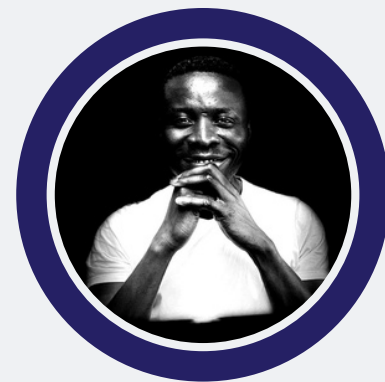


Improving developer  
productivity can't be a

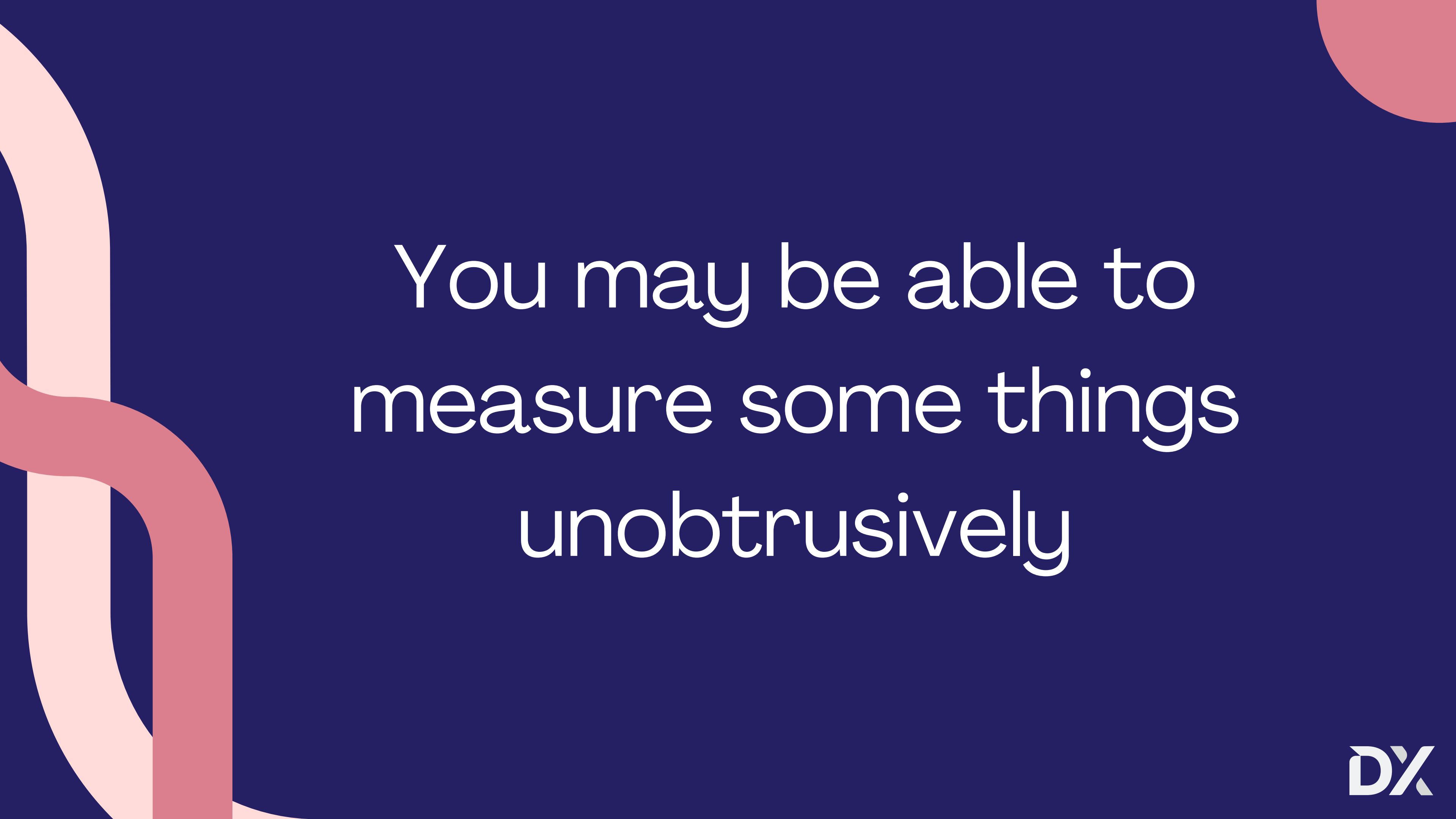
**SECRET  
MISSION**

We're not after a  
**SINGLE PANE  
OF GLASS**

In a way, DX has been like a mirror,  
reflecting back at us what we see as  
a group.



*Moses Mansaray*  
*GfK*



You may be able to  
measure some things  
unobtrusively

But you can't  
**CHANGE**  
them that way

Is developer productivity  
actually

**IMPROVING?**



# Measure for **CHANGE**

Laura Tacho, CTO @ DX // [getdx.com](https://getdx.com)

