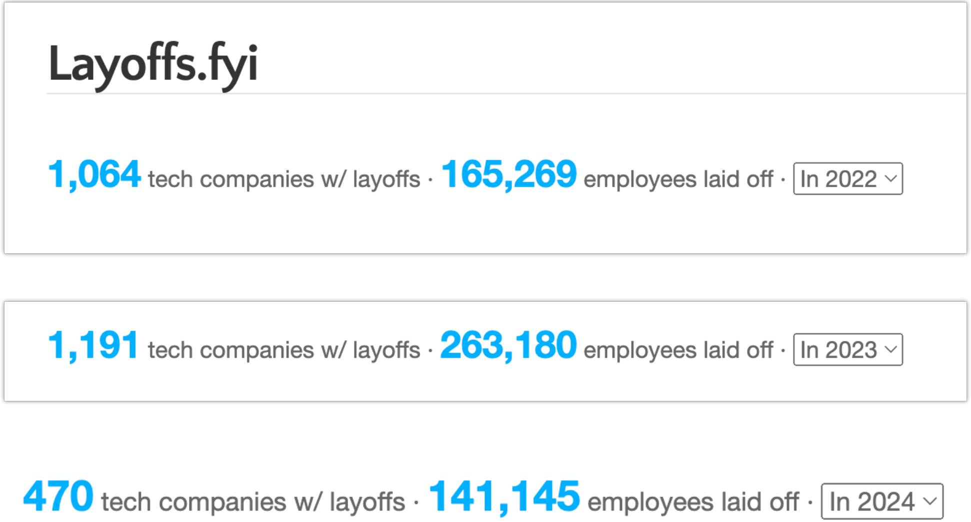
SURVIVING THE TECHPOCALYPSE

EMPOWERING SURVIVORS WITH DEVELOPER PRODUCTIVITY ENGINEERING



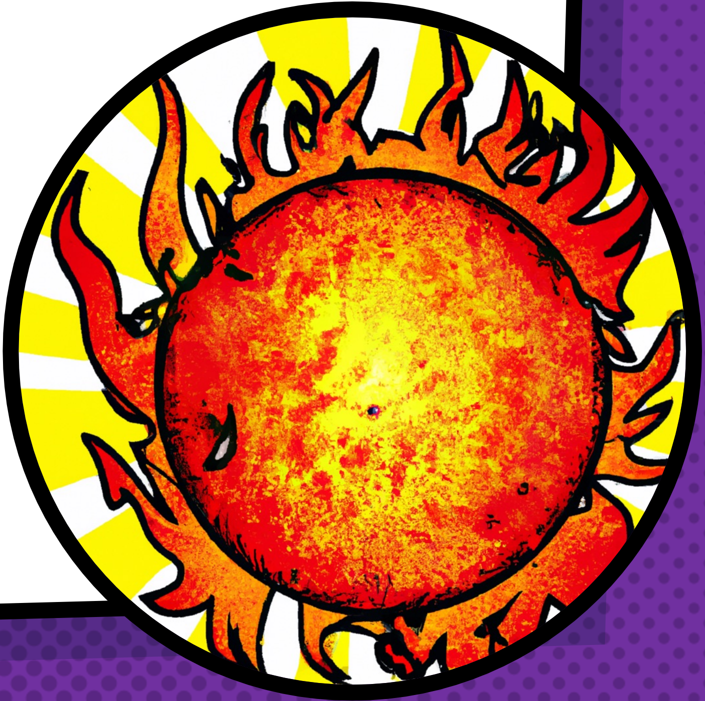
Layoffs.fyi

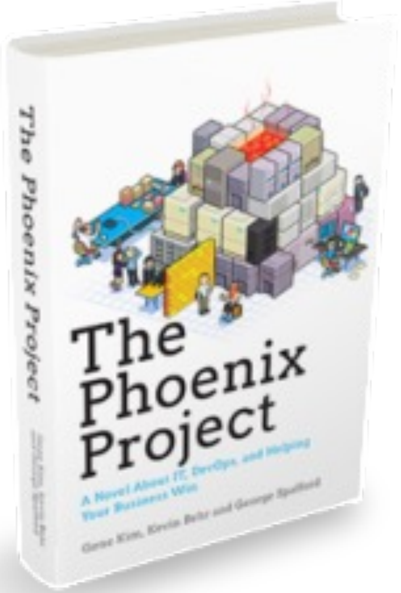
1,064 tech companies w/ layoffs · **165,269** employees laid off · In 2022

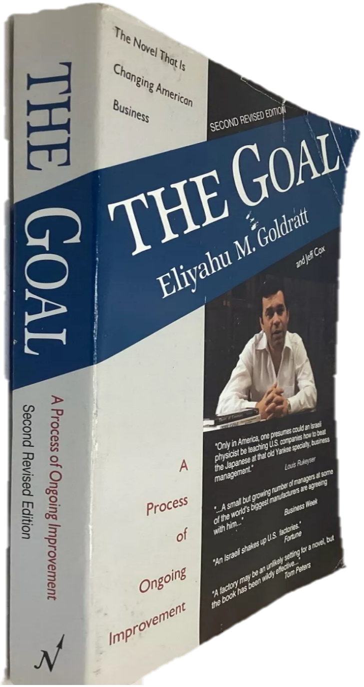
1,191 tech companies w/ layoffs · **263,180** employees laid off · In 2023

470 tech companies w/ layoffs · **141,145** employees laid off · In 2024

569,594




The Phoenix Project
The Phoenix Project
A Novel About IT, DevOps, and Helping Your Business Win
Gene Kim, Kevin Behr and George Spafford


THE GOAL
A Process of Ongoing Improvement
Second Revised Edition
N
The Novel That Is Changing American Business
SECOND REVISED EDITION
THE GOAL
Eliyahu M. Goldratt
and Jeff Cox
A Process of Ongoing Improvement

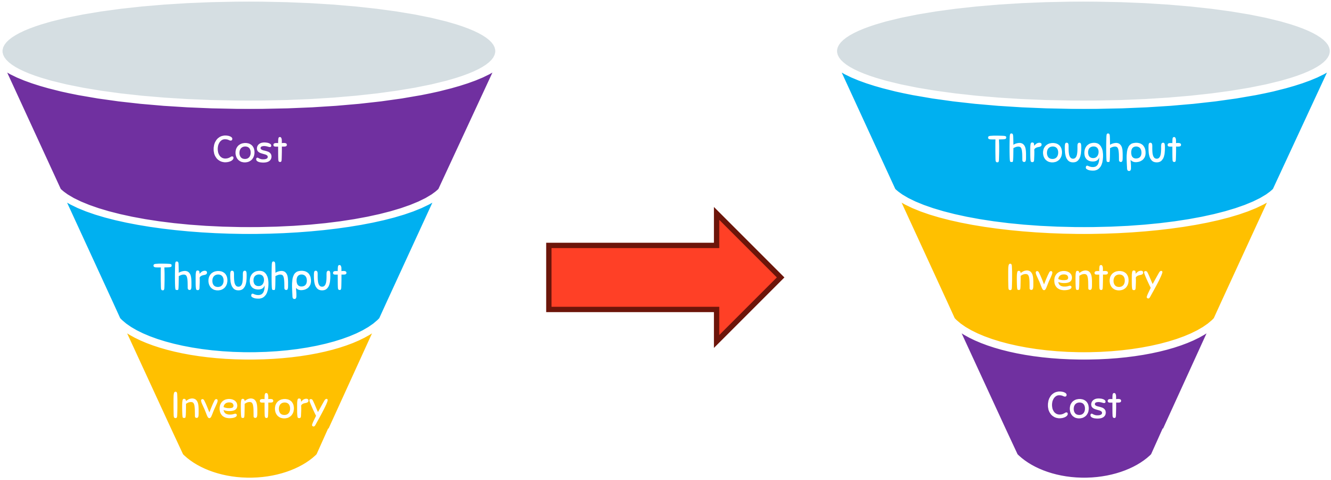
"Only in America, one presumes could an Israeli physicist be teaching U.S. companies how to beat the Japanese at that old Yankee specialty, business management."
Louis Rukeyser

"...A small but growing number of managers at some of the world's biggest manufacturers are agreeing with him..."
Business Week

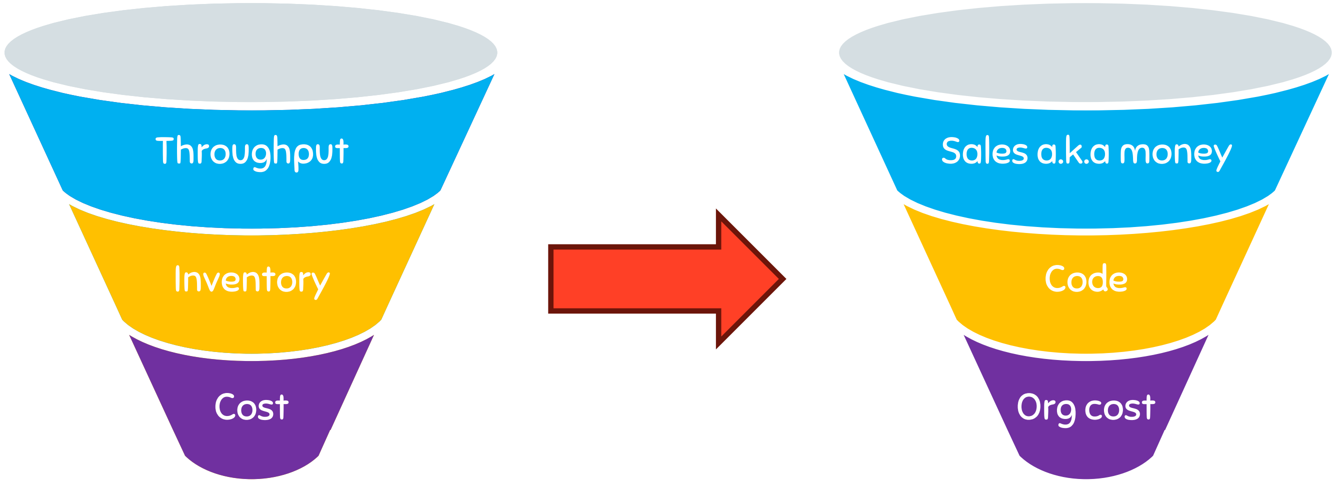
"An Israeli shakes up U.S. factories."
Fortune

"A factory may be an unlikely setting for a novel, but the book has been wildly effective..."
Tom Peters

FROM COST WORLD TO THROUGHPUT WORLD



IN TERMS OF IT COMPANY

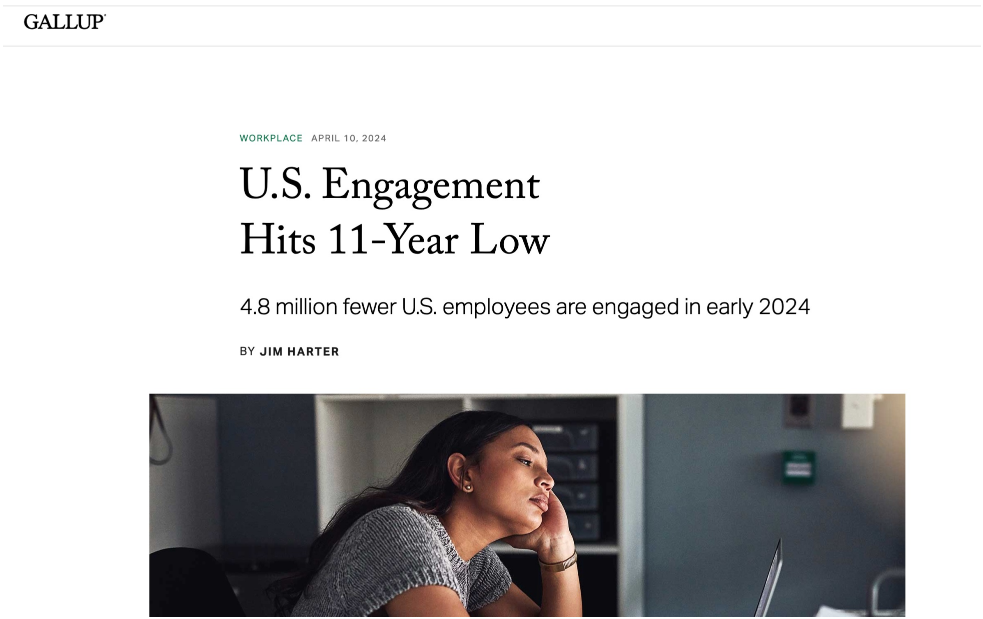


THE ONLY INITIATIVES THAT WILL POSITIVELY IMPACT PERFORMANCE ARE ONES THAT INCREASE THROUGHPUT WHILE SIMULTANEOUSLY DECREASING COST

CUTING COSTS (LAYOFFS) WITHOUT INCREASING THROUPUT BY MOVING INVENTORY (CREATING AND SELLING MORE SOFTWARE)

IS USELESS

BUT WHAT ABOUT SILENT QUITTING?



GALLUP

WORKPLACE APRIL 10, 2024

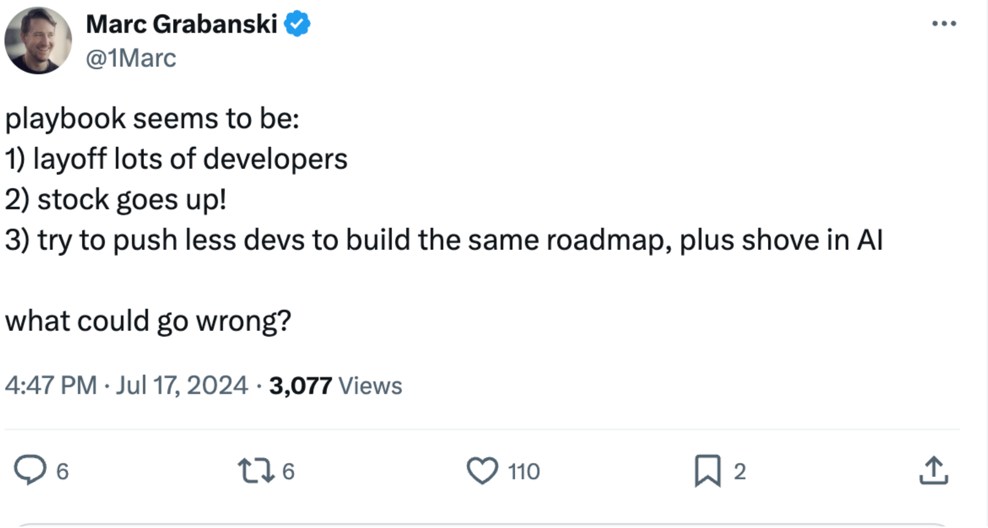
U.S. Engagement Hits 11-Year Low

4.8 million fewer U.S. employees are engaged in early 2024

BY JIM HARTER

But what about silent quitting?

YOU'D EXPECT SELECTIVE LAYOFFS THEN?



Marc Grabanski
@1Marc

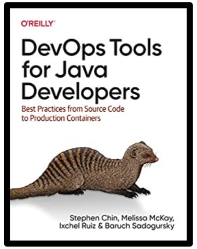
playbook seems to be:
1) layoff lots of developers
2) stock goes up!
3) try to push less devs to build the same roadmap, plus shove in AI

what could go wrong?

4:47 PM · Jul 17, 2024 · **3,077** Views

BARUCH SADOGURSKY - @JBARUCH

- Developer Productivity Advocate
- Gradle Inc
- Development -> DevOps -> #DPE

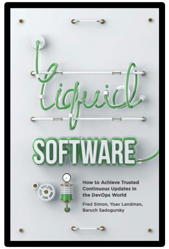


O'REILLY

DevOps Tools for Java Developers

Best Practices from Source Code to Production Containers

Stephen Chin, Melissa McKay, Ixchel Ruiz & Baruch Sadogursky



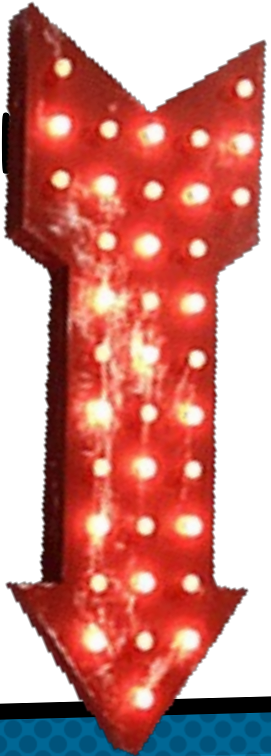
Liquid SOFTWARE

How to Achieve Trusted Continuous Updates in the DevOps World

Fred Simon, Yoav Landman, Baruch Sadogursky

SHOWNOTES

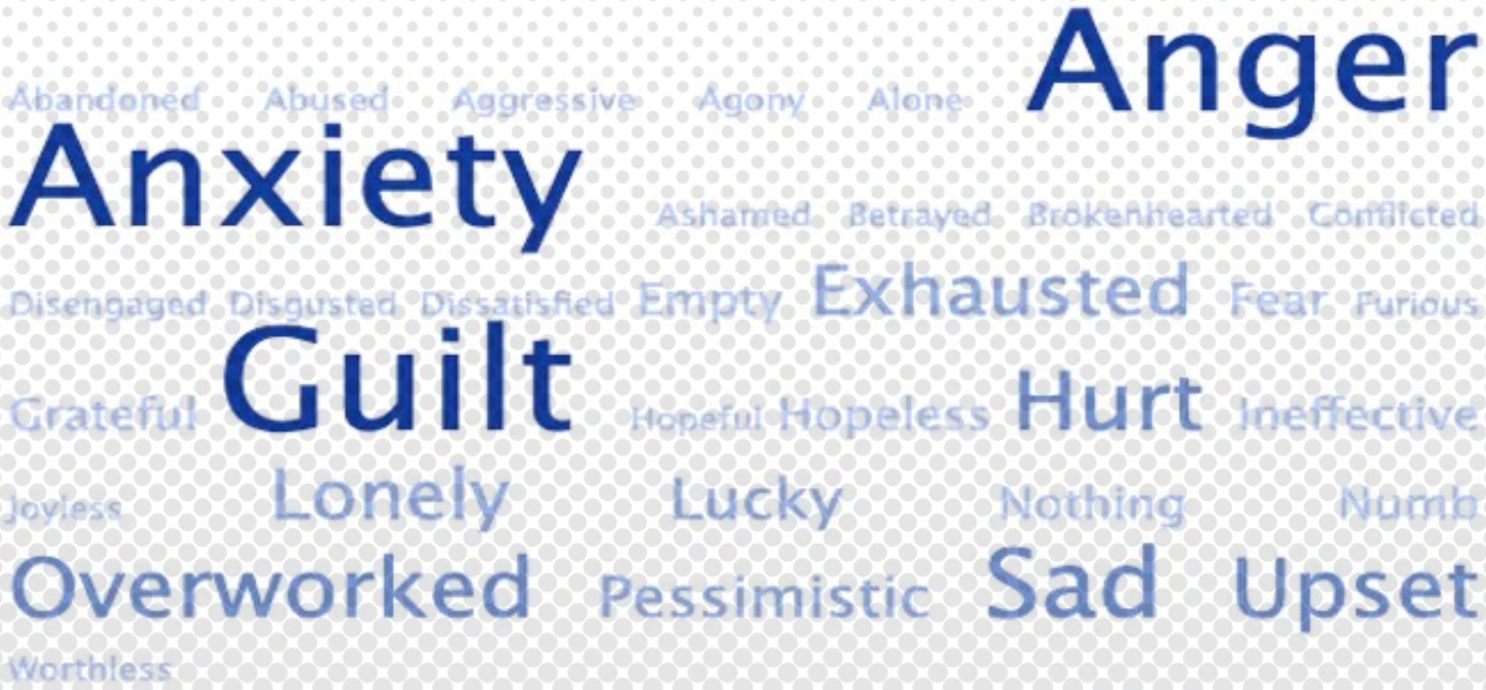
- speaking.jbaru.cl
- Slides
- Video
- All the links!



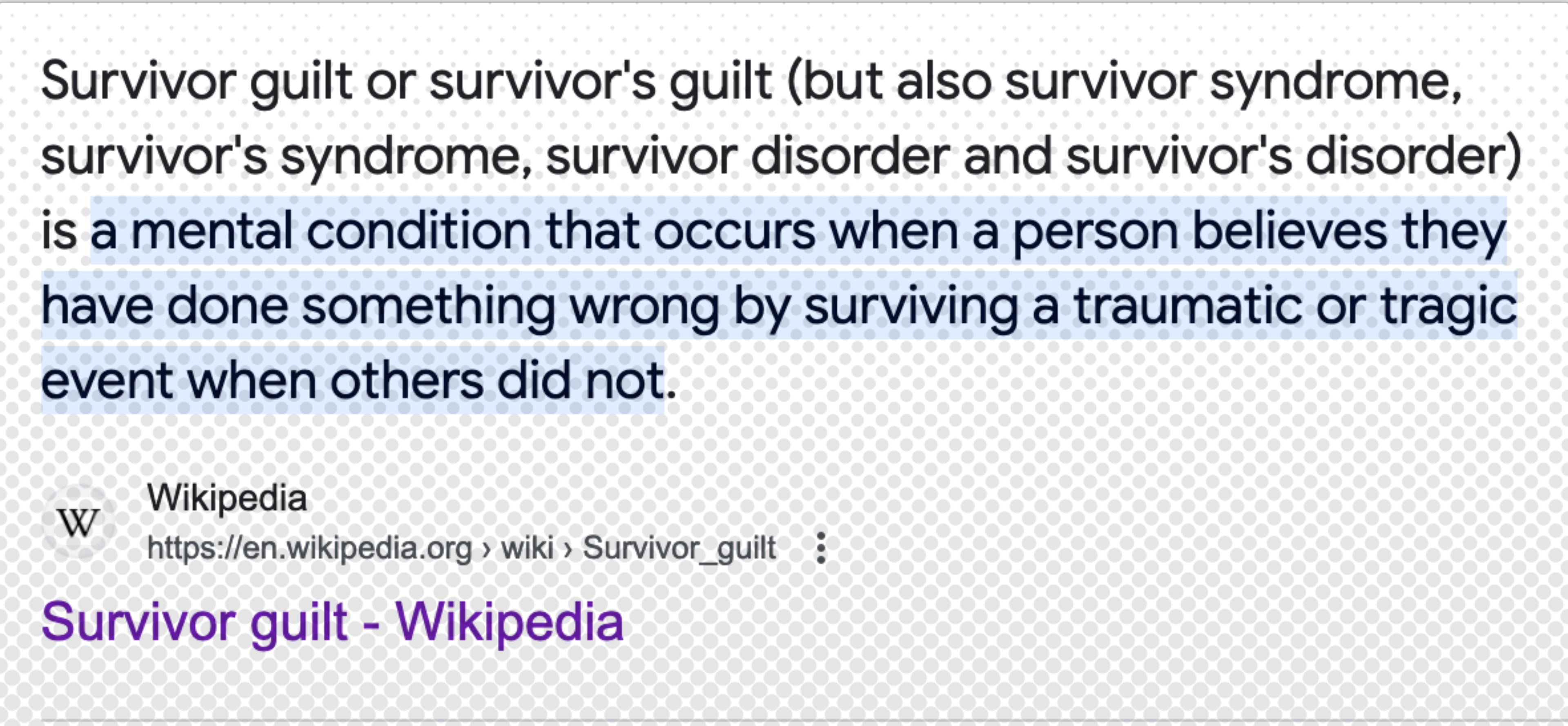
HUMAN RESOURCES ATTRITION

	VOLUNTARY	INVOLUNTARY
REGRETTABLE	– I quit – That's a bummer!	"The leadership takes full responsibility" 🤡
NON-REGRETTABLE	– I quit – Good riddance	– You're fired

Since the layoffs, how how would you describe your personal feelings?



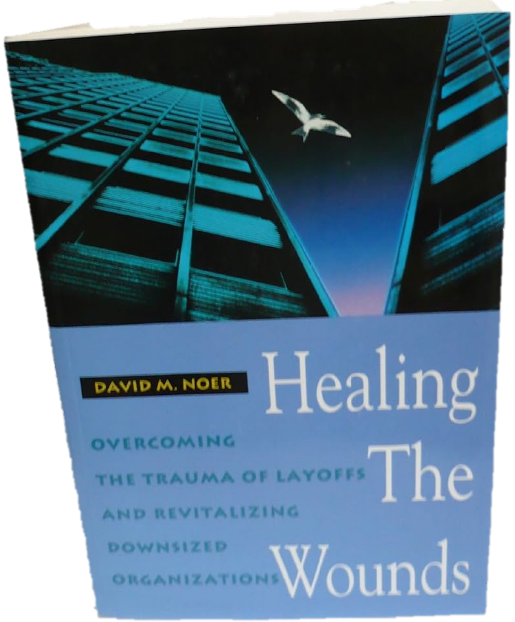
Source: LEADERSHIP IQ



Survivor guilt or survivor's guilt (but also survivor syndrome, survivor's syndrome, survivor disorder and survivor's disorder) is a mental condition that occurs when a person believes they have done something wrong by surviving a traumatic or tragic event when others did not.

Wikipedia
https://en.wikipedia.org › wiki › Survivor_guilt

Survivor guilt - Wikipedia



DAVID M. NOER

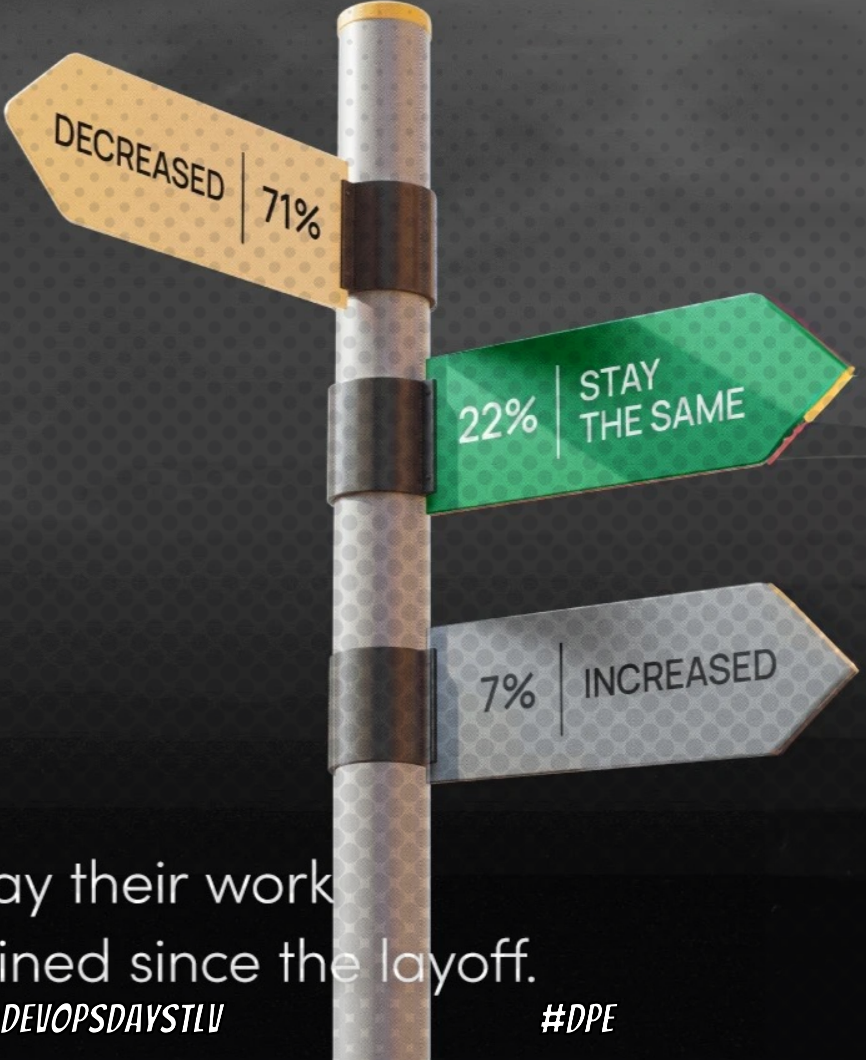
Healing The Wounds

OVERCOMING THE TRAUMA OF LAYOFFS AND REVITALIZING DOWNSIZED ORGANIZATIONS

LAYOFF SURVIVOR SICKNESS

Bizreport

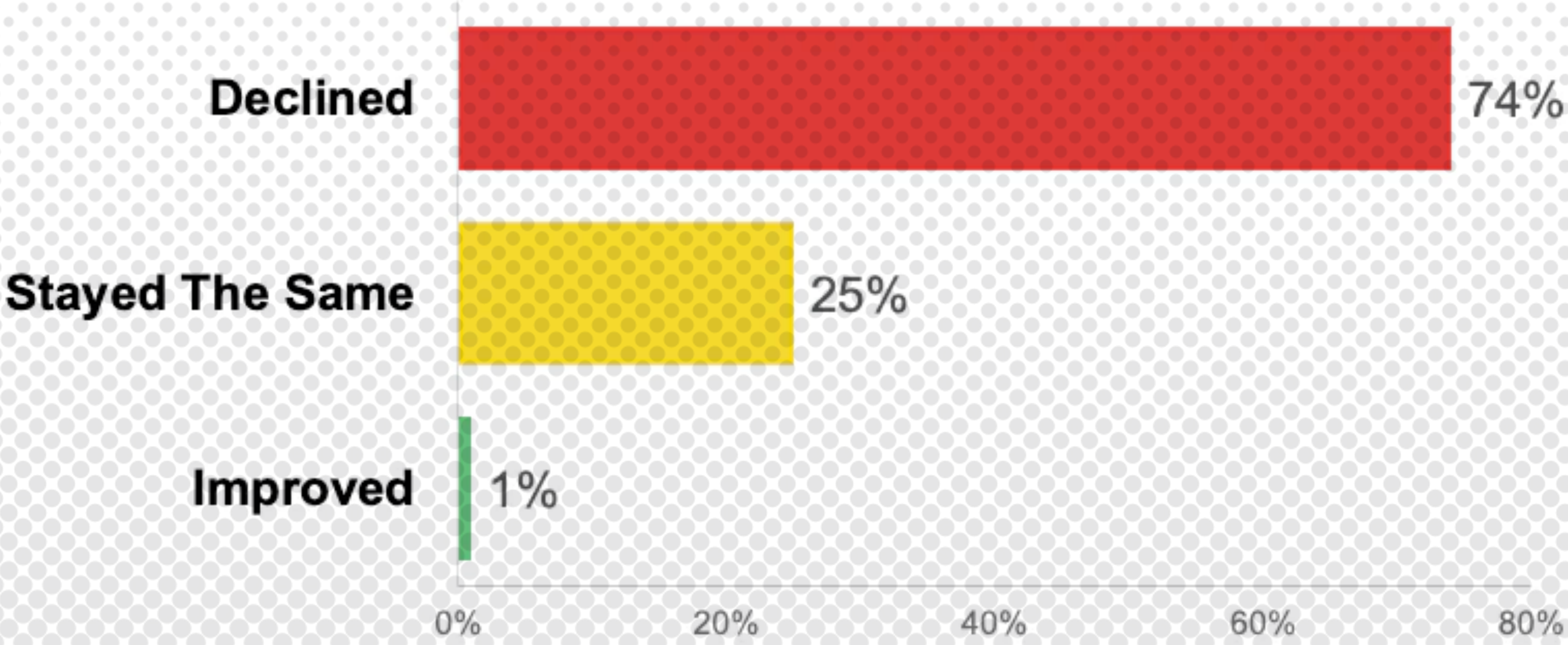
Data from a survey conducted by BizReport on November 22–27 of 2162 decision-makers in layoffs, layoff victims, and layoff survivors



71%

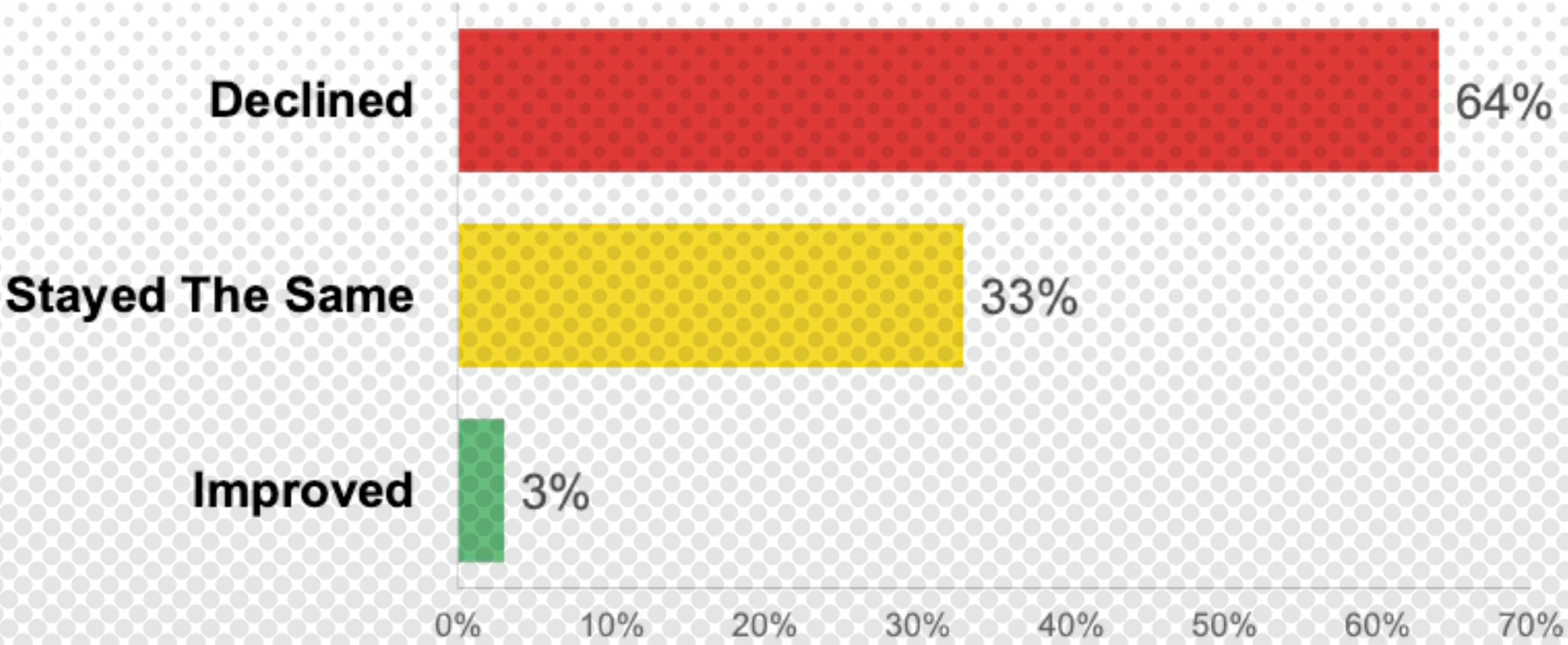
of layoff survivors say their work motivation has declined since the layoff.

Since the layoffs, my personal productivity has...



Source: LEADERSHIP IQ

Since the layoffs, from what I've seen, the productivity of my coworkers has...



Source: LEADERSHIP IQ

Bizreport

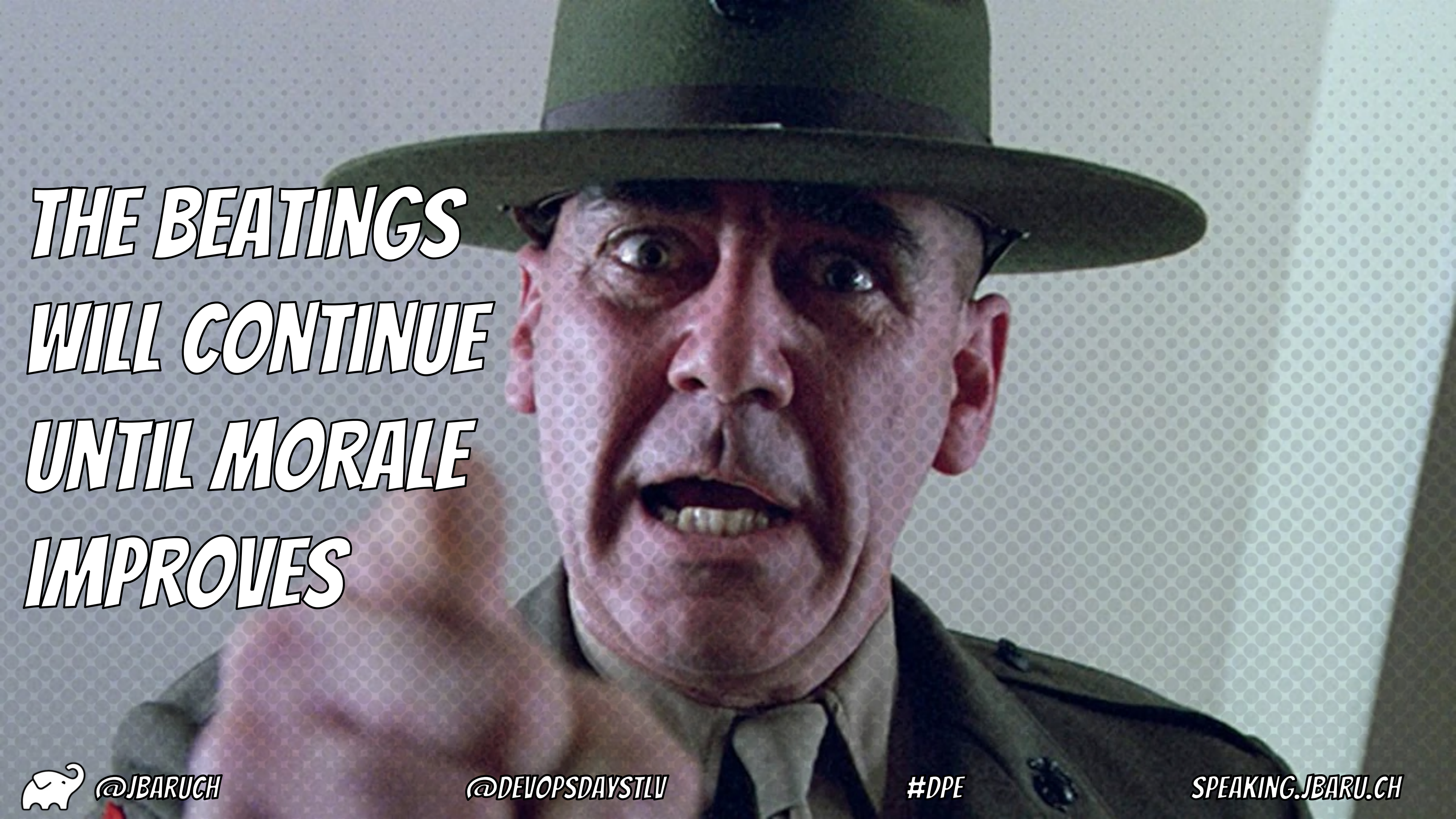
Data from a survey conducted by BizReport on November 22–27 of 2162 decision-makers in layoffs, layoff victims, and layoff survivors

65%

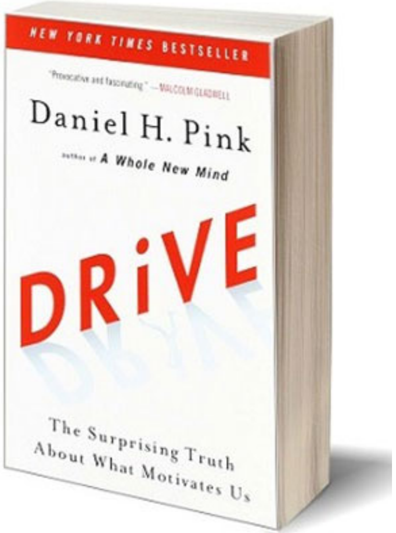
of layoff survivors say they are overworked since the layoff.

"LEAN AND MEAN"? MORE LIKE SAD AND ANGRY

The
Show
Must
Go
On



THE BEATINGS WILL CONTINUE UNTIL MORALE IMPROVES



NEW YORK TIMES BESTSELLER

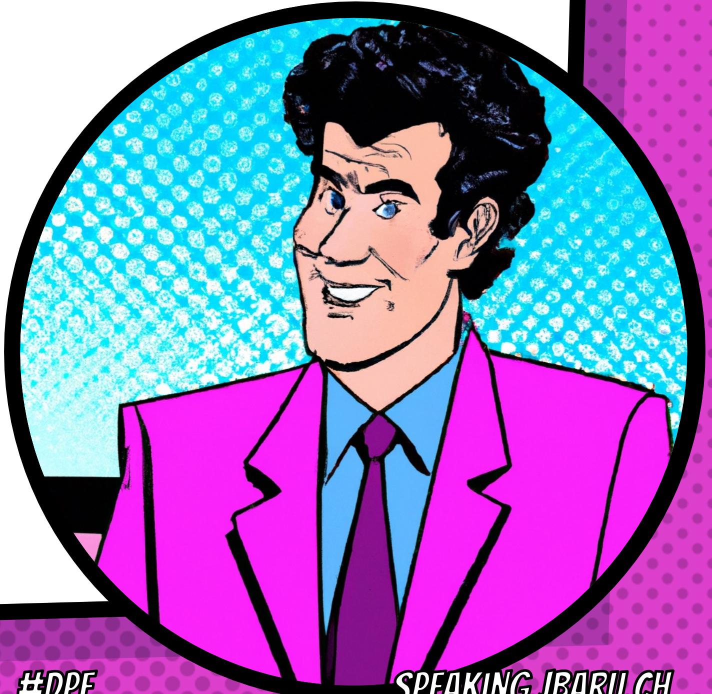
"Provocative and fascinating." —MALCOLM GLADWELL

Daniel H. Pink

author of *A Whole New Mind*

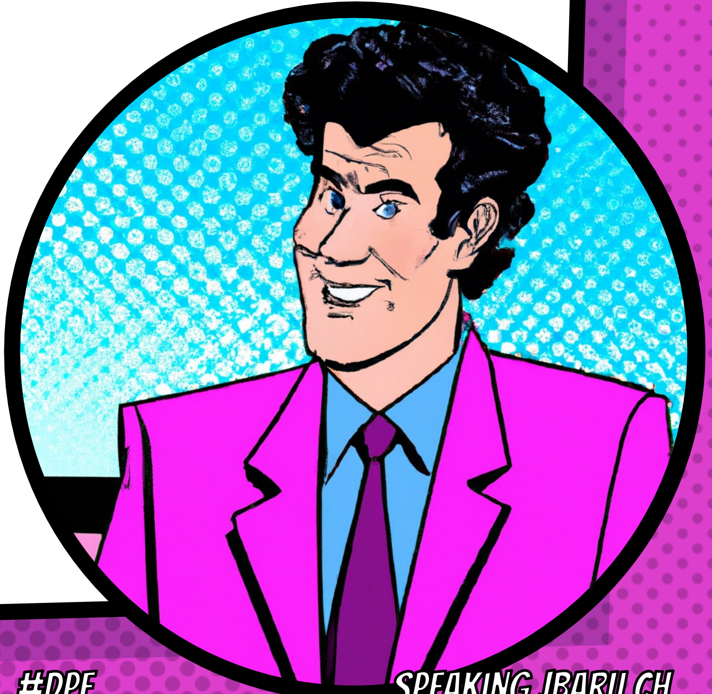
DRiVE

The Surprising Truth About What Motivates Us



3 FACTORS LEAD TO BETTER PERFORMANCE & PERSONAL SATISFACTION...

AUTONOMY
MASTERY
PURPOSE

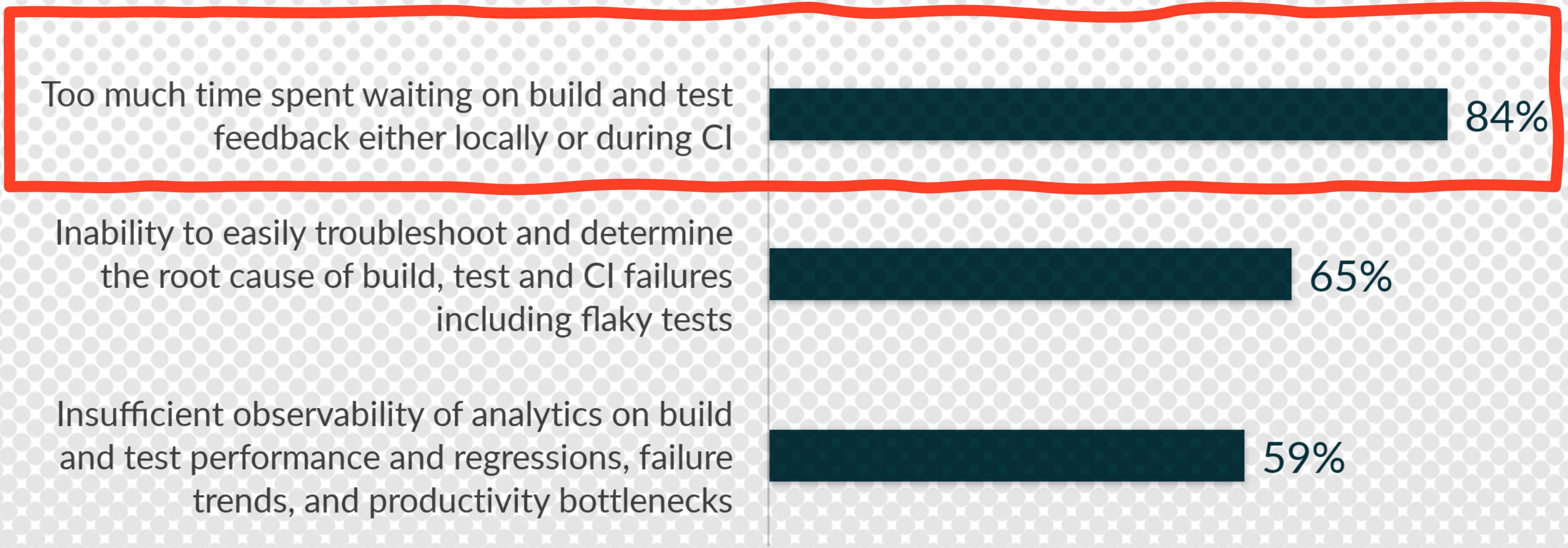


DON'T RUIN THE FLOW

"THE BUILD TAKES FOREVER, I AM DISTRACTED TO DO OTHER THINGS AND THE CONTEXT SWITCH IS TERRIBLE"

Development Pains are Widespread

Which of the following challenges or pain points did your organization experience prior to implementing Developer Productivity Engineering?

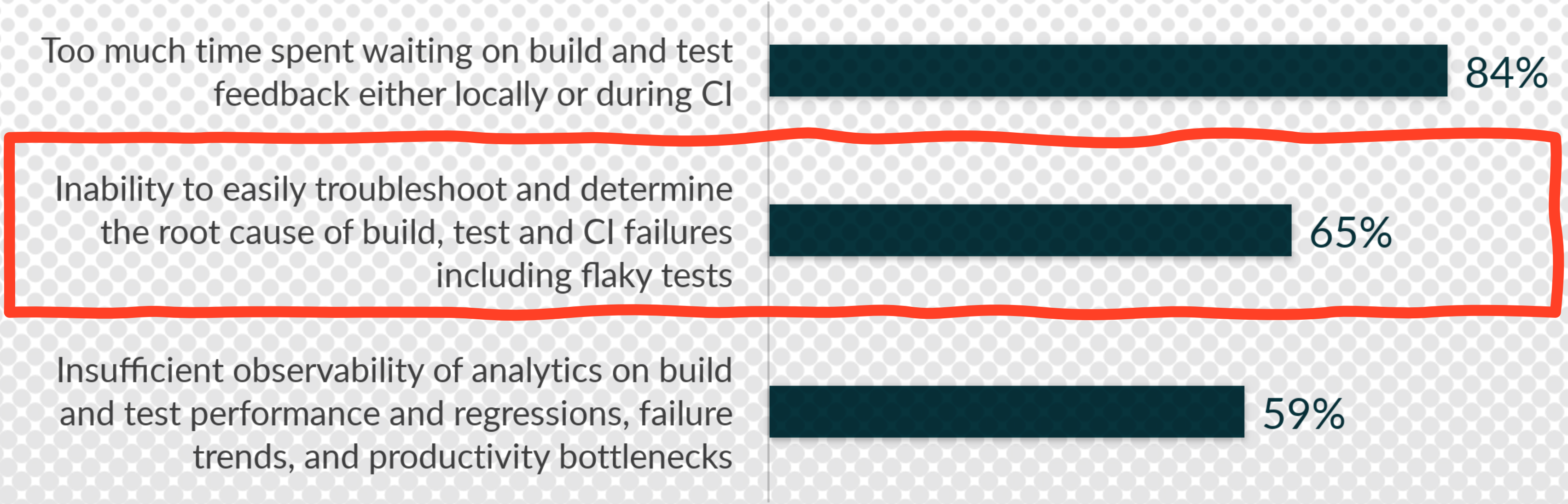


DON'T FRUSTRATE THE DEVELOPERS

"WE HAVE A FLANKY TEST, IT IS IRRELEVANT 99.5% OF THE TIME , BUT IT ALWAYS RUNS AND IT IS LAST IN THE SUITE"

Development Pains are Widespread

Which of the following challenges or pain points did your organization experience prior to implementing Developer Productivity Engineering?

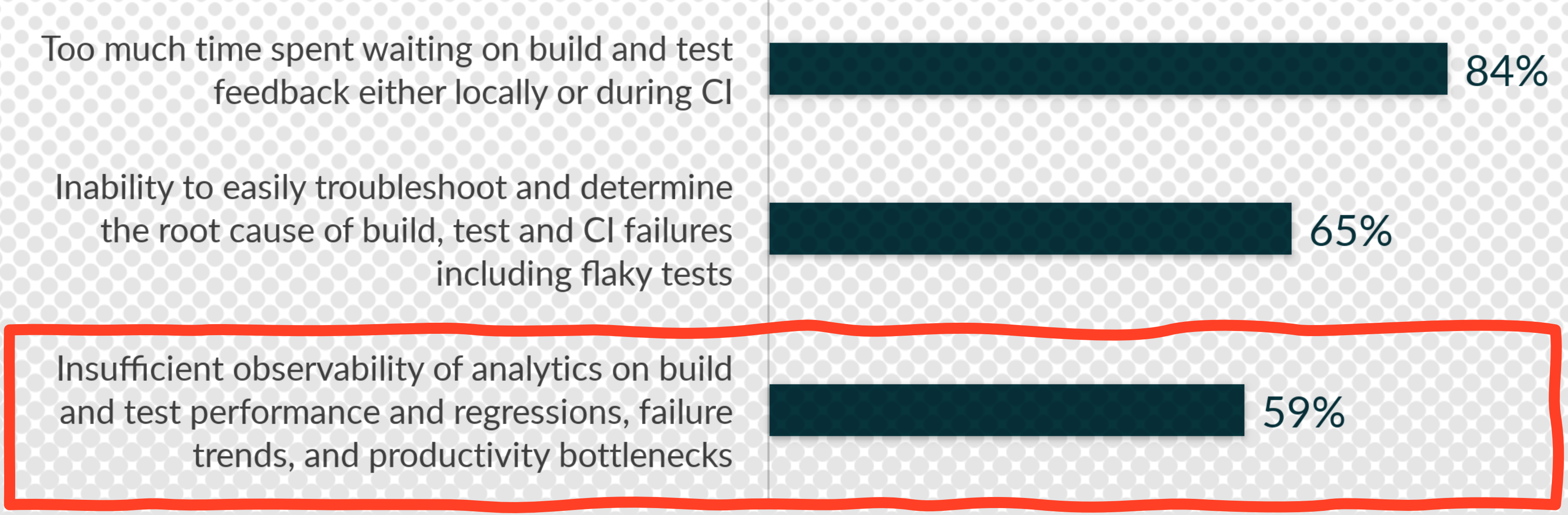


DON'T BOIL THE FROG

"I HAVE A FEELING THAT EVERYTHING IS SLOWER SOMEHOW..."

Development Pains are Widespread

Which of the following challenges or pain points did your organization experience prior to implementing Developer Productivity Engineering?



DEVELOPER PRODUCTIVITY :: A/M/P :: MOTIVATION

Autonomy → Tools and people aren’t in my way

Mastery → Tools and processes help me to excel

Purpose → I want to be *productive*, i.e. create the product

DEVELOPER PRODUCTIVITY ENGINEERING!

DEVELOPER PRODUCTIVITY ENGINEERING

Foster Faster Feedback

Collaborate through Effective Tooling

Embrace Rigorous Observability for Proactive Improvement

Eliminate Toil for Developers

Prioritize Automation and Eliminate Bottlenecks

Dedicated Organizational Mindset

Outcomes Over Output

TALK IS CHEAP, SHOW ME THE GOODS!

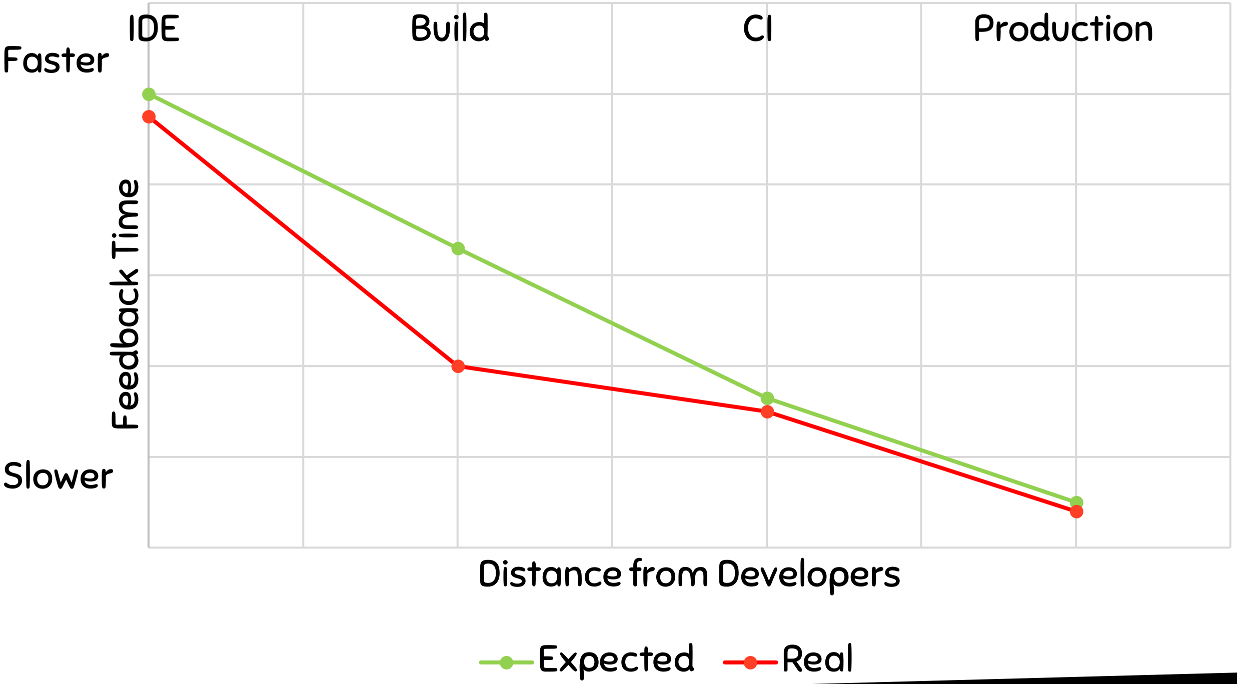
SMALL DPE IMPROVEMENTS MAKE A HUGE DIFFERENCE

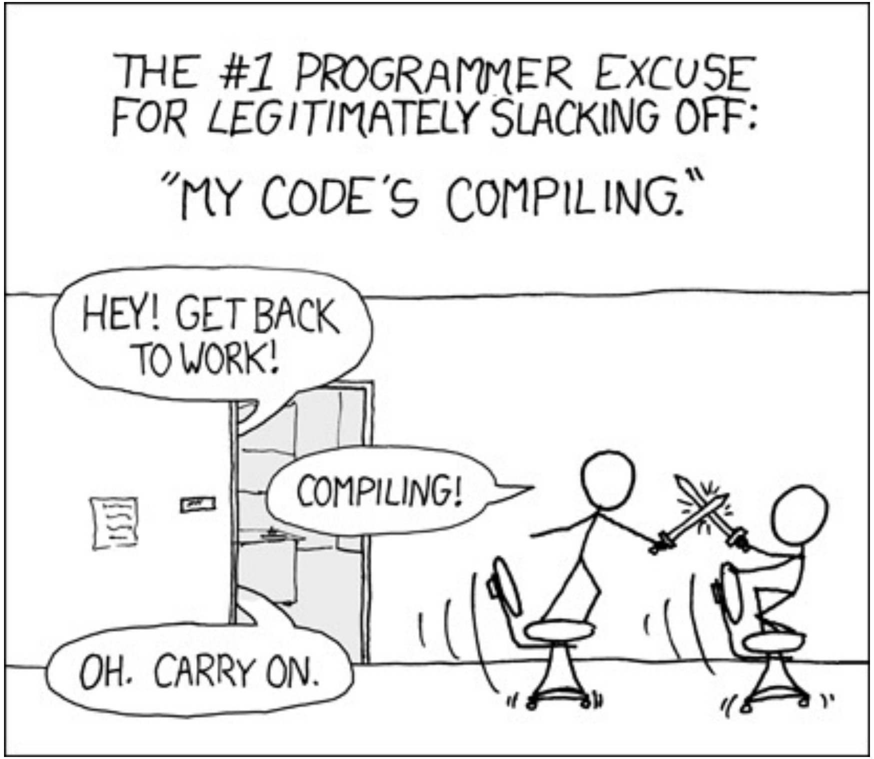
- Generate code faster: Better IDE
- Test better: Testcontainers
- Enforce better code: Sonar
- Test more reliably: Flaky test detection
- Foster Faster Feedback:

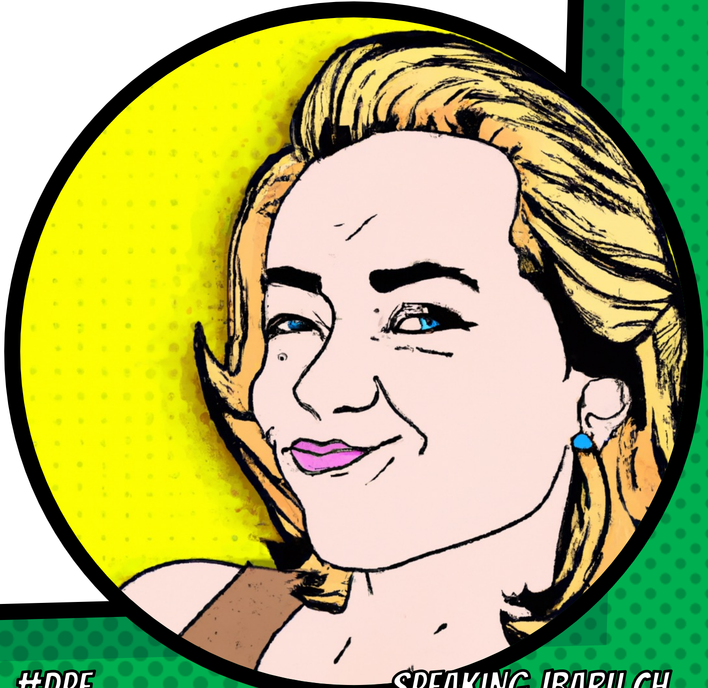
Feedback Efficiency

- IDE: Sub-seconds (I type, it marks it red)
- Build: Seconds
- CI: Minutes
- Production: Hours/Days

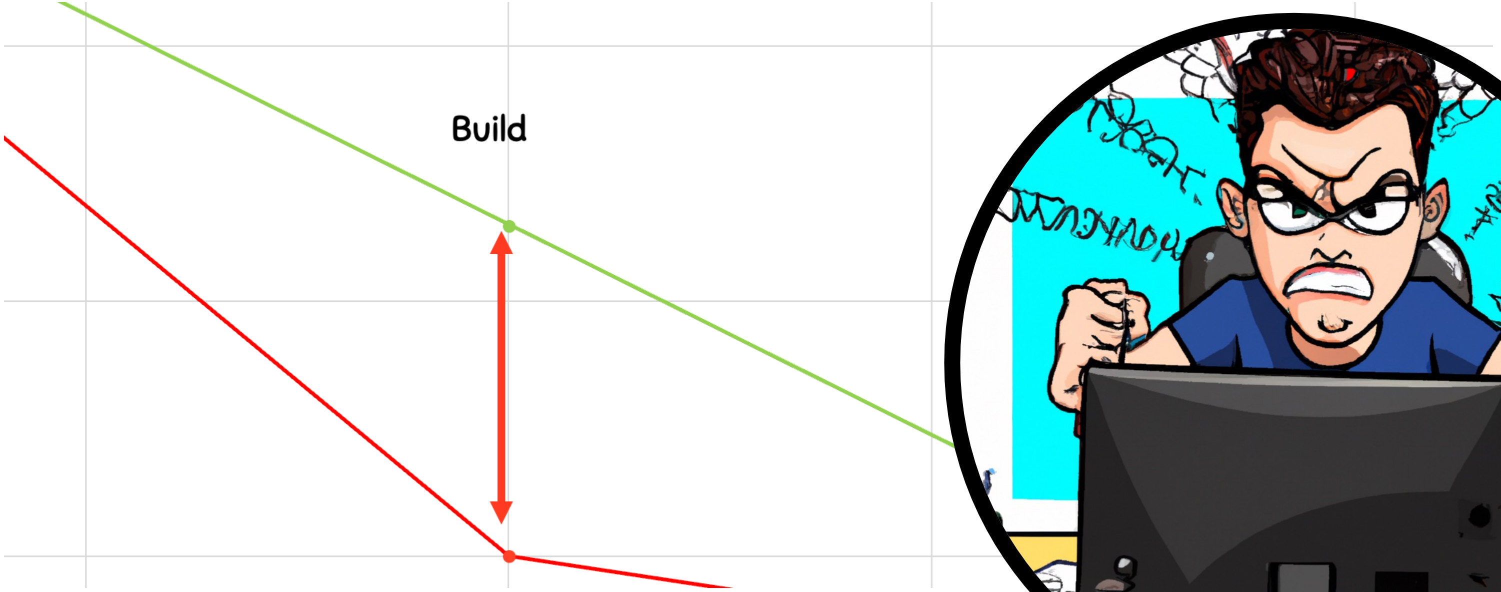
Reverse Dependency on Distance from Developers

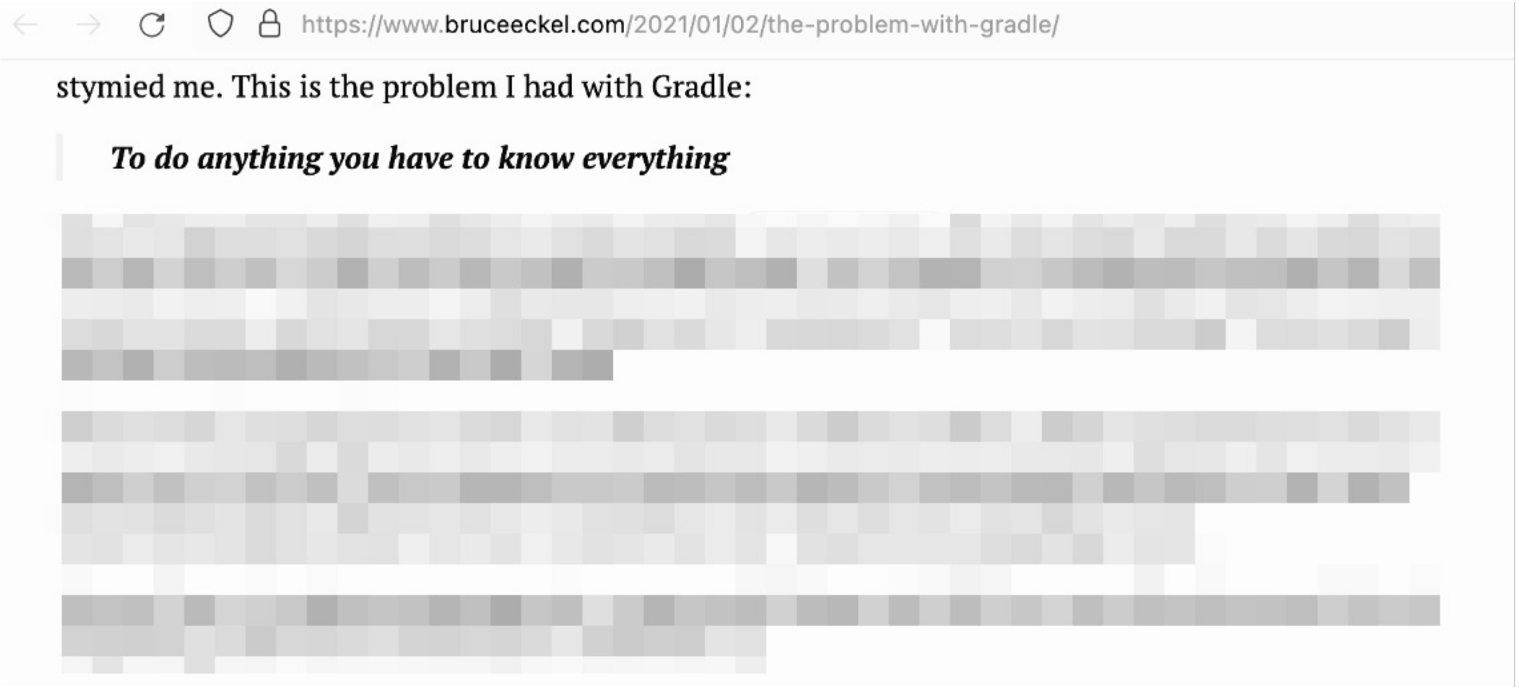



THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:
"MY CODE'S COMPILING."
HEY! GET BACK TO WORK!
COMPILING!
OH. CARRY ON.



IT IS SLOW!





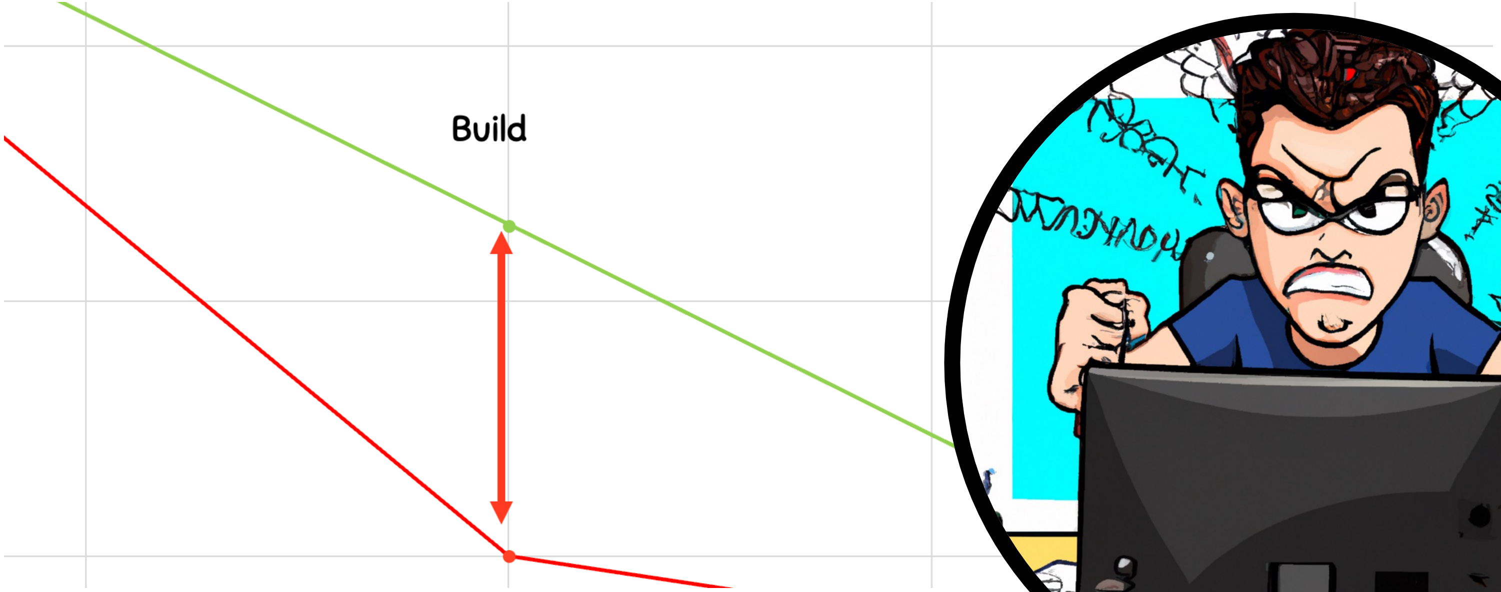
https://www.bruceeckel.com/2021/01/02/the-problem-with-gradle/

stymied me. This is the problem I had with Gradle:

> ***To do anything you have to know everything***

[redacted]

IT IS SLOW AND THE DEVELOPERS HAVE NO IDEA WHY!



WHAT IS BUILD?

- Project setup
- Downloading the Internet
- Artifact generation: Compilation, packaging, etc
- Tests
- Artifact deployment

WHAT CAN GO WRONG?

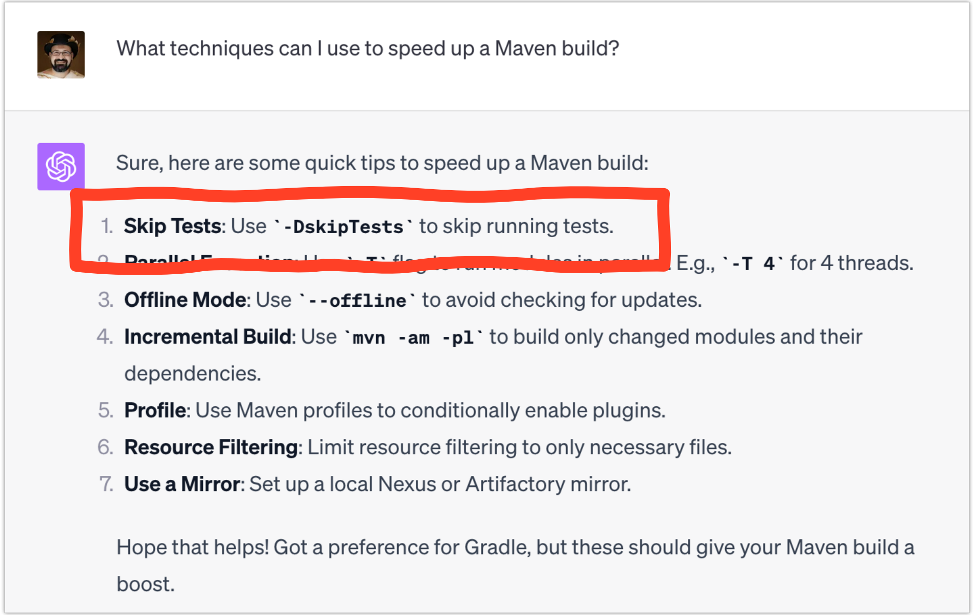
- Project setup
- Downloading the Internet
- Artifact generation: Compilation, packaging, etc
- Tests
- Artifact deployment

WHEN CAN IT GO WRONG?

AT ANY TIME.

THE BUILD FRUSTRATES THE DEVELOPERS

LET'S ASK CHATGPT



What techniques can I use to speed up a Maven build?

Sure, here are some quick tips to speed up a Maven build:

1. **Skip Tests**: Use `-DskipTests` to skip running tests.
2. **Parallel Execution**: Use `-T` flag to run modules in parallel. E.g., `-T 4` for 4 threads.
3. **Offline Mode**: Use `--offline` to avoid checking for updates.
4. **Incremental Build**: Use `mvn -am -pl` to build only changed modules and their dependencies.
5. **Profile**: Use Maven profiles to conditionally enable plugins.
6. **Resource Filtering**: Limit resource filtering to only necessary files.
7. **Use a Mirror**: Set up a local Nexus or Artifactory mirror.

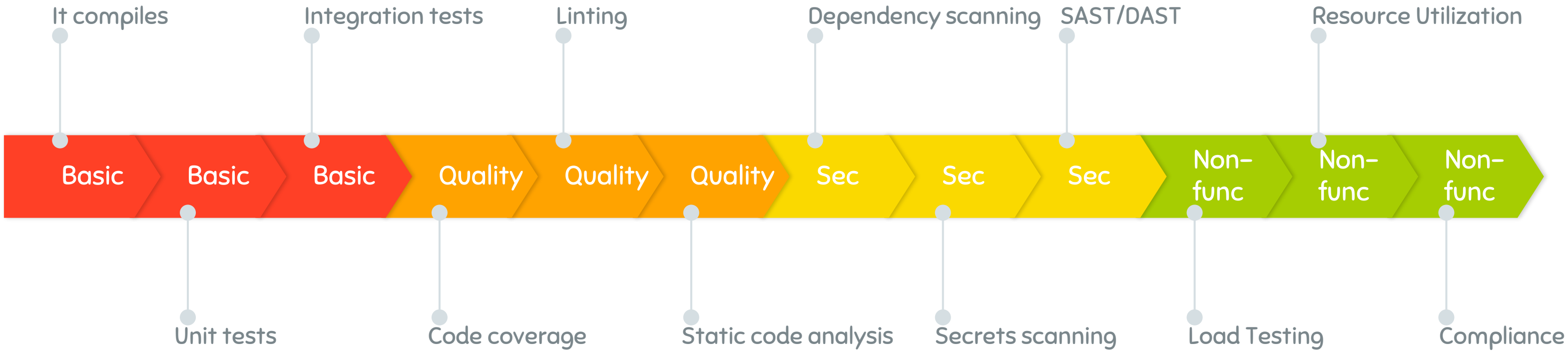
Hope that helps! Got a preference for Gradle, but these should give your Maven build a boost.

WHAT THE ACTUAL F*CK?!

- Skipping tests defeats the purpose of the build!
- How about skipping compilation?
- We want faster feedback, not less feedback

WHAT FEEDBACK DO WE WANT?

CI/CD PIPELINE QUALITY GATES



It compiles

Integration tests

Linting

Dependency scanning

SAST/DAST

Resource Utilization

Basic Basic Basic Quality Quality Quality Sec Sec Sec Non-func Non-func Non-func

Unit tests

Code coverage

Static code analysis

Secrets scanning

Load Testing

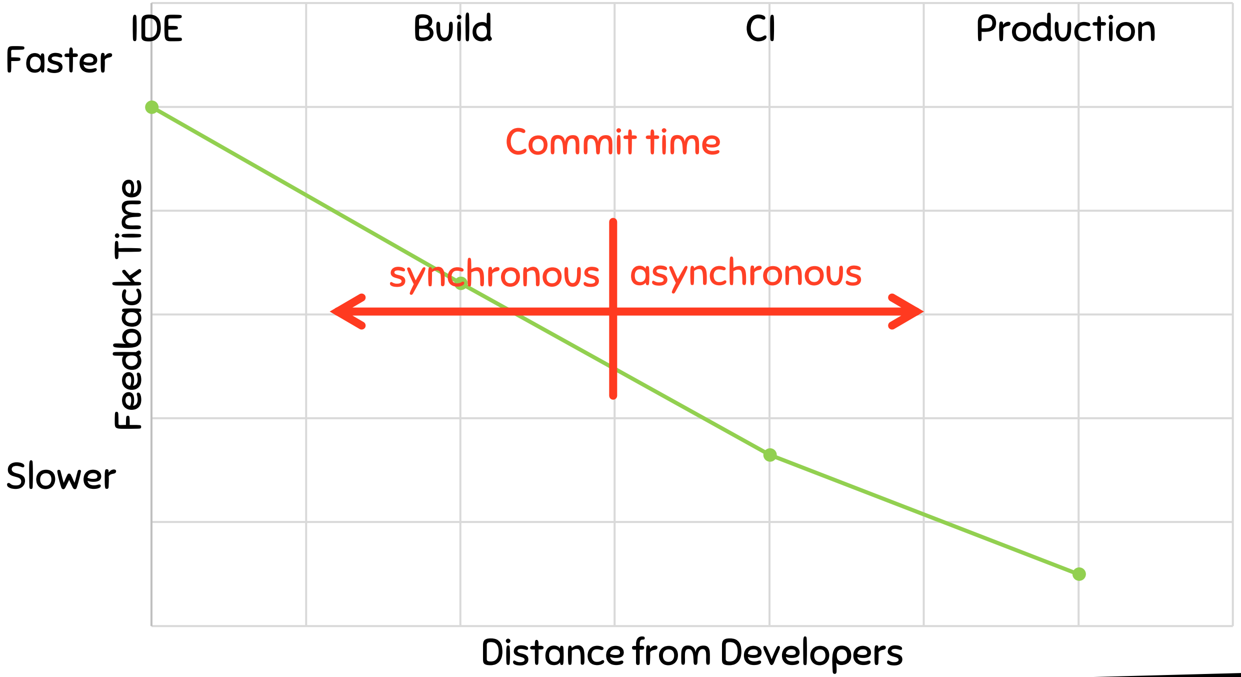
Compliance

CI/CD? NOT GREAT, NOT TERRIBLE.

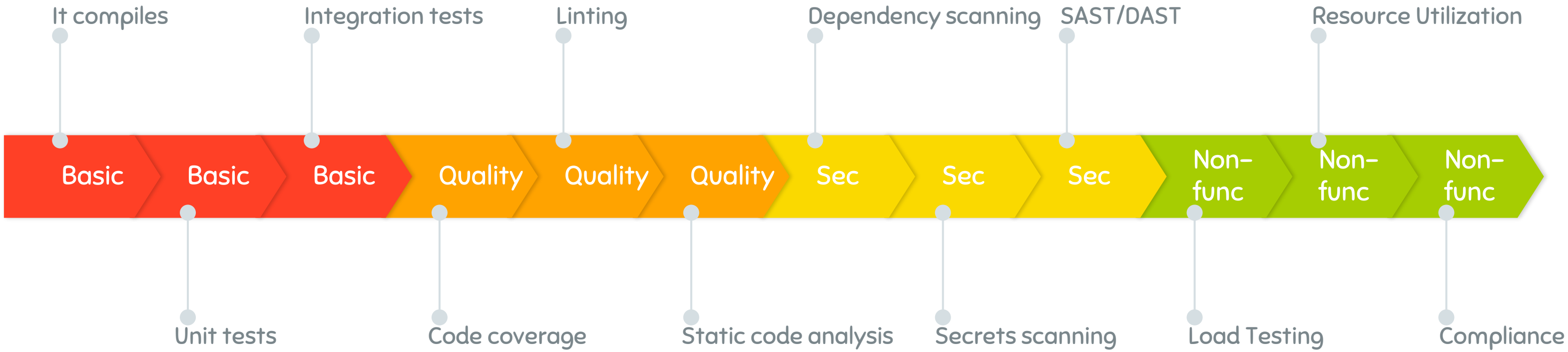
TWO TYPES OF FEEDBACK

ASYNCHRONOUS	× e.g., CI/CD × we never wait for it × results are distracting
SYNCHRONOUS	× e.g., build × we'll wait for it in the flow × we'll be pissed off when it's slow

REVERSE DEPENDENCY ON DISTANCE FROM DEVELOPERS



Ideal Build Time Feedback



Basic
It compiles

Basic
Unit tests

Basic
Integration tests

Quality
Code coverage

Quality
Linting

Quality
Static code analysis

Sec
Dependency scanning

Sec
Secrets scanning

Sec
SAST/DAST

Non-func
Load Testing

Non-func
Resource Utilization

Non-func
Compliance

BUT WON'T IT SLOW DOWN THE BUILD?!

DELIGHTFUL BUILD (PICK TWO):

☑ PROVIDES MAX FEEDBACK
☑ FAST

SKIP WHAT CAN BE SKIPPED (BUT NO MORE!)

AVOIDANCE: INCREMENTAL BUILD

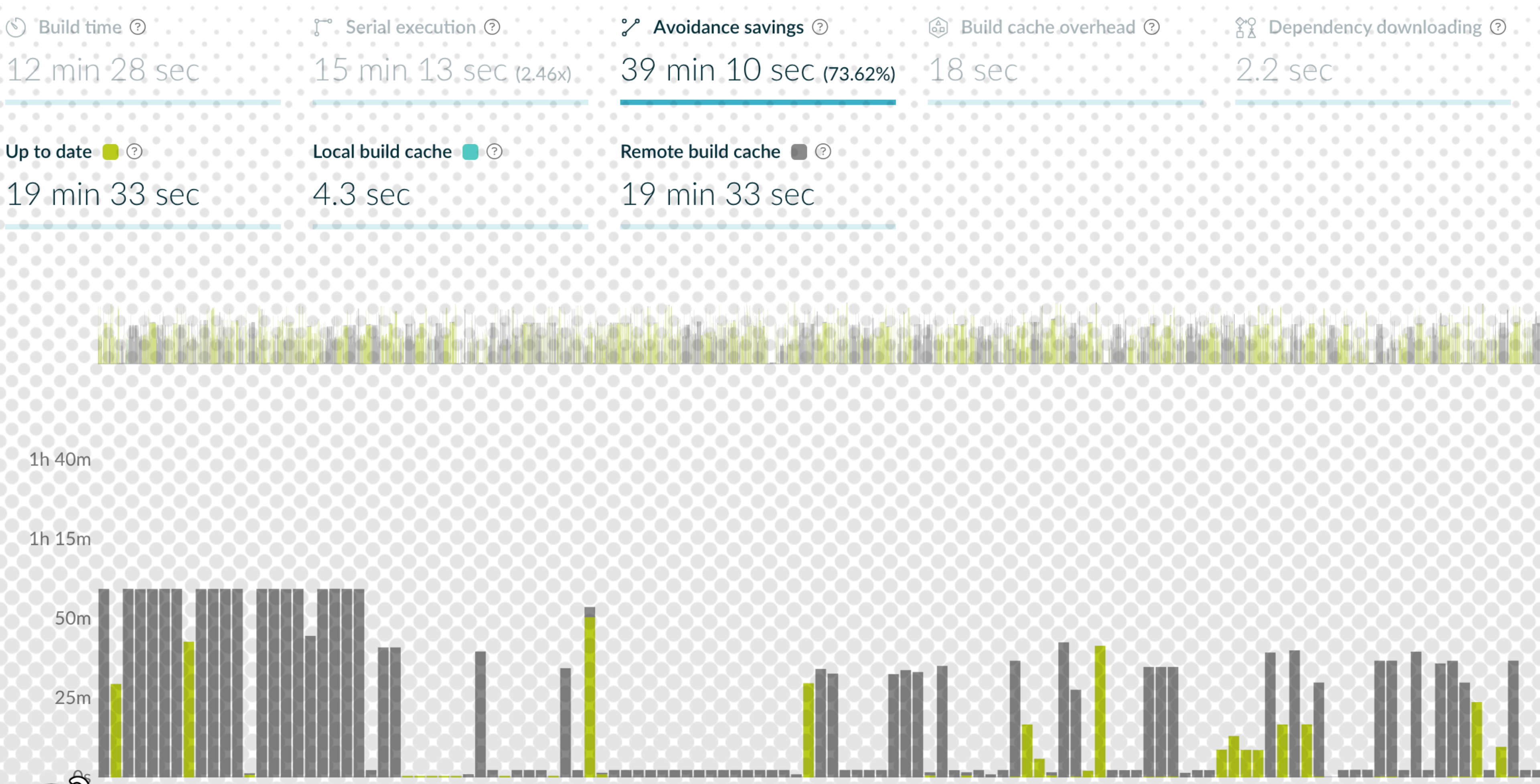
- Don’t build what didn’t changed
- Don’t build what isn’t affected

AVOIDANCE: INCREMENTAL BUILD SHORTCOMINGS

- × Relies on produced artifacts
- × Relies on architectural decisions

AVOIDANCE: CACHING

- Makes the build faster
- Makes the build faster for everybody
- Makes the build faster always
- Makes all parts of the build faster



Build time ⓘ
12 min 28 sec

Serial execution ⓘ
15 min 13 sec (2.46x)

Avoidance savings ⓘ
39 min 10 sec (73.62%)

Build cache overhead ⓘ
18 sec

Dependency downloading ⓘ
2.2 sec

Up to date 🟩 ⓘ
19 min 33 sec

Local build cache 🟦 ⓘ
4.3 sec

Remote build cache ■ ⓘ
19 min 33 sec

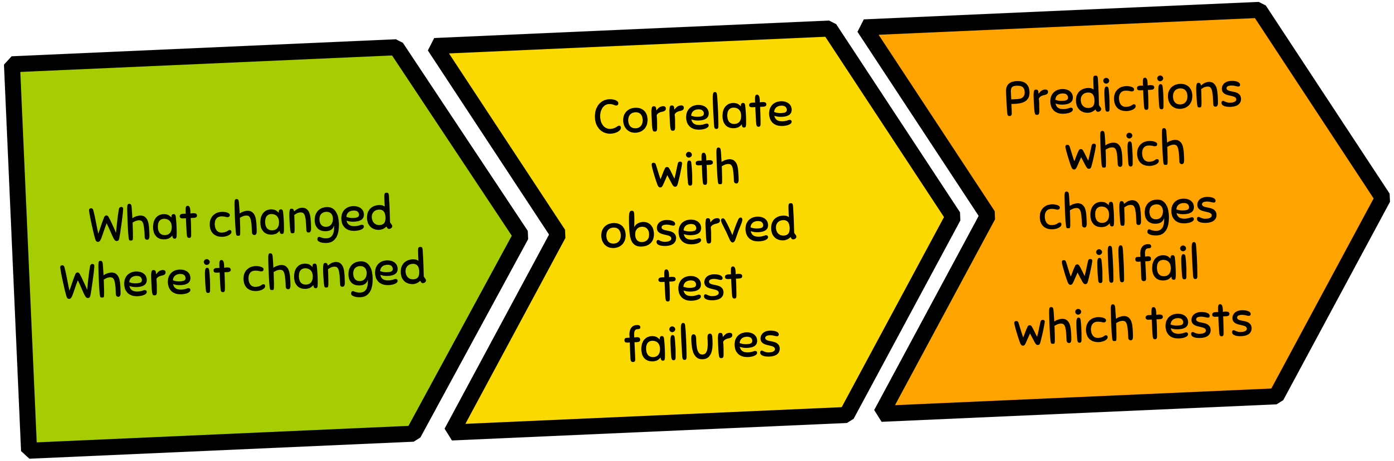
AVOIDANCE: PREDICTIVE TEST SELECTION

- Learns code changes effects de-facto
- Skips tests with high degree of confidence

HOW TEST PREDICTION WORKS

- Code changes and test results are thrown into learning model
- After a while, the model predicts which changes fail which tests

TEST PREDICTION



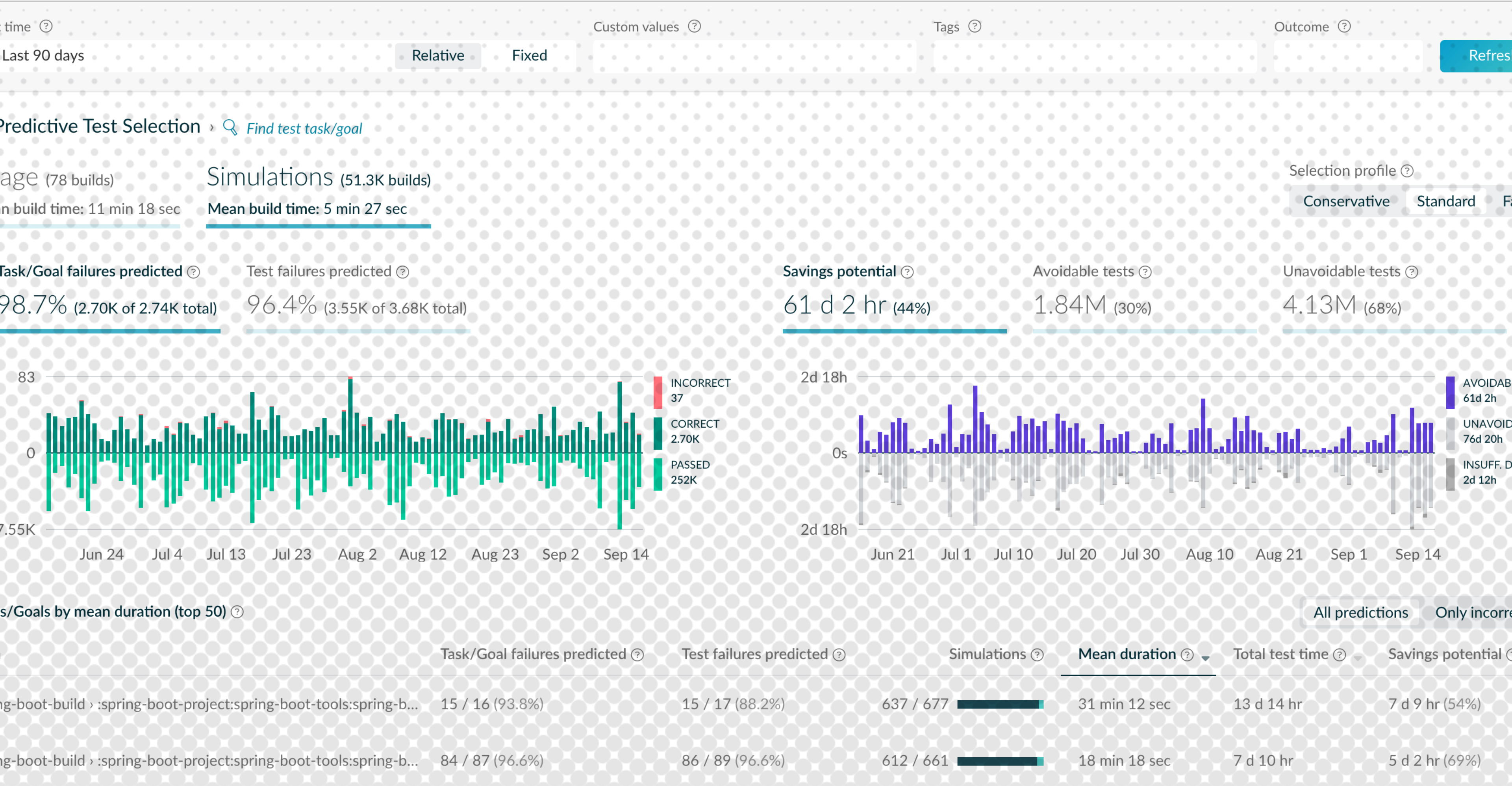
What changed
Where it changed

Correlate with observed test failures

Predictions which changes will fail which tests

BLACK MAGIC IN ACTION

- The more tests a project has, the less they break
- Refactorings in Java break tests less than in JavaScript



time ⓘ | Custom values ⓘ | Tags ⓘ | Outcome ⓘ

Last 90 days | Relative | Fixed | Refres

Predictive Test Selection › Find test task/goal

age (78 builds)
n build time: 11 min 18 sec

Simulations (51.3K builds)
Mean build time: 5 min 27 sec

Selection profile ⓘ
Conservative | Standard | Fa

Task/Goal failures predicted ⓘ
98.7% (2.70K of 2.74K total)

Test failures predicted ⓘ
96.4% (3.55K of 3.68K total)

Savings potential ⓘ
61 d 2 hr (44%)

Avoidable tests ⓘ
1.84M (30%)

Unavoidable tests ⓘ
4.13M (68%)

s/Goals by mean duration (top 50) ⓘ

All predictions | Only incorre

	Task/Goal failures predicted ⓘ	Test failures predicted ⓘ	Simulations ⓘ	Mean duration ⓘ	Total test time ⓘ	Savings potential ⓘ
g-boot-build › :spring-boot-project:spring-boot-tools:spring-b...	15 / 16 (93.8%)	15 / 17 (88.2%)	637 / 677	31 min 12 sec	13 d 14 hr	7 d 9 hr (54%)
g-boot-build › :spring-boot-project:spring-boot-tools:spring-b...	84 / 87 (96.6%)	86 / 89 (96.6%)	612 / 661	18 min 18 sec	7 d 10 hr	5 d 2 hr (69%)

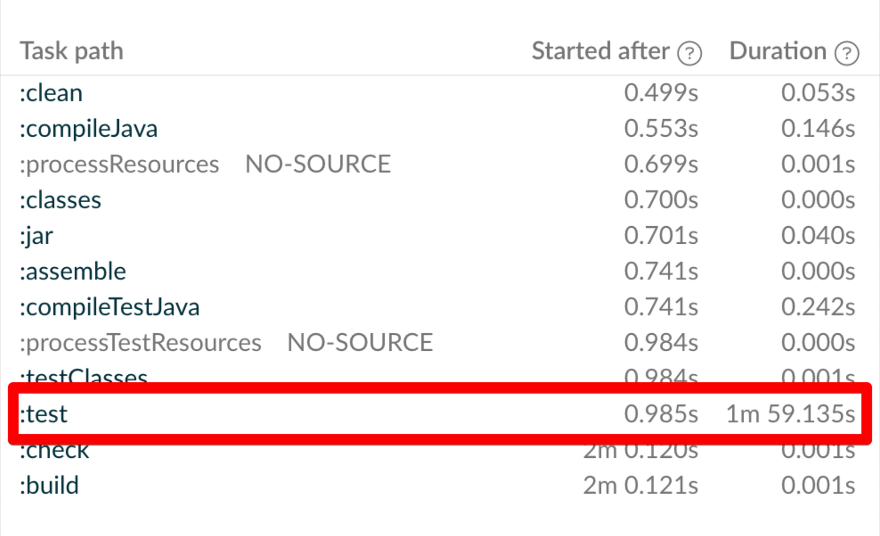
2.2 | © Gradle Inc. 2023 | Help and Feed

@JBARUCH @DEVOPSDAYSTLV #DPE SPEAKING.JBARU.CH

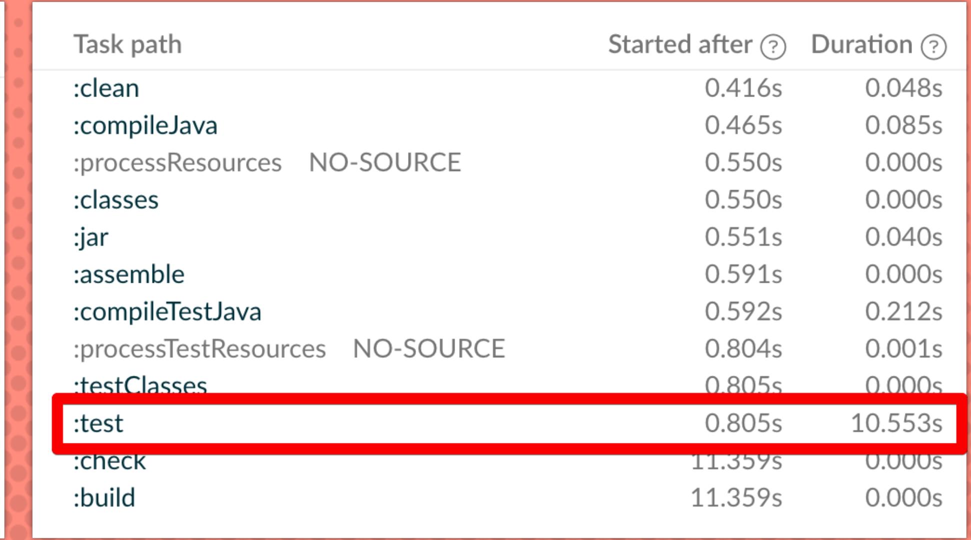
SPEED UP WHAT CAN'T BE SKIPPED

Test Parallelization

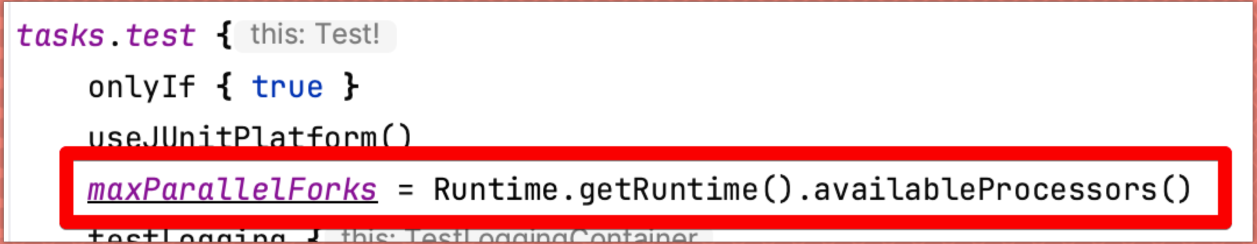
- Use max power of local machine
- (Yes, your boss should buy you the bleeding edge)



Task path		Started after ?	Duration ?
:clean		0.499s	0.053s
:compileJava		0.553s	0.146s
:processResources	NO-SOURCE	0.699s	0.001s
:classes		0.700s	0.000s
:jar		0.701s	0.040s
:assemble		0.741s	0.000s
:compileTestJava		0.741s	0.242s
:processTestResources	NO-SOURCE	0.984s	0.000s
:testClasses		0.984s	0.001s
:test		0.985s	1m 59.135s
:check		2m 0.120s	0.001s
:build		2m 0.121s	0.001s



Task path		Started after ?	Duration ?
:clean		0.416s	0.048s
:compileJava		0.465s	0.085s
:processResources	NO-SOURCE	0.550s	0.000s
:classes		0.550s	0.000s
:jar		0.551s	0.040s
:assemble		0.591s	0.000s
:compileTestJava		0.592s	0.212s
:processTestResources	NO-SOURCE	0.804s	0.001s
:testClasses		0.805s	0.000s
:test		0.805s	10.553s
:check		11.359s	0.000s
:build		11.359s	0.000s



```
tasks.test { this: Test!
    onlyIf { true }
    useJUnitPlatform()
    maxParallelForks = Runtime.getRuntime().availableProcessors()
    testLogging { this: TestLoggingContainer
```

TEST DISTRIBUTION

- CI uses fan-out to speed-up tests
- Shouldn't you enjoy it for local tests?
- Use the cloud to distribute test load
- RUN ALL THE TESTS!

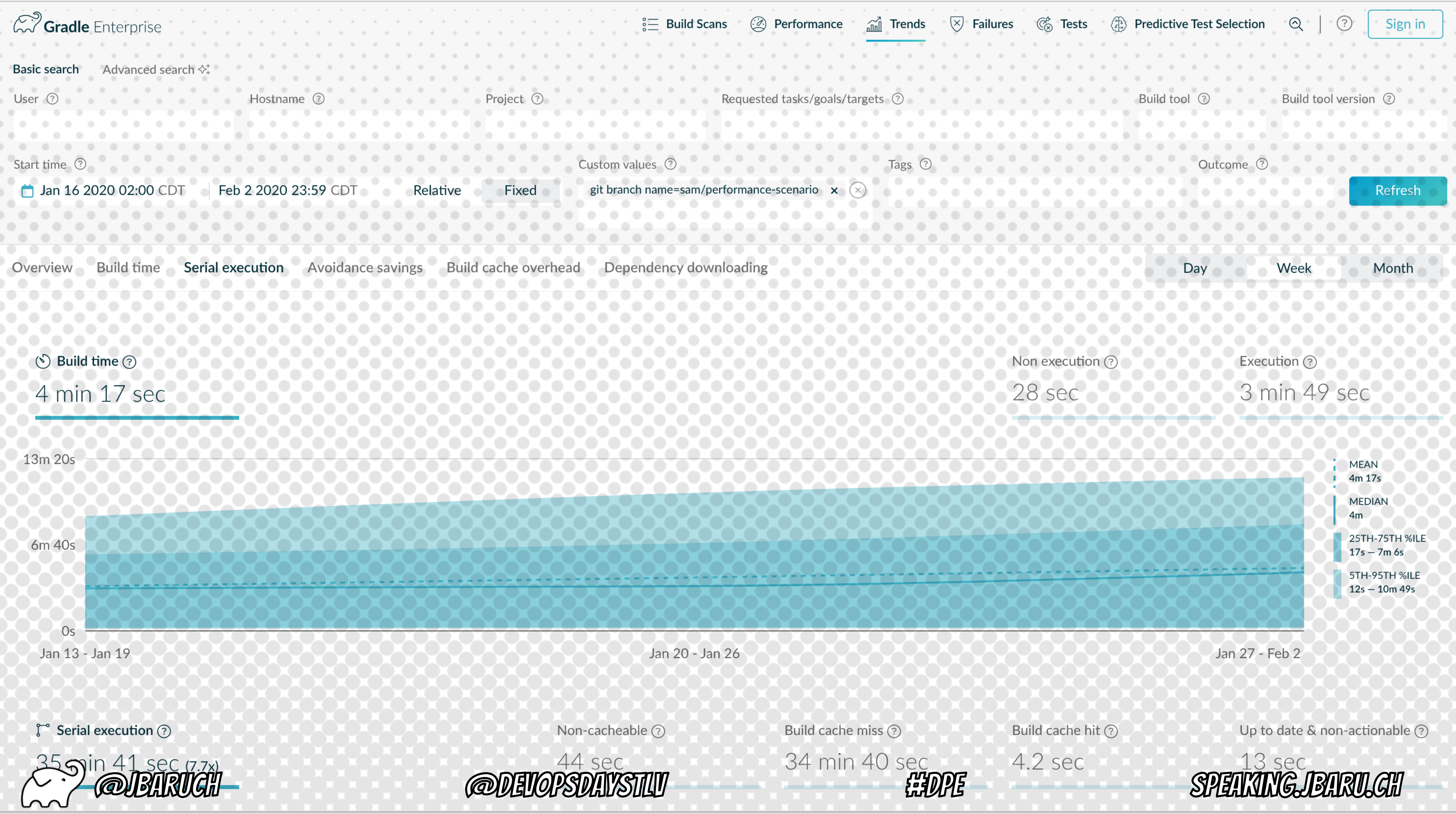
WHY NOT JUST USING CI FAN-OUT?

- Relying on shared CI infrastructure
- CI infrastructure is not optimized for real-time feedback!
- Are the agents as fast as they can be?

DON'T LET IT SLIDE

OBSERVE AND IMPROVE

- Measure local build times across time and environments
- Detect downfacing trends
- Find root causes and improve



User ?

Hostname ?

Project ?

Requested tasks/goals/targets ?

Build tool ?

Build tool version ?

Start time ?: Jan 16 2020 02:00 CDT — Feb 2 2020 23:59 CDT

Custom values ?: git branch name=sam/performance-scenario ×

Tags ?

Outcome ?

Build time ?
4 min 17 sec

Non execution ?
28 sec

Execution ?
3 min 49 sec

MEAN
4m 17s

MEDIAN
4m

25TH-75TH %ILE
17s — 7m 6s

5TH-95TH %ILE
12s — 10m 49s

Serial execution ?
35 min 41 sec (7.7x)

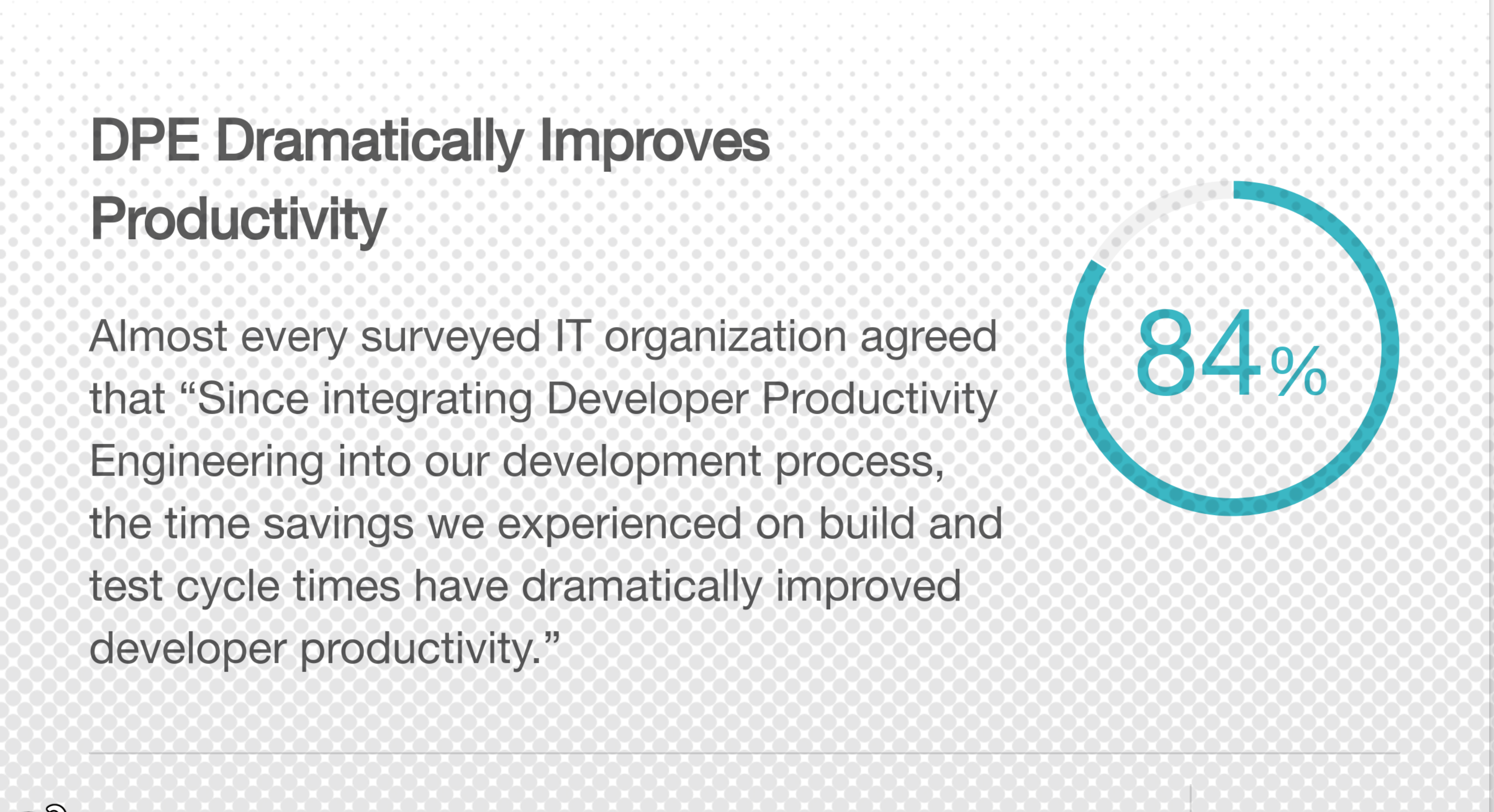
Non-cacheable ?
44 sec

Build cache miss ?
34 min 40 sec

Build cache hit ?
4.2 sec

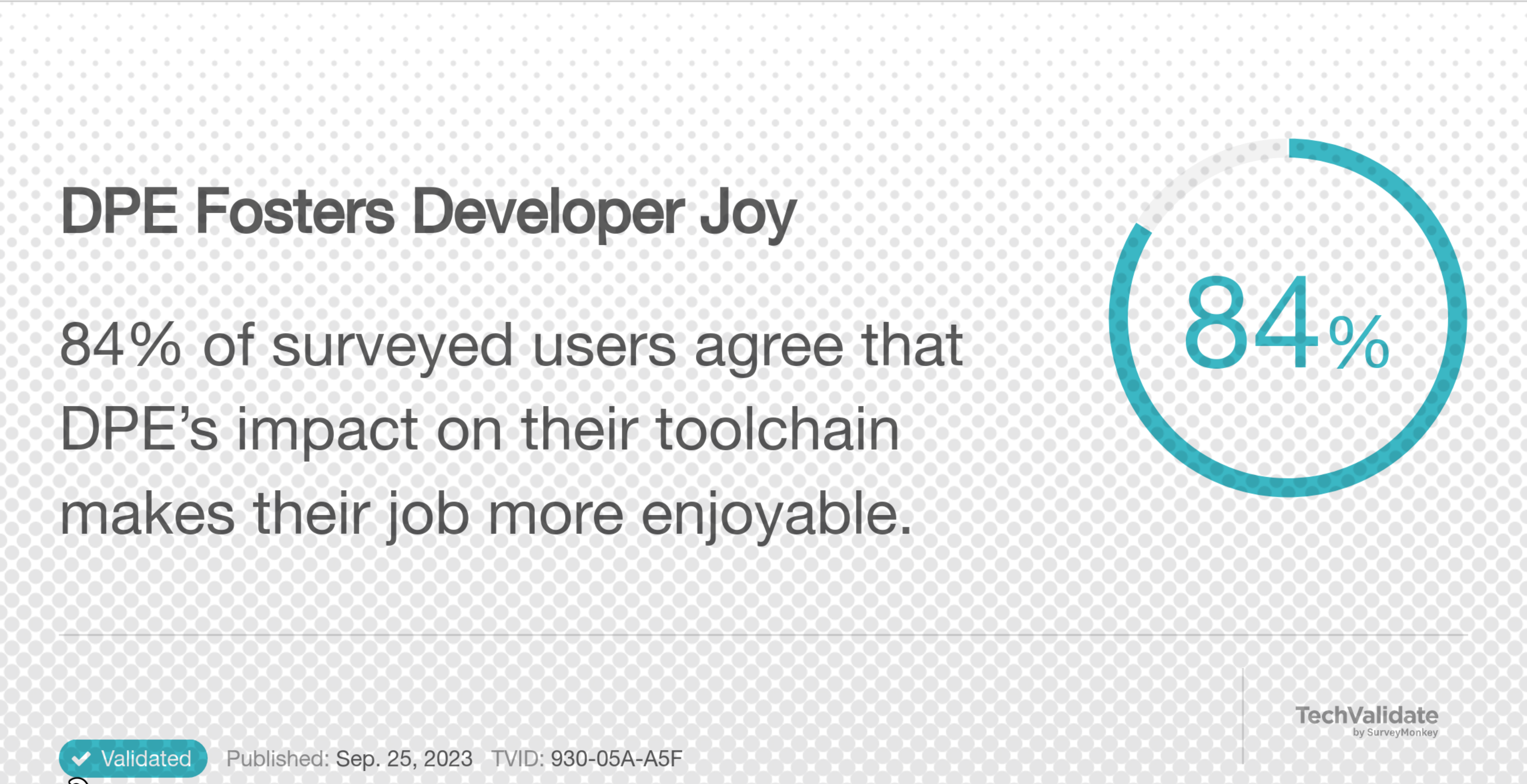
Up to date & non-actionable ?
13 sec

THE GAINS ARE REAL!



DPE Dramatically Improves Productivity

Almost every surveyed IT organization agreed that “Since integrating Developer Productivity Engineering into our development process, the time savings we experienced on build and test cycle times have dramatically improved developer productivity.”



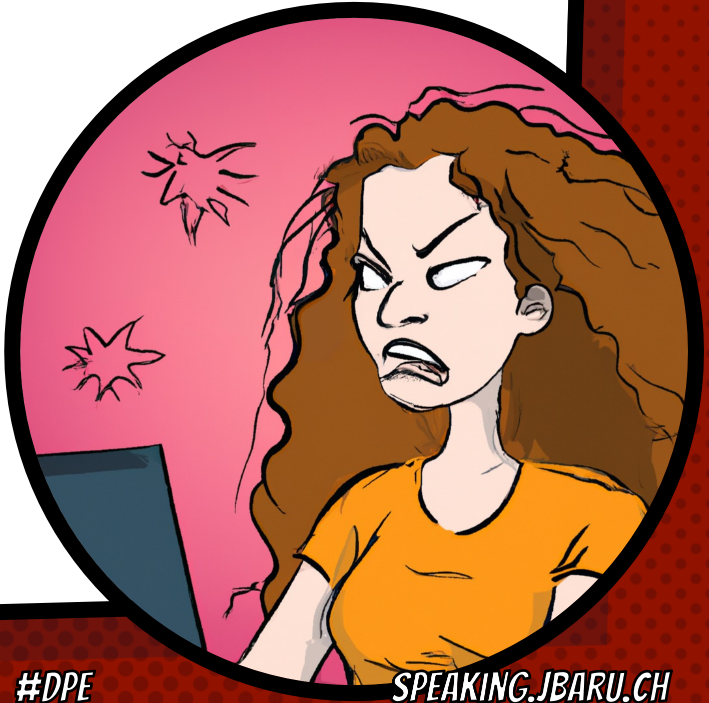
DPE Fosters Developer Joy

84% of surveyed users agree that DPE’s impact on their toolchain makes their job more enjoyable.

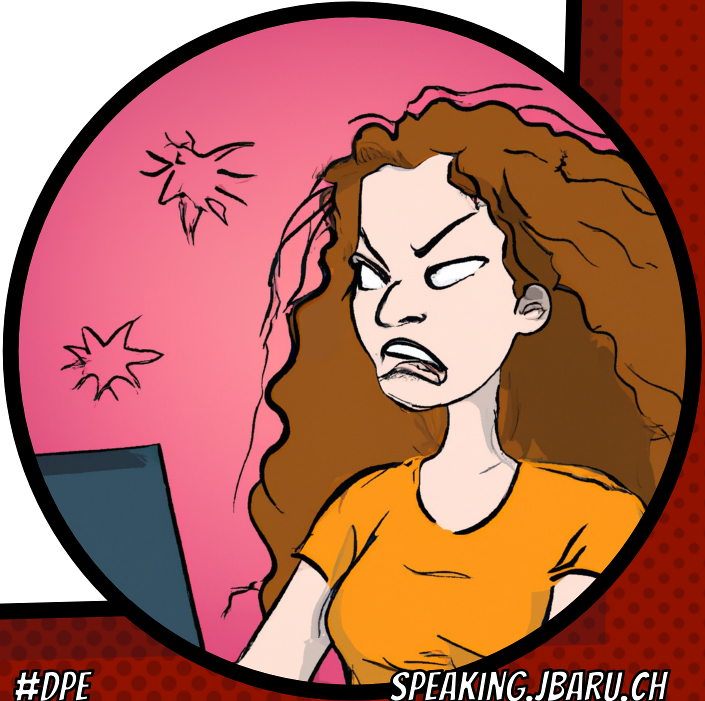
✓ Validated Published: Sep. 25, 2023 TVID: 930-05A-A5F

TechValidate by SurveyMonkey

BUT DPE IS NOT MY JOB!



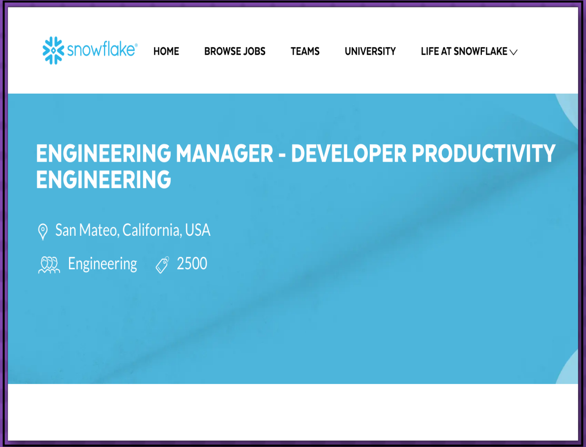
BUT DEVOPS IS NOT MY JOB!



DEVOPS VS DPE

	DEVOPS	DPE
OUTCOME	The product	More productive work
URGENCY	On everybody's minds	Slips away
ORG STRUCTURE	Organizational culture	Specialized team

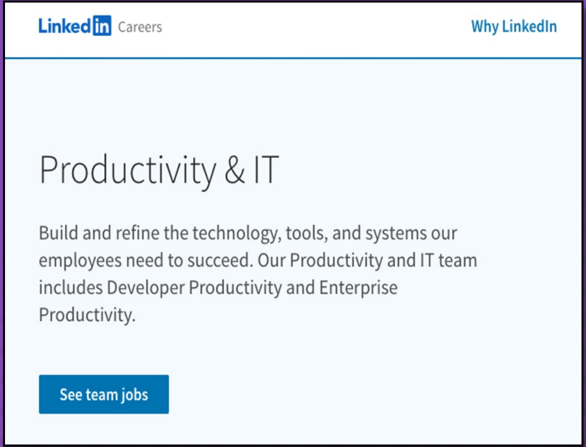
THE BEST DPE ORGANIZATIONS BUILD FOCUSED TEAMS



ENGINEERING MANAGER - DEVELOPER PRODUCTIVITY ENGINEERING

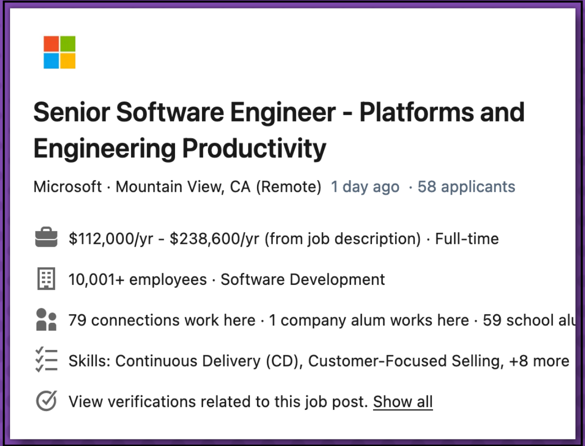
San Mateo, California, USA

Engineering 2500



Productivity & IT

Build and refine the technology, tools, and systems our employees need to succeed. Our Productivity and IT team includes Developer Productivity and Enterprise Productivity.


Senior Software Engineer - Platforms and Engineering Productivity

Microsoft · Mountain View, CA (Remote) 1 day ago · 58 applicants

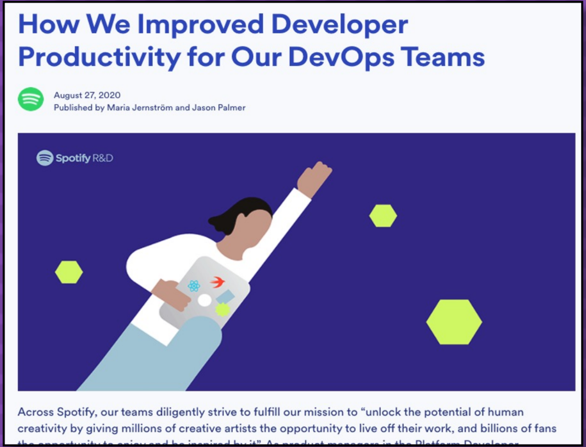
$112,000/yr - $238,600/yr (from job description) · Full-time

10,001+ employees · Software Development

79 connections work here · 1 company alum works here · 59 school alu

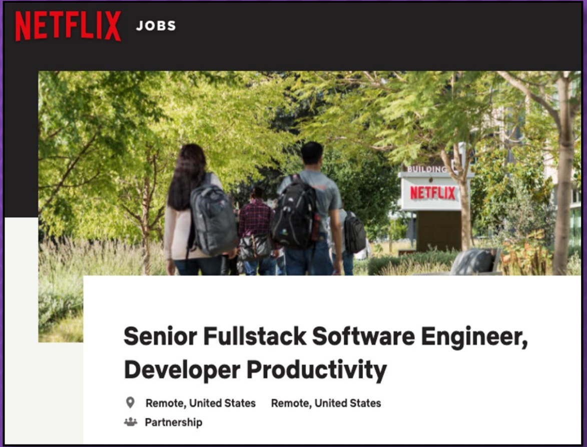
Skills: Continuous Delivery (CD), Customer-Focused Selling, +8 more

View verifications related to this job post. Show all


How We Improved Developer Productivity for Our DevOps Teams

August 27, 2020
Published by Maria Jernström and Jason Palmer

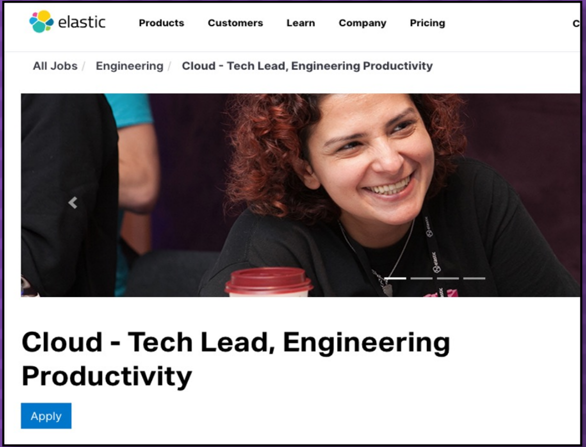
Across Spotify, our teams diligently strive to fulfill our mission to “unlock the potential of human creativity by giving millions of creative artists the opportunity to live off their work, and billions of fans the opportunity to enjoy and be inspired by it.” As product managers in the Platform Developer



Senior Fullstack Software Engineer, Developer Productivity

Remote, United States Remote, United States

Partnership



All Jobs / Engineering / Cloud - Tech Lead, Engineering Productivity

Cloud - Tech Lead, Engineering Productivity

FIGHT DOWNSIZING SURVIVOR GUILT WITH DPE

- It improves morale
- It covers productivity gaps
- It's the right thing to do for both the employees and the business
- But it requires an organizational mindset and a dedicated effort!

LEARN MORE AND TRY IT TODAY!

- Take the Gradle/Maven Speed Challenge!
- Be DPE Agent of Change!
- Read the DPE Handbook!
- Watch the DPE Summit videos!

Scan me

SPEAKING.JBARU.CH

Q&A AND SOCIAL ADS

- @JBARUCH
- @DEVOPSDAYSTLV
- SPEAKING.JBARU.CH