Measuring Developer Productivity

All models are wrong, but some are useful

DPE SUMMIT

Hi

Collin
Senior Staff Researcher

Background in cognitive psychology

Leads the user experience researchers on the team

Previous roles in aerospace, medicine, and consumer products

Ciera
Senior Staff Software Engineer

Background in software engineering

Leads the engineers on the team

Previous roles in program analysis, software architecture, professor of software engineering

“All models are wrong but some are useful.”

—George Box, 1976

“When you construct a model you leave out all the details which you, with the knowledge at your disposal, consider inessential…

Models should not be true, but it is important that they are applicable…”

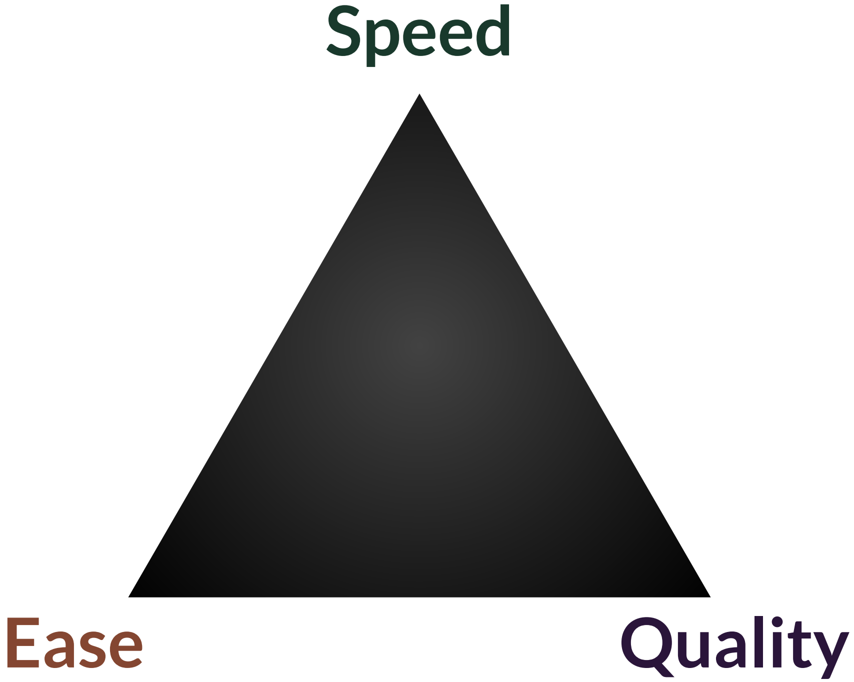
—Georg Rasch, 1960

“What’s the best metric for measuring engineering productivity?”

—lots of people we meet

A single metric doesn’t acknowledge the inherent tradeoffs that must be consistently evaluated.

“I can improve developer velocity by removing code review.”



Speed

Getting value (features and new products) to our users as quickly as possible, keeps us relevant in the marketplace.

Ease

Making the product development process efficient, friction-free, and well-supported. Ease supports a strong and engaged talent pool by creating a great development environment that attracts and retains top tech talent.

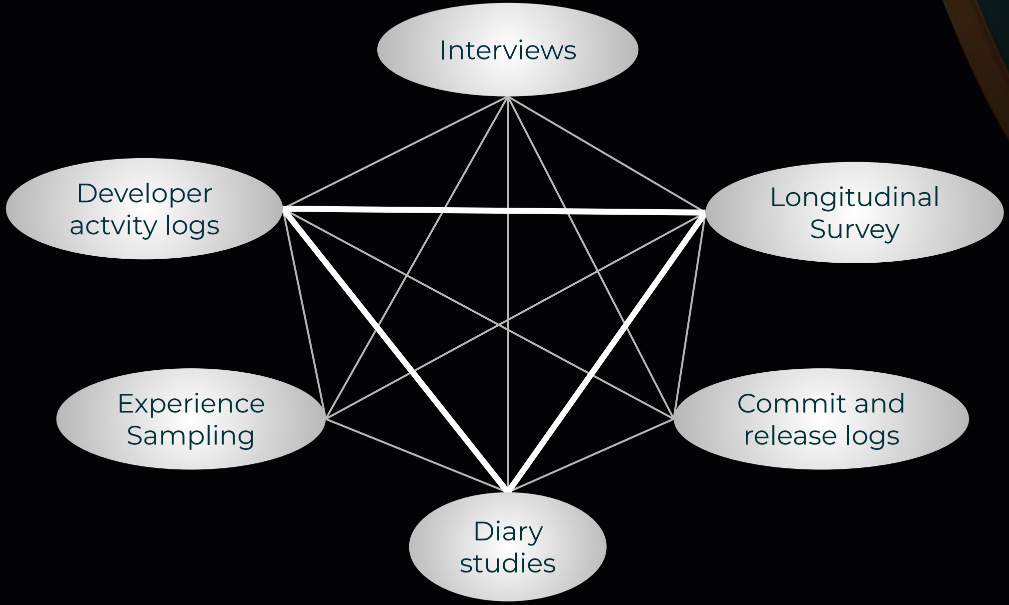
Quality

Building excellent products helps us attract and retain customers. Quality also acts as a guardrail to ensure that gains in speed and ease don't come at the expense of excellence.

Triangulating on productivity methologically helps us understand what's really happening.

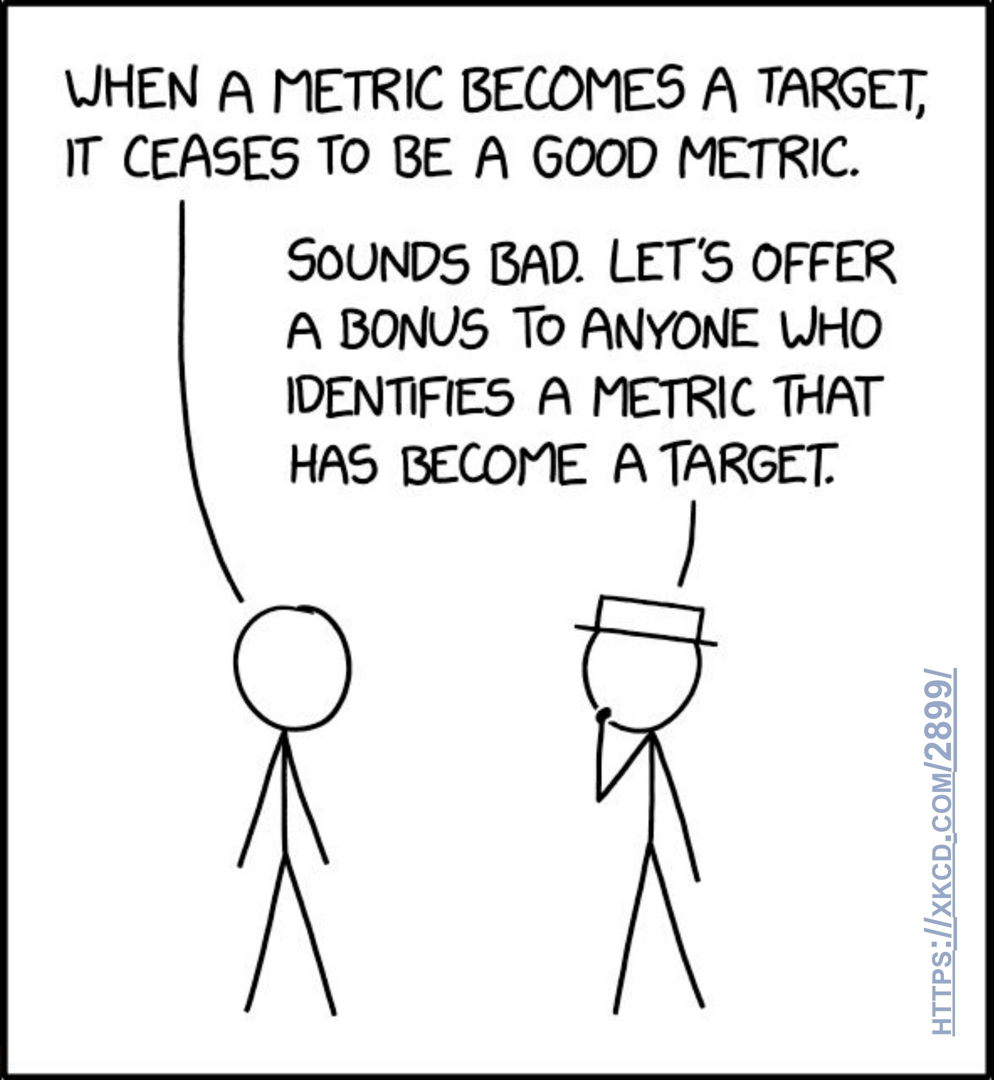
Quantitative and qualitative methods enable you to measure more things , to look for discrepancies in measurements, and to understand the context of our measurements (and blindspots).

Mixing methods is particularly helpful.



Single metrics are more susceptible to creating undesirable incentives.

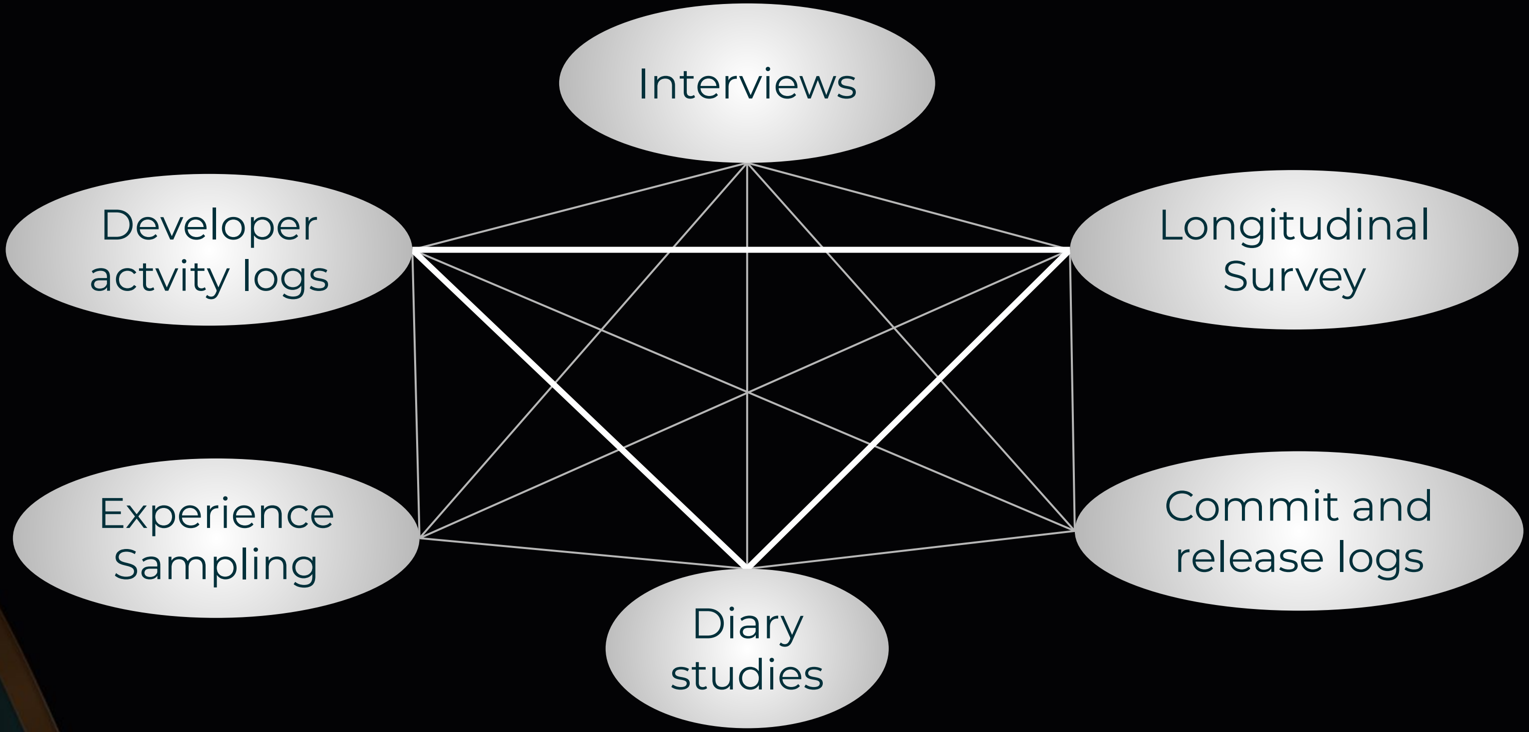
(Goodhart’s Law)


WHEN A METRIC BECOMES A TARGET, IT CEASES TO BE A GOOD METRIC.
SOUNDS BAD. LET'S OFFER A BONUS TO ANYONE WHO IDENTIFIES A METRIC THAT HAS BECOME A TARGET.
HTTPS://XKCD.COM/2899/

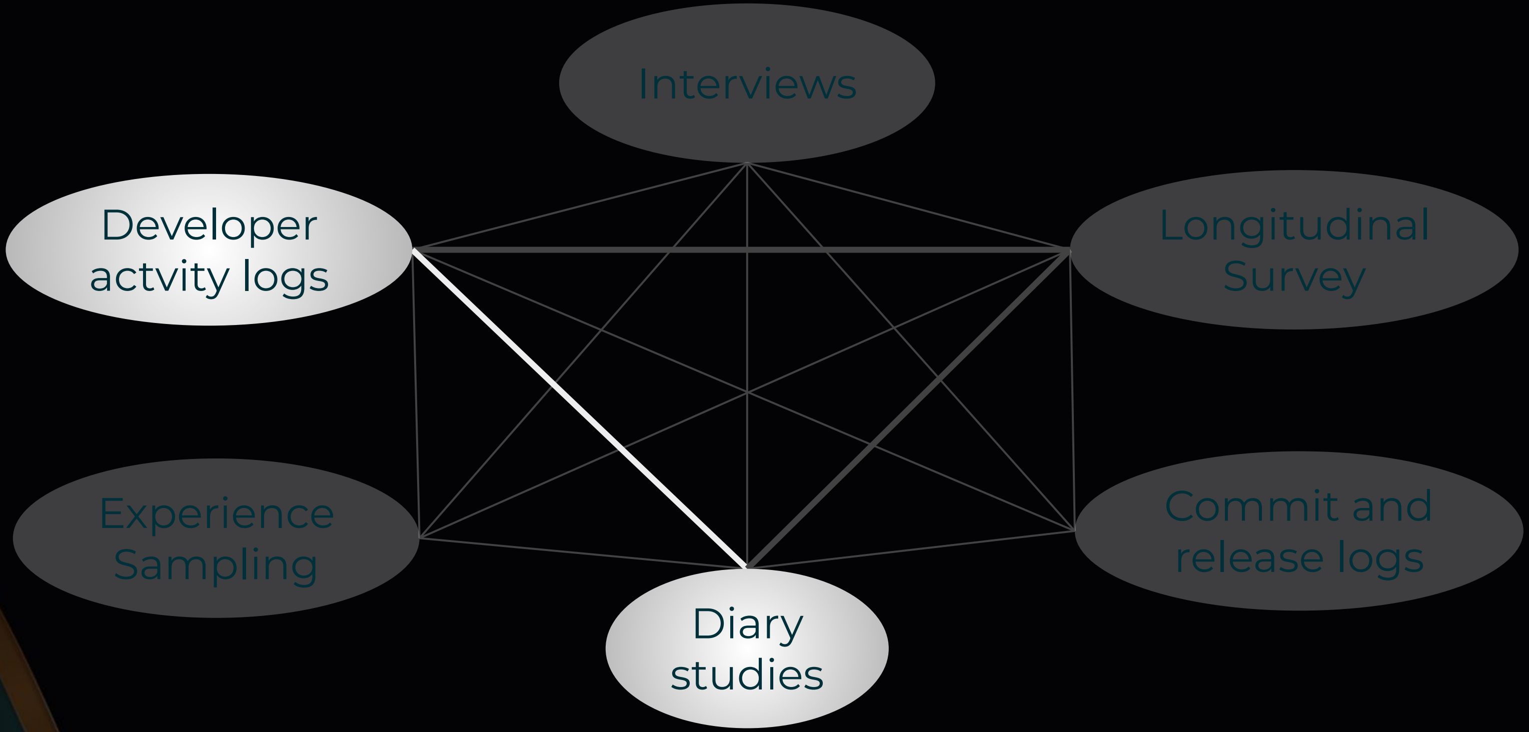
No Single Metric

A single metric doesn’t acknowledge the inherent tradeoffs that must be consistently evaluated.	“I can improve developer velocity by removing code review.”
All metrics are proxy measures of the underlying phenomenon we want to evaluate.	***Quantitative*** and ***qualitative*** methods enable you to measure more things, to look for discrepancies in measurements, and to understand the context of our measurements (and blindspots). ***Mixing methods*** is particularly helpful
Single metrics are more susceptible to creating undesirable incentives.	Goodhart’s Law: “When a measure becomes a target, it ceases to be a good measure.”

Measuring with multiple methods



Measuring with multiple methods



DPE SUMMIT

What's in our developer activity logs?

DEVELOPER TOOLS, including

Code Review	IDE
Code Search	Command Line Tools
Distr. Filesystem	Launch Tools
Documentation	Monitoring & Alerting
Build/Test Logs	Logging
Bug tracker	Frameworks
Task/Project management	

OTHER TOOLS/DATA

- Mail/Chat (timestamps)
- Calendar (free/busy times)
- Docs, Decks, Sheets (timestamps and ownership)
- Search (timestamps)
- Data Analysis tools

USER ATTRIBUTES and WORK CONTEXT

Team Hierarchy	Job code
Languages & readability	Level
Seat Location & Colocation	Tenure

KNOWN MISSING PIECES

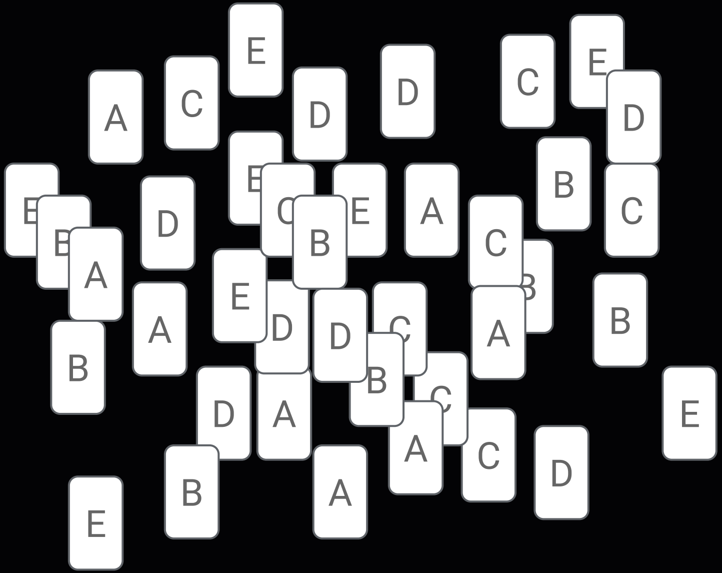
- Production management
- Experimentation
- ML modeling/training

Avoid collecting metadata for tools with mixed-use purposes (email, docs).

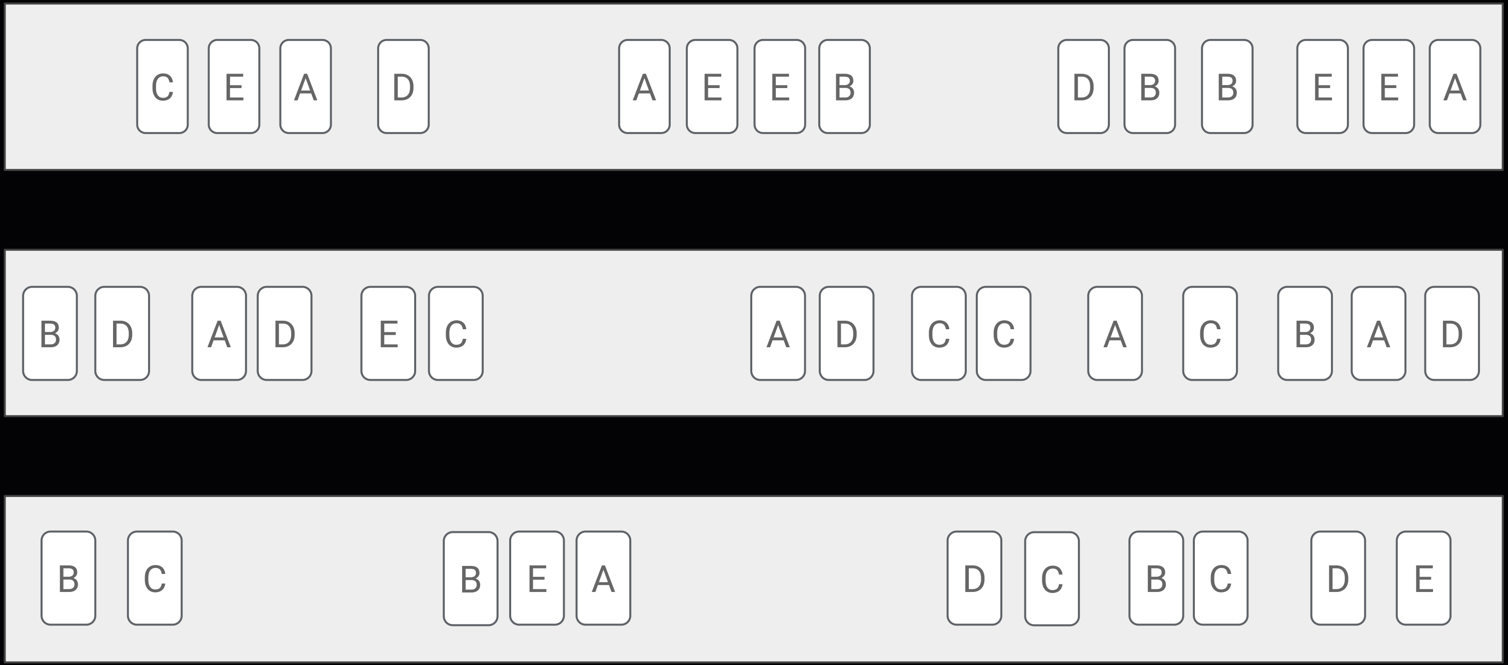
Logs data are **not** used for performance evaluation and are **always** presented in aggregate to protect privacy.

Goal is to raise productivity for ***everyone***.

Start with a pile of logs...

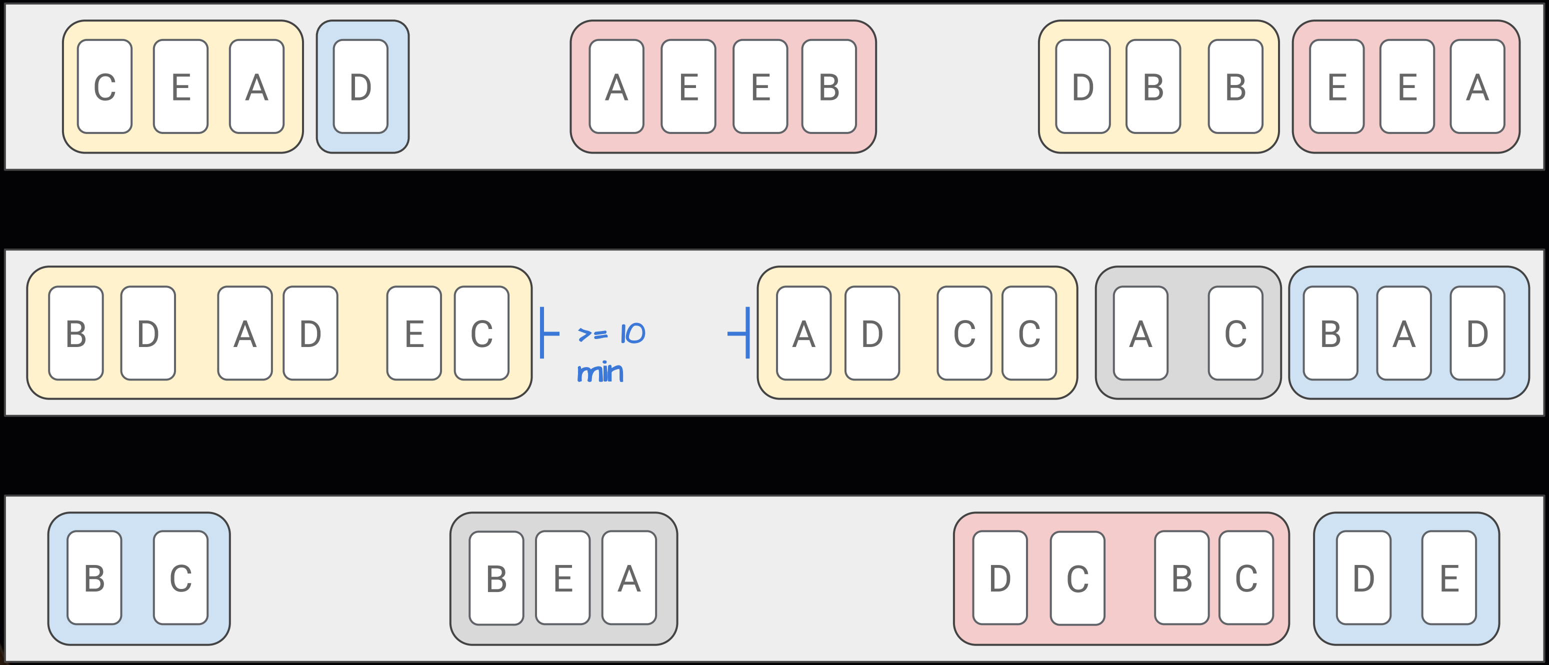


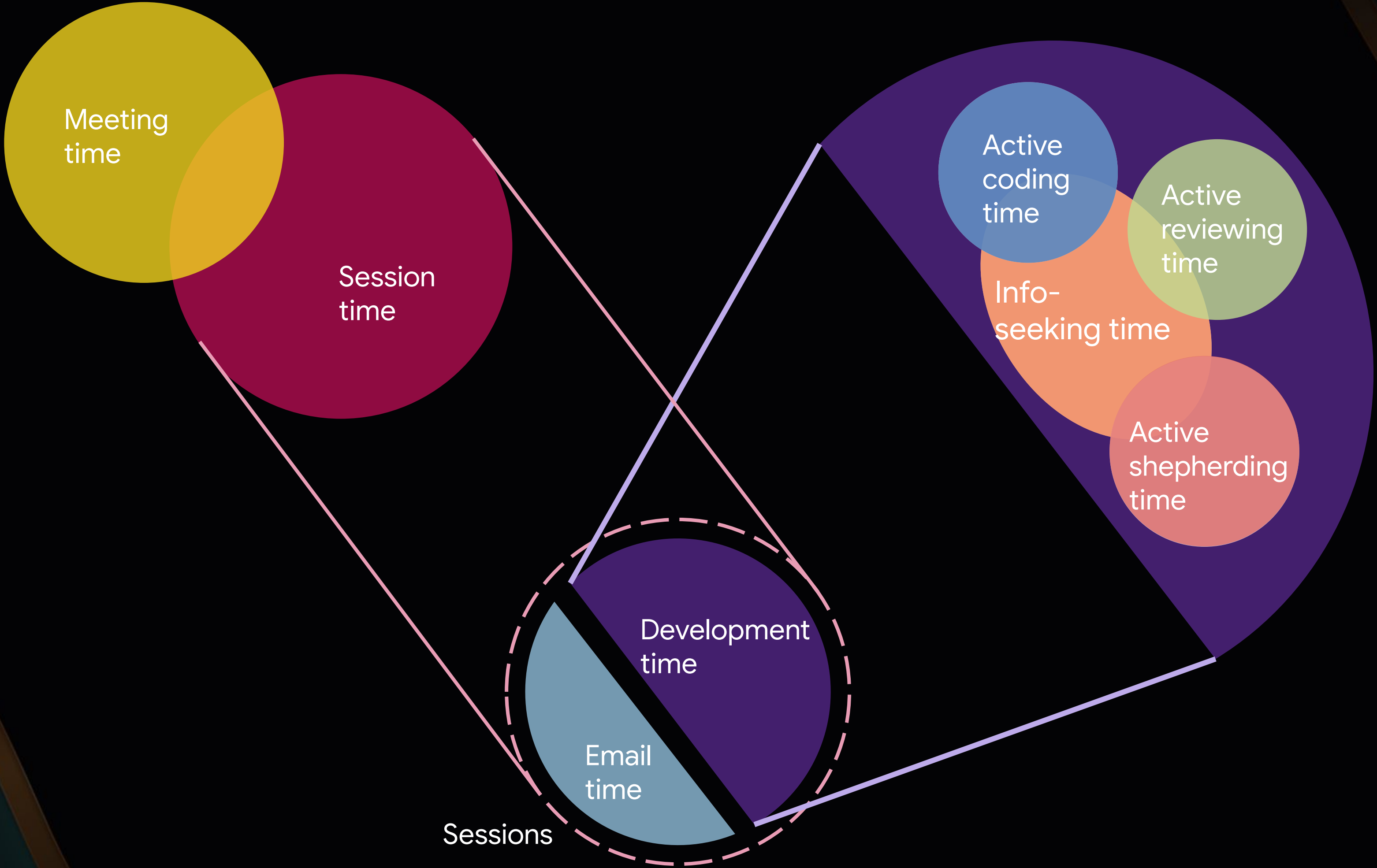
Partition by user, sort by time...

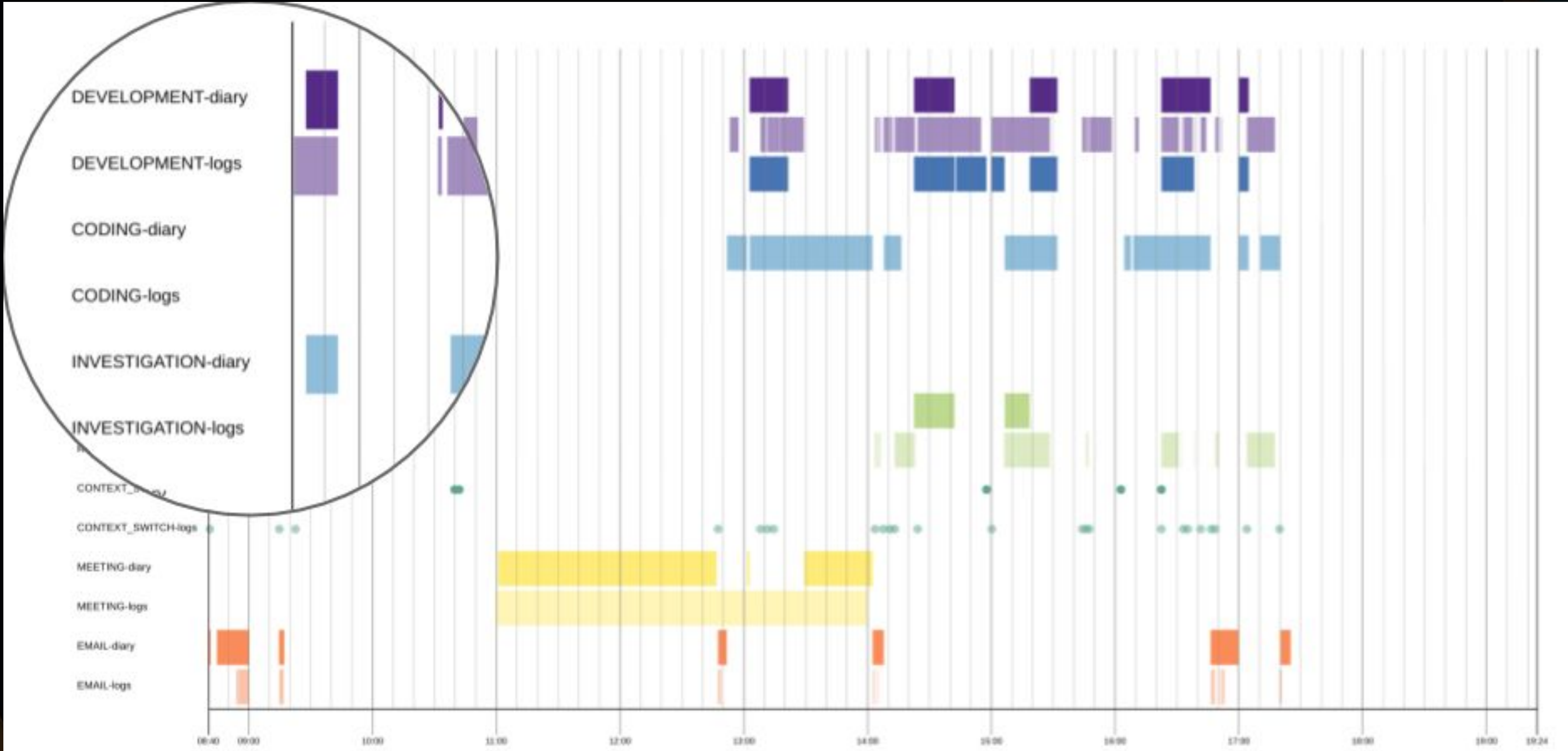


DPE SUMMIT

Group by proximity and task...



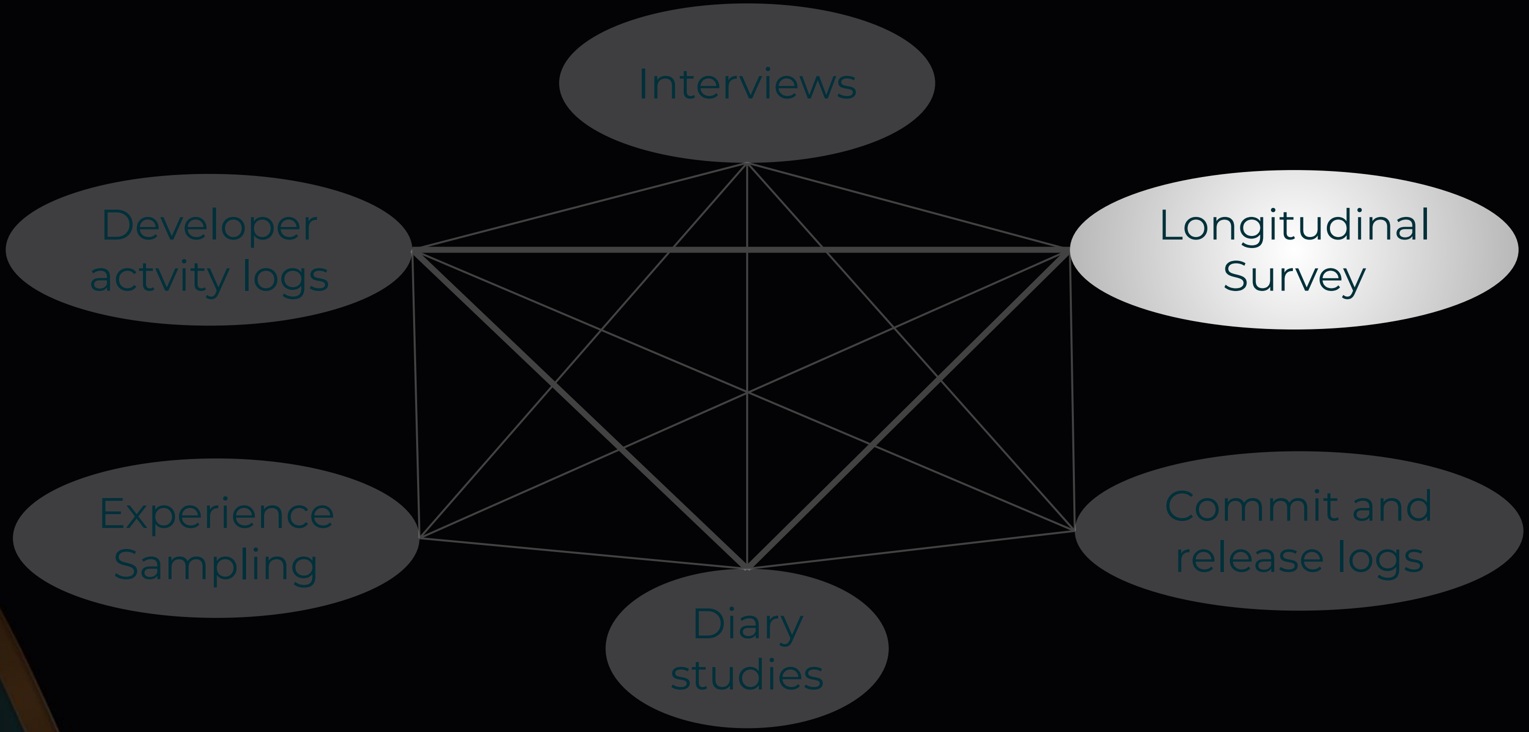

Meeting time
Session time
Development time
Email time
Sessions
Active coding time
Active reviewing time
Info-seeking time
Active shepherding time



Activity	COHENS_KAPPA	PABAK	BIAS_INDEX	PREVALENCE_INDEX	OBSERVED_AGREEMENT
DEVELOPMENT	0.31	0.48	-0.19	-0.53	0.74
REVIEWING	0.25	0.69	-0.09	-0.78	0.85
MEETING	0.81	0.86	-0.06	-0.5	0.93
EMAIL	0.46	0.9	0.05	-0.9	0.95

DPE SUMMIT

Measuring with multiple methods

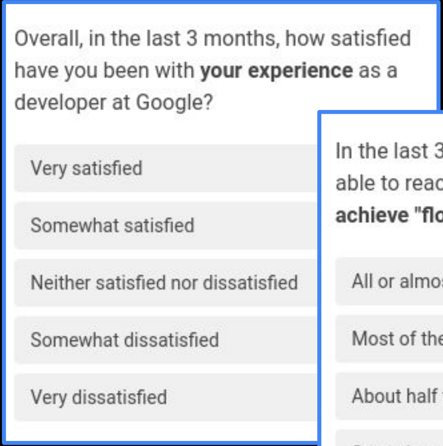


DPE SUMMIT

EngSat

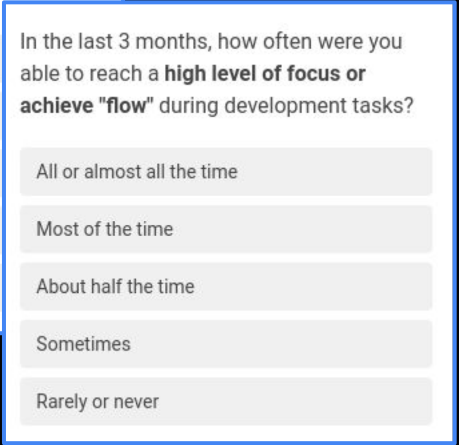
Quarterly survey to ⅓ of the engineers

Providing longitudinal data since 2018



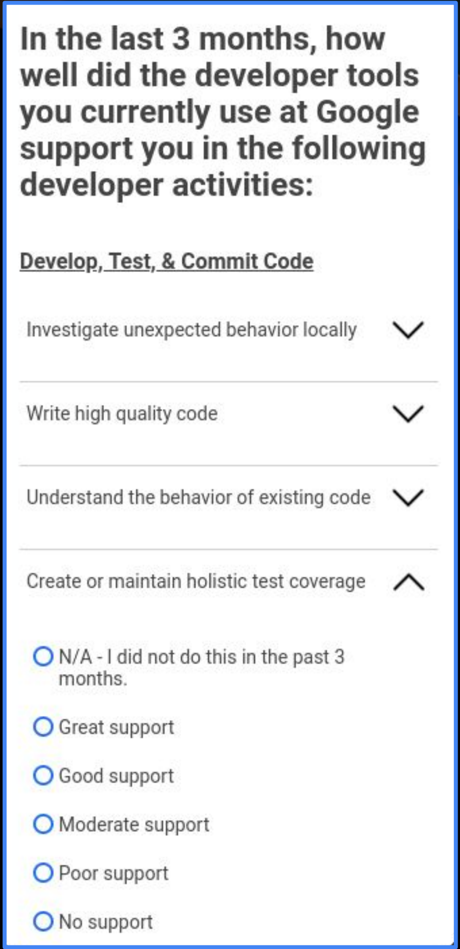
Overall, in the last 3 months, how satisfied have you been with **your experience** as a developer at Google?

- Very satisfied
- Somewhat satisfied
- Neither satisfied nor dissatisfied
- Somewhat dissatisfied
- Very dissatisfied



In the last 3 months, how often were you able to reach a **high level of focus or achieve "flow"** during development tasks?

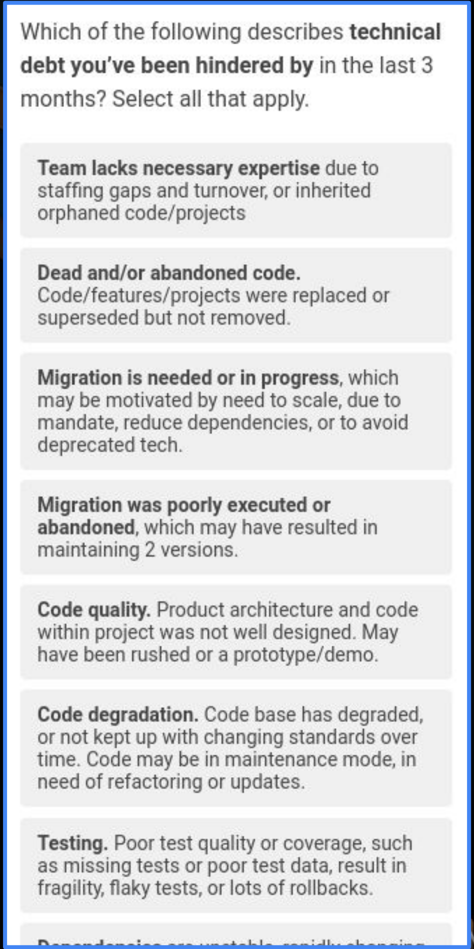
- All or almost all the time
- Most of the time
- About half the time
- Sometimes
- Rarely or never



In the last 3 months, how well did the developer tools you currently use at Google support you in the following developer activities:

Develop, Test, & Commit Code

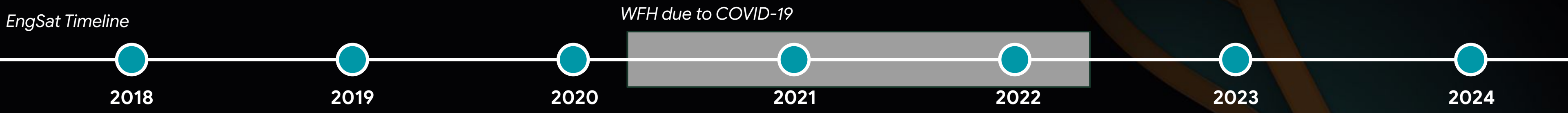
- Investigate unexpected behavior locally
- Write high quality code
- Understand the behavior of existing code
- Create or maintain holistic test coverage
  - ○ N/A - I did not do this in the past 3 months.
  - ○ Great support
  - ○ Good support
  - ○ Moderate support
  - ○ Poor support
  - ○ No support



Which of the following describes **technical debt you've been hindered by** in the last 3 months? Select all that apply.

- **Team lacks necessary expertise** due to staffing gaps and turnover, or inherited orphaned code/projects
- **Dead and/or abandoned code.** Code/features/projects were replaced or superseded but not removed.
- **Migration is needed or in progress**, which may be motivated by need to scale, due to mandate, reduce dependencies, or to avoid deprecated tech.
- **Migration was poorly executed or abandoned**, which may have resulted in maintaining 2 versions.
- **Code quality.** Product architecture and code within project was not well designed. May have been rushed or a prototype/demo.
- **Code degradation.** Code base has degraded, or not kept up with changing standards over time. Code may be in maintenance mode, in need of refactoring or updates.
- **Testing.** Poor test quality or coverage, such as missing tests or poor test data, result in fragility, flaky tests, or lots of rollbacks.

Meeting the needs of the moment



EngSat was starting its 3rd year of data collection when the COVID-19 pandemic began.

This experience revealed the **power of a stable survey instrument**. It allowed us to measure change on key measures before, during and after COVID.

We were also able to **add new questions** to better understand engineers experience adapting to WFH and returning to the office as well as partner with POps to get a holistic picture.

DPE SUMMIT

Build latency

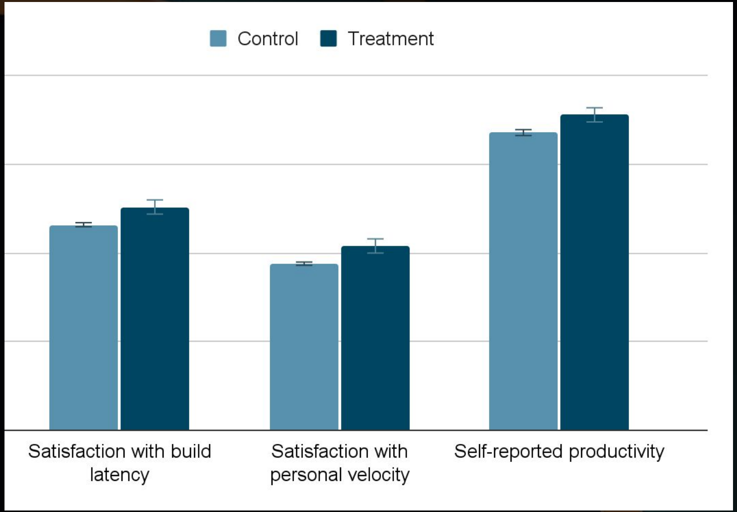
Is there a “magic number” for build latency?

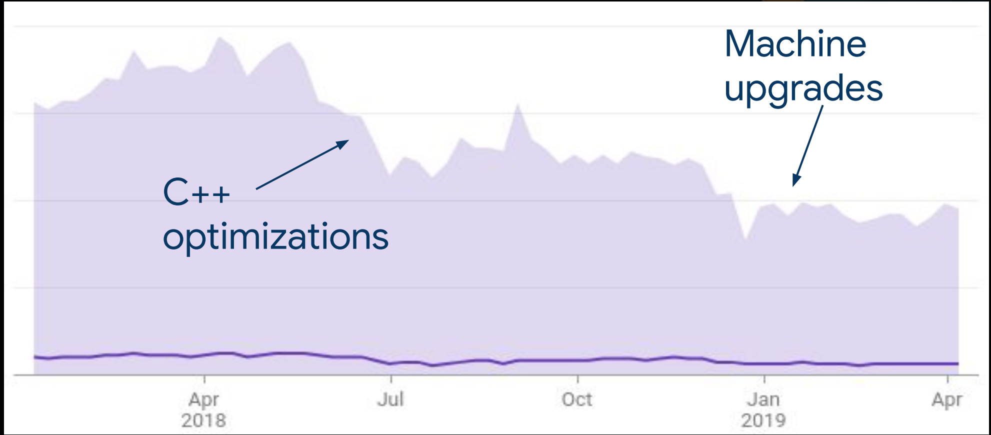
Data analysis says no: every improvement makes a difference.

Ran a covert A/B experiment on whether modest improvements to build latency would impact build speed.

Improved both logs-based and sentiment metrics.

Lead to multiple improvements to build latency.





Readability

We studied the impact (positive and negative) of readability and the readability certification process. The goal was to evaluate the costs and benefits of readability, in order to make an informed decision about modifying or eliminating the readability process.

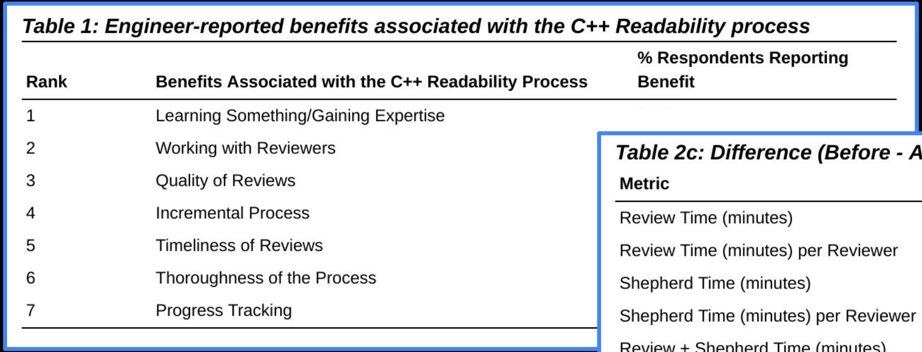


Table 1: Engineer-reported benefits associated with the C++ Readability process

Rank	Benefits Associated with the C++ Readability Process	% Respondents Reporting Benefit
1	Learning Something/Gaining Expertise	
2	Working with Reviewers	
3	Quality of Reviews	
4	Incremental Process	
5	Timeliness of Reviews	
6	Thoroughness of the Process	
7	Progress Tracking	

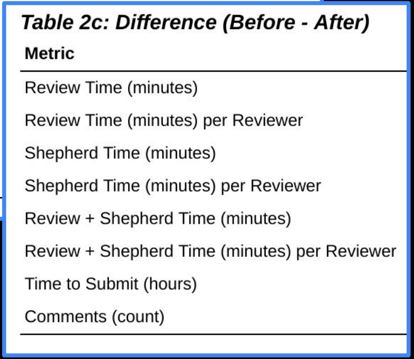
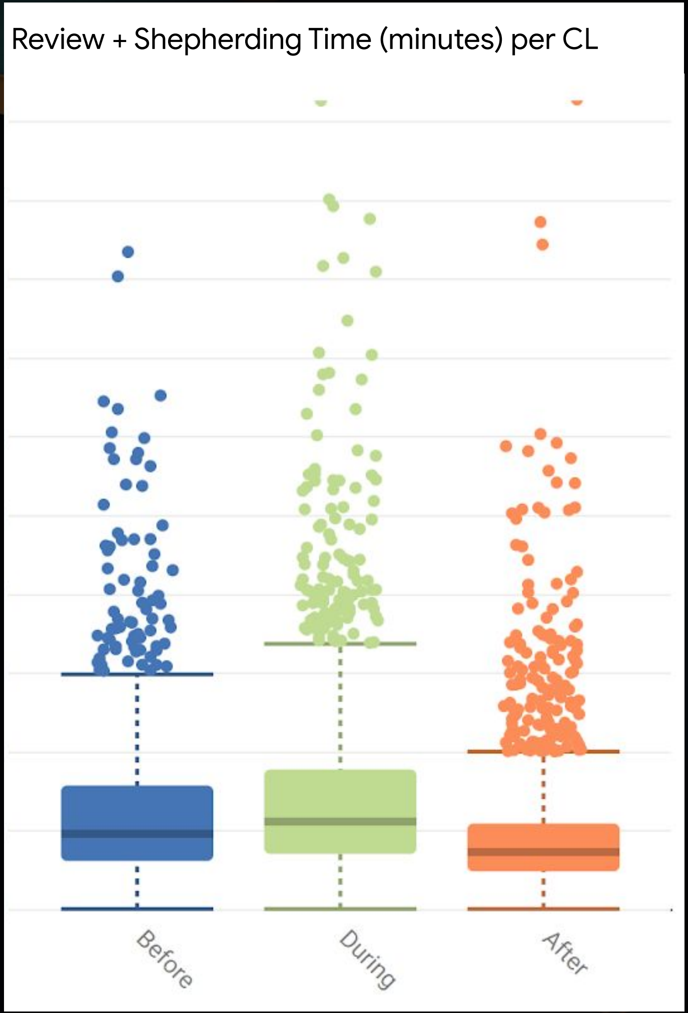


Table 2c: Difference (Before - After)

Metric
Review Time (minutes)
Review Time (minutes) per Reviewer
Shepherd Time (minutes)
Shepherd Time (minutes) per Reviewer
Review + Shepherd Time (minutes)
Review + Shepherd Time (minutes) per Reviewer
Time to Submit (hours)
Comments (count)



What can you do here at DPE?

Keep in mind: all models are wrong.

What's not being measured?

What are the potential tradeoffs among speed, ease, and quality (or other outcomes)?

Would different methods of measurement lead to a different conclusion?

Ask whether the model is useful.

Do the shortcomings of this approach matter?

How might this approach be combined with another?

What is the goal of this approach?

What can you do at your company?

Large scale?

Staff a multi-disciplinary team. Connect logs data with surveys, interviews, and diary studies.

Can't staff a 20 person team for measurement?

Surveys give you broad state for relatively low effort.

Fit in a large conference room?

Just have a discussion. Don't waste time over-optimizing or over-measuring.

In all cases....what's your goal?

Are you doing performance monitoring…or actually improving developer productivity?

Are you targeting what's easy to measure…or what's going to represent the human impact?

Are you triangulating across speed, ease, and quality?

Are you successfully identifying your **largest** and **most tractable** problems?

Related Reading

C. Jaspan et al. "Enabling the Study of Software Development Behavior With Cross-Tool Logs," IEEE Software, 2020.

C. Jaspan and C. Green. “Developer Productivity for Humans, Part 4: Build latency, predictability, and developer productivity”. IEEE Software, 2023.

A. Brown et al. “Using Logs Data to Identify When Software Engineers Experience Flow or Focused Work.” CHI Conference on Human Factors in Computing Systems (CHI '23), 2023.

C. Jaspan and C. Green, "A Human-Centered Approach to Developer Productivity," IEEE Software, 2023.

C. Green, C. Jaspan, M. Hodges, and J. Lin. “Developer Productivity for Humans, Part 7: Software Quality”. IEEE Software, 2023.

ciera@google.com
&
colling@google.com

DPE SUMMIT