

Empowering Productivity



Karim Nakad

Software Engineer

Empowering Productivity at Meta



Karim Nakad
Staff Software Engineer



MEET THE TEAM



Rui Abreu



Imad Ahmad



Moritz Beller



Drew Carlson



Miaoqing Fang



Jessica Fu



Jonathan
Huang



Noah Lee



Chandra
Maddila



Vijayaraghavan
Murali



Nachi
Nagappan



Abhimanyu
Nagpal



Amanda Park



Akshay Patel



Peter Rigby



Jiaqi Wu

WHY ARE WE HERE?



Agenda

01. Measuring Productivity

02. Debbie the developer

a. Velocity

b. Reliability

c. Knowledge

03. Other Metrics

04. Team Productivity Dashboard

05. What's Next?

06. Q&A

HOW DO WE MEASURE PRODUCTIVITY?



HOW DO WE MEASURE PRODUCTIVITY?

Code review time
Burnout
Coding Time
Amount of code written
Meeting Time
Build Runtime
Number of code reviews
Code Quality
Number of wikis published
Deployment Time
Number of commits
API Latency
Number of sev resolutions
Organization attrition
Deployment frequency
Number of Meetings
Words per minute
Service uptime
Time to fix bugs
Test Runtime
Time to resolve sevs
Time to close ticket
Satisfaction
Number of bug fixes
Tickets closed

HOW DO WE MEASURE PRODUCTIVITY?

```
// Defined in MyClass
public async
function callTheFunction(string $my_string): string

public async function callTheFunction(string $my_string):
string {
    return "Yay!";
}

<<__Memoize>>
public async function
return await $this->callTheFunction(self::CONSTANT_VALUE);    (you) * Lo

}

<<__Memoize>>
public async function callTheFunction(string $my_string): string {
    return "Yay!";
}
```

```
// Defined in MyClass
public async
function wrongFunction(string $my_string): string

public async function wrongFunction(string $my_string):
string {
    return "Oh No!";
}

<<__Memoize>>
public async function
return await $this->callTheFunction(self::CONSTANT_VALUE);    (you) * Lo

}

<<__Memoize>>
public async function callTheFunction(string $my_string): string {
    return "Yay!";
}
```


PRODUCTIVITY FRAMEWORKS

DevOps Research and Assessment

D

O

R

A



Deployment
Frequency



Lead Time for
Changes



Mean Time to
Recovery (MTTR)



Change Failure Rate

PRODUCTIVITY FRAMEWORKS

S

P

A

C

E



Satisfaction and
Well-Being



Performance



Activity



Communication
and Collaboration



Efficiency and
Flow

META'S FRAMEWORK

HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)



Debbie

01. Measuring Productivity

02. Debbie the developer

- a. Velocity
- b. Reliability
- c. Knowledge

03. Other Metrics

04. Team Productivity Dashboard

05. What's Next?

06. Q&A

DEBBIE



facebook

Connect with friends and the world
around you on Facebook.



Log In

[Forgot password?](#)

Create new account

Create a **Page** for a celebrity, brand or business.

DEBBIE



WHAT DOES DEBBIE FIX?

- Split mega-classes into component classes
- Fix architectural design flaws
- Remove unused variables
- Migrate to the latest frameworks
- Update your dependencies
- Upgrade tooling
- Write scripts to automate common work
- Improve documentation
- Fix bugs
- Fix papercuts
- ...

WHICH CODE TO FOCUS ON?

```
1 module GuessWhatIDo
2   DEFAULT_RANGE = 1..100
3   DEFAULT_TRIGGERS = [
4     ['Fizz', ->(i){ i % 3 == 0 }],
5     ['Buzz', ->(i){ i % 5 == 0 }],
6   ]
7
8
9   def self.range(range=DEFAULT_RANGE, triggers=DEFAULT_TRIGGERS)
10     enumerator(range.first, triggers).take(range.size)
11   end
12
13
14   def self.enumerator(start=DEFAULT_RANGE.first, triggers=DEFAULT_TRIGGERS)
15     Enumerator.new do |yielder|
16       i = start
17       loop do
18         parts = triggers.select{ |_, predicate| predicate.call(i) }
19         i_result = parts.size > 0 ? parts.map(&:first).join : i.to_s
20         yielder.yield(i_result)
21         i += 1
22       end
23     end
24   end
25
26 end
27
```

```
1 upto(100) do |i|
2   if i % 3 == 0 && i % 5 == 0
3     puts 'FizzBuzz'
4   elsif i % 3 == 0
5     puts 'Fizz'
6   elsif i % 5 == 0
7     puts 'Buzz'
8   else
9     puts i
10  end
11 end
12
```

HOW DO WE MEASURE EASE OF WRITING CODE?



Velocity

01. Measuring Productivity

02. Debbie the developer

a. Velocity

b. Reliability

c. Knowledge

03. Other Metrics

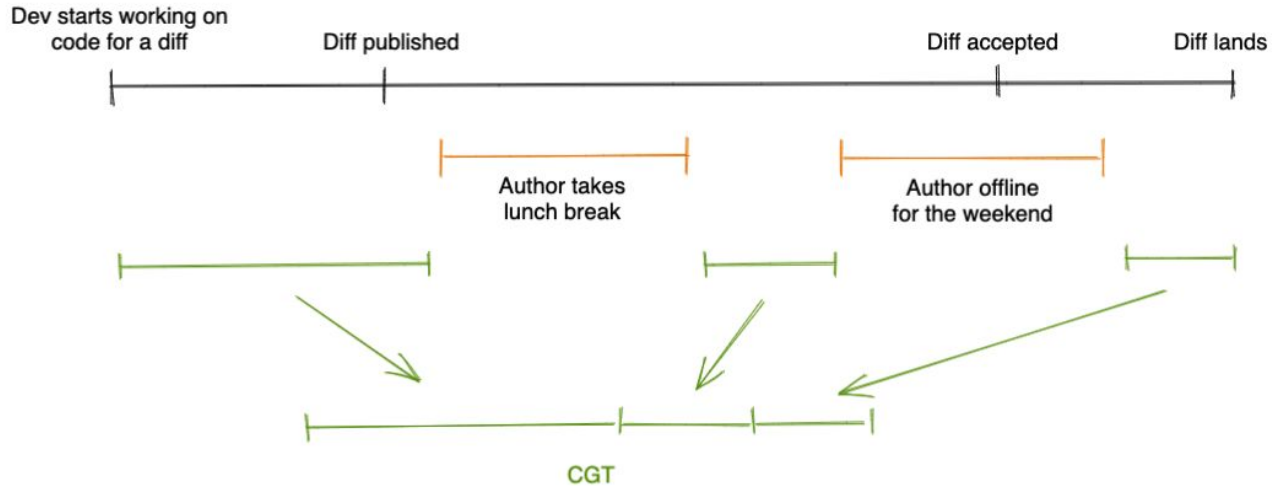
04. Team Productivity Dashboard

05. What's Next?

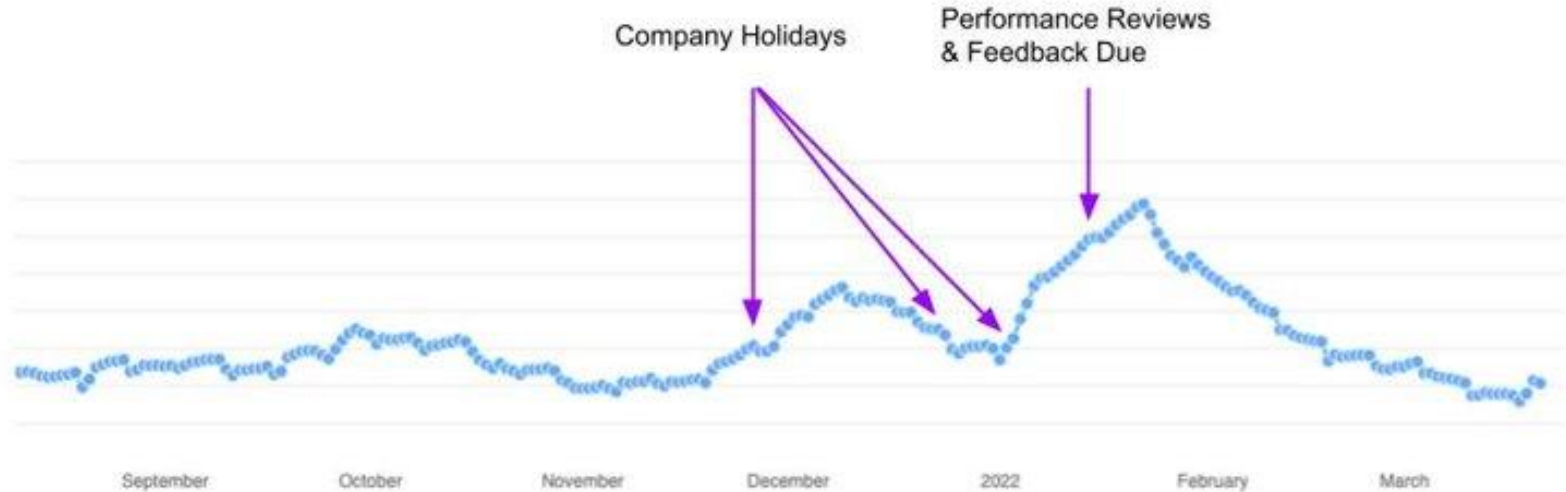
06. Q&A

CODE GESTATION TIME (CGT)

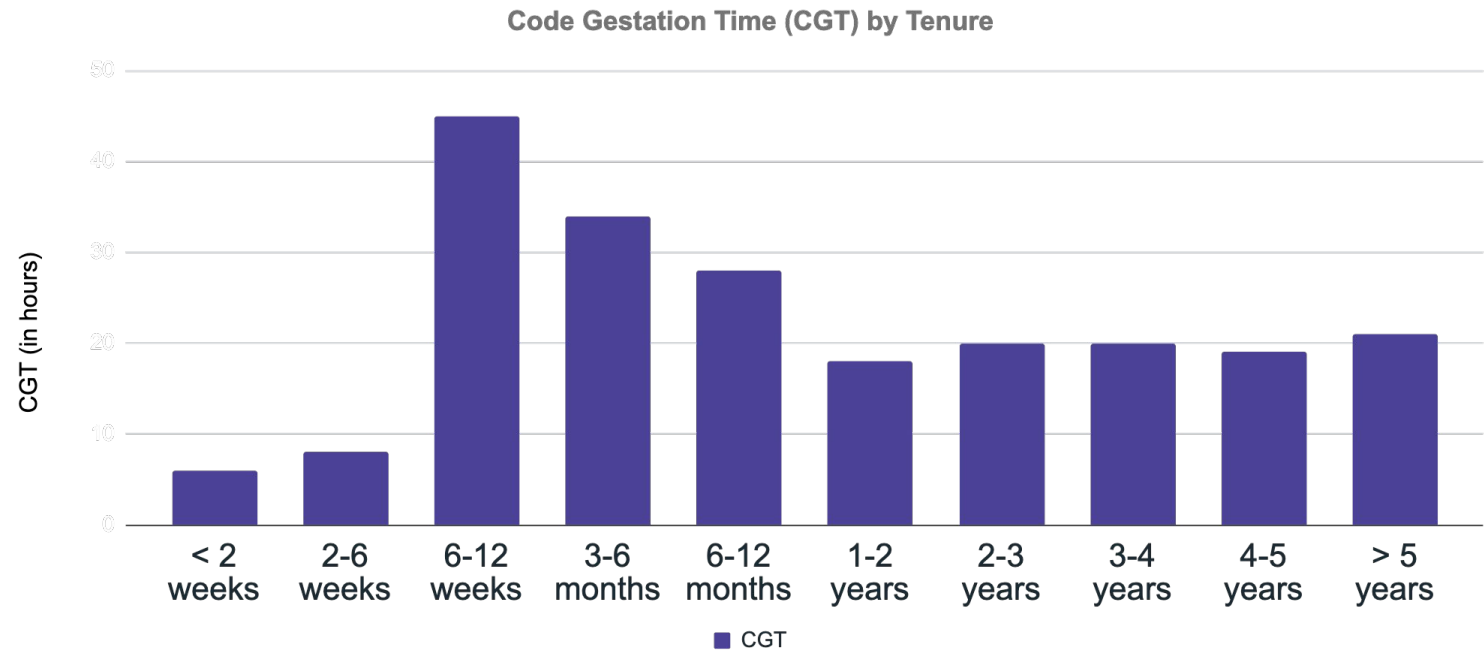
How many working hours does it take to write and land a diff?



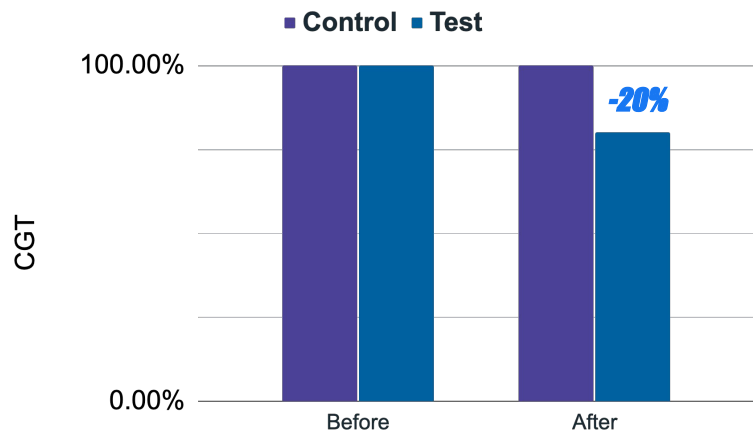
CGT AND FEEDBACK SEASON



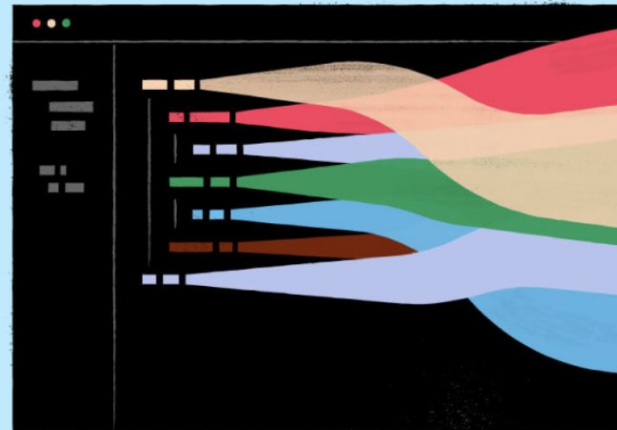
CGT AND ONBOARDING



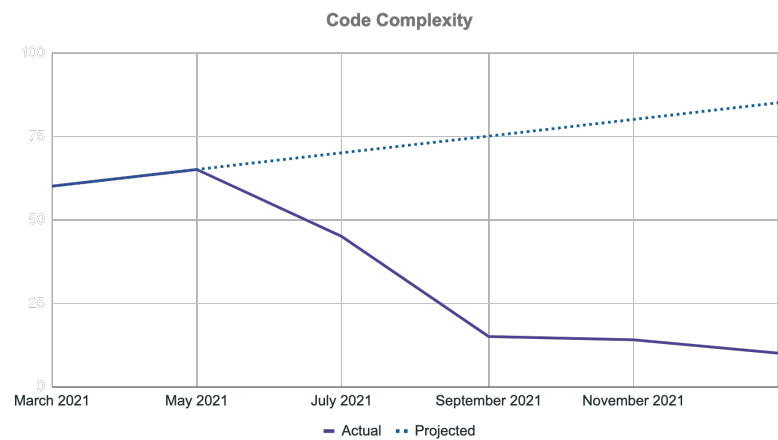
CGT AND WASABI



Enabling
Faster Python Authoring
With Wasabi

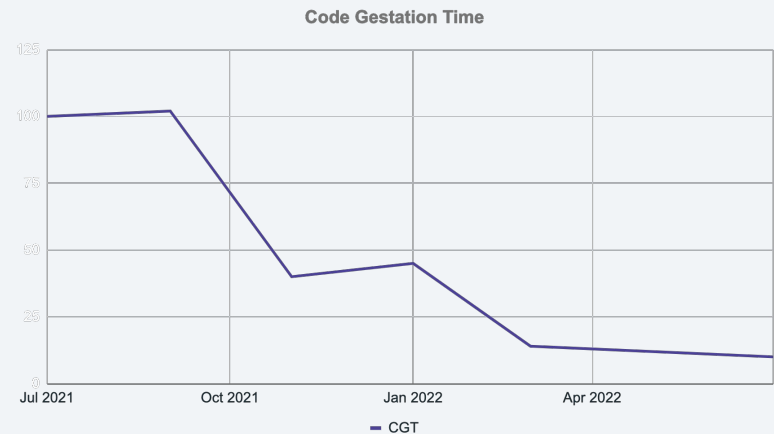


CGT AND CODE COMPLEXITY



60% reduction in complexity

42% reduction in CGT



FILE CGT



File	Effort Cost	CGT	File Edits
fileB.py	12	6	2
fileA.py	3	3	1
fileC.py	1	1	1

DEBBIE



File	Effort Cost	CGT	File Edits
friendGenerator.py	300	10	30
friendRecommender.php	120	6	20
mathComputer.py	10	1	10

Reliability

01. Measuring Productivity

02. Debbie the developer

a. Velocity

b. Reliability

c. Knowledge

03. Other Metrics

04. Team Productivity Dashboard

05. What's Next?

06. Q&A

MACHINE LEARNING IS NEAT: PARSLEY OR CILANTRO?

Parsley



Cilantro



Parsley



DIFF INTENT, CGT, AND RELIABILITY



File	Bug Fixes	Features
fileB.py	2	0
fileA.py	1	0
fileC.py	0	1

DEBBIE



Knowledge

01. Measuring Productivity

02. Debbie the developer

a. Velocity

b. Reliability

c. Knowledge

03. Other Metrics

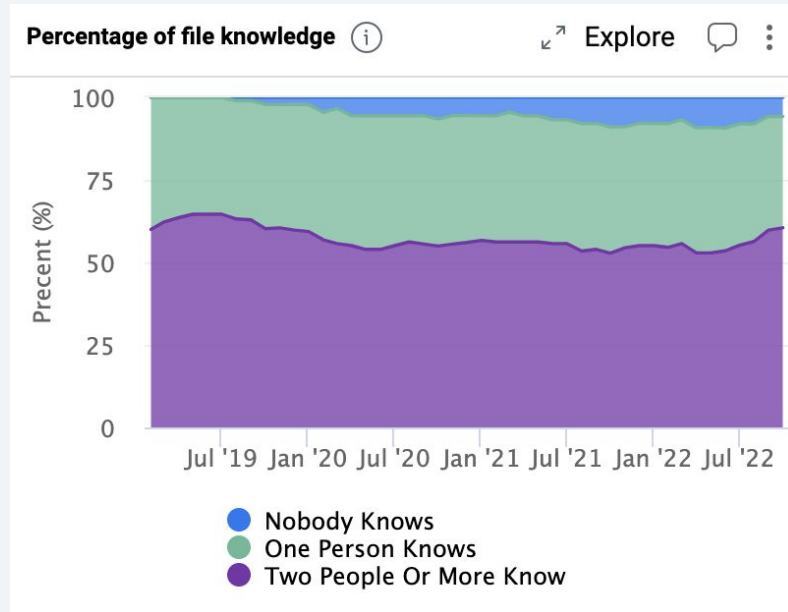
04. Team Productivity Dashboard

05. What's Next?

06. Q&A

FILE KNOWLEDGE

File	Knowledgeable Individuals
UILayer.js	['Debbie', 'Shrek']
CreditCardProcessor.java	['Debbie']
GiftCardProcessor.java	['Shrek']
BankProcessor.java	[]



FILE RELIABILITY AND INTENT



Reviewers

Debbie

Recommended reviewers:

Shrek

Other Metrics

01. Measuring Productivity

02. Debbie the developer

a. Velocity

b. Reliability

c. Knowledge

03. Other Metrics

04. Team Productivity Dashboard

05. What's Next?

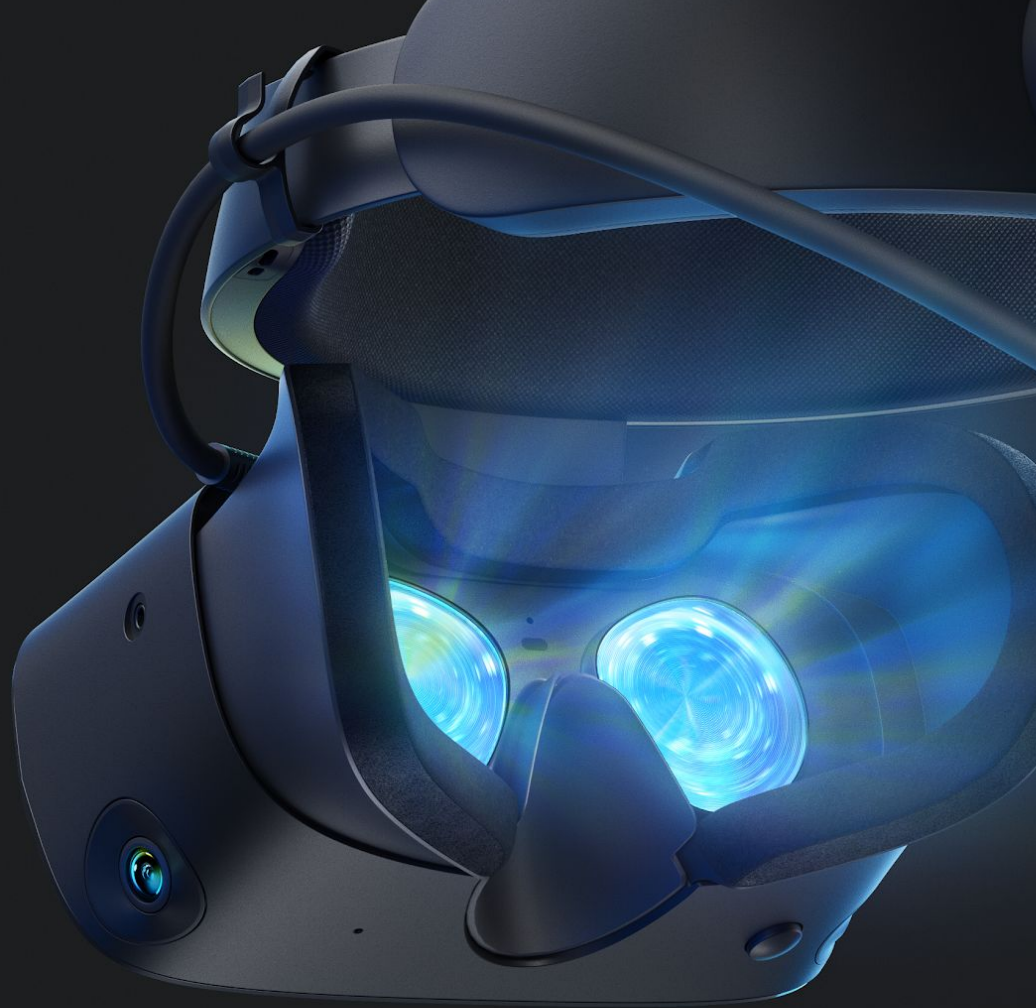
06. Q&A

WHAT CAN DEBBIE DO, NOW?



- Velocity
- Reliability
- Knowledge
- More!

PERSONAS



Team Productivity Dashboard

01. Measuring Productivity

02. Debbie the developer

- a. Velocity
- b. Reliability
- c. Knowledge

03. Other Metrics

04. Team Productivity Dashboard

05. What's Next?

06. Q&A

TEAM PRODUCTIVITY DASHBOARD

Productivity Metrics

- Velocity
- Oncall Load
- Sentiment

Investment Area Metrics

- Code Quality Metrics
 - Code Complexity
 - Test Coverage
- Tooling Duration
- Documentation Quality
- Meeting Hours
- Focus Time
- ...

TEAM PRODUCTIVITY DASHBOARD

Team Productivity Dashboard

[Mark] Code Size



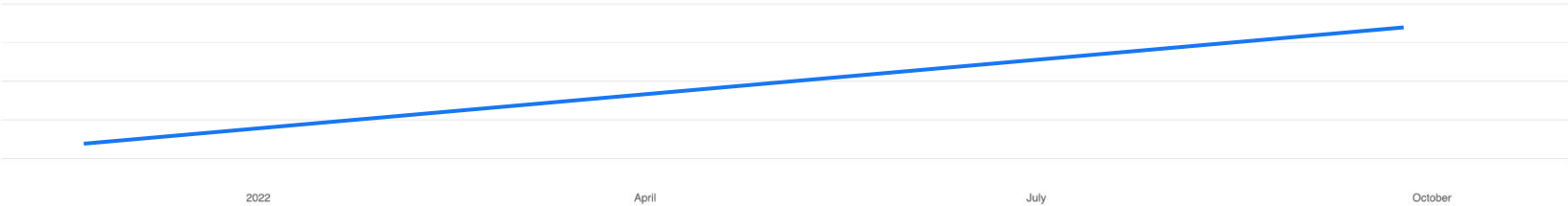
Filters(1) manager:is-any-of:[Mark Zuckerberg]



Latest Value

Compare

Last 360 days - now



Teams

Days

2 items selected

Find

Team	Sparkline	Latest Value	Goal	vs Goal
[Mark Zuckerberg] Team Metrics				
[Mark] Team Velocity Signals				
[Mark] Code Gestation Time				
[Mark] Investment Opportunities				
[Mark] Tooling Duration				
[Mark] Test Coverage				
[Mark] Code Size				
[Mark] Reference Data				

What's Next?

01. Measuring Productivity

02. Debbie the developer

a. Velocity

b. Reliability

c. Knowledge

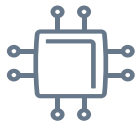
03. Other Metrics

04. Team Productivity Dashboard

05. What's Next?

06. Q&A

WHAT'S NEXT?



Predictions



Integrations

Q&A

- 01. Measuring Productivity
- 02. Debbie the developer
 - a. Velocity
 - b. Reliability
 - c. Knowledge
- 03. Other Metrics
- 04. Team Productivity Dashboard
- 05. What's Next?
- 06. Q&A

Workgraph: Personal Focus vs. Interruption for Engineers at Meta



- Paper scheduled for release soon!

Source:

<https://2022.esec-fse.org/details/fse-2022-industry/23/Workgraph-Personal-Focus-vs-Interruption-for-Engineers-at-Meta>

