Developer productivity 2.0

What works, and what does not?

Gergely Orosz / The Pragmatic Engineer

Hi, I'm Gergely

I worked at

I write

The Pragmatic Engineer

By Gergely Orosz

Big Tech and high-growth startups, from the inside. Highly relevant for software engineers and engineering managers, useful for those working in tech.

Developer productivity, platform teams: frequent enough topics

Access the full archive at
PragmaticURL.com/LeadDev

Developer productivity… what is it?


story point
jira
spent more time
optimised calendar
great dev
given initiative
slack
rapid triage
gained efficency
lost time
paramount importance
smart code completion
immature new feature
agile
argued decision
lead time
complicated workflow
dispersed team
important email
formatted code
velocity
automated testing
clear story
too many custom fields
noone has an overview
too many steps
incredible tool
great fan of tools
written doc
unwanted meeting
people have time
agile board
static analysis tool
good spreadsheet
team
unreliable tool
great burden
so many tools
too many different technologies
shared doc for guidelines
not committed company
improper use of tools
useful meeting
written standup communication
too many legacy
shared project plan
productive ci tool
poor grammar
used jira
terrible ux

What if we just started over?

Let's go on a journey

1. Three stories on developer productivity
2. What practices do productive teams follow?
3. Why is developer productivity such a hard topic?
4. What are things that don't work?
5. What are things that DO work?

1. Three stories on developer productivity


Skype

The year is 2012

- ~900 people
- Shipping about once per quarter
- “How can we speed things up?”

“Let’s roll out Scrum across the company”

- The best training money can buy
- From one of the Agile Manifesto founders
- (Probably) the biggest Agile/Scrum transformation up to that date

It was a win!

We got a lot faster!

- Visibly feeling faster!
- Shipping every 2-4 weeks
- Stack ranking projects across the company

#1 stack ranked project across Skype


XBOX ONE

Lesson at Skype: for better developer productivity, use Scrum!


WhatsApp

Back to Skype, in 2012

- “Should we be worried about WhatsApp?”
- “WhatsApp is really gaining on us.”
- “WhatsApp just launched video chat as well…”

WhatsApp overtook Skype, over time

So did WhatsApp use Scrum?

I asked cofounder and CEO Jan Koum

“to be honest i have no idea what this scrum garbage is and we never uttered the word Scrum when i was at WhatsApp”

Lesson at WhatsApp: for better developer productivity, you don't need Scrum

Laura Tacho

Formerly Senior Director of Engineering at CloudBees

From ‘Measuring Software Engineering Productivity’

- Introducing a tool to measure developer productivity at CloudBees (~300 people)
- “It was a flop”
- “My team felt like I didn’t trust them, and that this data was going to be weaponized against them if they had a slow week”

From ‘Measuring Software Engineering Productivity’

- “I bought into the idea that quantitative data is always better than qualitative data, though my teams had plenty to share about what slowed them down.”
- “The most useful insights into our development workflows – and specifically, what we could do to reduce friction and ship more code – came from 1-1s and retrospectives. My team knew where the pain was; they experienced it every day. “
- Full article: PragmaticURL.com/LeadDev

Lesson: tools that help measure developer productivity don't always work as you hope

What connects all these stories?

- Skype: use Scrum
- WhatsApp: don't use Scrum
- CloudBees: listen to your team, over using a vendor

???

There's no real connection

- in fact, some contradict each other!

Let's gather more data, shall we?

2. What makes productive teams? A survey.

Gathering more data

- A survey on practices that (un) productive engineering teams use
- 17 questions
- ~100 pages worth of text responses

Who filled out the survey?

- Engineering managers, software engineers, PMs
- Google, Facebook, Databricks, Auth0, Hopin, Cloudera, Elastic, Proton, Volvo, Bosch
- From small startups to large enterprises

Who filled out the survey?

- 75 responses
- 35 engineering managers
- 34 software engineers
- 5 product managers & 1 design manager

“What are tactics that made your team more efficient?”

Engineering managers:

Engineers:


test
team brainstorming session
project
mountain view team
relevant ticket
proactive weekly meeting
topical standup
full day work dev
meeting
relevant production db copy
productivity experience
awesome code review
good speccing
daily standup version
task of stories
team member
handled ongoing work
po for sync
ready to submit
effective asynchronous collaboration practice
team
focused disagreement
flexible schedule
good devop tooling
weekly alignment meeting
task
change
less merge conflicts
friendly chat
caring relationship
pair programming
accepted question
motivating team member
good testing approach
written stand-up
huge new user problem
fast prototyping
mountain view time
critical button
work
cl
prioritized backlog
round of reviews
good note
asynchronous collaboration practice
more frequent integrations
prioritize pr review
not blocked team
> knowledge transfer
critical pr comment

PMs:


preparation
manager
right people
daily
refinement
clear leadership
review work
single representative
retrospective
daily stand
realistic target
engineer
mvp top
target
context
team member
stand-up
meeting
action plan
owner
retrospective has an owner
reflective retro
larger organisations
work
pr
dev
faster review works
sprint goal
scrum
team
note
office day
bone
focus day
term of goals
depth problem
home
waste of time
strategy
right context
session
week coaching
ticket
review
point
retro
leadership
standup

All responses:

“What are strategies that helped increase productivity?”

All responses:

What are strategies that helped increasing productivity?

- Ownership & autonomy
- Focus
- Clarity, clear vision
- Having a plan
- Great people: hiring well, growing them

What are strategies that helped increasing productivity?

No mention of:

- Processes like Scum, Kanban, Agile (by the book)
- Meetings mentioned surprisingly little

“What are strategies and tactics that reduced productivity?”

All responses:

“What are tools that increased productivity?”

All responses:

“What are tools that decreased productivity?”

All responses:

"What was the most productive engineering team you've worked on?"

“What was the most productive engineering team?”

- Clear goals, clear direction
- Small team
- Good balance between autonomy & business needs
- Healthy team, good practices
- Focus

~100 pages worth of responses

Still hard to see things
that work for all teams

3. Why is developer productivity such a hard topic?

Max Kanat-Alexander
Principal staff engineer at LinkedIn

Formerly tech lead on Google Code Health

Because it's a people problem

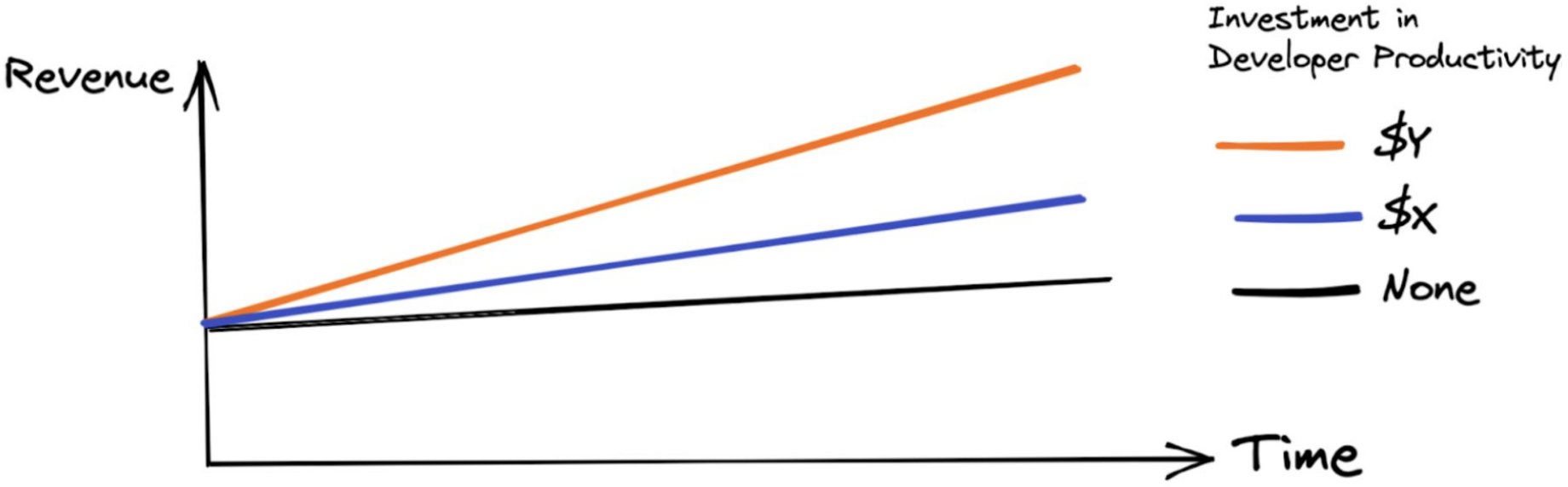
- Developer productivity is not a technical problem
- It's a people one
- People are complicated!

Because it's hard to make the right argument on what to fix

- Leadership wants to fix "developer productivity"
- This is what they will be happy to fund:

Revenue generated by the engineering team

(assuming flat headcount)



Because it's hard to make the right argument on what to fix

- But we cannot deliver on that promise!
- The reality is that we can improve what we actually control...
- And we don't control a lot of things that we would need to improve
- Like distractions/interruptions

Because we are probably not even measuring the right thing.

- What IS developer productivity? What is productivity?
- It means something different for an ML engineer vs an Go backend engineer....

4. What are things that don't work?

Gergely Orosz
@GergelyOrosz

Incredible how most engineering managers don't realize how measuring developer productivity by visualizing JIRA+git stats is a dead-end if you want truly high-performing teams.

This path works just like mandating that teams use Scrum. Yes: it helps bad teams get better.

4:34 PM · Apr 28, 2022 · Twitter Web App

View Tweet analytics

323 Retweets **61** Quote Tweets **2,428** Likes

5. What are things that DO work?

1. Focus on value created for the business

This is why we are here!

2. Get feedback from your team

What works? What does not?

3. Qualitative feedback

- People describing what works best
- Hard to do this well, because it's not numbers.
- A startup with a promising approach: **DX** (GetDX.com).
  Disclaimer: I'm an investor & advisor

4. Eliminate the biggest time wasters!

Your team already knows what these are.

5. Invest for leverage

- How can engineers focus more?
- In-house: Developer Experience, Platform teams
- Vendors: tools to iterate faster, catch regressions faster
- Shift left in testing/regressions.
  (e.g. what **mobile.dev** is doing for mobile engineering.
  Disclaimer: I'm an advisor)

6. Take your team on the journey with you

“You need to take your team along with you on this journey. In all of these scenarios, success starts before measuring a single thing. If you’re tempted to jump on a sales call to evaluate Pluralsight Flow, LinearB, or any of the other dashboard-as-a-service companies, I urge you to put away your credit card for now. “

- Laura Tacho

7. Iterate, iterate, iterate!

- What works today won't work when your team is bigger/smaller
- Keep improving, one thing at a time!

The things that work:

1. Focus on value created for the business
2. Get feedback from your team
3. Qualitative feedback
4. Eliminate the biggest time wasters!
5. Invest for leverage
6. Take your team on the journey with you
7. Iterate, iterate, iterate!

Thank you!

Gergely Orosz / The Pragmatic Engineer

Get a free month of The Pragmatic Engineer:

PragmaticURL.com/LeadDev