DOT NET MLG

Measuring Developer Productivity: the good, the bad and the data

Verisk™ SECTOR ALARM prodware LEMONCODE METADEV plain concepts

Marco Pierobon

Technologist, tinkerer and animal rights activist.

Principal dev @ Thoughtworks_Spain

@pierobond.bsky.social

thoughtworks

Thoughtworks' purpose

To create an extraordinary impact on the world through our culture and technology

Our core values:

Integrity / Pursuit of Excellence / Global First / Cultivation / Curiosity / Autonomous Teams / Inclusivity / Courageous

Be an awesome partner for clients and their ambitious missions

Revolutionize the technology industry

Amplify positive social change and advocate for an equitable tech future

Foster a vibrant community of diverse and passionate technologists

Achieve enduring commercial success and sustained growth

Technology excellence then, now and next

Thoughtworks
Incorporated

1993

ThoughtMobile
Hits the streets

1996

1st office outside U.S. for distributed agile

2000

cruisecontrol

First continuous integration server

2001

Inc. 5000

Named as one of the fastest-growing private companies for the fourth consecutive year

Se

Created Selenium, setting a new industry standard for cross platform cross browser web-app functional testing

500+
Thoughtworks

2004

Thoughtworks University
Run 60+ times since

2005

gauge

go

2006

25th book by Thoughtworkers

2008

1st edition of **Technology Radar**

2010

3,000+
Thoughtworks

2014

ANITA B.ORG

AnitaB.org names Thoughtworks Top Company for Women Technologists

2016 - 2018 and 2020

Apax PARTNERS

Apax Partners acquires Thoughtworks

2017

7,000+
Thoughtworks

2019

GEMINI SOLUTIONS FOURKIND

Thoughtworks acquires Gemini Solutions and Fourkind

2021

12,000+
Thoughtworks

2022

TWKS
Nasdaq Listed

Thoughtworks lists on NASDAQ

2021

Thoughtworks acquires Connected in Canada and Handmade in Brazil

2022

Connected Handmade

Named a leader in The Forrester Wave™: Modern Application Development Services, Q3 2022

Named a leader in The Forrester Wave™: Cloud Migration & Managed Service Partners in China, Q4 2022

Named a leader in The Forrester Wave™: Cloud Migration & Managed Service Partners in Asia Pacific, Q4 2022

2022

We wrote the book on it

Books written and digital publications

Rooted in a culture of learning and sharing, we believe that knowledge should be accessible to all. We are committed to improving the tech industry and are passionate about sharing our expertise across technology, business and culture.

100+

books written

Perspectives
A publication for digital leaders
Learn more

Technology Radar
An opinionated guide to today's technology landscape
Learn more

Digital Fluency Model
Discover your digital fluency
Learn more

Decoder
The A-Z guide to tech for business executives
Learn more

Looking Glass
The trends your business should focus on today and in the future
Learn more

Agenda

- Intro
- The Good
- The Bad
- The Data

thoughtworks

INTRO

thoughtworks

/thoughtworks

Notions (1 / 2)

“When a measure becomes a target, it ceases to be a good measure.”
— **Goodhart’s Law**

“Culture eats strategy for breakfast”
— **Peter Drucker**

“Observe that for the programmer, as for the chef, the urgency of the patron may govern the scheduled completion of the task, but it cannot govern the actual completion.”
— Frederick P. Brooks Jr., The Mythical Man-Month: Essays on Software Engineering

When we say “developer productivity”, we mean the degree to which developers' can do their work in a frictionless manner – not the individual performance of developers.

/thoughtworks

Notions (2/2)

- Inputs/Effort
- Outputs
- Outcomes
- Impacts

THE GOOD

thoughtworks

Focus on Outcomes, Impacts & Quality

Considering outcomes and impacts

Tracking delivery speed, customer oriented outcomes

Productivity is the amount of value delivered per unit of time or cost

Check ‘value per unit time’

Compare team’s productivity?

Holistic, Real-time, Data-Driven Methods

Data-driven methods helps in accurate measurement of developer productivity

Qualitative and quantitative metrics like SPACE and DORA provide a more holistic perspective

How do you measure tech debt?

EEBO Metrics to create a line of sight from engineering excellence to business outcomes

Team-Based Productivity Measurement

More accurate insights due to the collaborative nature of software development

Avoid focusing on individual performances

Prioritise work-life balance, reducing stress and physical and mental health

Invest in the best tools and processes

THE BAD

thoughtworks

Misuse of Productivity Metrics

Lose developers' trust

Focusing on simple inputs, outputs or efforts

Gaming the system

Developers should just code 70% of the time

Negative impact on engineering culture/surveillance culture

Focus on Individual Performance

Manipulative behaviours

Wrong perception of the team's productivity

Rank developers

Outdated Mindsets & Vanity Metrics

Inaccuracies in tracking

Not taking a human centered approach

Not considering qualitative data

Command & Control

Lack of Focus on Key Aspects

Employee wellbeing and job satisfaction

Inefficient workflows

Tech debt

Insufficient documentation

The data

Strong positive association	+ +
Positive association	+
Not associated	+ −
Negative association	−
Strong negative association	− −

thoughtworks

Establish a healthy culture

Generative culture

Team, org & op performance	+ +
SW delivery performance	+ +
CI & CD	+ +
Job satisfaction and productivity	+ +
Burnout	– –

Leaders create incentive structures to promote a generative culture

↕

Individual contributors drive technical capabilities that improve performance

Teams with generative cultures have 30% higher organizational performance than teams without

Build with users in mind

Product, Ops and Platform teams need to think of their users

Team performance and productivity + +

Focusing on the user leads to 20% higher job satisfaction

Teams with strong user-focus have 40% higher organizational performance

Unlock software delivery performance with faster code reviews

Code Review speed

Delivery and op performance	+ +
Burnout	– –

Loosely coupled architectures

Team and org performance	+ +
Burnout	– –

AI

Individual well being measures	+
Group level outcomes	+ –

Teams with shorter code review times have 50% better software delivery performance

Amplify technical capabilities with quality documentation

Individual metrics

Job satisfaction and productivity + +

Burnout – –

Group level outcomes

Team, Org and Op performance + +

12x performance increase for teams using TBD

Watch out for underrepresented minorities

Reliability unlocks performance

Even small levels of reliability practices can result in operational performance improvements

Stick it out through the lull.

- Reducing toil
- Blameless postmortems
- Team autonomy
- Sublinear scaling of teams

Nonlinear relationship between operational performance and practices like automation

Increase infrastructure flexibility with cloud

Flexible infrastructures predict 30% higher organizational performance than inflexible infrastructures

How does a team use the cloud?

Public clouds?

Flexible infrastructures

Org and team performance + +

Software delivery performance + +

Distribute work fairly

Significantly increases the quality of documentation

Group level outcomes

Team and org performance + +

TBD, CI, CD + +

Individual metrics

Job satisfaction and productivity +

Burnout –

Eliminates the difference in burnout across genders

DevEx

Developer productivity increases with:

- Clear goals for projects
- Psychological safety

Improving DevEx increases:

- Productivity
- Satisfaction
- Engagement
- Employee retention

Org level:

- Build/Delivery time
- Complex infrastructure
- Self service

Team level:

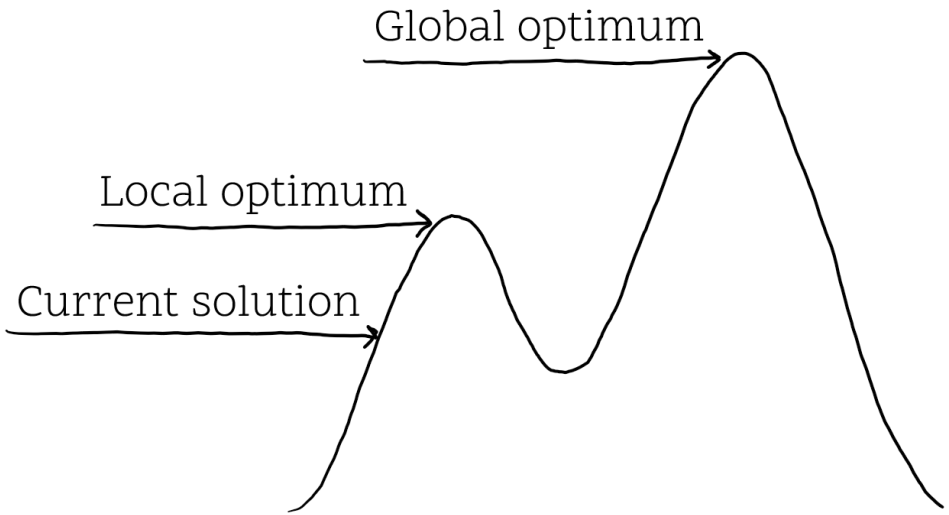
- Productivity
- Satisfaction
- Engagement
- Employee retention

SUMMARY & CLOSING THOUGHTS

thoughtworks

/thoughtworks

Local vs Global Optima



/thoughtworks

How to select your metrics

1. Charter for the DevEx team
2. Define the signals to look out for to achieve the charter.
   How easy?
   How quickly?
   With what quality?

/thoughtworks

How to select your metrics

3. For each category, define metrics to help track how it's going.

Speed: Perceived Delivery Speed, Perceived Productivity

Ease: Ease of Delivery, Deployment Lead Time, Build Failure Rate

Quality: Incident frequency, Perceived Software Quality

/thoughtworks

How to select your metrics

4. Add operational metrics

e.g.

- Developer satisfaction with specific tools
- Adoption rate of a particular service

Please give me feedback!

REFERENCES & EXTRA MATERIAL

thoughtworks

References

Yes, you can measure software developer productivity (mckinsey.com)
Measuring developer productivity? A response to McKinsey (pragmaticengineer.com)
Measuring Developer Productivity: Real-World Examples (pragmaticengineer.com)
A new way to measure developer productivity – from the creators of DORA and SPACE (pragmaticengineer.com)
What McKinsey got wrong about developer productivity (leaddev.com)
Can developer productivity be measured? (stackoverflow.blog)
A Human-Centered Approach to Developer Productivity (ieee.org)
The End to the Myth of Individual Programmer Productivity (ieee.org)
How Developers and Managers Define and Trade Productivity for Quality (arxiv.org)
You Are What You Measure (hbr.org)

References

A Guide to Measuring Developer Productivity (thenewstack.io)
The 10x Programmer: Is Individual Productivity Overrated? (thenewstack.io)
Measuring Developer Productivity via Humans (martinfowler.com)
Is it worth measuring software developer productivity? CIOs weigh in (cio.com)
Should You Measure Developer Productivity? (devops.com)
Eebo Metrics (eebo.org)
State of DevOps 2023 (google)
Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations (IT Revolution)

DORA Metrics

SPACE Metrics

	Satisfaction & Well-being	Performance	Activity	Communication & Collaboration	Efficiency & Flow
Individual	• Developer Satisfaction • Retention	• Code Review Velocity	• Focus Time • # Commits • # Issues / PBIs • Lines of Code	• Code Review Score (quality) • PR Merge Times	• Knowledge Sharing • X-Team Reviews
Team	• Developer Satisfaction • Retention	• Velocity (shipped) • Delivery Lead Time	• Cycle Time • Velocity (done) • # Issues / PBIs	• Code Review Engagement • PR Merge Times • Meeting Quality	• Code Review Stale Time • Handoffs
System	• Satisfaction with Engineering System	• Velocity • Lead Time • Customer Satisfaction • MTTR	• Deployment Frequency	• Knowledge Sharing • X-Team Reviews	• Lead Time • Velocity

Qualitative Metrics (e.g.)

How easy or difficult is it for you to do work as a developer or technical contributor at [INSERT ORGANIATION NAME]?

- Very difficult
- Somewhat difficult
- Neither easy nor difficult
- Somewhat easy
- Very easy

For the primary application or service you work on, what is your lead time for changes (that is, how long does it take to go from code committed to code successfully running in production)?

- More than one month
- One week to one month
- One day to one week
- Less than one day
- Less than one hour

How often do you feel highly productive in your work?

- Never
- A little of the time
- Some of the time
- Most of the time
- All of the time

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
My team follows development best practices	◻	◻	◻	◻	◻
I have enough time for deep work.	◻	◻	◻	◻	◻
I am satisfied with the amount of automated test coverage in my project.	◻	◻	◻	◻	◻
It's easy for me to deploy to production.	◻	◻	◻	◻	◻
I'm satisfied with the quality of our CI/CD tooling.	◻	◻	◻	◻	◻
My team's codebase is easy for me to contribute to.	◻	◻	◻	◻	◻
The amount of technical debt on my team is appropriate based on our goals.	◻	◻	◻	◻	◻
Specifications are continuously revisited and reprioritized according to user signals.	◻	◻	◻	◻	◻

Please share any additional feedback on how your developer experience could be improved

[open textarea]

McKinsey's metrics

Adding a focus on opportunities to software developer productivity metrics can offer clearer paths to improvement.

Focus areas by level

● DORA[1] metrics ● SPACE[2] metrics ● Opportunity-focused metrics

	Outcomes focus *Are you delivering products satisfactorily?*	**Optimization focus[3]** *Are you delivering products in an optimized way?*	**Opportunities focus** *Are there specific opportunities to improve how you deliver products, and what are they worth?*
System level	● Deployment frequency ● Customer satisfaction ● Reliability (uptime)	● Code-review timing ● Velocity/flow through the system	● Satisfaction with engineering system ● Inner/outer loop time spent
Team level	● Lead time for changes ● Change failure rate ● Time to restore service ● Code-review velocity	● Story points completed ● Handoffs	● Quality of documentation ● Developer Velocity Index benchmark[4] ● Contribution analysis
Individual level	● Developer satisfaction ● Retention	● Interruptions	● Contribution analysis ● Talent capability score

[1]Google's DevOps research and assessment team, which developed these outcome metrics.
[2]Satisfaction and well-being, performance, activity, communication and collaboration, and efficiency and flow; GitHub and Microsoft Research developed these metrics, which aim to look at developer well-being as a measurement at the individual level.
[3]Nonexhaustive.
[4]Benchmarks an organization's technology, working practices, and organizational enablement; see Shivam Srivastava, Kartik Trehan, Dilip Wagle, and Jane Wang, "Developer Velocity: How software excellence fuels business performance," McKinsey, Apr 20, 2020.

Metrics used by Tech companies

Developer Productivity Metrics at 17 Tech Companies

Company	Name of function	What they measure
Amplitude	Platform Engineering	Ease of Delivery, Engineer Engagement, Deploys per Service, Change Failure Rate
ATLASSIAN	Developer Experience	DSAT (Developer Satisfaction), Time from Commit to Deploy, Pull Request Cycle Time, Self-Serve Documentation, Self-Serve Dependency Maintenance
chime	Engineering Operations	Developer CSAT, Perceived Rate of Delivery, Ease of Delivery, Sentiment and Quantitative Metrics for 16+ DevEx factors
DOORDASH	Developer Platform	Developer Satisfaction, Adoption Rate, Stability of Services/Apps
Etsy	Enablement	Experiment Velocity, Availability, Performance, Developer Sentiment
GoodRx	Engineering Solutions	Engagement, Weekly Time Loss, Deep Work, Ease of Delivery, Speed, Stability, % Adoption of Standards
Google	Engineering Productivity	Speed, Ease, and Quality *Specific metrics for each of these dimensions vary
INTERCOM	Developer Experience	Ease of Delivery, Perceived Productivity, Engineer Engagement
Lattice	Developer Experience	Ease of Delivery, CI Pipeline Failure Rate, CI Pipeline P50 Build Time, Number of Deployments, Change Failure Rate
LinkedIn	Productivity & Happiness	Developer Build Time, Code Reviewer Response Time, Post-Commit CI Speed, CI Determinism, Deployment Success Rate, Developer NSAT (Net User Satisfaction)
Microsoft	Engineering Thrive	SPACE metrics *Specific metrics for each of these dimensions vary
Notion	Developer Infrastructure	Developer survey with a focus on perceptions of friction
PELOTON	Tech Enablement	Developer Satisfaction Score, Time to 1st and 10th PR, Lead Time, Deployment Frequency, % of PRs under 250 lines, Line Coverage, Change Failure Rate, Time to Restore Services
POSTMAN	Platform Engineering	Ease of Delivery, Perceived Productivity, Weekly Time Loss, Engineer Engagement, Cycle Times, Satisfaction
Spotify	Developer Experience	Self-Reported Productivity, % Adoption to Golden Standards, Deployment Frequency, Weekly Deployments per Weekly Active Developer
stripe	Developer Tools	Avg. Number of Days With Sufficient Focus Time, Branch Creation to Master Merge Time, Sentiment, Weekly # of PRs per Developer
Uber	Developer Platform	Weekly # of Diffs/PRs per Engineer, Weekly # of Code Reviews per Engineer, Weekly # of Design Docs Generated per Engineer, Avg. Weekly Focus Time per Engineer

Metrics that can be used to report to the C-suite

1. Business Impact What are we working on right now, and why?	**2. System Performance** How performant and reliable are our systems?	**3. Developer Effectiveness** How effective is the engineering organization and how can it be improved?
Current and planned projects, including: • Business rationale and success metrics • Project status (e.g. delivered, on track, at-risk)	• Application latency and uptime • Number of incidents • Server costs • User NPS	Measures of speed, ease, and quality: • Perceived Delivery Speed • Perceived Productivity • Weekly Time Loss • Deployment Lead Time • Ease of Delivery • Developer Satisfaction • Incident Frequency • Perceived Software Quality • Change Failure Rate

pragmaticengineer.com

EEBO Metrics

3 key metrics of excellence in software development

- Build failure rate
- Security warnings
- Tech debt

4 key metrics of excellence in production deployment

- Mean time to restore
- Change fail percentage
- Lead time
- Deployment frequency

4 key categories for business outcomes

- Improvement in efficiency & effectiveness
- Improvement in experience
- Increase in influence
- Market sensing