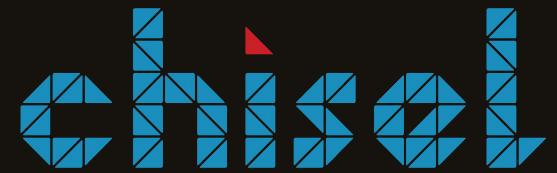


# **Measuring the Unmeasurable:**

## *A Design Science Approach to Understanding and Improving Developer Productivity*

Margaret-Anne (Peggy) Storey  
May 21, 2025  
Lund University



## **SOFTWARE ENGINEERING PRODUCTIVITY RESEARCH CONDUCTED AT STANFORD**

**9.5% OF  
SOFTWARE ENGINEERS  
ARE GHOSTS**

- Performance: <0.1x median engineer
- Do virtually no work
- Might work multiple jobs

The Quest to Measure Software Developer Productivity

The Space of Development Productivity

A Design Science Approach to Understanding and Improving Productivity

Developer Productivity in the age of GenAI

# ④ The Quest to Measure Software Developer Productivity

The Space of Development Productivity

A Design Science Approach to Understanding and Improving Productivity

Developer Productivity in the age of GenAI

1968 NATO Conference  
on the Software Crisis



### What is measured?

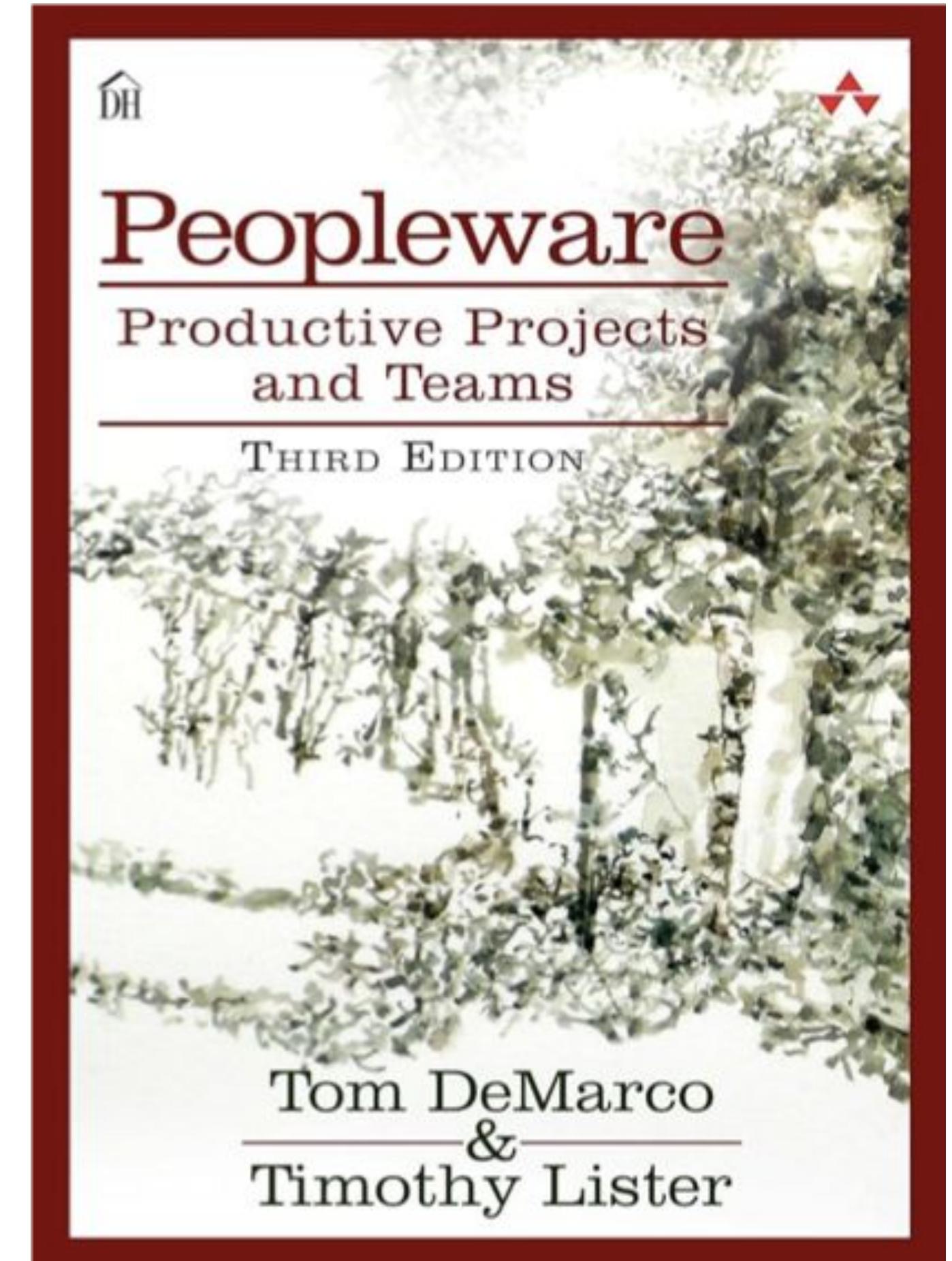
- a. **Performance:** Space, speed, throughput, turn around.
- b. **Language:** compliance with requirements, accuracy of object system. External Function: error isolation, configuration modularity/clear documentation, availability, installation ease, modification ease.
- d. **Internals:** serviceability, reliability (freedom from mistakes), conformity to standards.

# Dawn of software metrics

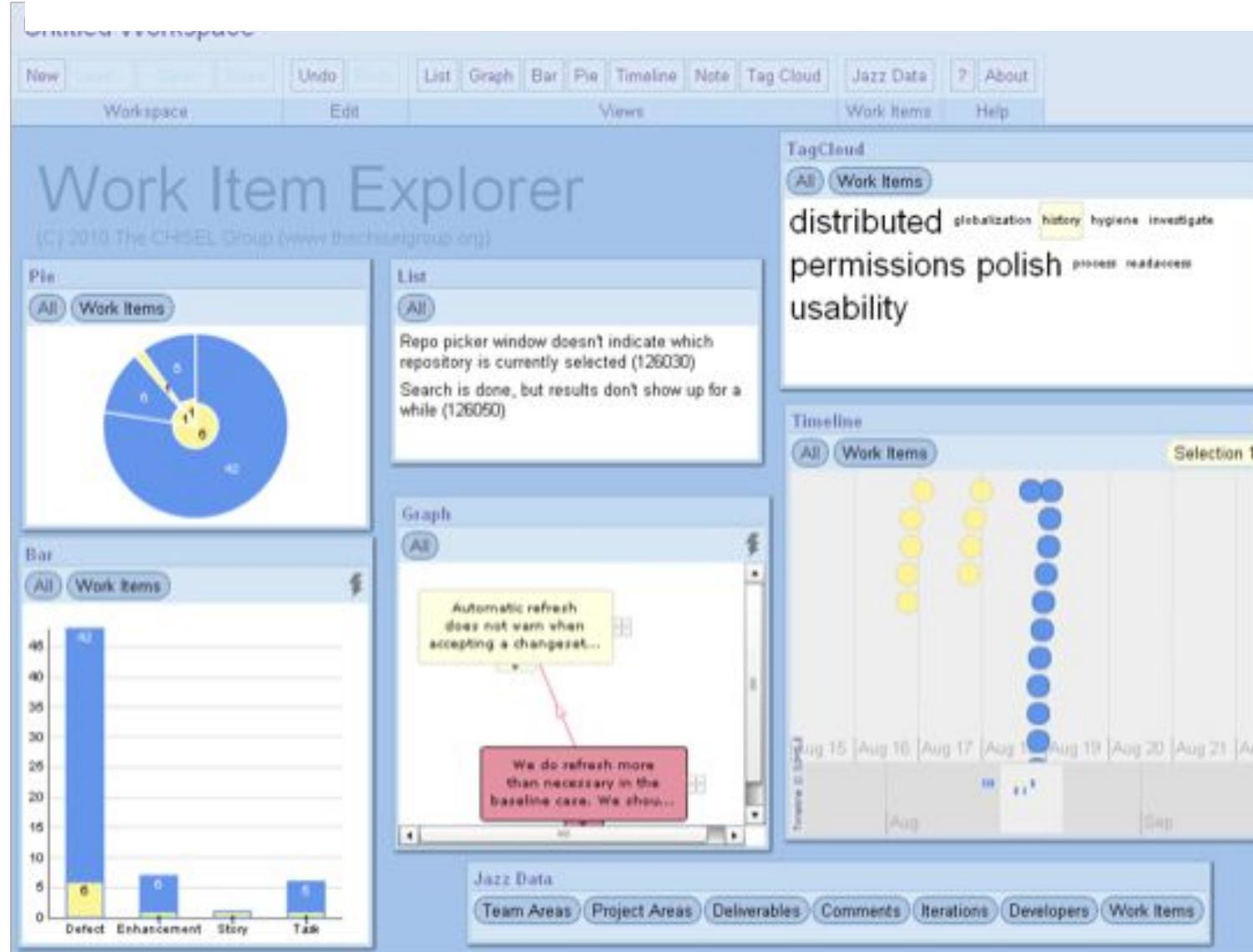


*"If you can't measure it,  
you can't manage it"*  
[Tom de Marco, 1982]

*"Only metric worth  
counting is **defects**"*  
[Tom de Marco, 1997]



# Early dashboards showed behavioral data from repositories



Treude & Storey, "Awareness 2.0: staying aware of projects, developers and tasks using dashboards and feeds", ICSE 2010

The figure shows two developer dashboards. The left dashboard is titled 'Issues' and lists a series of bugs with their IDs, summaries, and last touch dates. The right dashboard is titled 'Patches and Reviews' and lists patches with their IDs, bug IDs, flags, requesters, and last touched dates. Both dashboards have tabs for Activity, Assigned, Reported, CC, Comments, Patch Log, and Reviews.

BugID	Summary	Last Touched
862998	Add glue to allow Firefox first run page to highlight UI elements	Yesterday
872617	[meta] Australis Customization	Yesterday
924004	Ghost entry in customization palette, causes weirdness in menu panel.	Yesterday
912172	Call to xpconnect wrapped JSObject produced this error: * [Exception... "JavaScript Error: \"this.view.displayedFolder is null\" {file: \"chrome://messenger/content/folderDisplay.js\" line: 1071}]" when calling method: [nsIMsgSearchNotify:onSearchDone]	Yesterday
923165	Switch from the toolkit loading_16.png to our own thropper with retina support.	Yesterday
923857	Australis: Cus... [attachment_added(None->None)] [flags(feedback?(richard.marti@gmail.com)->None] buildArea	Yesterday
904719	items is unde... [flags(None->feedback?(richard.marti@gmail.com))]	Yesterday
546932	Add support for online address books using the CardDav format	2 days ago
922847	Move downloads animations into their own element rather than in a stack inside the button	2 days ago
881937	The Australis panel menu should be keyboard accessible	2 days ago
923738	Move the Awesomebar dropdown marker to the right of the go/stop/reload button.	2 days ago
768802	[adbe 3223393] Firefox window loses focus every time Flash plugin processes are (re)launched	2 days ago
428943	Site identity popup should link to explanation/support	2 days ago
900541	Contacts side bar: First address wrongly pre-selected when changing address book (risk of sending message to unintended recipients)	2 days ago

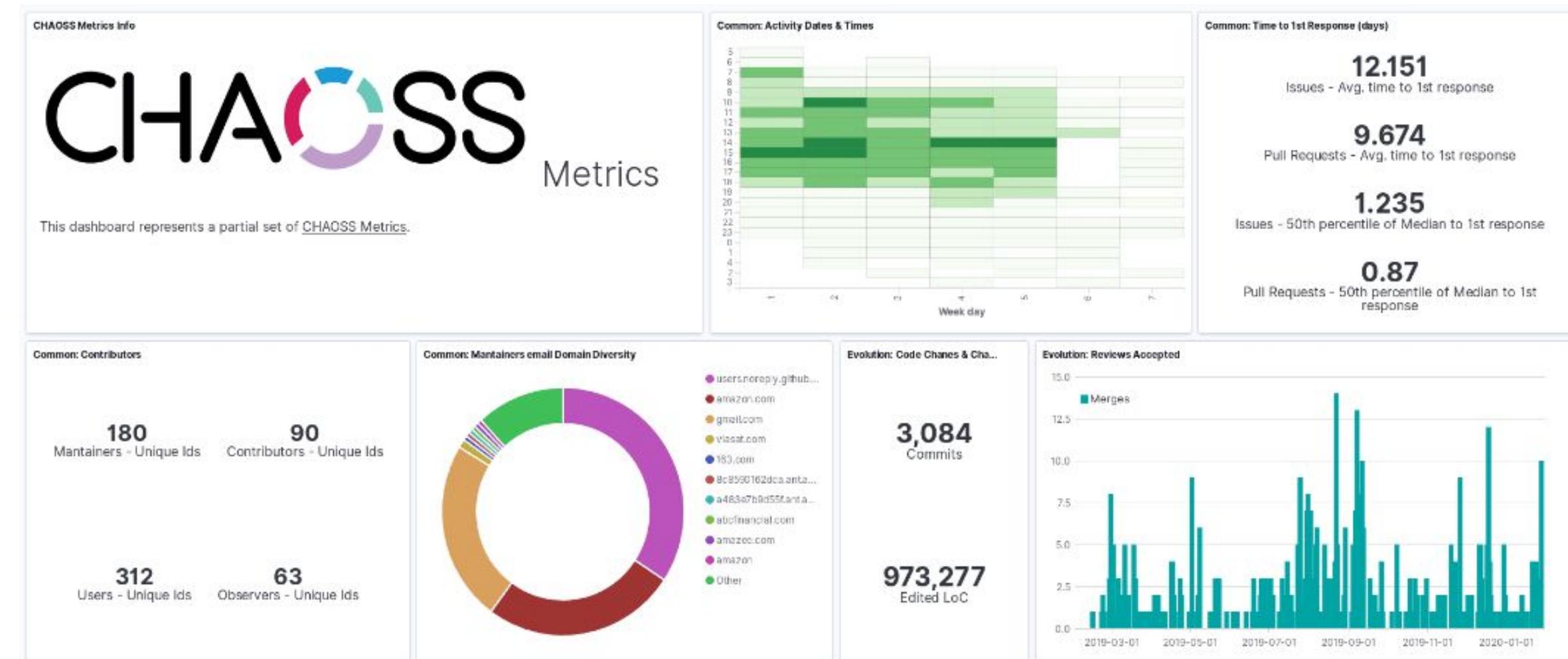
Patch ID	Bug ID	Flag	Requester	Last Touched
804788	912172	review-	ishikawa@yk.rim.or.jp	Yesterday
813055	922847	review+	gjjskruitbosch+bugs@gmail.com	2 days ago
778452	881937	review+	gjjskruitbosch+bugs@gmail.com	2 days ago
778503	881937	review+	gjjskruitbosch+bugs@gmail.com	2 days ago
783101	428943	review-	lie.r.min.g@gmail.com	2 days ago
813839	428943	review?	lie.r.min.g@gmail.com	2 days ago
805789	900541	review+	syshaganwal@gmail.com	2 days ago
806860	682901	review+	acelists@atlas.sk	3 days ago
804910	884805	review+	philipp@bugzilla.kewis.ch	3 days ago
813198	923186	review+	gjjskruitbosch+bugs@gmail.com	4 days ago
803332	529584	feedback+	acelists@atlas.sk	5 days ago
785696	529584	ui-review+	bugzilla2007@duellmann24.net	5 days ago
785696	529584	review+	bugzilla2007@duellmann24.net	5 days ago
804925	916482	review-	archaeopteryx@coole-files.de	5 days ago
804781	916358	review-	ishikawa@yk.rim.or.jp	5 days ago
802386	914610	review+	neil@httl.net	5 days ago
812120	845408	review+	gjjskruitbosch+bugs@gmail.com	5 days ago
806833	845408	feedback+	gjjskruitbosch+bugs@gmail.com	5 days ago
812976	733535	review?	ishikawa@yk.rim.or.jp	5 days ago
801321	733535	review-	ishikawa@yk.rim.or.jp	5 days ago
803361	897476	review+	philipp@bugzilla.kewis.ch	6 days ago
780415	897476	review+	philipp@bugzilla.kewis.ch	6 days ago

Kononenko, et al., "DASHboards: Enhancing Developer Situational Awareness", ICSE 2014

"We have all this data, the **problem** is what to do with it." [A Software Engineering Researcher]

# Developer experience/productivity dashboards today

## How People use Processes to create Products



<https://chaoss.community/>

# Working from Home: A Tale of Two Cities

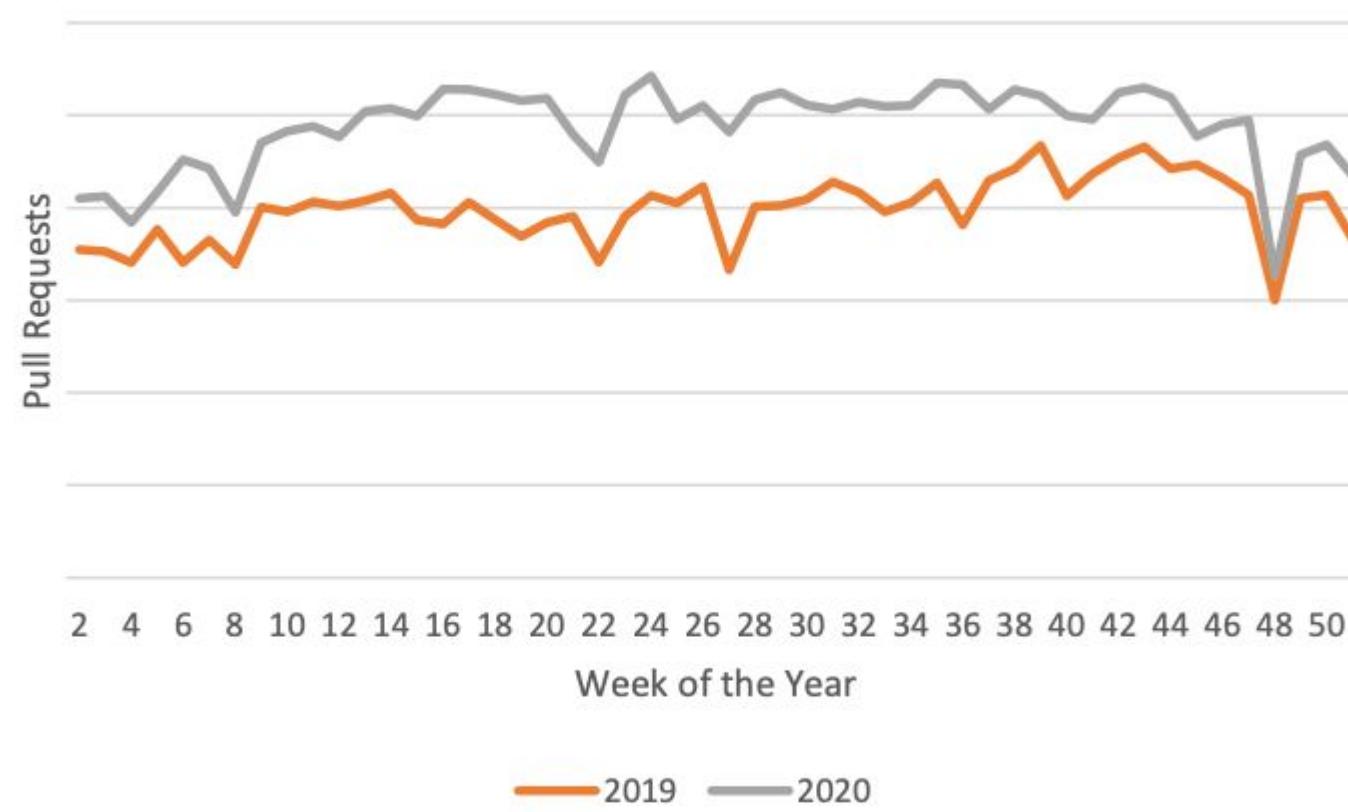
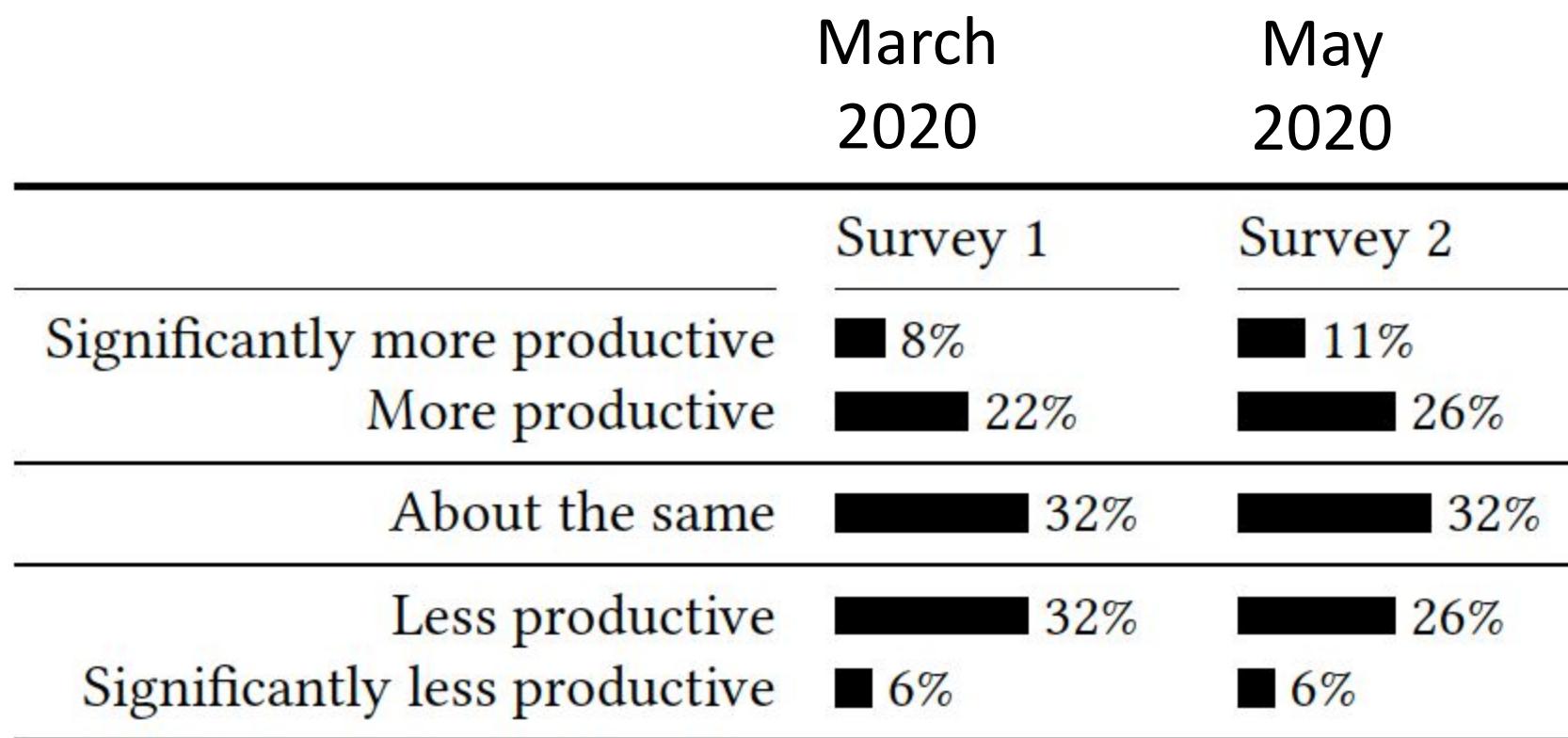


Figure 3: Weekly pull requests over time.



*“We’re doing very well, we’re barely hanging in there.”*  
— Shane O’Flynn

# Working from Home: A Tale of Two Cities



*“We’re doing very well, we’re barely hanging in there.”*  
— Shane O’Flynn

# Working from Home: A Tale of Two Cities

**Less distraction** from others, especially coming from an open office scenario.

Teams meetings for some meetings where very little participation is required lets you continue to **work while “attending”** and listening in, which is better than being idle in a conference room.”

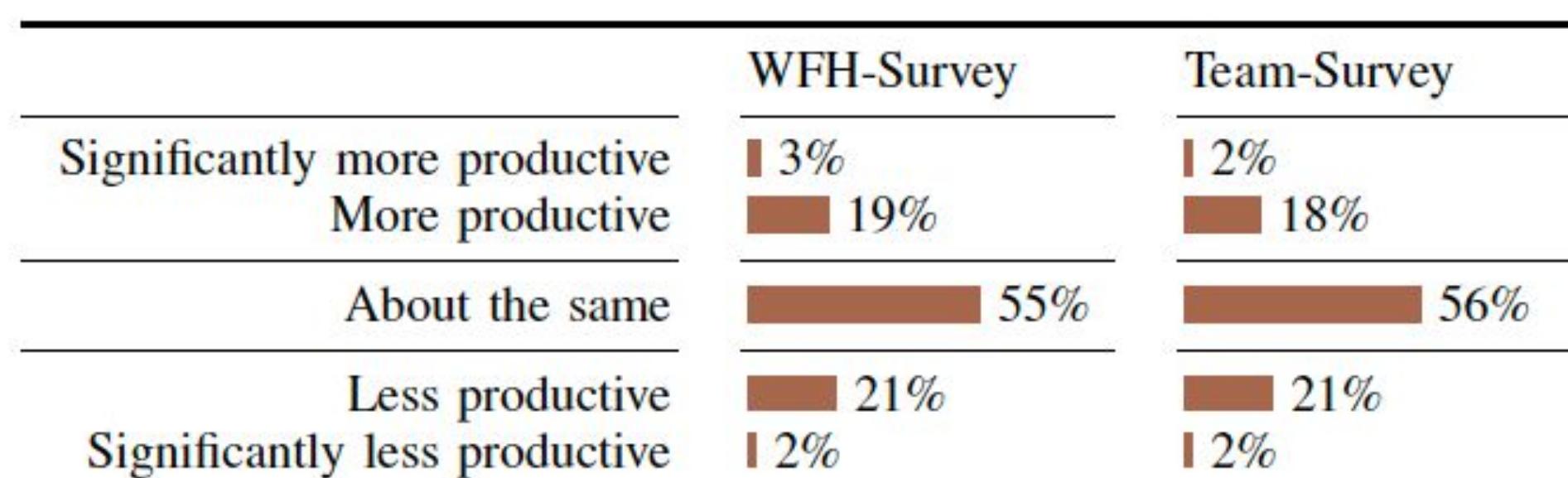
Dichotomous variables:  
focus time, worklife balance, childcare needs...

*I think we spend a lot **more time trying to coordinate** with each other and driving for clarity and **shared understanding is harder**. This means more time solidifying what we need to be doing and less time doing/building it.”*

***Interruptions** and concentration as I can [only] be reached on **Teams** and by email vs someone walking over for a question. Harder to keep **tabs on direct reports**.*

# *Team* productivity during work from home

CHANGES IN PERCEIVED TEAM PRODUCTIVITY.



*"I think we spend a lot **more time trying to coordinate** with each other and driving for clarity and **shared understanding is harder**. This means more time solidifying what we need to be doing and less time doing/building it."*

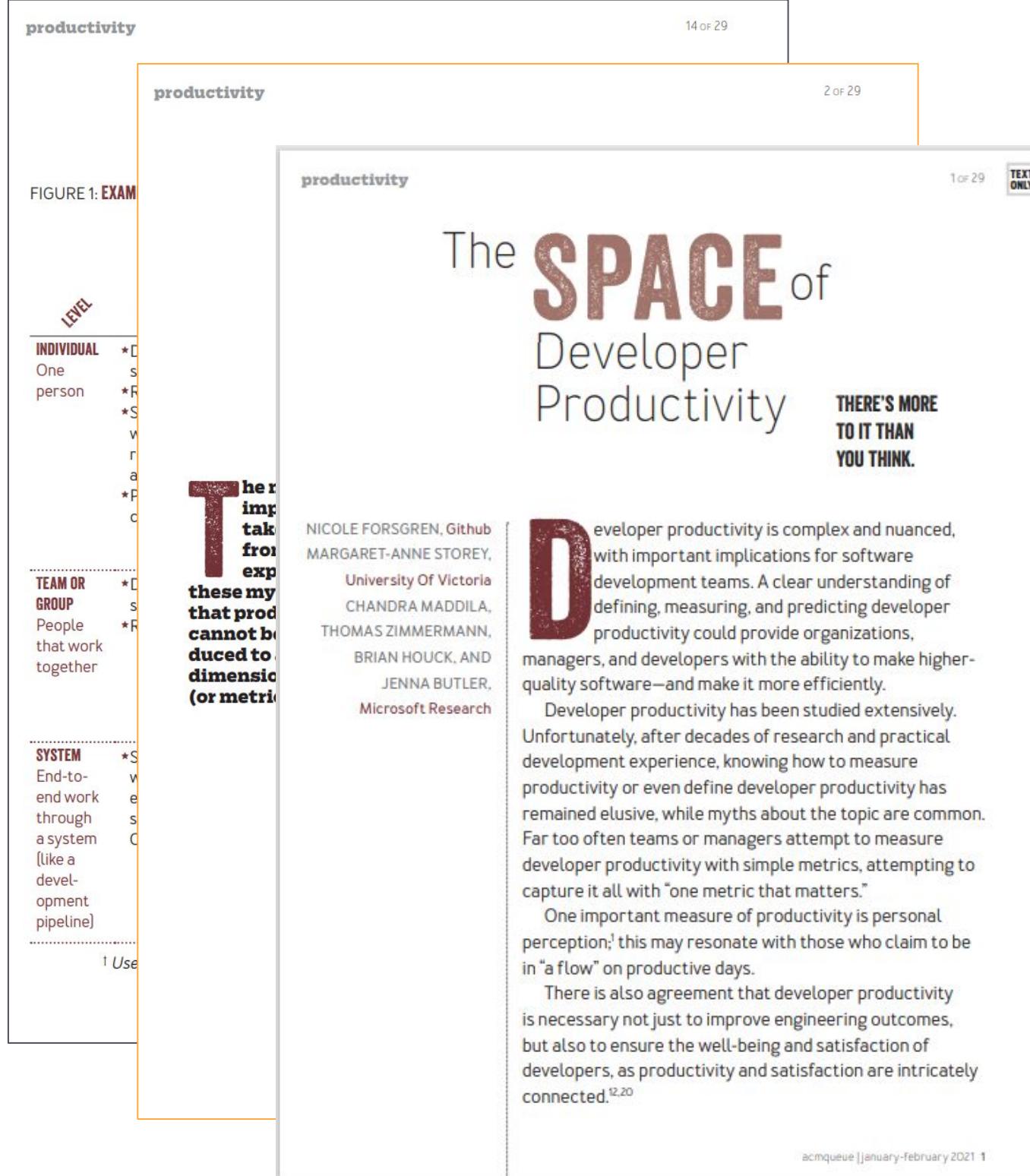
# The Quest to Measure Software Developer Productivity

## ④ The Space of Development Productivity

A Design Science Approach to Understanding and Improving Productivity

Developer Productivity in the age of GenAI

# SPACE: A framework for understanding productivity



# SPACE: A framework for understanding productivity



Satisfaction and well-being



Performance



Activity



Communication and collaboration



Efficiency and flow

How fulfilled developers feel with their work, team, tools, or culture

How healthy and happy developers are

Storey, Zimmermann, Bird, Czerwonka, Murphy. Kalliamvakou: Towards a Theory of Software Developer Job Satisfaction and Perceived Productivity. IEEE TSE 2021.

# SPACE: A framework for understanding productivity



Satisfaction and well-being



Performance



Activity



Communication and collaboration



Efficiency and flow

The outcome of a system or process. Hard to quantify performance because there are so many variables

# SPACE: A framework for understanding productivity



Satisfaction and well-being



Performance



Activity

The number of actions or outputs completed while performing work



Communication and collaboration



Efficiency and flow

# SPACE: A framework for understanding productivity



Satisfaction and well-being



Performance



Activity



Communication and  
collaboration

How people and teams communicate and  
work together



Efficiency and flow

# SPACE: A framework for understanding productivity



Satisfaction and well-being



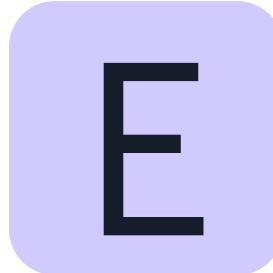
Performance



Activity



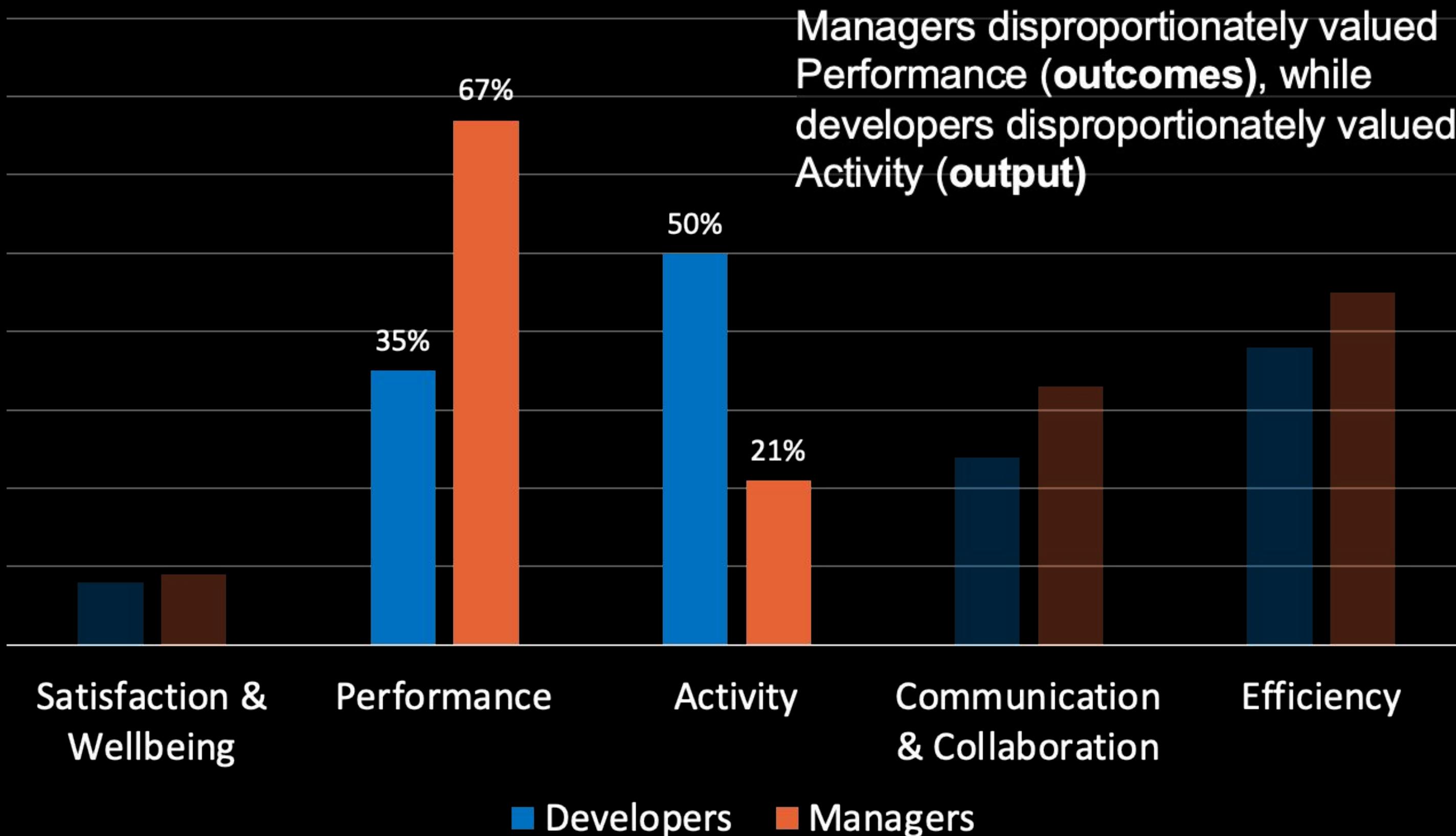
Communication and collaboration



Efficiency and flow

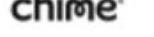
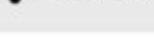
How well developers and teams can make progress and complete work without interruptions or delays

# How aligned are definitions of productivity?



Storey, Houck, Zimmermann:  
How Developers and Managers  
Define and Trade Productivity  
for Quality. CHASE 2022.

## Developer Productivity Metrics at 15+ Top Companies

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

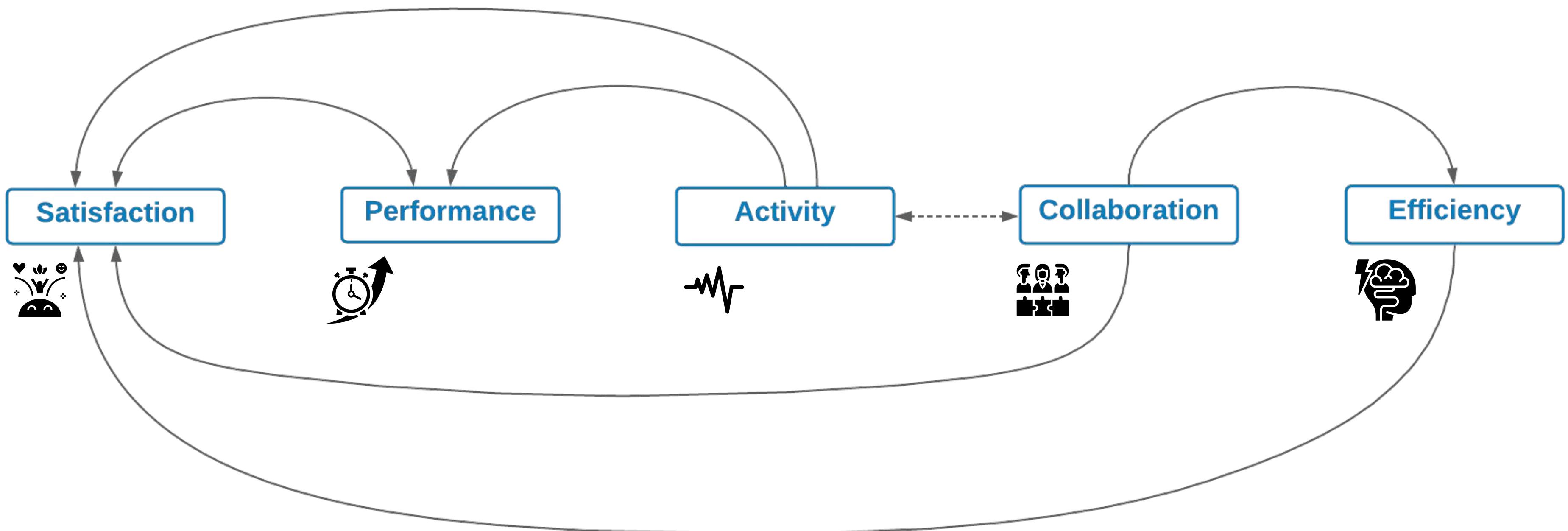
getdx.com

Ghosts may be  
doing lots of this!

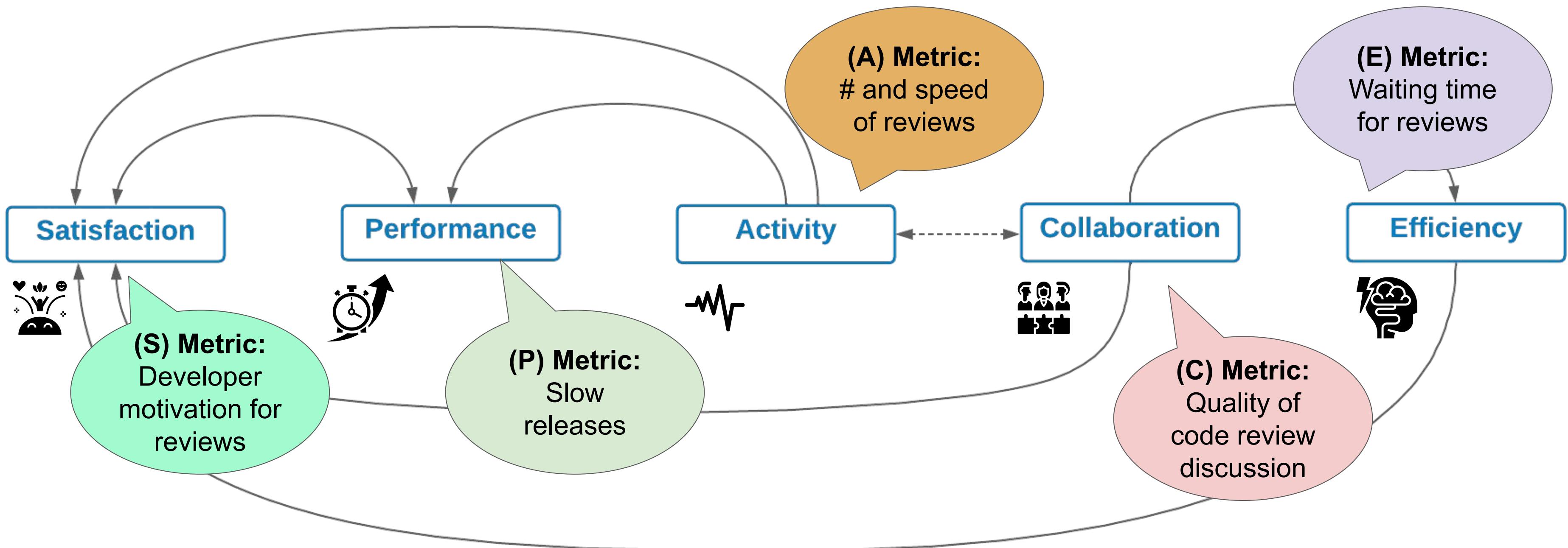
S  
P  
A  
C  
E

<https://getdx.com/uploads/developer-productivity-metrics-at-top-tech-companies.pdf>

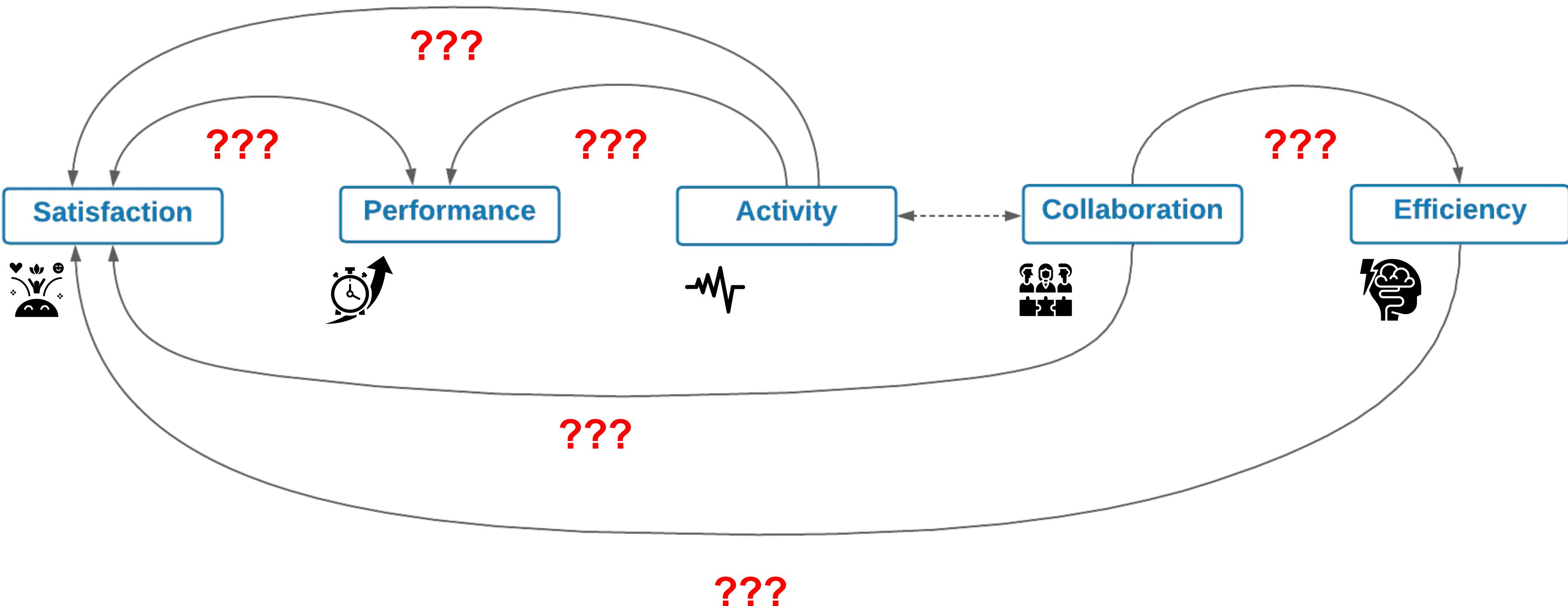
# Understanding slow **code reviews** across the five dimensions?



# Understanding slow code reviews across the five dimensions?



# Understanding how interventions may improve code reviews across the dimensions?



The Quest to Measure Software Developer Productivity

The Space of Development Productivity

④ **A Design Science Approach to Understanding and Improving Productivity**

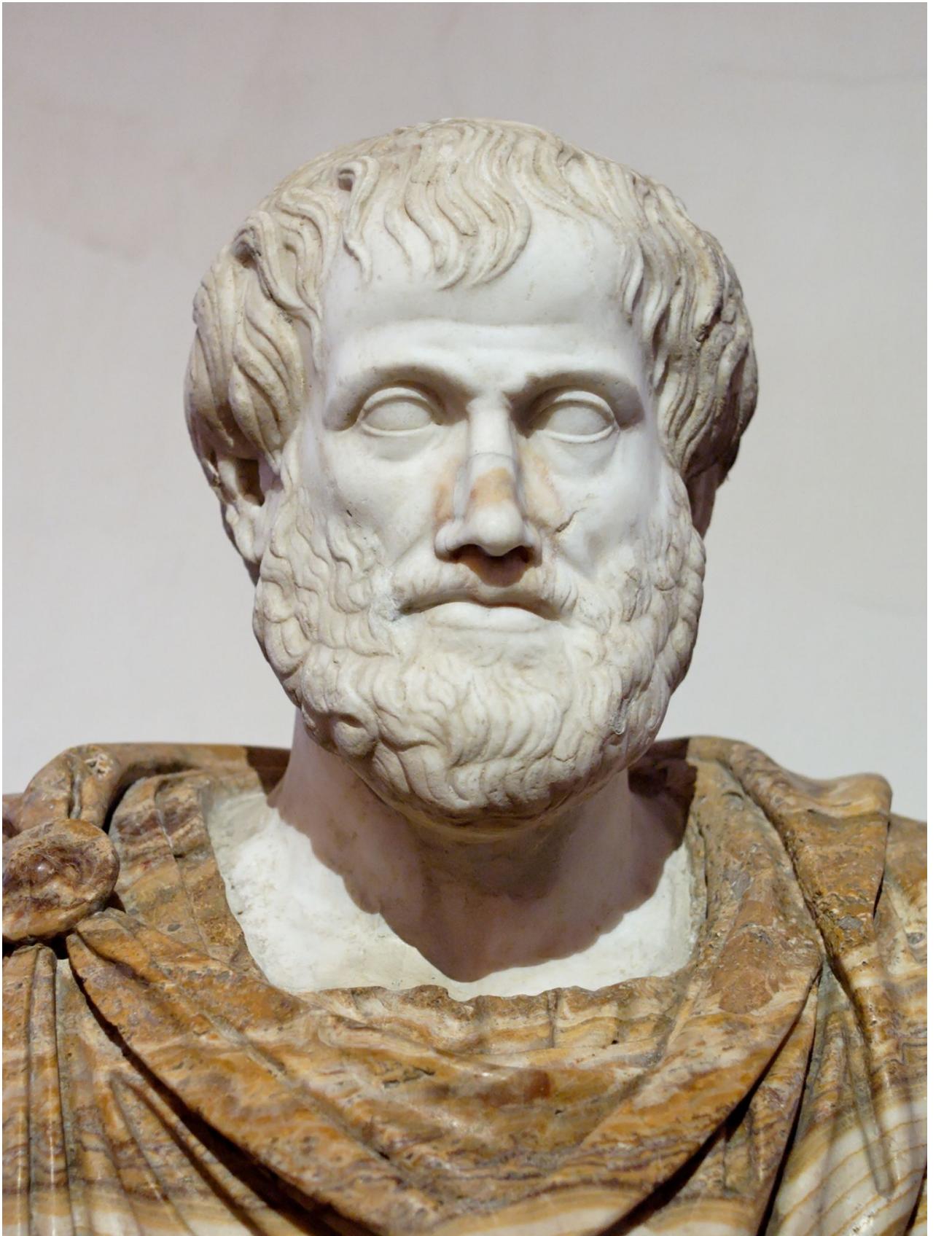
Developer Productivity in the age of GenAI

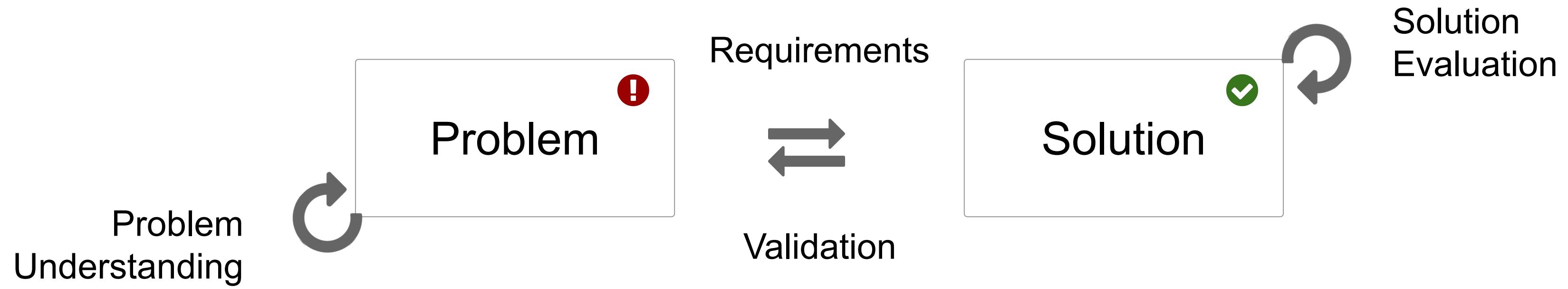
# Theories in software engineering research

**Descriptive** (what *is* happening)

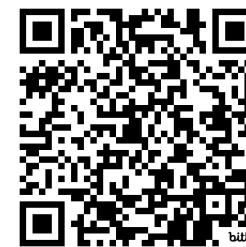
**Predictive** (what *will* happen)

**Prescriptive** (what *should* happen  
if we do something)

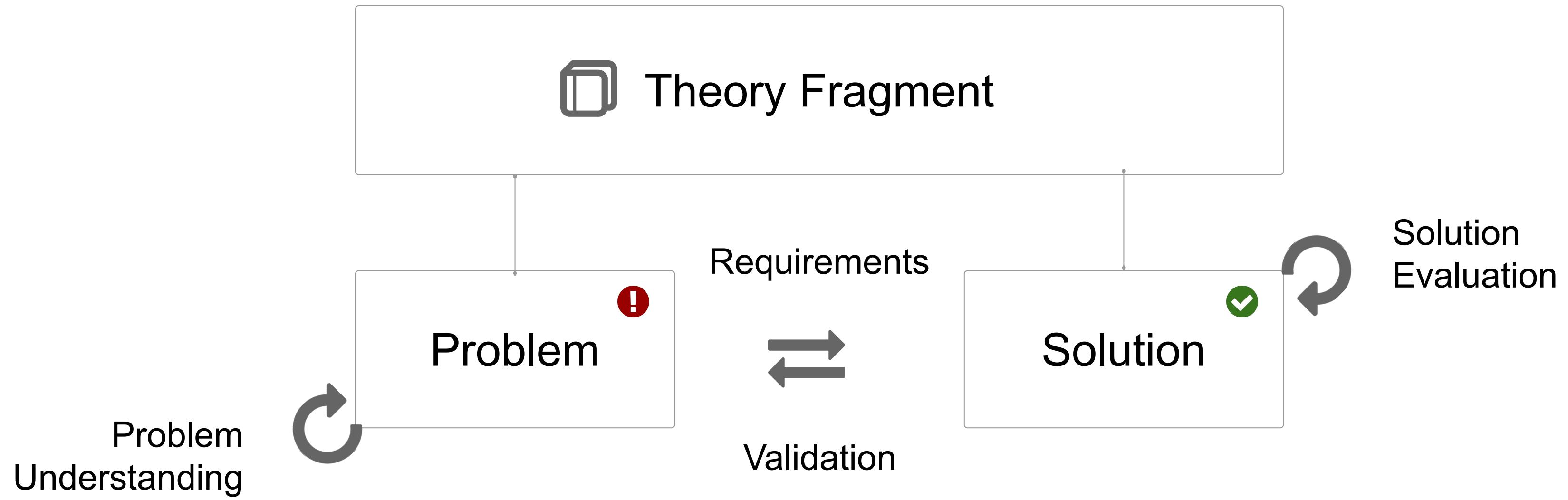




# Design Science



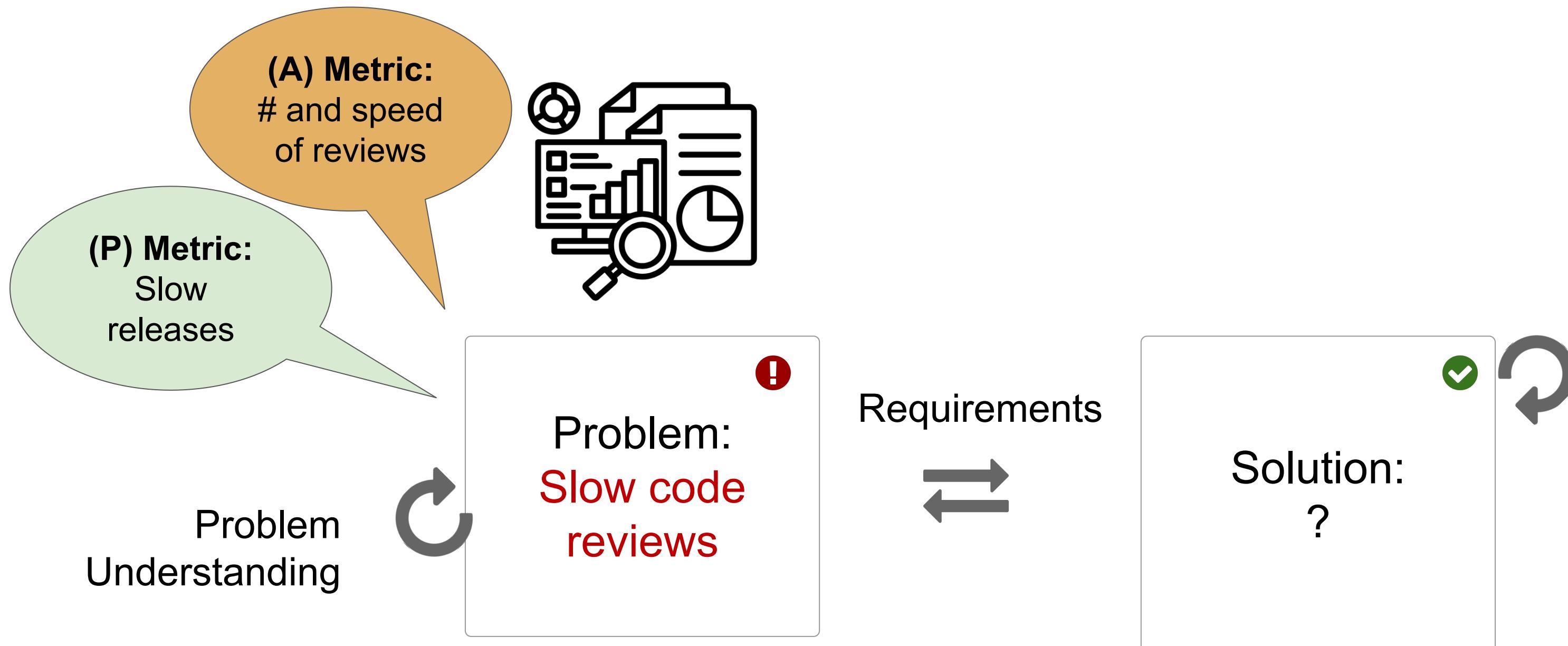
Engström, E., Storey, M.A., Runeson, P., Host M., Baldassera M.,  
How software engineering research aligns with design science: a review.  
*Empirical Software Eng* (2020).



# Design Science



Engström, E., Storey, M.A., Runeson, P., Host M., Baldasserra M.,  
How software engineering research aligns with design science: a review.  
*Empirical Software Eng* (2020).

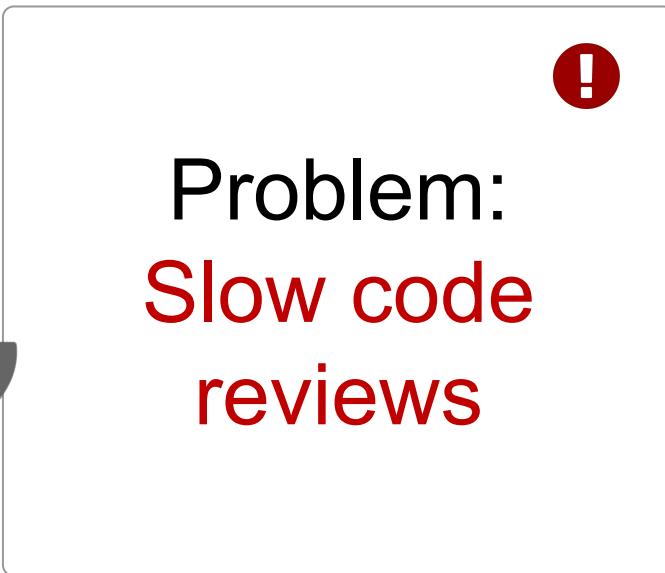
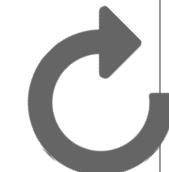


# Design Science in action

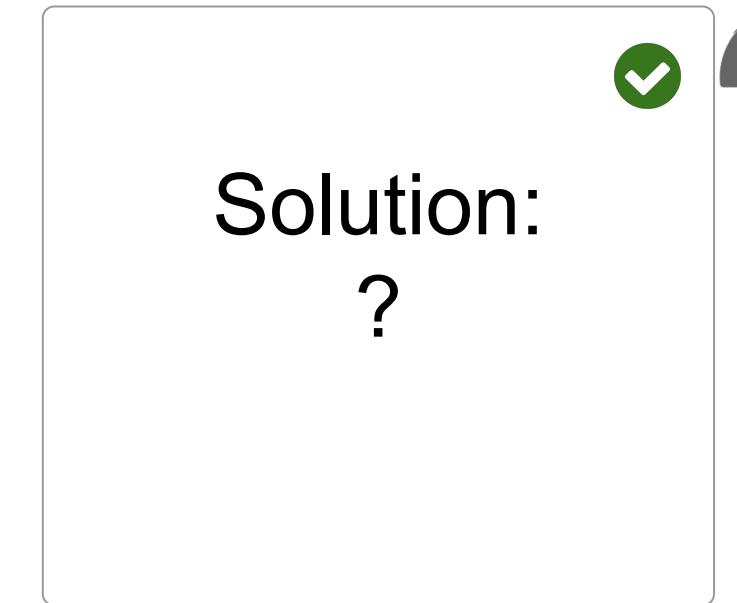


## Problem Understanding

*Code changes are too complex, hard to test, too many requests, low motivation, overwhelmed...*

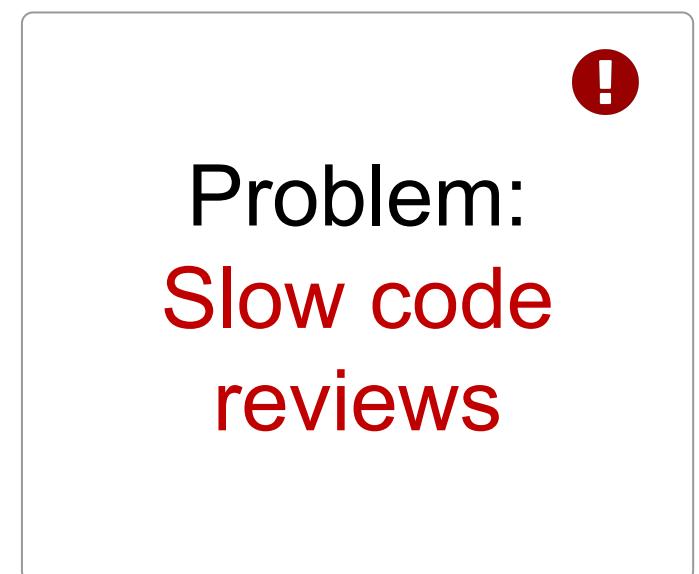


Requirements

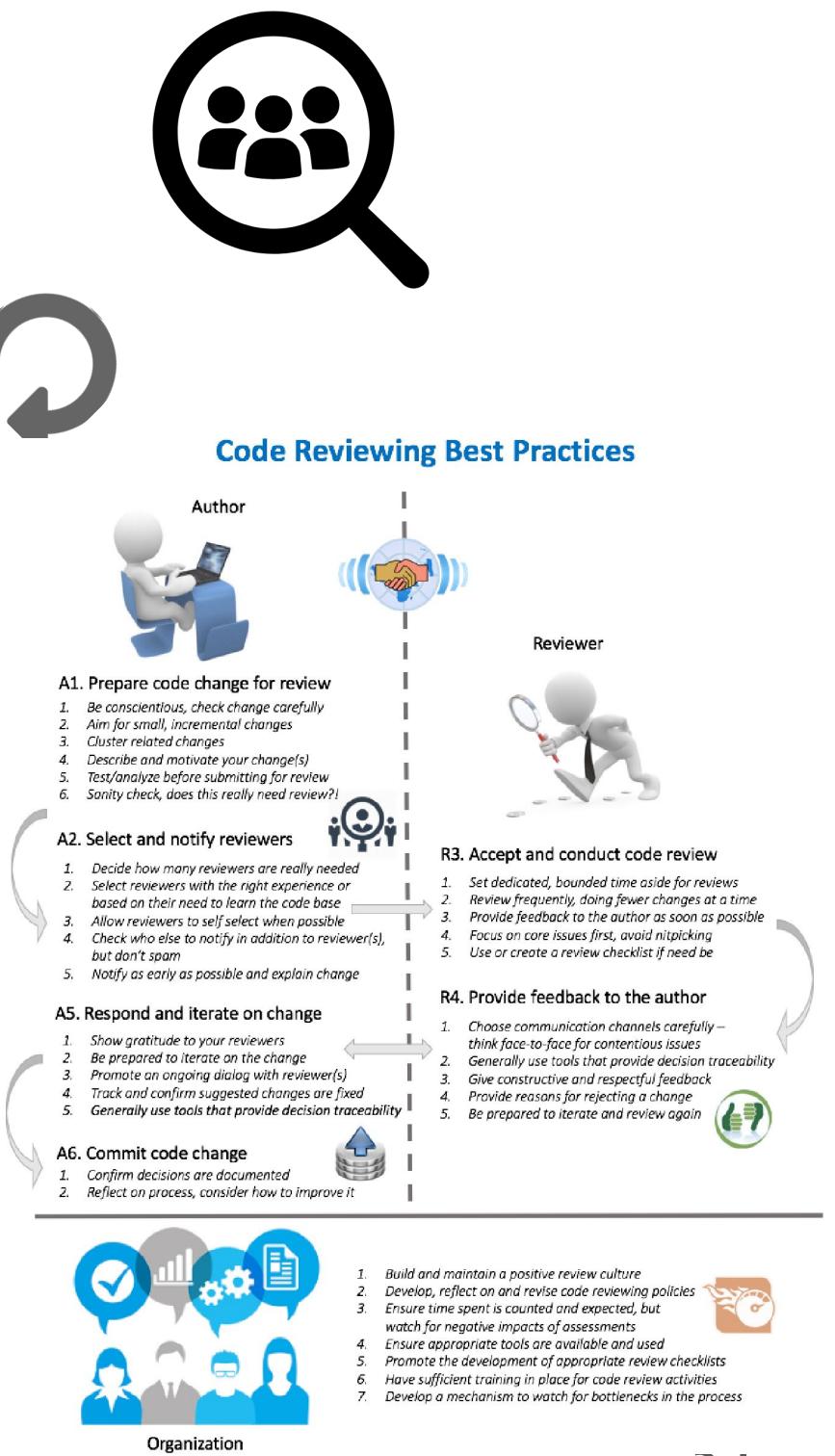
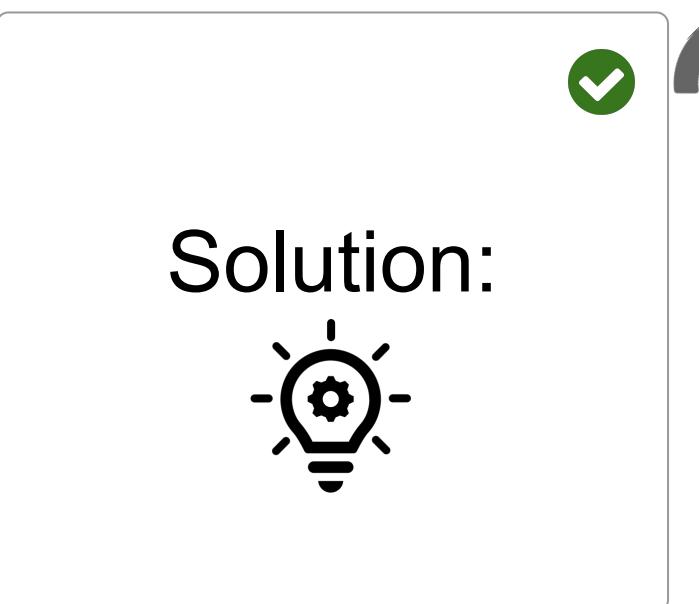


# Design Science in action

# Design Science in action



Requirements  
↔



**(S) Metric:**  
Developer motivation  
for reviews

**(P) Metric:**  
Slow  
releases

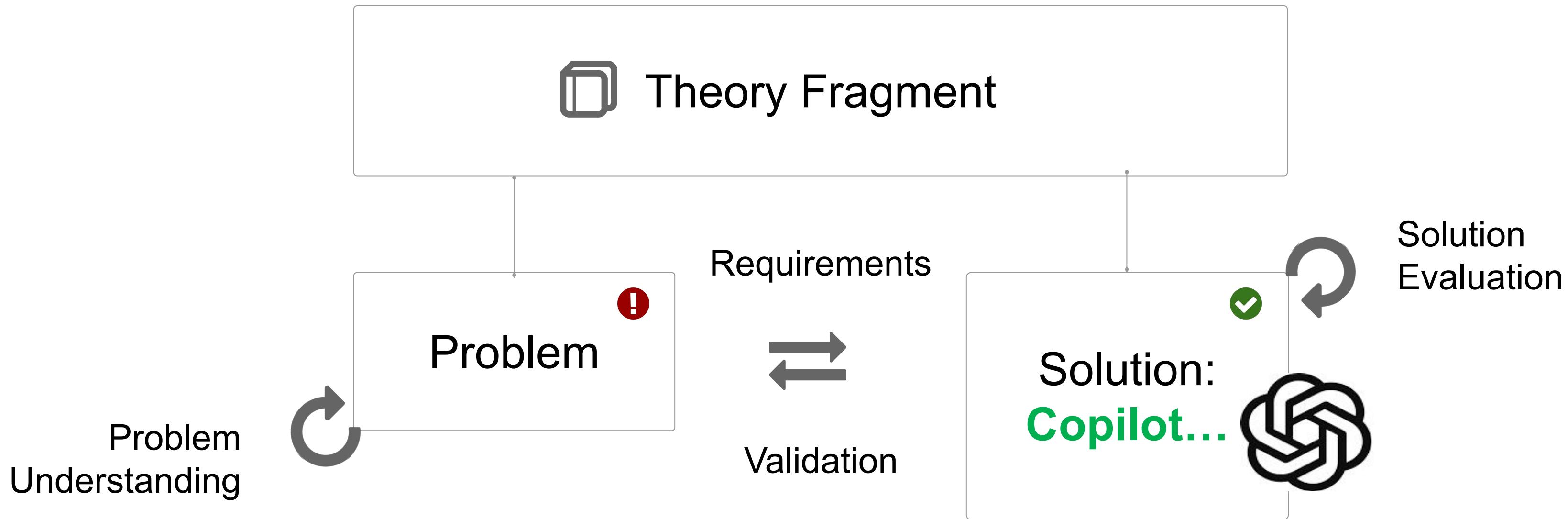
*Theory Fragment:* Developers will be more likely to accept and complete quality reviews more quickly if changes are smaller, changes are clustered, not too many and are documented well

**(A) Metric:**  
# and speed  
of reviews

**Problem:**  
**Slow code  
reviews**

Requirements  
↔  
Validation

**Solution:** ✓  
**Best practices:**  
small changes,  
cluster them,  
documented...



What problems or opportunities for improvement  
are we **adopting GenAI** for?

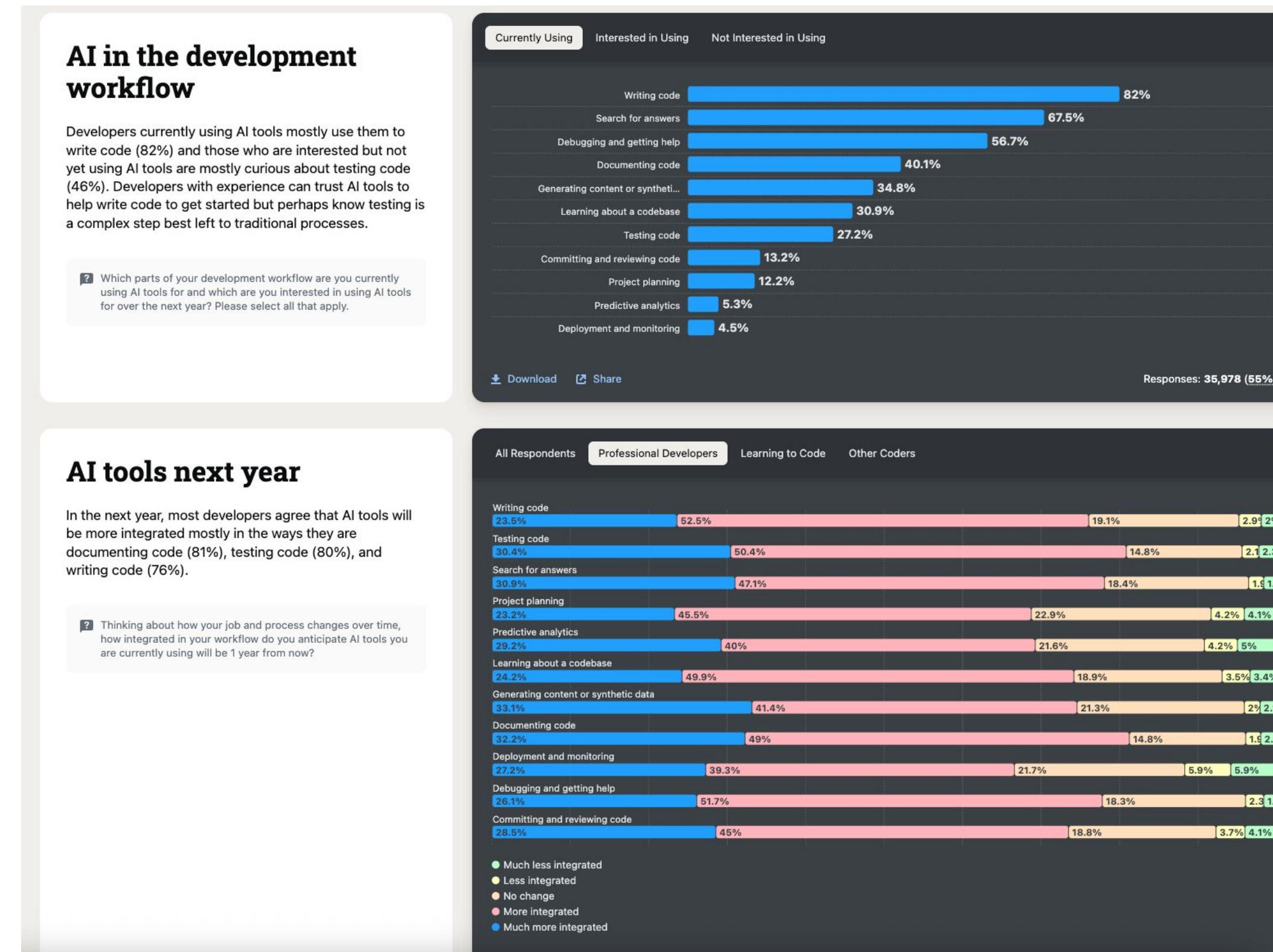
The Quest to Measure Software Developer Productivity

The Space of Development Productivity

A Design Science Approach to Understanding and Improving Productivity

④ **Developer Productivity in the age of GenAI**

# The disruption of Generative AI on Software Engineering is here...



StackOverflow Developer Survey 2024

# Inspiring Movements

## Project /Product Mgmt

Build/CI Deployment

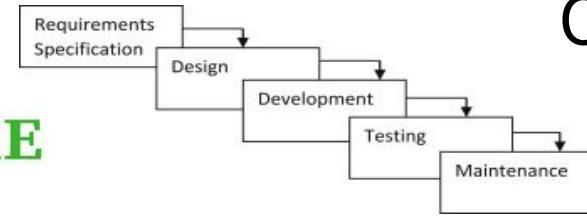
Versioning & Review

Testing & Analysis

Code Authoring

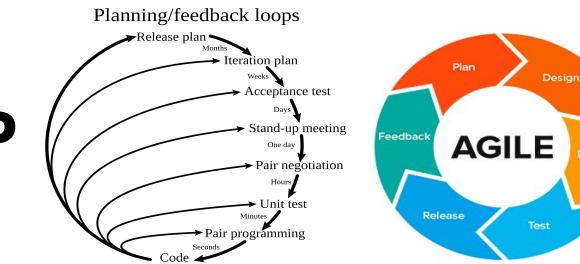
Require-  
Ments &  
Design

## SOFTWARE CRISIS



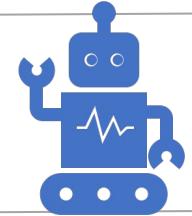
## Open source Code Review

XP

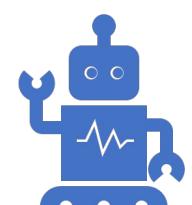


?

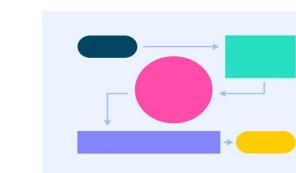
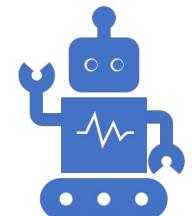
Continuous Integration



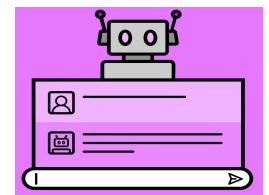
Test Automation



Version control (RCS, CVS)



Formal Methods



1968-1970

1980

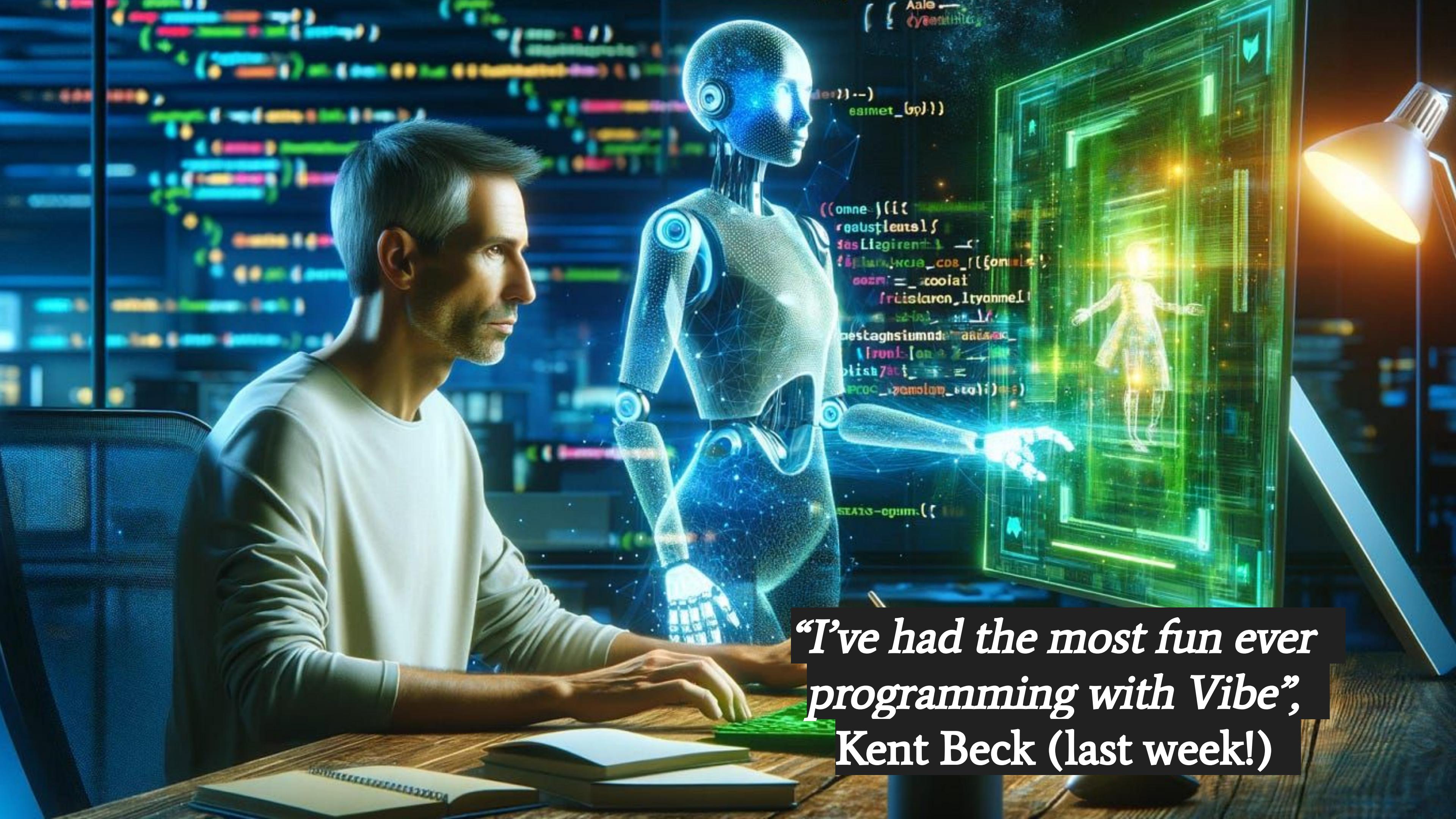
1990

2000

2010

2020

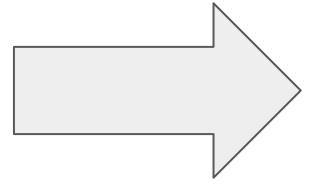
2030



*“I’ve had the most fun ever  
programming with Vibe”,  
Kent Beck (last week!)*

*“Ade” is finding their way of programming in which you do **a lot of prompt engineering, less satisfying, more dull, and less likely to put them into flow**. And they think it is because of the structure of the activity. At the end, they are less productive because they cannot force themselves as much to stay focused on the task.*

[private communication from a colleague]



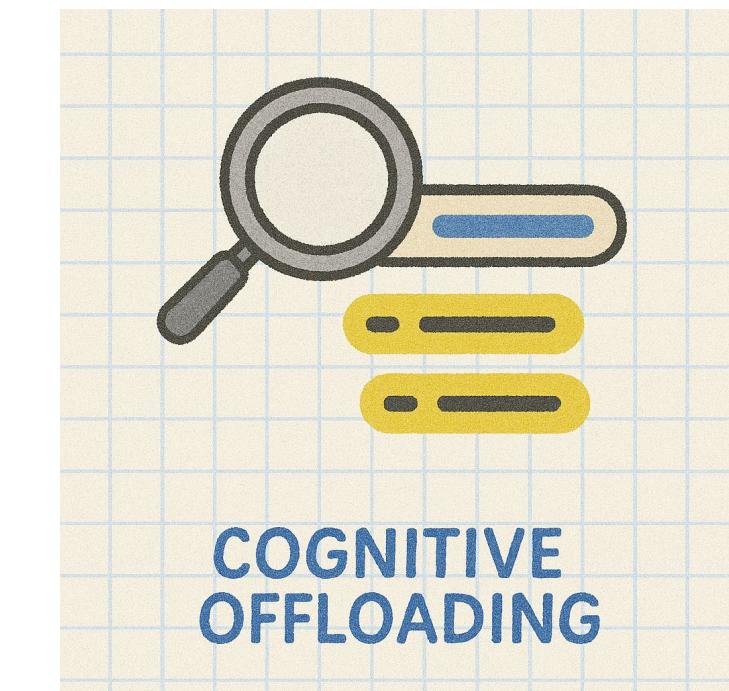
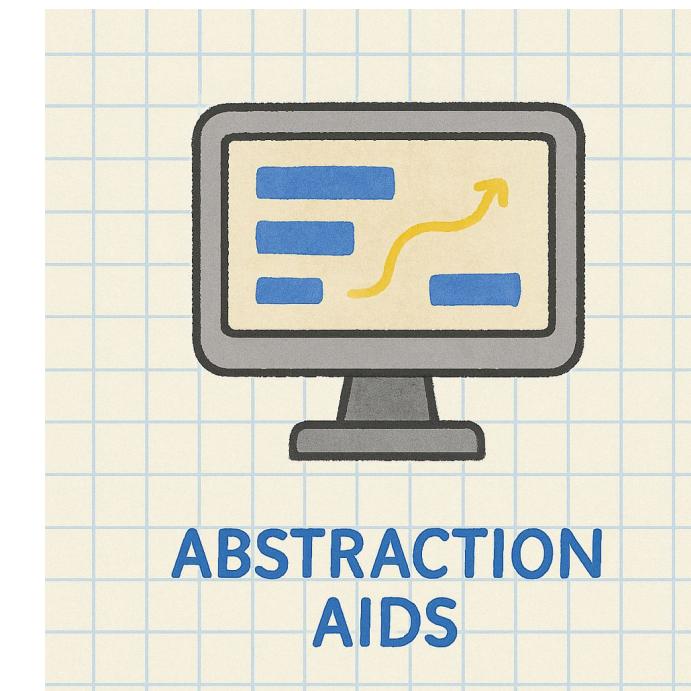
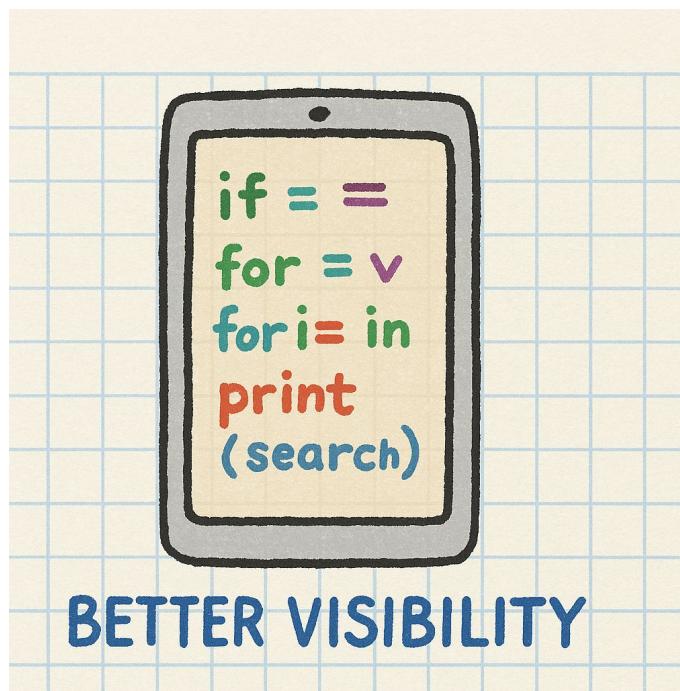
*How to measure the impact of **Generative AI** on:*

Cognitive Augmentation  
Flow Experiences  
Collaboration  
...Creativity

# Cognitive Support Theories [Walenstein, 2000]

Humans perform cognitive tasks like searching, matching, recalling, reasoning....

Tools support cognition by asking “What is hard about this task, and how can this tool help?”



## Types of cognitive support:

[Walenstein, 2000]

- **Search** support (locate relevant information)
- **Inference** support (e.g., showing dependencies)
- **Transformation** support (e.g., refactor code)
- **Traceability** support (e.g., helps users understand changes over time)

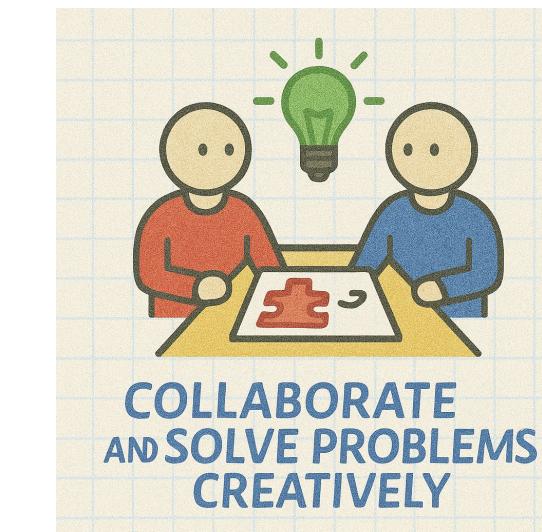
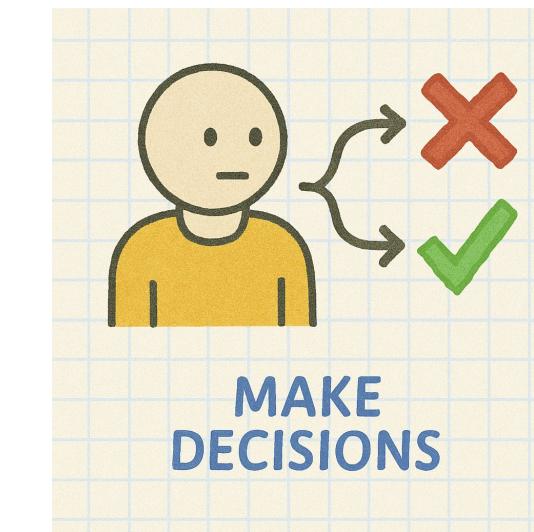
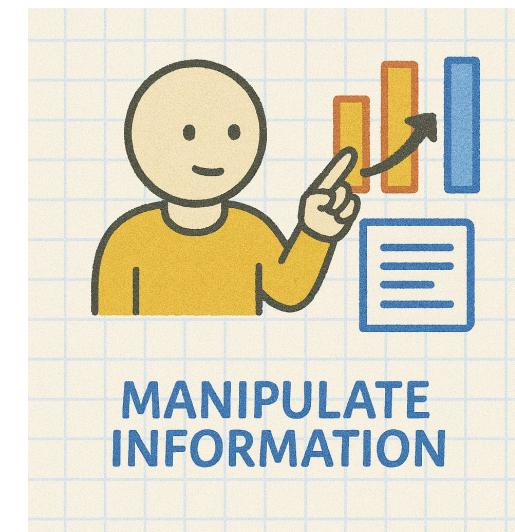
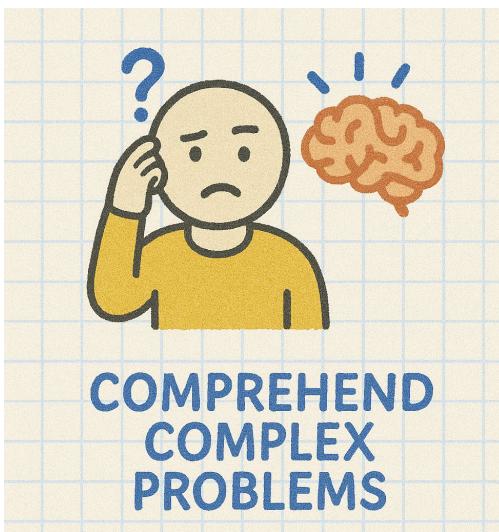
## Evaluating cognitive support:

- Task-Artifact **fit**
- Does it reduce **cognitive cost** or does it overload memory or attention

*How can cognitive support theories help us understand the impact of GenAI ?*

# Augmenting Human Intellect: A Conceptual Framework

Engelbart (1962): technology can **amplify** human intellect, improving our ability to:



**Augmentation vs. Automation:** Engelbart emphasized augmenting human capabilities (e.g., thought, learning, reasoning), rather than replacing them.

# Engelbart's theory of Cognitive Augmentation cont.

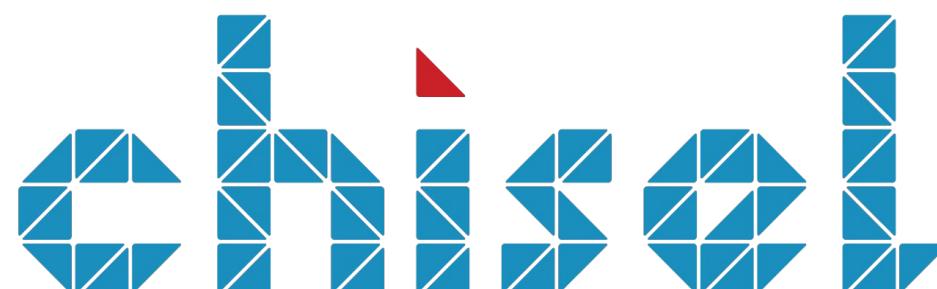
A **co-evolution of humans and tools**, with cognitive work distributed across the individual and external systems (tools, representations, environments)

**Bootstrapping Strategy:** Improving tools that help us improve tools — a recursive model for accelerating intellectual growth (GPTs!)

*How can we improve tools that help improve tools to augment developer intellect - but without forgetting the goal is to improve developer intellect?*

*How to measure the impact of **Generative AI** on:*

Cognitive Augmentation  
Flow Experiences  
Collaboration  
...Creativity



<https://thechiselgroup.org/>

Margaret-Anne Storey  
<https://www.linkedin.com/in/margaret-anne-storey-8419462/>

# Bonus Slides

# Code Reviewing Best Practices

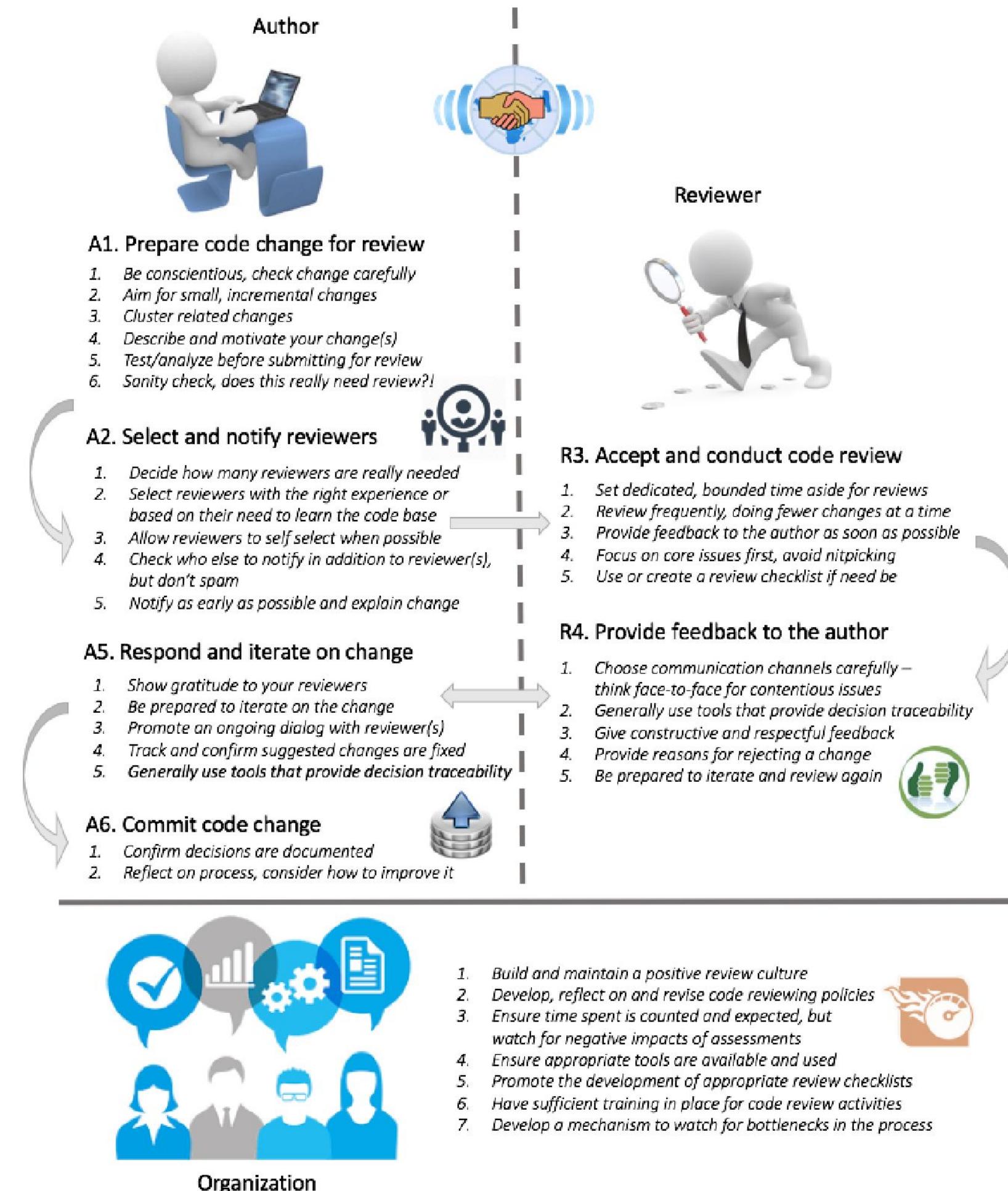
RESEARCH-ARTICLE

## Code Reviewing in the Trenches: Challenges and Best Practices

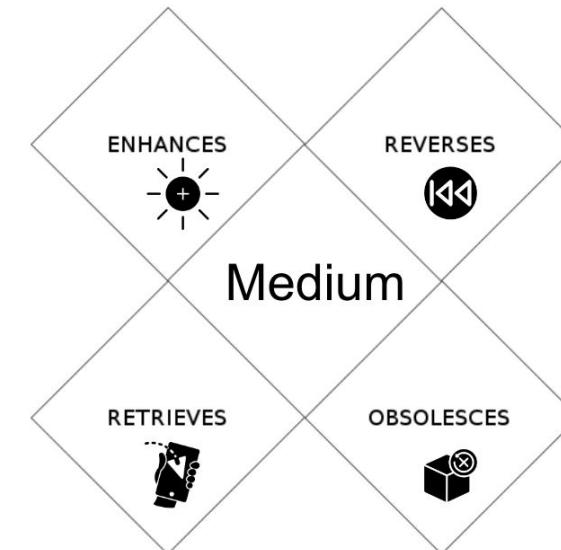
Authors:  [Laura MacLeod](#),  [Michaela Greiler](#),  [Margaret-Anne Storey](#),  [Christian Bird](#),  [Jacek Czerwonka](#)  
[& Claims](#)

IEEE Software, Volume 35, Issue 4 • Pages 34 - 42 • <https://doi.org/10.1109/MS.2017.265100500>

Published: 01 July 2018 [Publication History](#)

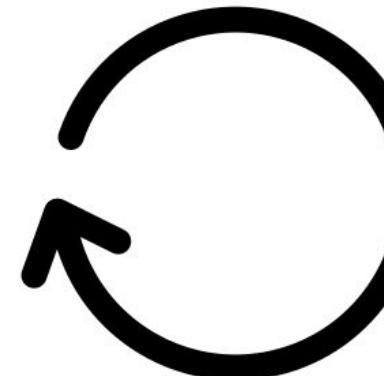


# A Research Playbook for Studying the Impacts of a Disruptive Technology

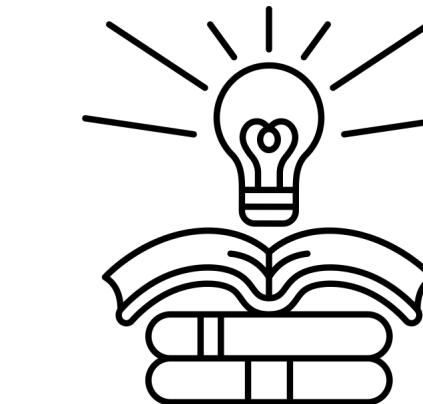


1

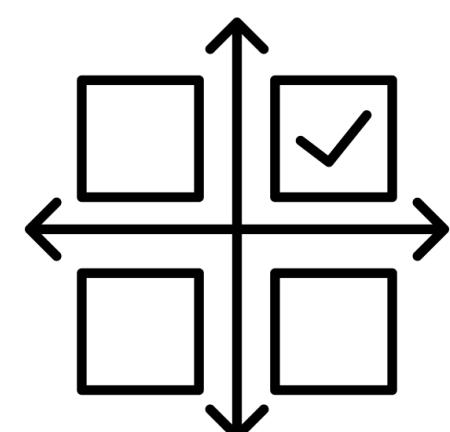
Use McLuhan's tetrad to **map out different impacts** of a specific application of the disruptive technology



2



Consider which **phenomenon and ideas** about these phenomena are relevant to study

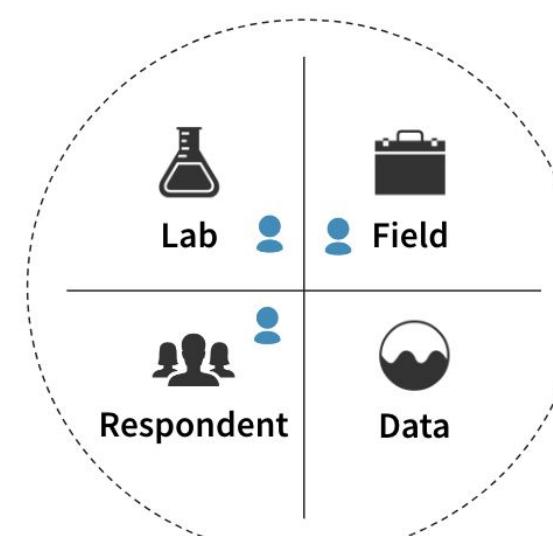


3



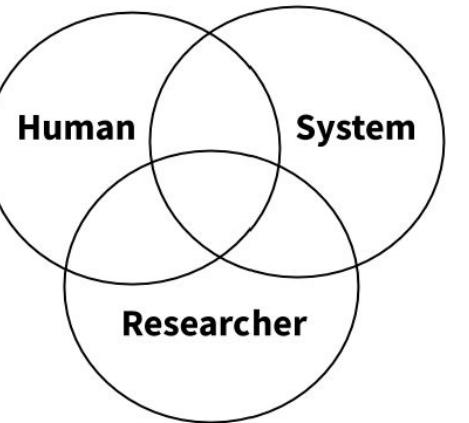
Develop specific **research questions** selecting units of analysis and determine the desired theoretical contributions

4



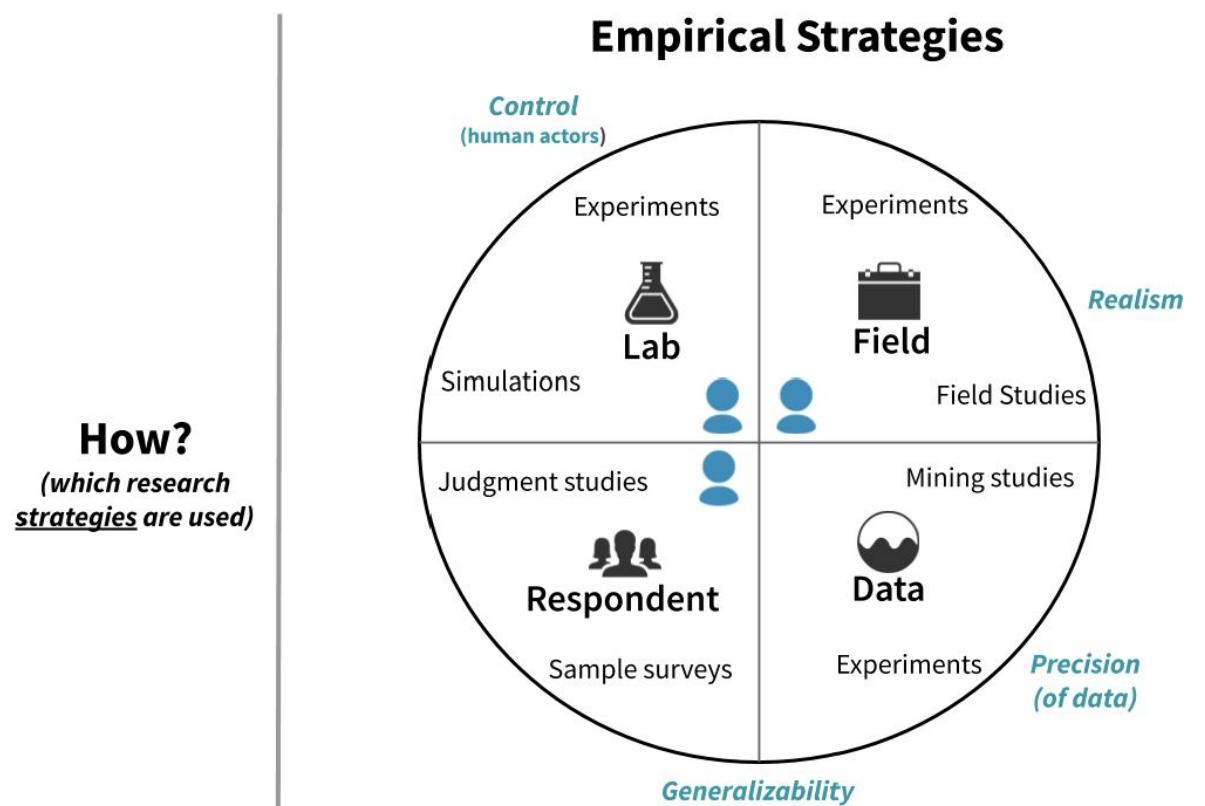
Select suitable **research strategies** that align with the research questions and phenomena to be studied

**Who?**  
*(is the main beneficiary)*



**What?**  
*(type of research contribution)*

Descriptive      Solution



**How?**  
*(which research strategies are used)*

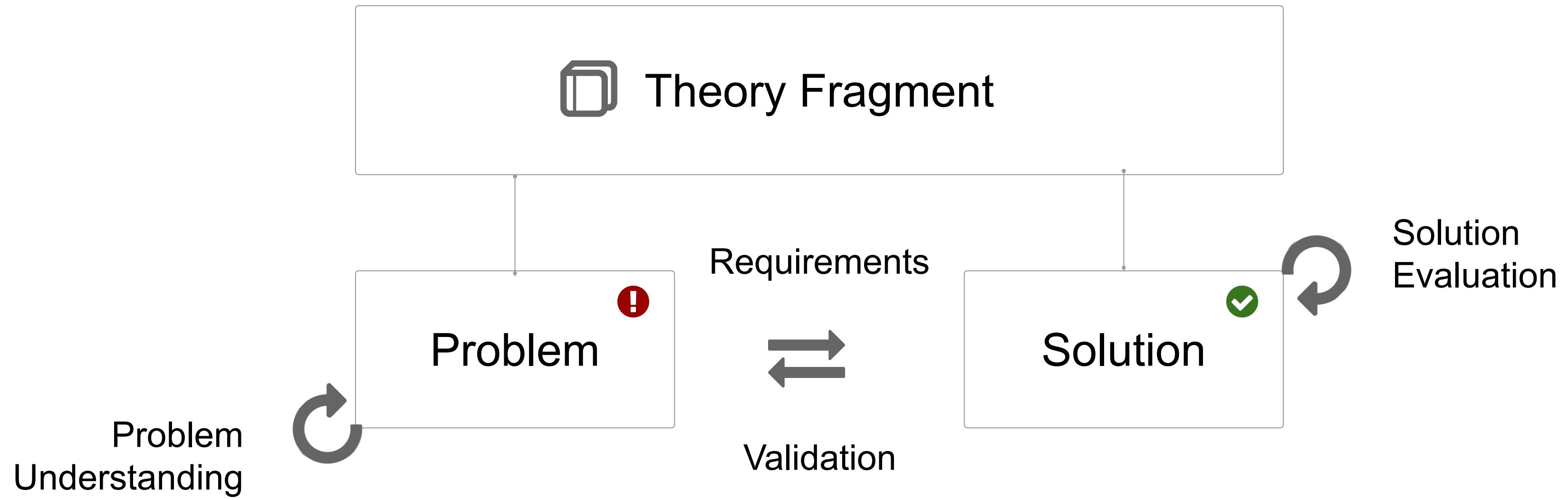
**Non-Empirical Strategies**

Formal Theory      Meta



<https://bit.ly/wwh-framework>

Storey, M., Ernst, N.A., Williams, C. et al. The who, what, how of software engineering research: a socio-technical framework. *Empir Software Eng* 25, 4097–4129 (2020).

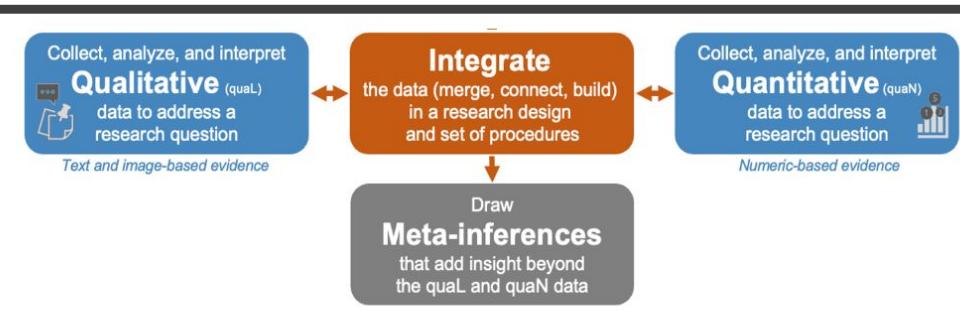


# Design Science

Engström, E., Storey, M.A., Runeson, P., Host M., Baldasserra M.,  
How software engineering research aligns with design science: a review.  
*Empirical Software Eng* (2020).

## Mixed Methods Research Definition

Mixed methods research (**MMR**) is a research approach where multiple methods are used to collect, analyze, and **integrate both qualitative and quantitative** data to address a research problem and **produce novel insights**.



## Landscape of MMR Design

### Design Properties

- Research Questions
- Planned or Emergent
- Inductive or Deductive Dominance
- Timing (Sequential or Concurrent)

### Integration Strategies

- Sequential integration
- Results-based integration
- Data-based integration
- Transformation-based integration

### Research Designs

- Exploratory Sequential
- Explanatory Sequential
- Convergent Parallel
- Embedded

## Principles to Guide MMR in SE

### 1 Methodological Rationale

*Why did we select a particular MMR research design and set of procedures?*

- Complementarity
- Expansion
- Development
- Triangulation

- Credibility
- Explanation
- Increased design flexibility

### 2 Novel Integrated Insights

*What did we gain from using MMR?*

- Improved problem understanding
- Greater depth and breadth
- Explaining unexpected results
- Complementary storytelling

### 3 Procedural Rigor

*How well did we conduct the study?*

- Is the use of mixed methods justified?
- Are the methods effectively integrated to answer the RQs?
- Are the findings from the mixed methods integrated?
- Are the different methods used rigorously conducted?

### 4 Ethical Research

*How responsibly did we do it?*

- Considering the why
- Privacy and confidentiality
- Respect and cultural sensitivity
- Safety and welfare

## Antipatterns of MMR Designs

### Presentation Antipatterns

- Uninvited guest or party crasher
- Smoke and mirrors
- Limitation shirker

### Study Design Antipatterns

- Missing the mark
- Selling your soul
- Cargo cult research
- Sample contamination
- Lost opportunity
- Integration failure
- Questionable ethics

EMSE To Appear

## Guiding Principles for Mixed Methods Research in Software Engineering

Margaret-Anne Storey · Rashina Hoda ·  
Alessandra Maciel Paz Milani ·  
Maria Teresa Baldassarre

March 24, 2025

**Abstract** Mixed methods research is often used in software engineering, but researchers outside of the social or human sciences often lack experience when using these designs. This paper provides guiding principles and advice on how to design mixed method research, and to encourage the intentional, rigorous, and innovative application of mixed methods in software engineering. It also presents key properties of core mixed method research designs. Through a number of fictitious but recognizable software engineering research scenarios, we showcase how to choose suitable mixed method designs and consider the inevitable trade-offs any design choice leads to. We describe several antipatterns that illustrate what to avoid in mixed method research, and when mixed method research should be considered over other approaches.

**Keywords** Mixed methods · Research methods · Methodology · Guiding Principles · Guidelines

Margaret-Anne Storey  
University of Victoria  
Victoria, BC, Canada  
E-mail: mstorey@uvic.ca

Rashina Hoda  
Monash University  
Melbourne, VIC, Australia  
E-mail: rashina.hoda@monash.edu

Alessandra Maciel Paz Milani  
University of Victoria  
Victoria, BC, Canada  
E-mail: amilani@uvic.ca

Maria Teresa Baldassarre  
University of Bari  
Bari, Italy  
E-mail: mariateresa.baldassarre@uniba.it

arXiv:2404.06011v4 [cs.SE] 24 Mar 2025



<https://bit.ly/mmrse>

# Developer satisfaction and productivity theory

