Software Productivity Decoded

How Data Science helps to Achieve More

Thomas Zimmermann, Microsoft Research, USA


Microsoft
MARINERS
THANKS

What makes software engineering research at Microsoft unique?

Easy access to industrial problems and data

Easy access to engineers

Near term impact

Collaborations with Microsoft researchers

Collaborations with external researchers

Product group engagement

Windows

Office 365

Bing

Visual Studio

XBOX

What metrics are the **best predictors of failures**?

If I increase **test coverage**, will that actually increase software quality?

What is the **data quality** level used in empirical studies and how much does it actually matter?

Are there any **metrics that are indicators of failures** in both Open Source and Commercial domains?

I just submitted a **bug report**. Will it be fixed?

Should I be writing **unit tests** in my software project?

How can I tell if a piece of software will have **vulnerabilities**?

Do **cross-cutting concerns** cause defects?

Is strong **code ownership** good or bad for software quality?

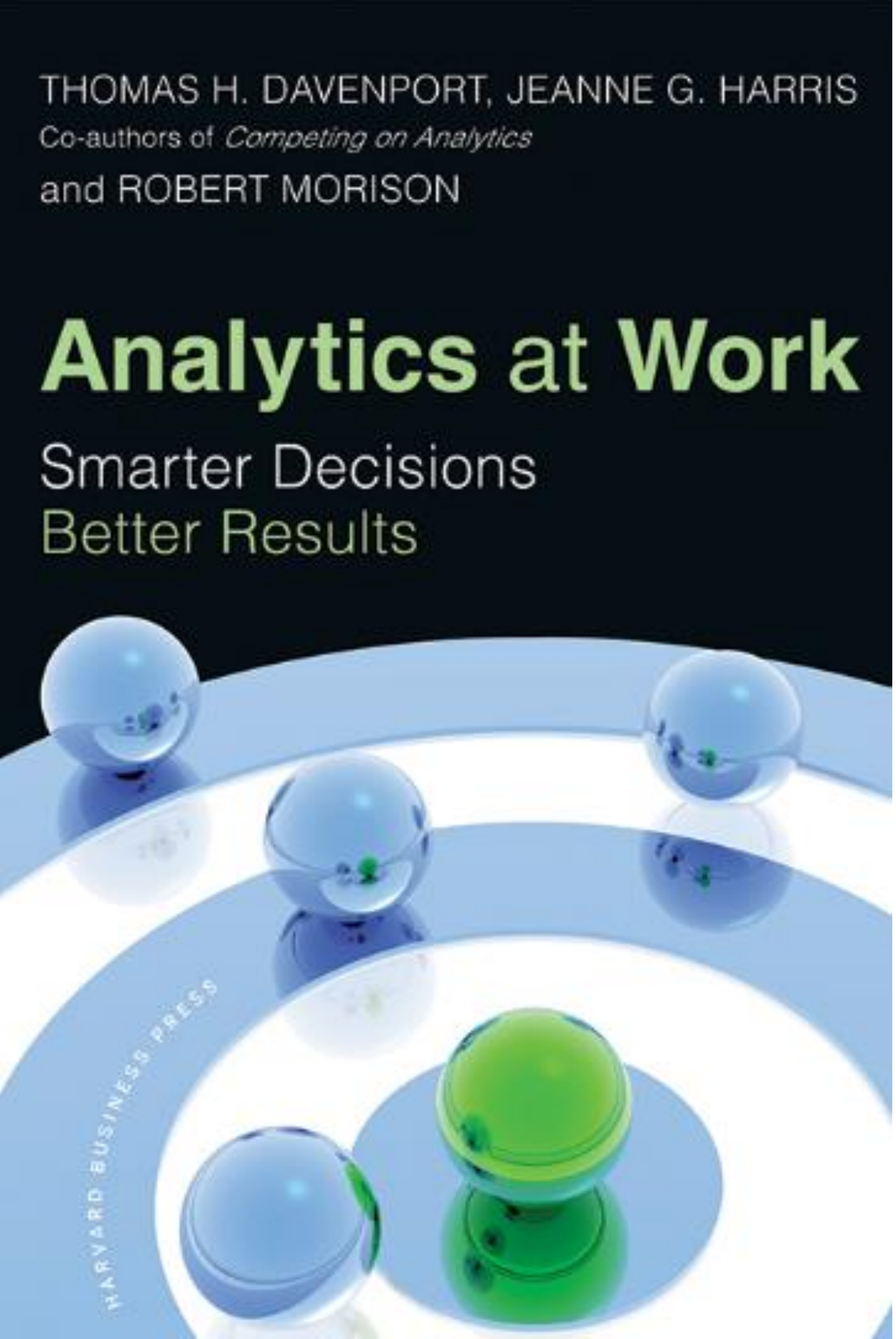
Does **Distributed/Global software development** affect quality?

Does **Test Driven Development** (TDD) produce better code in shorter time?

Data Science

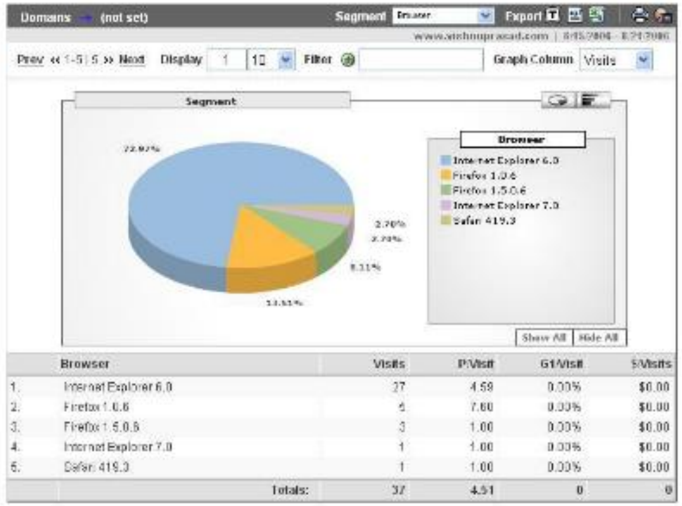
Productivity


Data Science



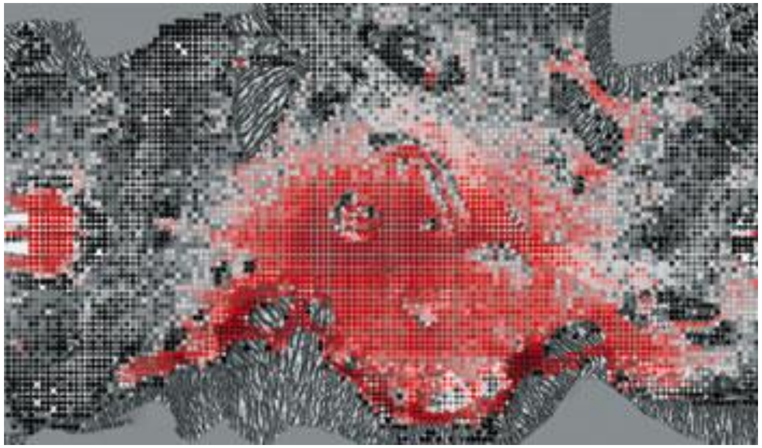
Use of data, analysis, and systematic reasoning to [inform and] make decisions

Web analytics



(Slide by Ray Buse)

Game analytics

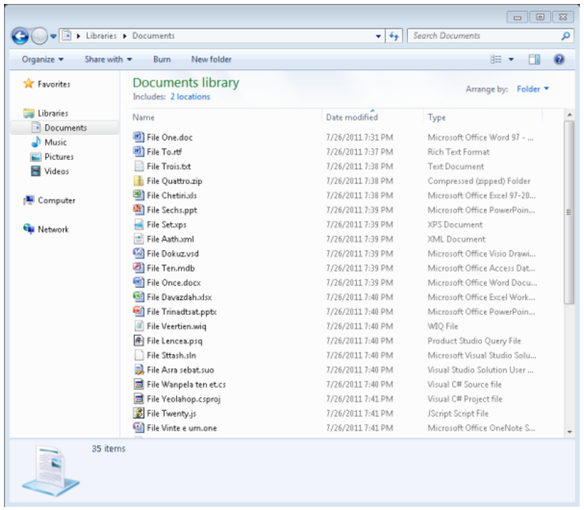


Halo heat maps

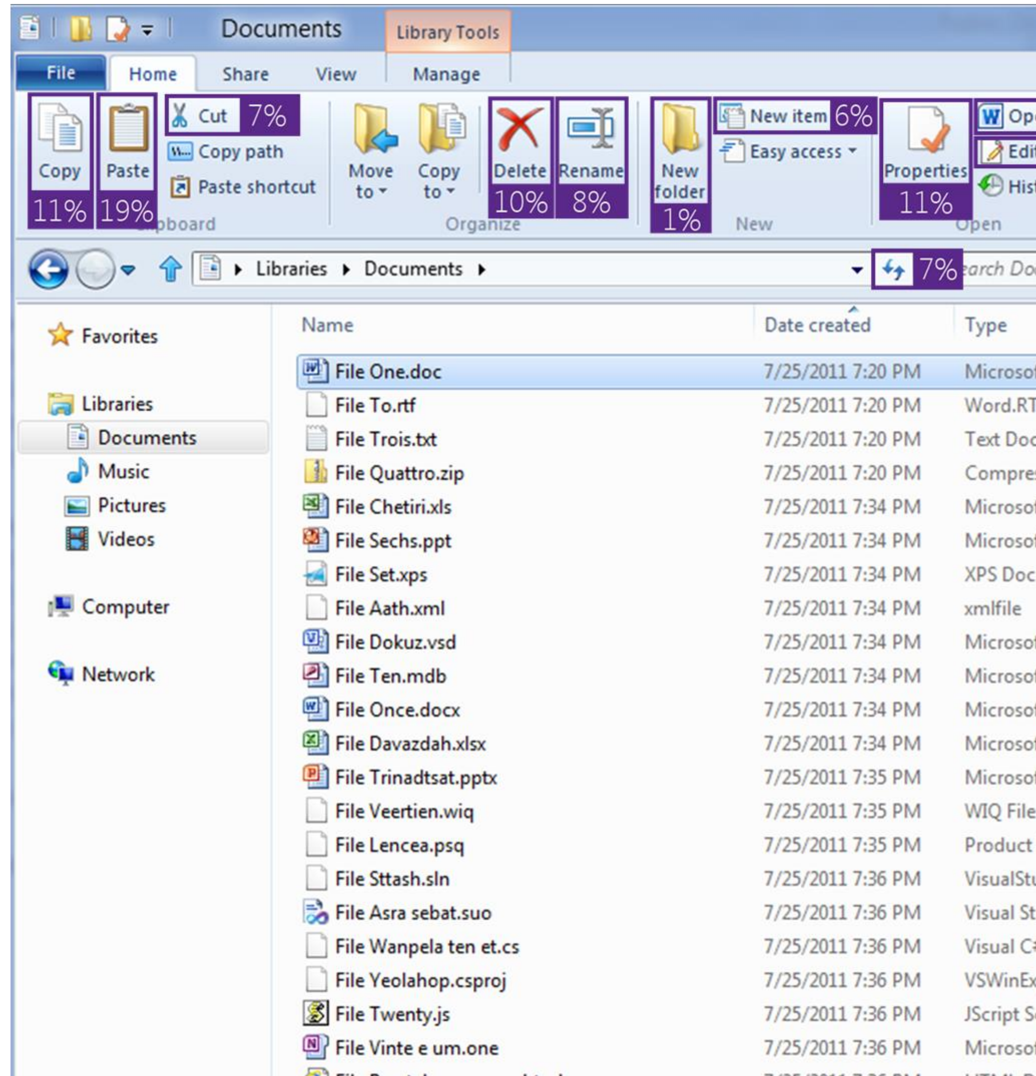
Free to play

Usage analytics

Improving the File Explorer for Windows 8

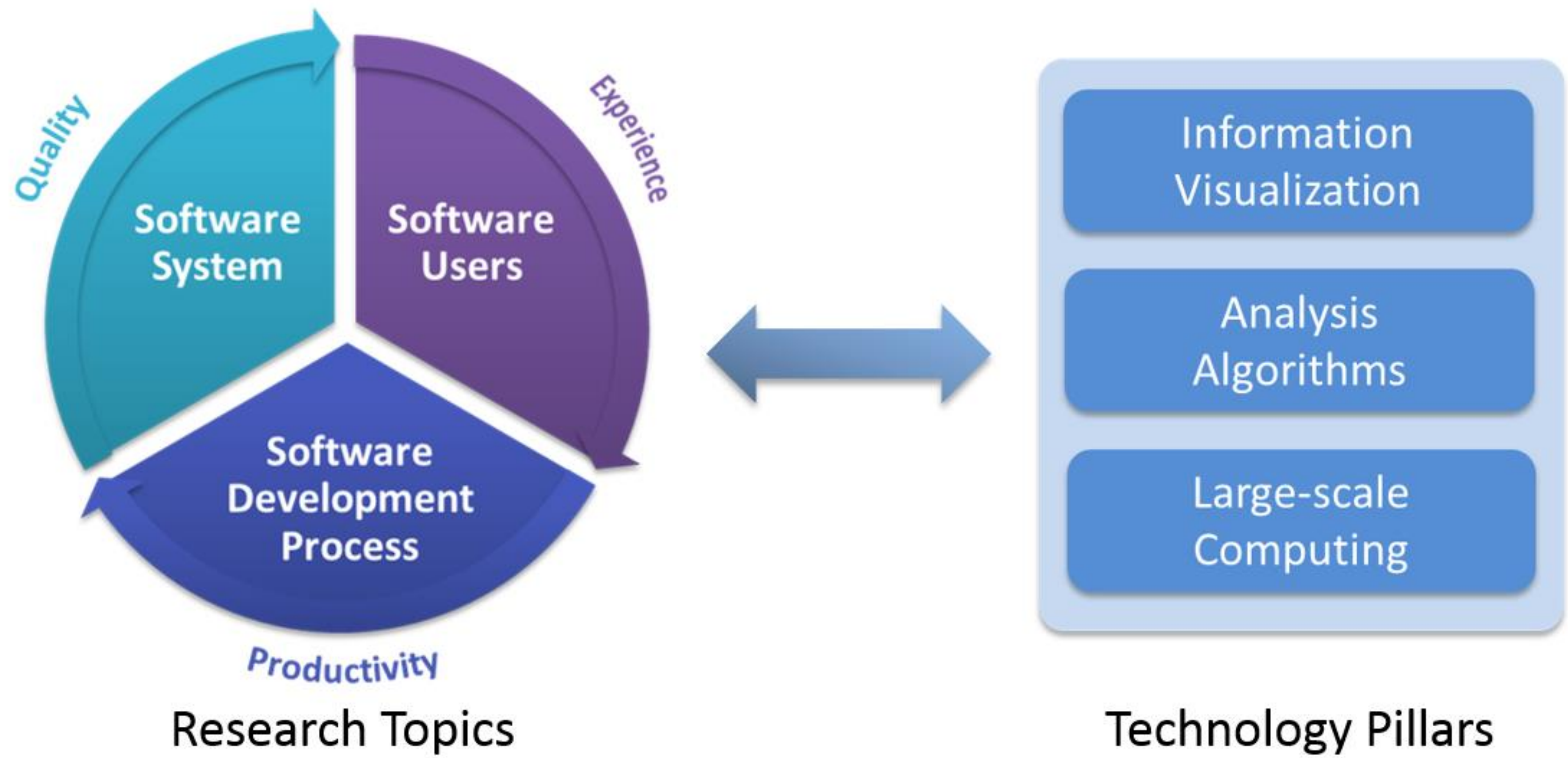


Explorer in Windows 7



Alex Simons: Improvements in Windows Explorer. http://blogs.msdn.com/b/b8/archive/2011/08/29/improvements-in-windows-explorer.aspx

Trinity of software analytics



Dongmei Zhang, Shi Han, Yingnong Dang, Jian-Guang Lou, Haidong Zhang, Tao Xie: Software Analytics in Practice. IEEE Software 30(5): 30-37, September/October 2013.

MSR Asia Software Analytics group: http://research.microsoft.com/en-us/groups/sa/

History of software analytics

EARLY "GLOBAL" MODELS AND SOFTWARE ANALYTICS

As soon as people started programming, it became apparent that programming was an inherently buggy process. As recalled by Maurice Wilkes,[1] speaking of his programming experiences from the early 1950s: "It was on one of my journeys between the EDSAC room and the punching equipment that 'hesitating at the angles of stairs' the realization came over me with full force that a good part of the remainder of my life was going to be spent in finding errors in my own programs."

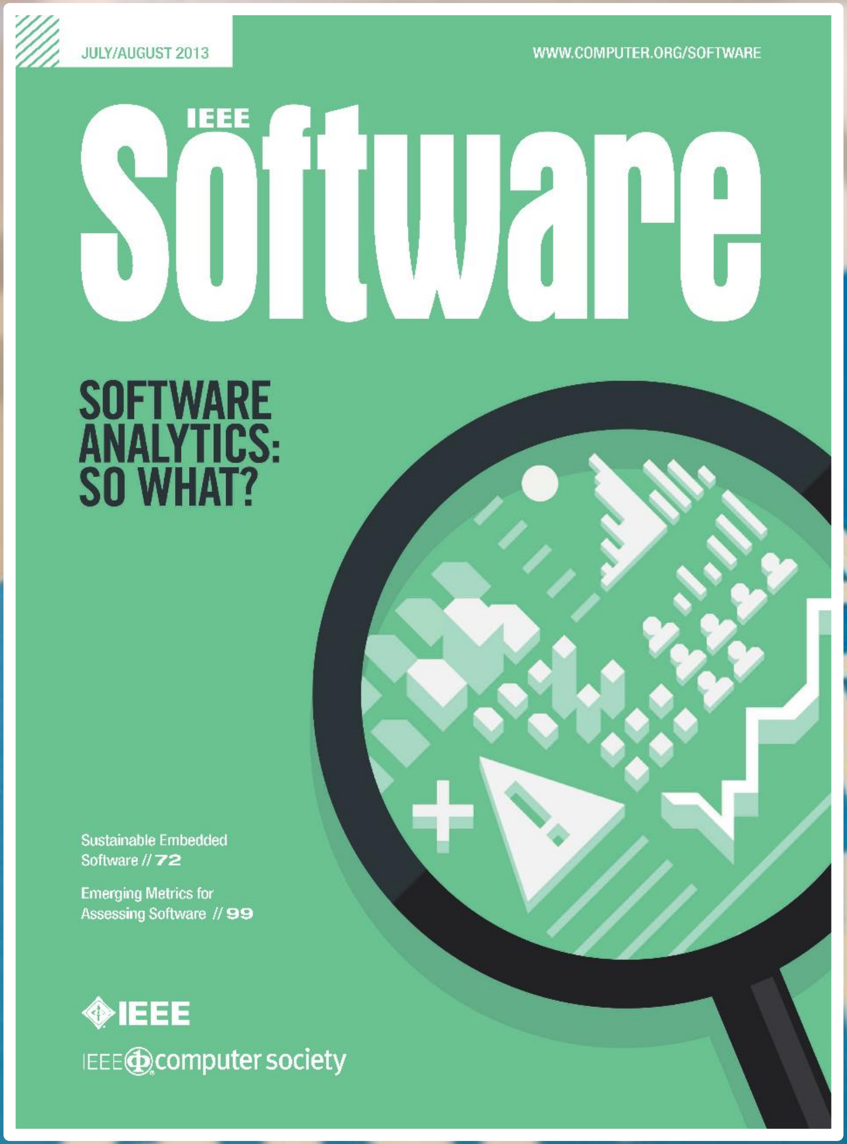
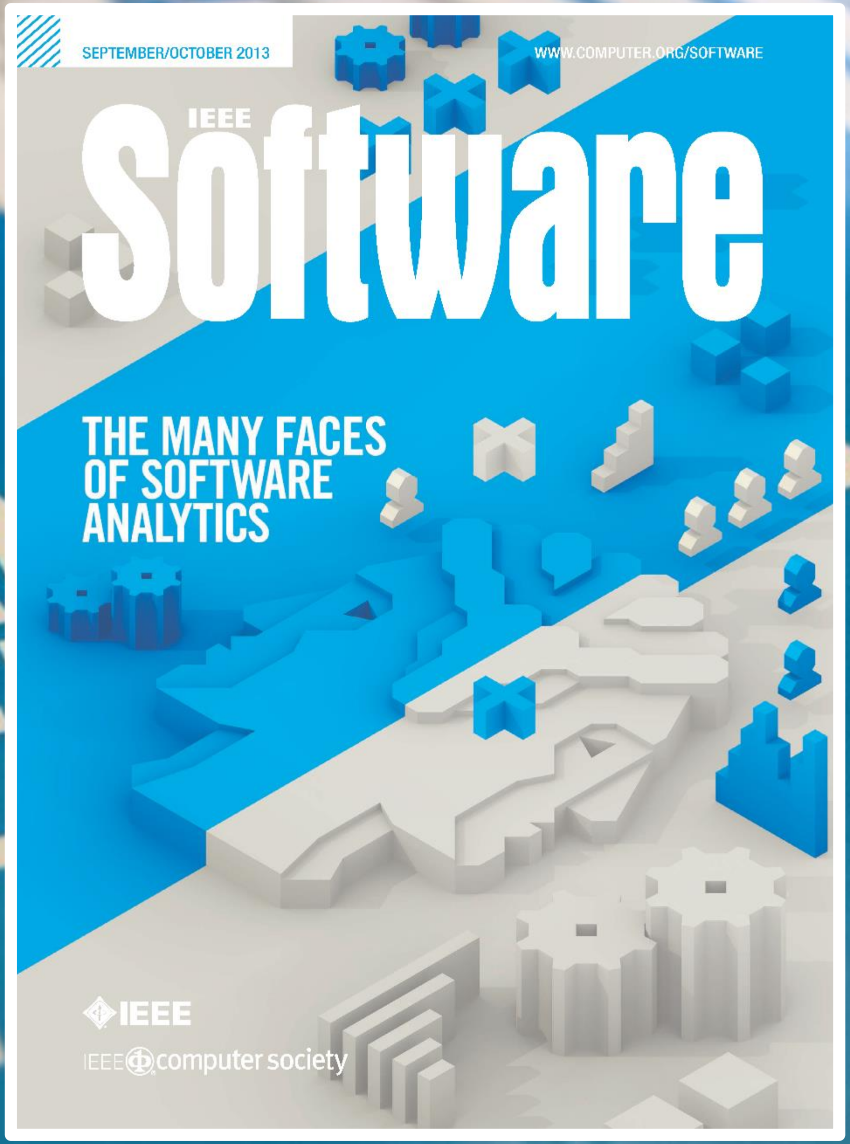
It took several decades to gather the experience required to quantify the size/defect relationship. In 1971, Fumio Akiyama[2] described the first known "size" law, saying the number of defects D was a function of the number of LOC; specifically, $D = 4.86 + 0.018 * i$. In 1976, Thomas McCabe argued that the number of LOC was less important than the complexity of that code.[3] He argued that code is more likely to be defective when his "cyclomatic complexity" measure was over 10.

Not only is programming an inherently buggy process, it's also inherently difficult. Based on data from 63 projects, Barry Boehm[4] proposed in 1981 an estimator for development effort that was exponential on program size: effort = a * KLOCb * EffortMultipliers, where $2.4 \le a \le 3$ and $1.05 \le b \le 1.2$.

References

1. M. Wilkes, *Memoirs of a Computer Pioneer*, MIT Press, 1985.
2. F. Akiyama, "An Example of Software System Debugging," *Information Processing*, vol. 71, 1971, pp. 353–359.
3. T. McCabe, "A Complexity Measure," *IEEE Trans. Software Eng.*, vol. 2, no. 4, 1976, pp. 308–320.
4. B. Boehm, *Software Engineering Economics*, Prentice-Hall, 1981.

Tim Menzies, Thomas Zimmermann: Software Analytics: So What?
IEEE Software 30(4): 31-37 (2013)

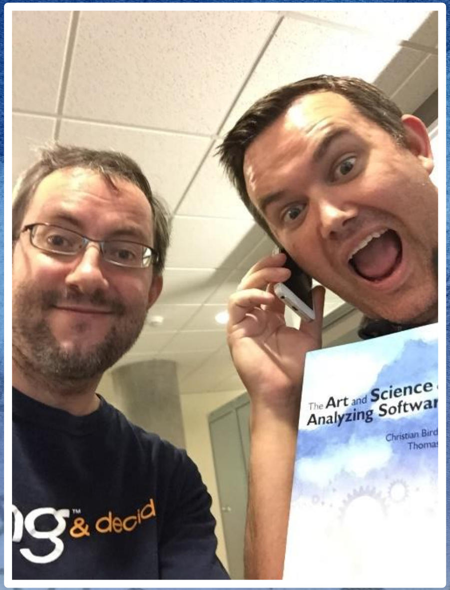

JULY/AUGUST 2013
WWW.COMPUTER.ORG/SOFTWARE
IEEE Software
SOFTWARE ANALYTICS: SO WHAT?
Sustainable Embedded Software // 72
Emerging Metrics for Assessing Software // 99
IEEE
IEEE computer society

SEPTEMBER/OCTOBER 2013
WWW.COMPUTER.ORG/SOFTWARE
IEEE Software
THE MANY FACES OF SOFTWARE ANALYTICS
IEEE
IEEE computer society

Alberto Bacchelli, Olga Baysal, Ayse Bener, Aditya Budi, Bora Caglayan, Gul Calikli, Joshua Charles Campbell, Jacek Czerwonka, Kostadin Damevski, Madeline Diep, Robert Dyer, Linda Esker, Davide Falessi, Xavier Franch, Thomas Fritz, Nikolas Galanis, Marco Aurélio Gerosa, Ruediger Glott, Michael W. Godfrey, Alessandra Gorla, Georgios Gousios, Florian Groß, Randy Hackbarth, Abram Hindle, Reid Holmes, Lingxiao Jiang, Ron S. Kenett, Ekrem Kocaguneli, Oleksii Kononenko, Kostas Kontogiannis, Konstantin Kuznetsov, Lucas Layman, Christian Lindig, David Lo, Fabio Mancinelli, Serge Mankovskii, Shahar Maoz, Daniel Méndez Fernández, Andrew Meneely, Audris Mockus, Murtuza Mukadam, Brendan Murphy, Emerson Murphy-Hill, John Mylopoulos, Anil R. Nair, Maleknaz Nayebi, Hoan Nguyen, Tien Nguyen, Gustavo Ansaldi Oliva, John Palframan, Hridesh Rajan, Peter C. Rigby, Guenther Ruhe, Michele Shaw, David Shepherd, Forrest Shull, Will Snipes, Diomidis Spinellis, Eleni Stroulia, Angelo Susi, Lin Tan, Ilaria Tavecchia, Ayse Tosun Misirli, Mohsen Vakilian, Stefan Wagner, Shaowei Wang, David Weiss, Laurie Williams, Hamzeh Zawawy, and Andreas Zeller

The Art and Science of Analyzing Software Data

Edited by

Christian Bird, Tim Menzies, Thomas Zimmermann



Perspectives on Data Science for Software Engineering

Edited by Tim Menzies, Laurie Williams, Thomas Zimmermann

Contributors: Bram Adams, A. Bacchelli, T. Barik, E.T. Barr, O. Baysal, A. Bener, G. Bergersen, C. Bird, D. Budgen, B. Caglayan, T. Carnahan, J. Czerwonka, P. Devanbu, M. Di Penta, S. Diehl, T. Dyba, T. Fritz, M.W. Godfrey, G. Gousios, P. Guo, K. Herzig, A. Hindle, R. Holmes, Zhitao Hou, J. Huang, Andrew J. Ko, N. Juristo, S. Just, M. Kim, E. Kocaguneli, K. Kuutti, Qingwei Lin, Jian-Guang Lou, N. Medvidovic, A. Meneely, T. Menzies, L.L. Minku, A. Mockus, J. Munch, G.C. Murphy, B. Murphy, E. Murphy-Hill, M. Nagappan, M. Nayebi, M. Oivo, A. Orso, T. Ostrand, F. Peters, D. Posnett, L. Prechelt, Venkatesh-Prasad Ranganath, B. Ray, R. Robbes, P. Rotella, G. Ruhe, P. Runeson, B. Russo, M. Shepperd, E. Shihab, D.I.K. Sjøberg, D. Spinellis, M.-A. Storey, C. Theisen, A. Tosun, B. Turhan, H. Valdivia-Garcia, S. Vegas, S. Wagner, E. Weyuker, J. Whitehead, L. Williams, Tao Xie, A. Zeller, Dongmei Zhang, Hongyu Zhang, Haidong Zhang, T. Zimmermann

http://tiny.cc/superdog


PEOPLE

ARTWORK: TAMAR COHEN, ANDREW J BUBOLTZ, 2011, SILK SCREEN ON A PAGE FROM A HIGH SCHOOL YEARBOOK, 8.5" X 12"

DATA

Data Scientist: The Sexiest Job of the 21st Century

by Thomas H. Davenport and D.J. Patil

FROM THE OCTOBER 2012 ISSUE

When Jonathan Goldman arrived for work in June 2006 at LinkedIn, the business networking site, the place still felt like a start-up. The company had just under 8 million accounts, and the number was growing quickly as existing members invited their friends and colleagues to join. But users weren't seeking out connections with the people who were already on the site at the rate executives had expected. Something was apparently missing in the social experience. As one LinkedIn manager put it, "It was like arriving at a conference reception and realizing you don't know anyone. So you just stand in the corner sipping your drink—and you probably leave early." Goldman, a PhD in physics from Stanford, was intrigued by the linking he did see going on and by the richness of the user profiles. It all made for messy data and unwieldy analysis, but as he began exploring people's connections, he started to see possibilities. He began forming theories, testing hunches, and finding patterns that allowed him to predict whose networks a given profile would land in. He could imagine that new features capitalizing on the heuristics he was developing might

WHAT TO READ NEXT
Big Data: The Management Revolution
Making Advanced Analytics Work for You
Google Flu Trends' Failure Shows Good Data > Big Data

VIEW MORE FROM THE
October 2012 Issue

Obsessing over our customers is everybody's job. I'm looking to the engineering teams to **build the experiences our customers love**. […] In order to deliver the experiences our customers need for the mobile-first and cloud-first world, we will modernize our engineering processes to be **customer-obsessed, data-driven, speed-oriented and quality-focused.**

http://news.microsoft.com/ceo/bold-ambition/index.html

Each engineering group will have **Data and Applied Science resources** that will focus on measurable outcomes for our products and predictive analysis of market trends, which will allow us to innovate more effectively.

http://news.microsoft.com/ceo/bold-ambition/index.html

The Data Scientists

Robert DeLine

Andrew Begel

Miryung Kim

Miryung Kim, Thomas Zimmermann, Robert DeLine, Andrew Begel:
The Emerging Role of Data Scientists on Software Development Teams. ICSE 2016.
Data Scientists in Software Teams: State of the Art and Challenges. To appear in TSE.

Methodology

Interviews

16 data scientists

- 5 women and 11 men from eight different Microsoft organizations

Snowball sampling

- data-driven engineering meet-ups and technical community meetings
- word of mouth

Coding with Atlas.TI

Clustering of participants

Survey

793 responses

- full-time data scientists
- employees with interest in data science

Questions about

- demographics
- skills
- self-perception
- working styles
- time spent
- challenges and best practices

Background of data scientists

Most CS, many interdisciplinary backgrounds

Many have higher education degrees

Survey: 41% have master's degrees, and 22% have PhDs

Strong passion for data

I've always been a data kind of guy. I love playing with data. I'm very focused on how you can organize and make sense of data and being able to find patterns. I love patterns. [P14]

"Machine learning hackers". Need to know stats

My people have to know statistics. They need to be able to answer sample size questions, design experiment questions, know standard deviations, p-value, confidence intervals, etc.

Background of data scientists

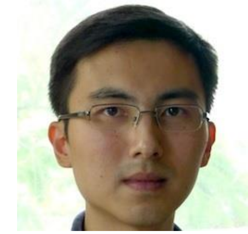
PhD training contributes to working style

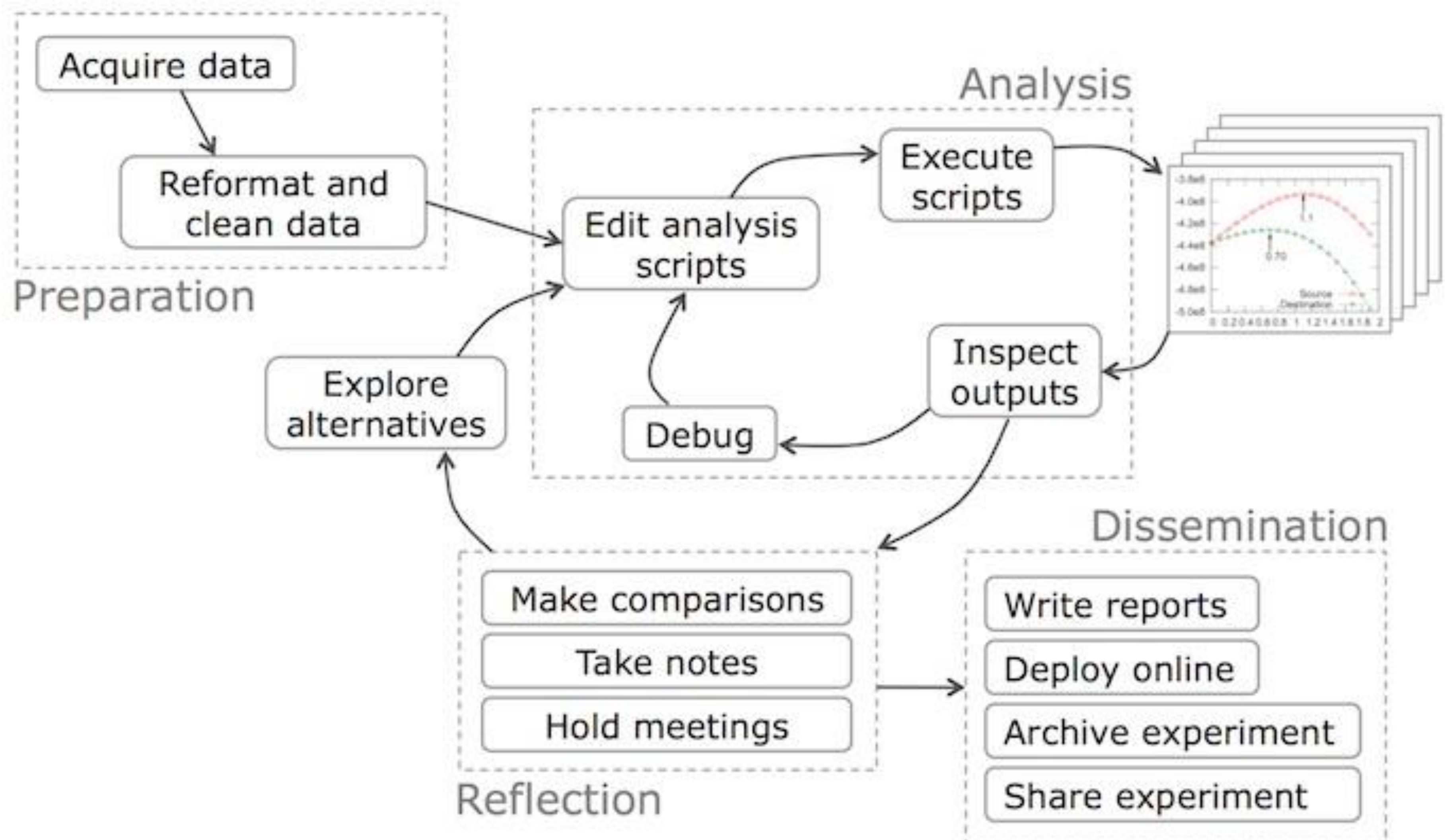
It has never been, in my four years, that somebody came and said, "Can you answer this question?" I mostly sit around thinking, "How can I be helpful?" Probably that part of your PhD is you are figuring out what is the most important questions. [P13]

I have a PhD in experimental physics, so pretty much, I am used to designing experiments. [P6]

Doing data science is kind of like doing research. It looks like a good problem and looks like a good idea. You think you may have an approach, but then maybe you end up with a dead end. [P5]

Typical data science workflow





https://cacm.acm.org/blogs/blog-cacm/169199-data-science-workflow-overview-and-challenges/

Activities of data scientists

Preparation

Data engineering platform; Telemetry injection; Experimentation platform

Analysis

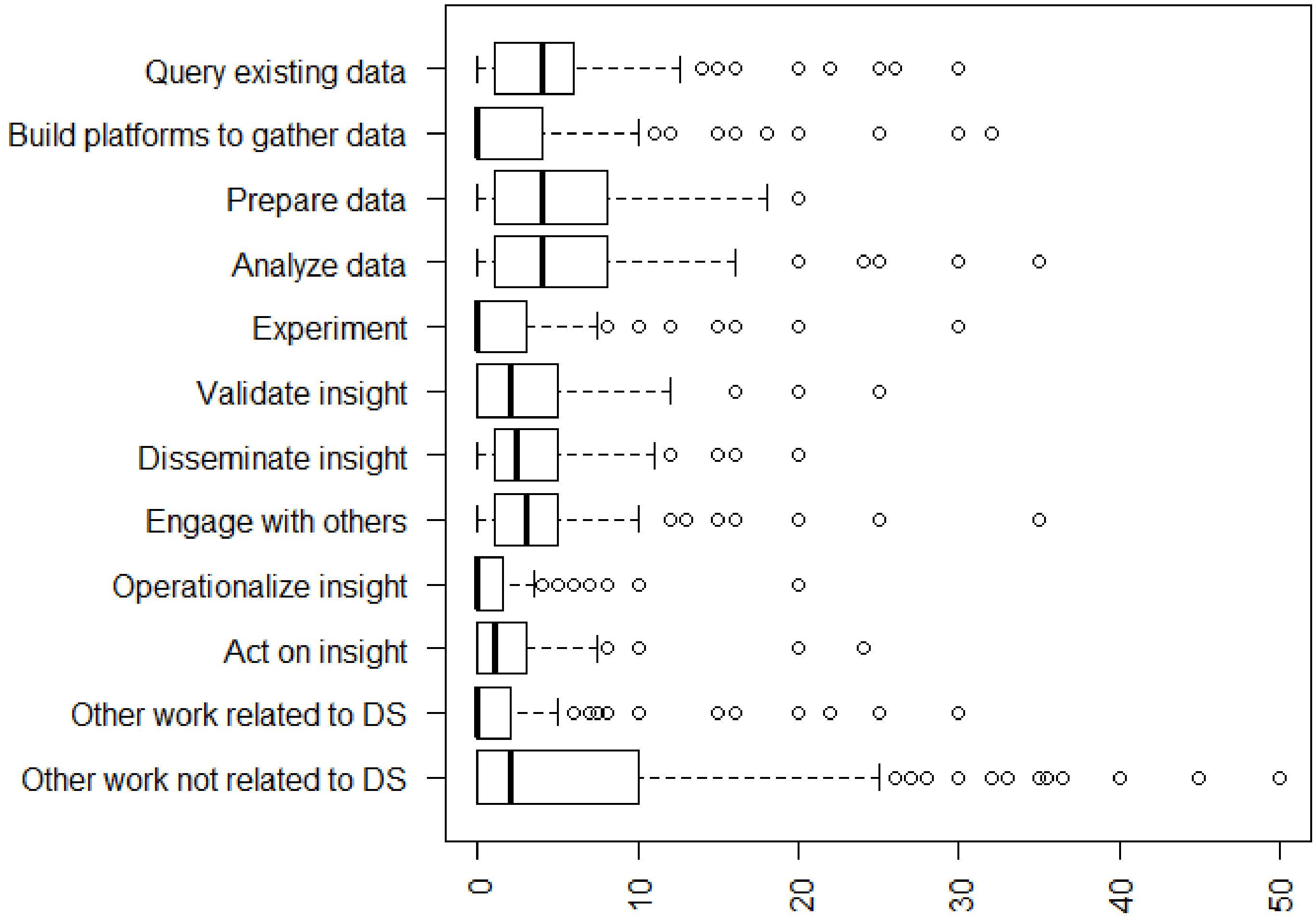
Data merging and cleaning; Sampling; Data shaping including selecting and creating features; Defining sensible metrics; Building predictive models; Defining ground truths; Hypothesis testing

Dissemination

Operationalizing predictive models; Defining actions and triggers; Translating insights and models to business values

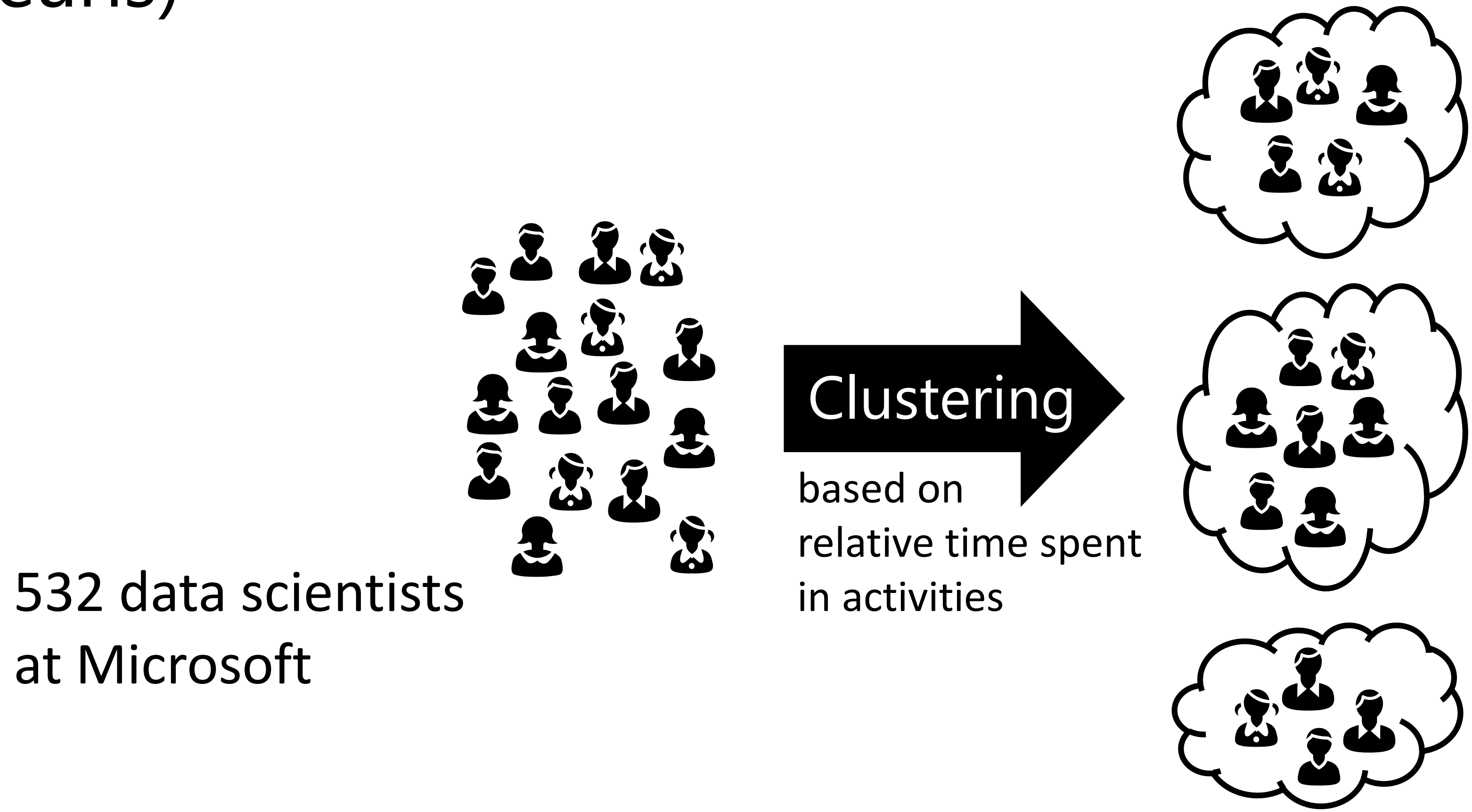
Time spent on activities

Hours spent on certain activities (self reported, survey, N=532)



Time spent on activities

Cluster analysis on relative time spent (k-means)



Time spent on activities

	Query existing data	Build platforms to gather data	Prepare data	Analyze data	Experiment	Validate insight	Disseminate insight	Engage with others	Operationalize insight	Act on insight	Other work related to DS	Other work not related to DS
Entire population 532 people	12.0% 4.7h	7.2% 2.9h	11.7% 4.9h	12.5% 5.2h	4.8% 2.1h	6.9% 3.0h	8.5% 3.5h	9.2% 3.8h	2.4% 1.1h	5.5% 2.1h	4.1% 1.9h	15.1% 6.7h
Cluster 1 Polymath 156 people	10.4% 4.4h	8.5% 3.6h	11.5% 5.1h	15.1% 6.7h	9.1% 4.0h	7.7% 3.6h	7.4% 3.5h	7.9% 3.6h	3.2% 1.5h	5.2% 2.3h	4.0% 2.0h	10.1% 4.5h
Cluster 2 Data Evangelist 71 people	6.8% 2.2h	2.1% 1.0h	6.7% 2.5h	7.7% 2.9h	2.4% 1.2h	7.0% 2.6h	12.0% 4.5h	23.0% 8.6h	3.7% 1.3h	9.5% 3.3h	13.4% 6.0h	5.7% 2.6h
Cluster 3 Data Preparer 122 people	24.5% 9.4h	4.9% 1.9h	19.6% 7.8h	10.0% 4.0h	3.0% 1.3h	9.0% 4.1h	11.6% 4.5h	8.8% 3.5h	1.5% 0.7h	3.9% 1.3h	1.5% 0.7h	1.8% 0.8h
Cluster 4 Data Shaper 33 people	5.6% 2.5h	1.8% 0.7h	27.0% 11.5h	25.7% 10.9h	6.0% 2.6h	8.9% 3.8h	7.6% 3.3h	7.5% 3.2h	2.1% 1.0h	3.3% 1.4h	2.5% 1.1h	1.9% 0.9h

Time spent on activities

	Query existing data	Build platforms to gather data	Prepare data	Analyze data	Experiment	Validate insight	Disseminate insight	Engage with others	Operationalize insight	Act on insight	Other work related to DS	Other work not related to DS
Entire population 532 people	12.0% 4.7h	7.2% 2.9h	11.7% 4.9h	12.5% 5.2h	4.8% 2.1h	6.9% 3.0h	8.5% 3.5h	9.2% 3.8h	2.4% 1.1h	5.5% 2.1h	4.1% 1.9h	15.1% 6.7h
Cluster 5 Data Analyzer 24 people	9.9% 3.7h	0.9% 0.3h	5.8% 2.4h	49.1% 18.4h	4.6% 2.2h	6.6% 2.7h	5.2% 2.2h	5.8% 2.4h	1.8% 0.9h	4.2% 1.6h	2.8% 1.3h	3.2% 1.3h
Cluster 6 Platform Builder 27 people	12.5% 4.4h	48.5% 18.4h	6.1% 2.6h	4.3% 1.9h	3.8% 1.1h	2.7% 1.2h	4.4% 2.0h	4.1% 1.9h	2.1% 0.9h	3.0% 1.1h	1.4% 0.6h	6.9% 3.1h
Cluster 7 Moonlighter 50% 63 people	7.3% 3.1h	5.0% 2.2h	5.0% 2.1h	5.5% 2.4h	2.8% 1.2h	4.2% 2.0h	7.8% 3.3h	5.9% 2.4h	1.8% 0.8h	5.7% 2.3h	2.5% 1.1h	46.5% 20.0h
Cluster 8 Moonlighter 20% 32 people	2.9% 1.2h	1.4% 0.6h	1.9% 0.9h	1.6% 0.7h	0.4% 0.2h	1.5% 0.7h	1.7% 0.8h	2.3% 1.0h	0.6% 0.3h	2.1% 1.0h	2.9% 1.3h	80.9% 36.1h
Cluster 9 Insight Actor 4 people	0.9% 0.1h	2.1% 1.0h	1.8% 0.2h		0.9% 0.1h	5.7% 1.5h	18.5% 4.8h	10.1% 1.6h	3.0% 1.1h	57.1% 11.8h		

Types of data scientists

		Harris et al. 2013
Generalists	Polymath “describes data scientists who ‘do it all’ ” 	Data Creatives “data scientists [who] can often tackle the entire soup-to-nuts analytics process on their own”
Specialists	Data Preparer	
	Data Shaper	
	Data Analyzer / Insight Provider “main task is to generate insights and to support and guide their managers in decision making” 	
	Platform Builder “build shared data platforms used across several product teams” 	Data Developer “people focused on the technical problem of managing data”
	Modelling Specialist “data scientists who act as expert consultants and build predictive models” 	
		Data Researcher people with “deep academic training in the use of data to understand complex processes”
Manager	Data Evangelist / Team Leader “senior data scientists who run their own data science teams act as data science ‘evangelists’ ” 	Data Businesspeople people who “are most focused on the organization and how data projects yield profit”
	Insight Actor	
Moonlighter	50% Moonlighter	
	20% Moonlighter	

Andrew Begel

Andrew Begel, Thomas Zimmermann:
Analyze this! 145 questions for data scientists in software engineering. ICSE 2014


Meet
Greg Wilson
from Mozilla
UNIVERSITY OF TORONTO

Ten Questions for Researchers

Posted Aug 22, 2012 by Greg Wilson

I gave the opening talk at MSR Vision 2020 in Kingston on Monday (slides), and in the wake of that, an experienced developers at Mozilla sent me a list of ten questions he'd really like empirical software engineering researchers to answer. They're interesting in their own right, but I think they also reveal a lot about what practitioners want from researchers in general; comments would be very welcome.

1. Vi vs. Emacs vs. graphical editors/IDEs: which makes me more productive?
2. Should language developers spend their time on tools, syntax, library, or something else (like speed)? What makes the most difference to their users?
3. Do unit tests save more time in debugging than they take to write/run/keep updated?

Let’s ask Microsoft engineers what they would like to know!

http://aka.ms/**145**Questions

1

SURVEY

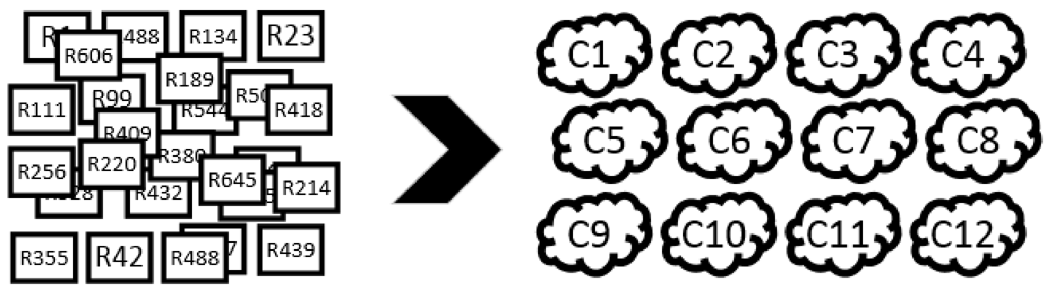
203 participants, 728 questions R1..R728

Suppose you could work with a team of data scientists and data analysts who specialize in studying how software is developed. Please list up to five questions you would like them to answer.

✶ CATEGORIES

12 categories C1..C12

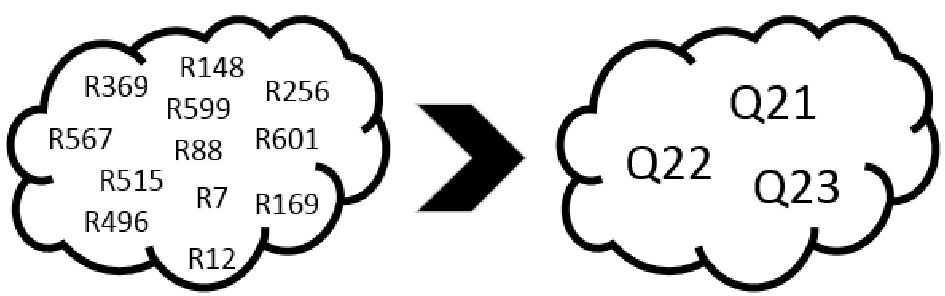
Use an open card sort to group questions into categories.



✶ DESCRIPTIVE QUESTIONS

145 questions Q1..Q145

Summarize each category with a set of descriptive questions.



2

SURVEY

607 participants, 16 765 ratings

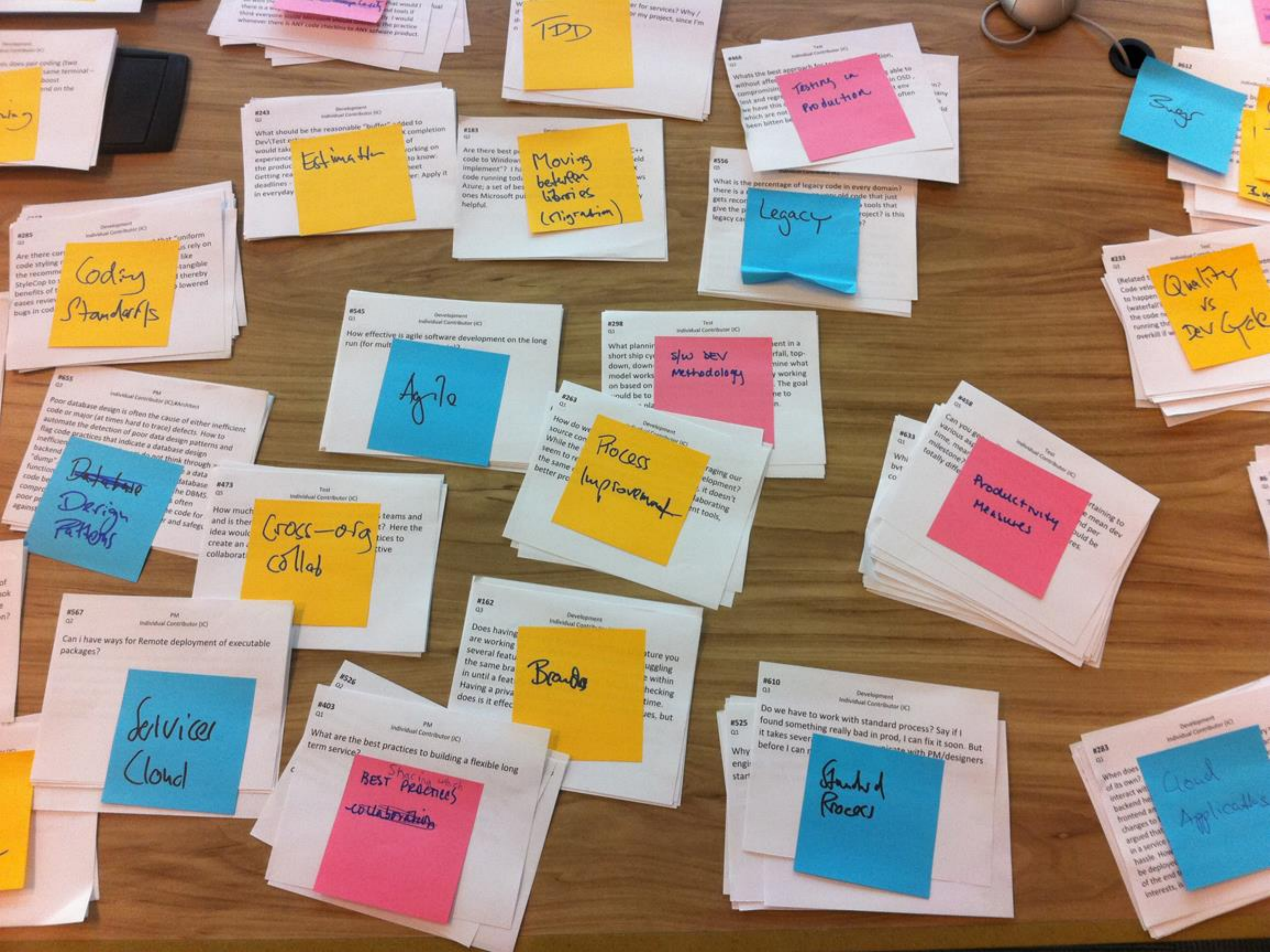
Split questionnaire design, where each participant received a subset of the questions Q1..Q145 (on average 27.6) and was asked:

In your opinion, how important is it to have a software data analytics team answer this question?
[Essential | Worthwhile | Unimportant | Unwise | I don't understand]

✶ TOP/BOTTOM RANKED QUESTIONS

✶ DIFFERENCES IN DEMOGRAPHICS

Discipline: Development, Testing, Program Management
Region: Asia, Europe, North America, Other
Number of Full-Time Employees
Current Role: Manager, Individual Contributor
Years as Manager
Has Management Experience: yes, no.
Years at Microsoft



TDD
Testing in Production
Bugs
Estimation
Moving between libraries (migration)
Legacy
Coding Standards
Quality vs Dev Cycle
Agile
S/W DEV Methodology
Process Improvement
Productivity Measures
~~Database~~ Design Patterns
Cross-org collab
Branches
Services Cloud
Sharing which BEST PRACTICES ~~collaboration~~
Standard Process
Cloud Applications
Can i have ways for Remote deployment of executable packages?
What are the best practices to building a flexible long term service?
Do we have to work with standard process? Say if I found something really bad in prod, I can fix it soon.

Microsoft's Top 10 Questions	Essential	Essential + Worthwhile
How do users typically use my application?	80.0%	99.2%
What parts of a software product are most used and/or loved by customers?	72.0%	98.5%
How effective are the quality gates we run at checkin?	62.4%	96.6%
How can we improve collaboration and sharing between teams?	54.5%	96.4%
What are the best key performance indicators (KPIs) for monitoring services?	53.2%	93.6%
What is the impact of a code change or requirements change to the project and its tests?	52.1%	94.0%
What is the impact of tools on productivity?	50.5%	97.2%
How do I avoid reinventing the wheel by sharing and/or searching for code?	50.0%	90.9%
What are the common patterns of execution in my application?	48.7%	96.6%
How well does test coverage correspond to actual code usage by our customers?	48.7%	92.0%

Microsoft's 10 Most Unwise Questions	Unwise
Which **individual measures correlate with employee productivity** (e.g. employee age, tenure, engineering skills, education, promotion velocity, IQ)?	25.5%
Which **coding measures correlate with employee productivity** (e.g. lines of code, time it takes to build software, particular tool set, pair programming, number of hours of coding per day, programming language)?	22.0%
What metrics can use used to **compare employees**?	21.3%
How can we **measure the productivity of a Microsoft employee?**	20.9%
Is the number of bugs a good **measure of developer effectiveness**?	17.2%
Can I generate 100% test coverage?	14.4%
Who should be in charge of creating and maintaining a consistent company-wide software process and tool chain?	12.3%
What are the benefits of a consistent, company-wide software process and tool chain?	10.4%
When are code comments worth the effort to write them?	9.6%
How much time and money does it cost to add customer input into your design?	8.3%


Productivity

What is productivity?

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction

Help
About Wikipedia
Community portal
Recent changes
Contact page

Not logged in Talk Contributions Create account Log in

Article Talk Read Edit View history Search Wikipedia

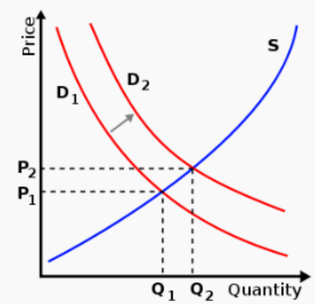
Productivity

From Wikipedia, the free encyclopedia

This article is about the economic and business concept. For other uses, see Productivity (disambiguation).

Productivity describes various measures of the efficiency of production. A productivity measure is expressed as the ratio of output to inputs used in a production process, i.e. output per unit of input. Productivity is a crucial factor in production performance of firms and nations. Increasing national productivity can raise living standards because more real income improves people's ability to purchase goods and services, enjoy leisure, improve housing and education and contribute to social and environmental programs. Productivity growth also helps businesses to be more profitable.[1] There are many different definitions of productivity and the choice among them depends on the purpose of the productivity measurement and/or data availability.

Economics

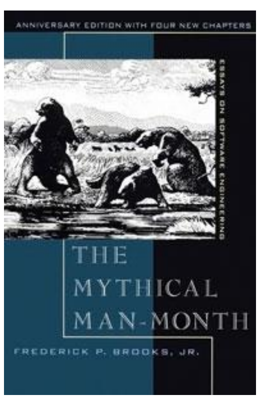


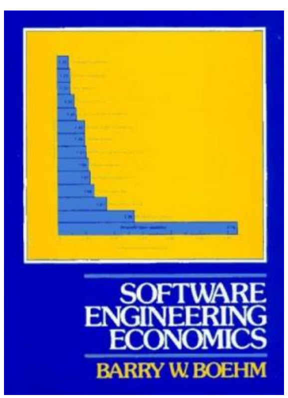
A supply and demand diagram, illustrating the effects of an increase in demand.

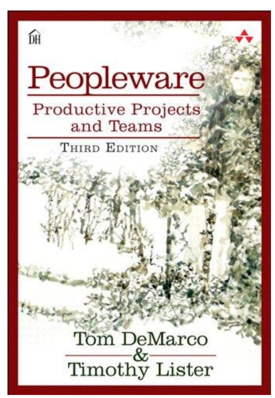
Index · Outline · Category

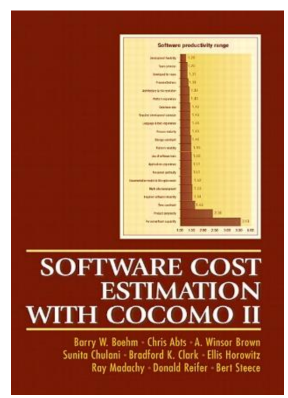
What is productivity?

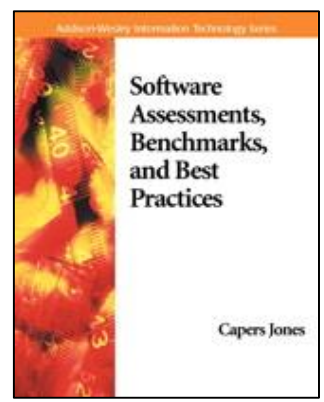
$$\text{Productivity} = \frac{\text{Output}}{\text{Input}}$$











Examples of productivity measures

From academic papers

- number of modification requests and added lines of code per year
- number of tasks per month
- number of function points per month
- number of source lines of code per hour
- number of lines of code per person month of coding effort
- amount of work completed per reported hour of effort for each technology
- ratio of produced logical code lines and spent effort
- average number of logical source statements output per month over the product development cycle
- total equivalent lines of code per person-month
- resolution time defined as the time, in days, it took to resolve a particular modification request
- number of editing events to number of selection and navigation events needed to find where to edit code

What influences productivity?

<u>Technical factors</u>

Product: complexity, quality, constraints

Process: maturity, completeness of design

Development environment:
tools, modern development practices, programming language, documentation

Wagner, S. and Ruhe, M. (2008) A Systematic Review of Productivity Factors in Software Development. In Proc. 2nd International Workshop on Software Productivity Analysis and Cost Estimation (SPACE 2008).

What influences productivity?

Social factors

Corporate culture: fairness, respect, credibility

Team culture: team cohesion, turnover

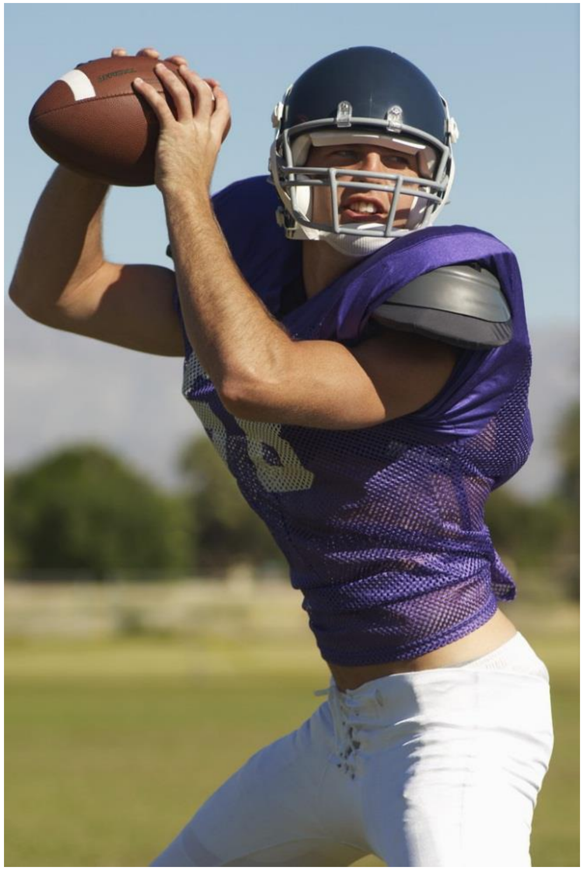
Capabilities and experiences:
programmer capability, experience with application domain, platform, language, and tool,

Work environment: fragmentation, separation

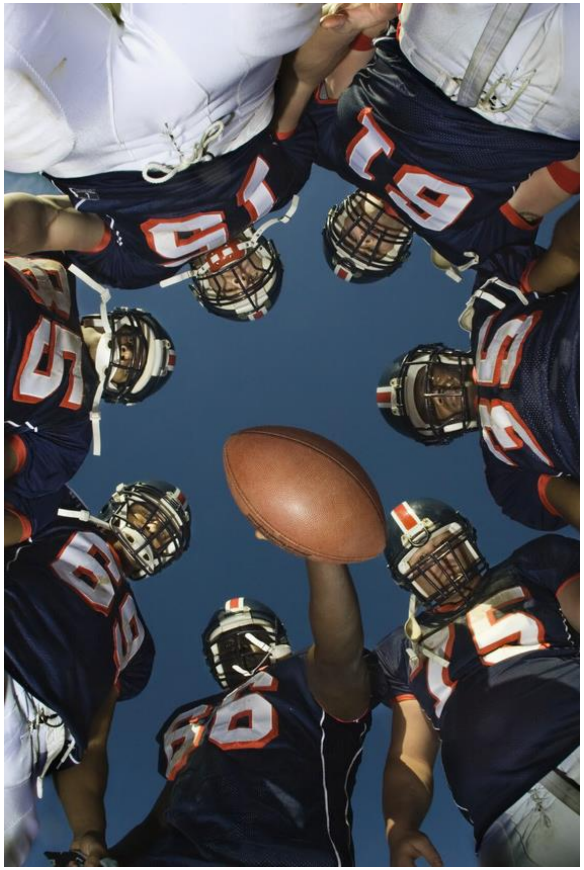
Project: average team size

Wagner, S. and Ruhe, M. (2008) A Systematic Review of Productivity Factors in Software Development. In Proc. 2nd International Workshop on Software Productivity Analysis and Cost Estimation (SPACE 2008).

Different levels of productivity



Individual

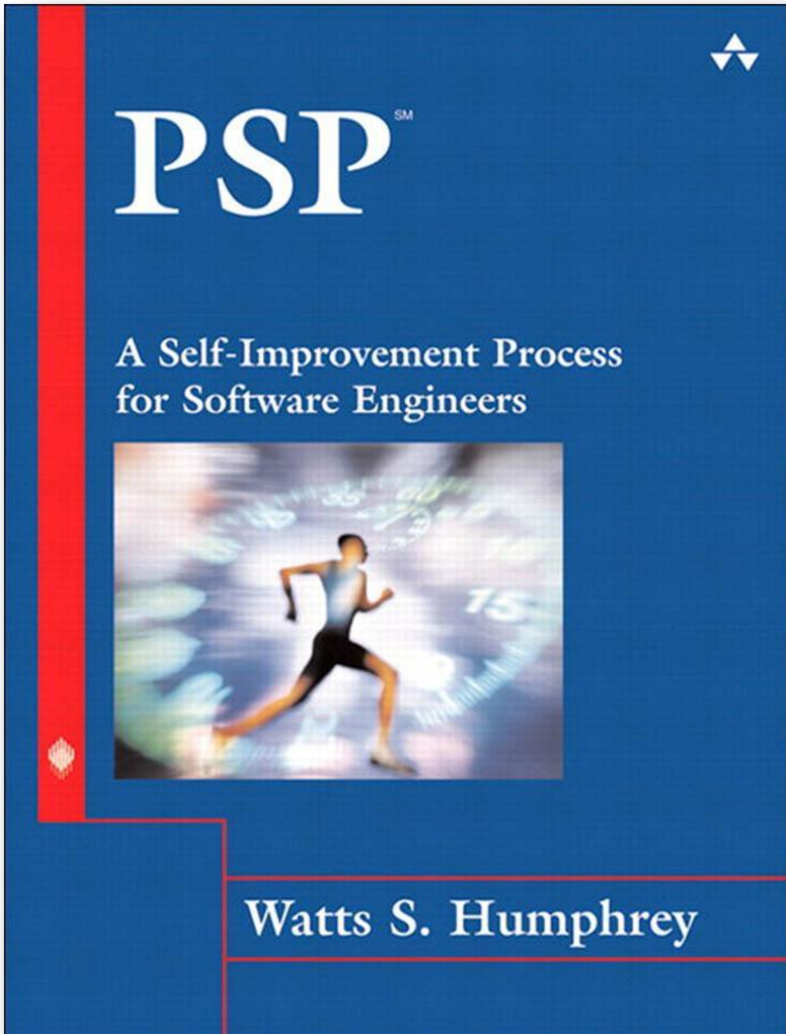


Team

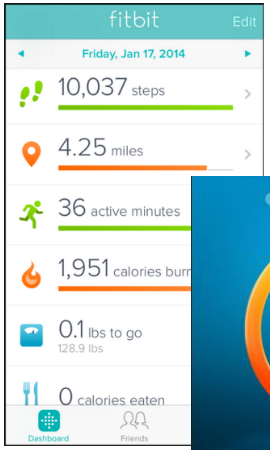
Organization


Individual
Productivity

Personal Software Process



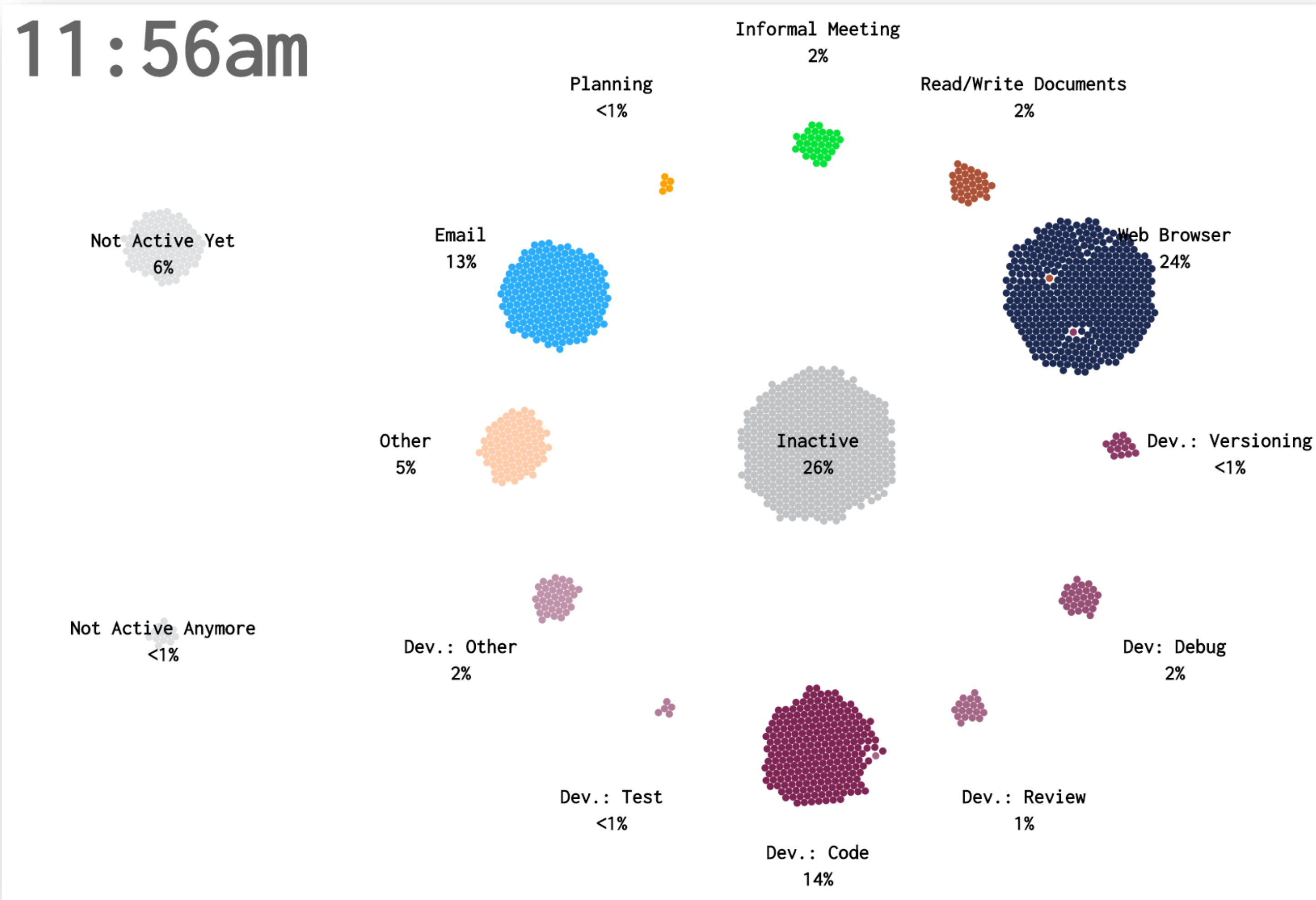
Fitness Tracking



The stereotypical software developer

100% of the time in front of a computer

A Day in the Life of Developers



Moritz Beller. Microsoft Internship 2017

Thomas
Fritz

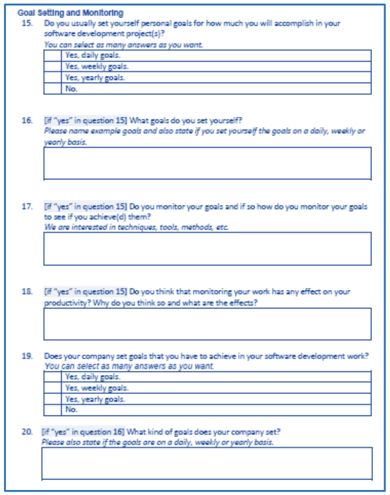
Gail
Murphy

Andre Meyer

André N. Meyer, Laura Barton, Gail C. Murphy, Thomas Zimmermann, Thomas Fritz. The Work Life of Developers: Activities, Switches and Perceived Productivity. IEEE TSE

André N. Meyer, Thomas Fritz, Gail C. Murphy, Thomas Zimmermann: Software developers' perceptions of productivity. SIGSOFT FSE 2014

Survey: Method & Participants



Goal Setting and Monitoring

15. Do you usually set yourself personal goals for how much you will accomplish in your software development project(s)?
You can select as many answers as you want.

	Yes, daily goals.
	Yes, weekly goals.
	Yes, yearly goals.
	No.

16. [if "yes" in question 15] What goals do you set yourself?
Please name example goals and also state if you set yourself the goals on a daily, weekly or yearly basis.

17. [if "yes" in question 15] Do you monitor your goals and if so how do you monitor your goals to see if you achieve(d) them?
We are interested in techniques, tools, methods, etc.

18. [if "yes" in question 15] Do you think that monitoring your work has any effect on your productivity? Why do you think so and what are the effects?

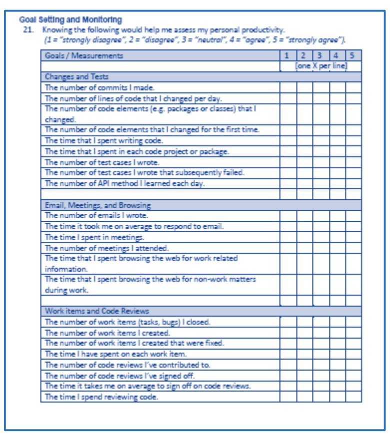
19. Does your company set goals that you have to achieve in your software development work?
You can select as many answers as you want.

	Yes, daily goals.
	Yes, weekly goals.
	Yes, yearly goals.
	No.

20. [if "yes" in question 16] What kind of goals does your company set?
Please also state if the goals are on a daily, weekly or yearly basis.

28 questions on background, perceptions, assessing, measuring and improving productivity

Recruitment through emails and posts in online forums



Goal Setting and Monitoring

21. Knowing the following would help me assess my personal productivity.
(1 = "strongly disagree", 2 = "disagree", 3 = "neutral", 4 = "agree", 5 = "strongly agree").

Goals / Measurements	1	2	3	4	5
	[one X per line]				
Changes and Tests					
The number of commits I made.					
The number of lines of code that I changed per day.					
The number of code elements (e.g. packages or classes) that I changed.					
The number of code elements that I changed for the first time.					
The time that I spent writing code.					
The time that I spent in each code project or package.					
The number of test cases I wrote.					
The number of test cases I wrote that subsequently failed.					
The number of API method I learned each day.					
Email, Meetings, and Browsing					
The number of emails I wrote.					
The time it took me on average to respond to email.					
The time I spent in meetings.					
The number of meetings I attended.					
The time that I spent browsing the web for work related information.					
The time that I spent browsing the web for non-work matters during work.					
Work items and Code Reviews					
The number of work items (tasks, bugs) I closed.					
The number of work items I created.					
The number of work items I created that were fixed.					
The time I have spent on each work item.					
The number of code reviews I've contributed to.					
The number of code reviews I've signed off.					
The time it takes me on average to sign off on code reviews.					
The time I spend reviewing code.					

379 participants

194 within Microsoft, 185 public survey

9.2 years of professional experience on average

I have a productive workday when

I complete the **goals** I set for myself that day.

I get through my list of **tasks** for the day.

I write more than 10 **lines of code**

I wrote at least 1 line of code

I get enough rest and can
take nap when I feel tired.

I am in a good **mood** and **slept** well :)

I am not randomized too much by **meetings** and **interruptions** from others

I am **focused** on one work item and have all the inputs I need to close on that work item.

I have a productive workday when I have **autonomy** on what I do and don't have to wait for response from other side of earth.

I have at most one meeting.
I have no code reviews to complete.

Get what I have planned done

When the **weather** is dry and I can work at a silent and well ventilated **place**.

Survey: Method & Participants

s. e. a. l.
software evolution & architecture lab

University of Zurich[UZH]

27%

Productivity

Please complete the following sentence in up to three ways:

I have a productive workday when…

1
2
3

Are you satisfied with your productivity of your prior workday?

- very unsatisfied
- unsatisfied
- undecided
- satisfied
- very satisfied

Are you satisfied with your productivity last week?

- very unsatisfied
- unsatisfied
- undecided
- satisfied
- very satisfied

I have a productive workday when

Complete tasks or goals	53.2%
Have no/few interruptions and distractions	50.4%
Have no meetings	21.9%
Have clear goals	19.9%
Plan my workday	17.2%

Developers feel productive when they make progress on tasks with few context switches.

Productive/unproductive activities

Productive activities

Coding	72%
Meetings	17%
Planning	7%

Unproductive activities

Meetings	58%
Emails	19%
Unplanned work	18%

Based on a survey of 379 professional developers from many different companies

Measures to assess productivity

Which measures are helpful to assess your productivity?

Number of

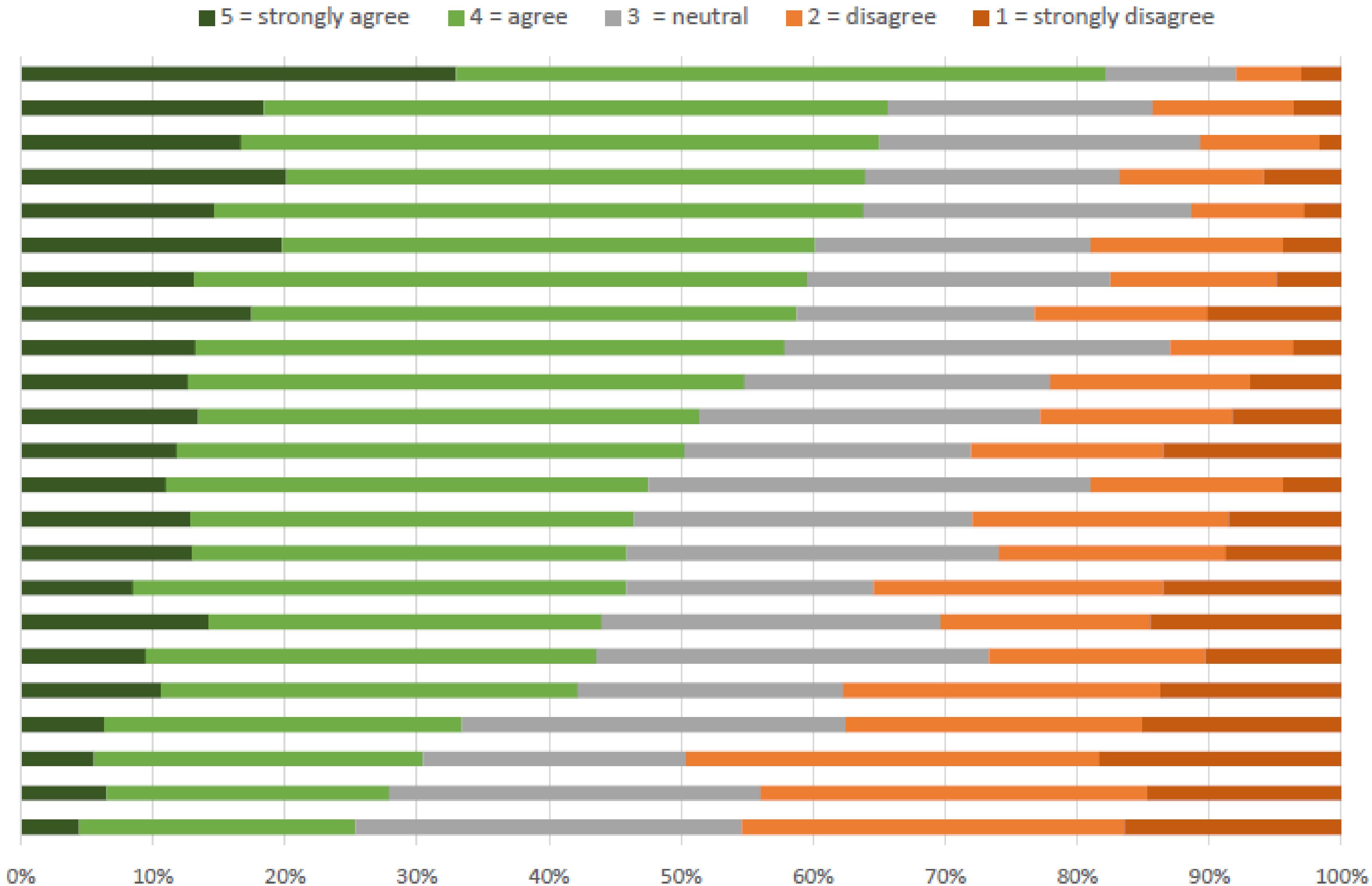
API methods I learned
Code elements I changed
Code elements I changed
for the first time
Code reviews I've contributed to
Code reviews I've signed off
Commits I made
Emails I wrote
Lines of code I changed per day
Meetings I attended
Test cases I wrote
Test cases I wrote
that subsequently failed
Work items I closed
Work items I created
Work items I created that were fixed

Time spent

Writing code
Reviewing code
Meetings
Browsing for work
Browsing for personal matters
Per code project or package
Per work item
Sign off on code reviews (average)
Respond to email (average)

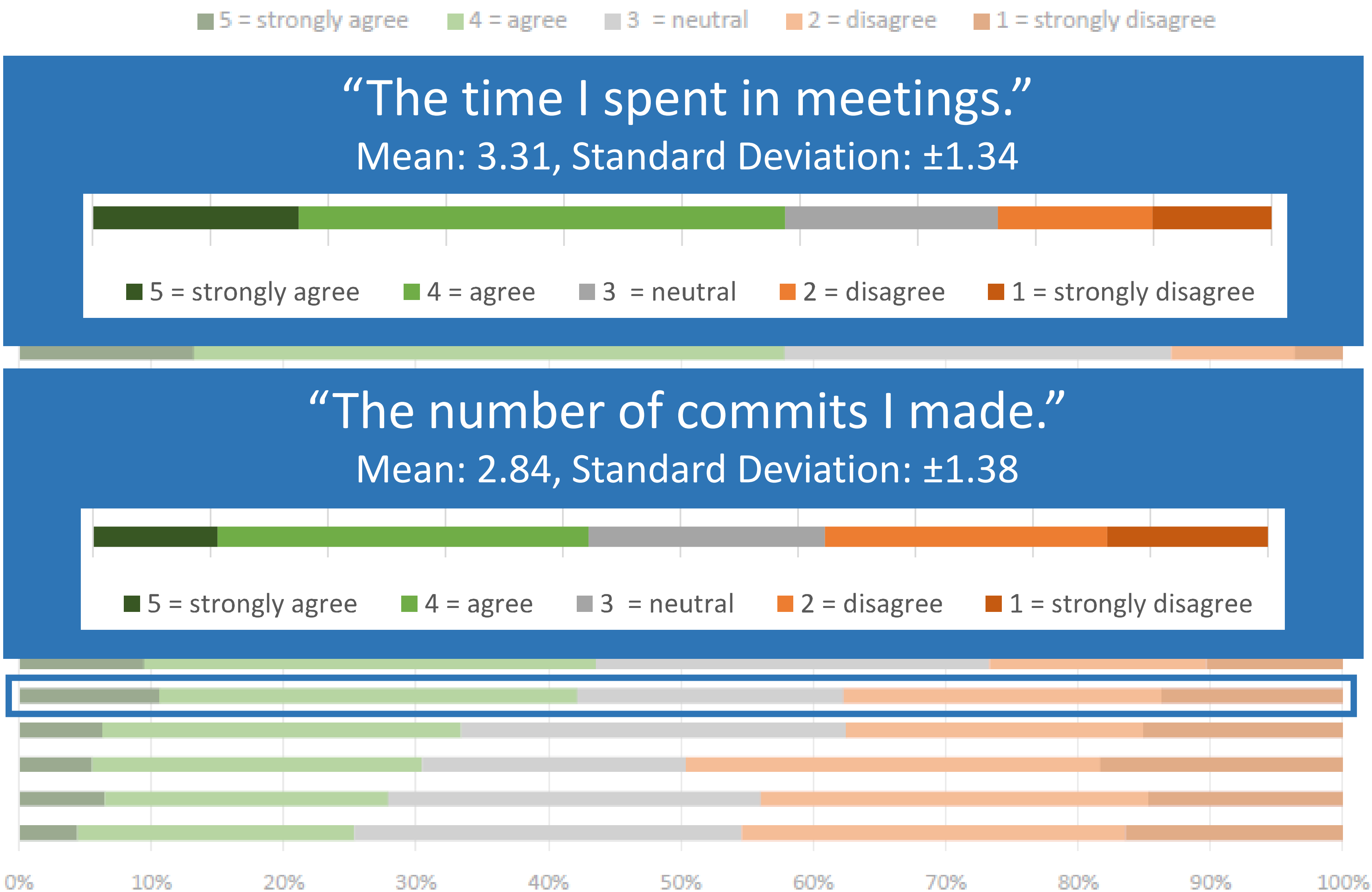
Measures to assess productivity

Which measures are helpful to assess your productivity?



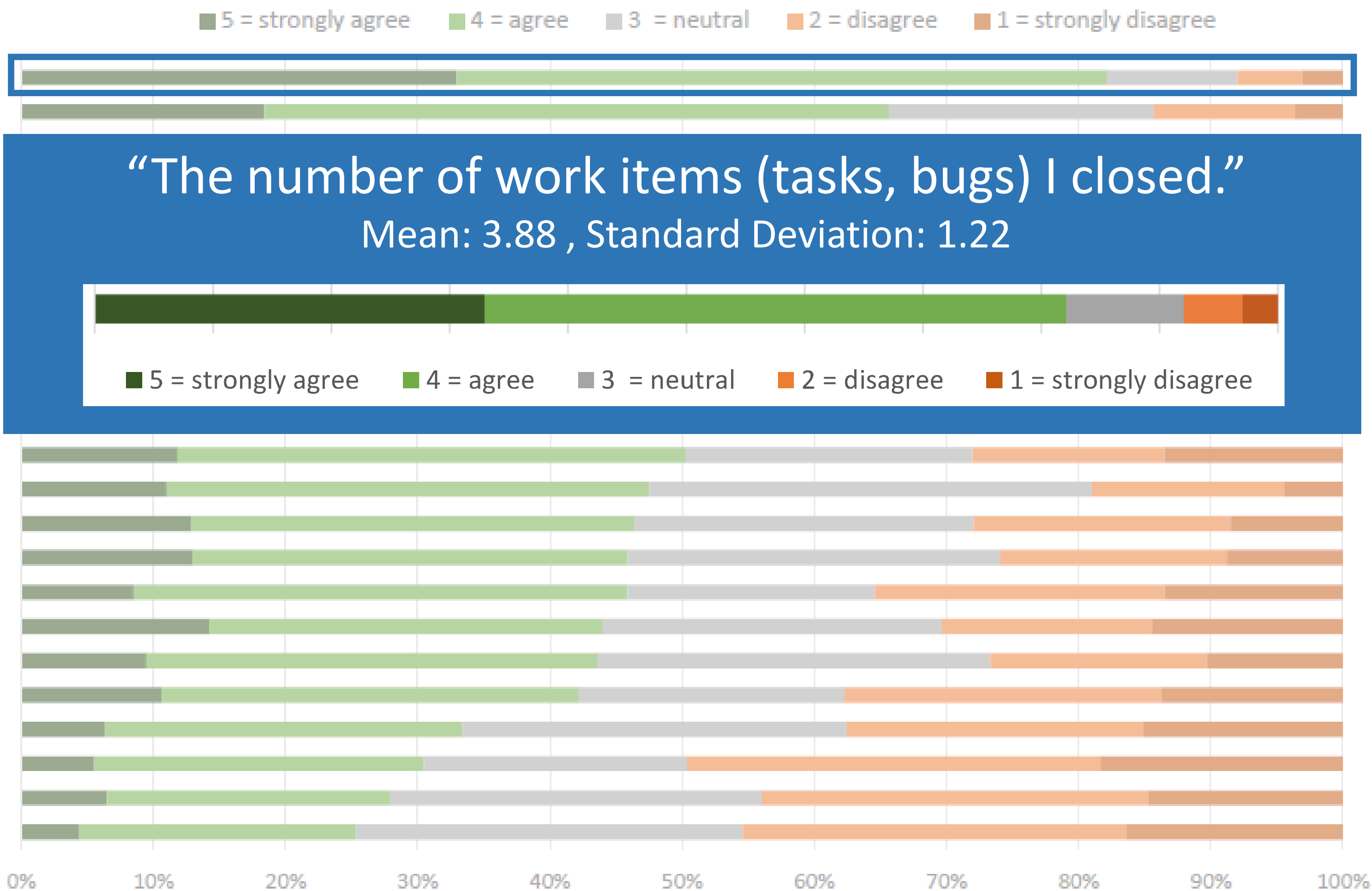
Measures to assess productivity

Which measures are helpful to assess your productivity?

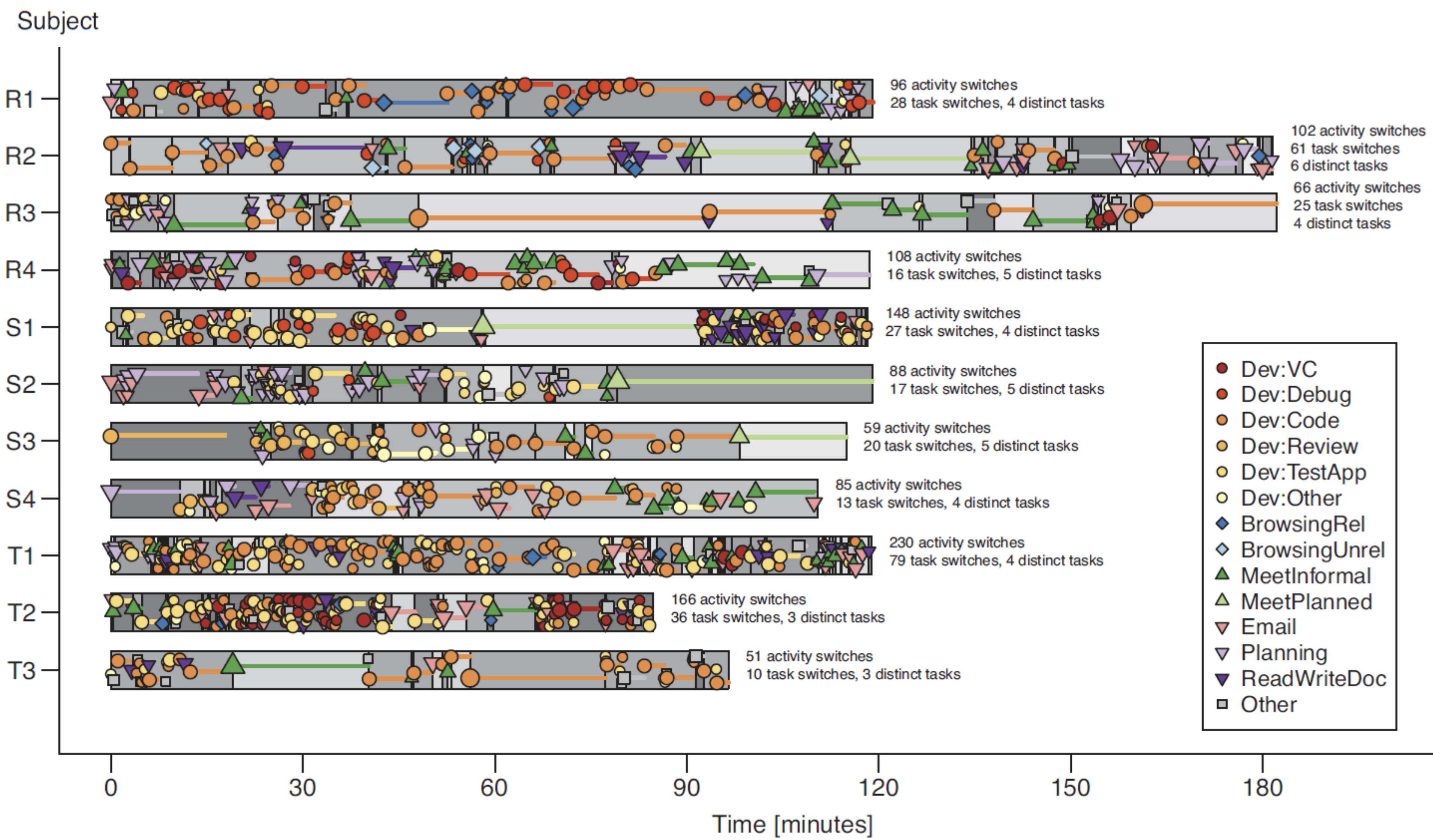


Measures to assess productivity

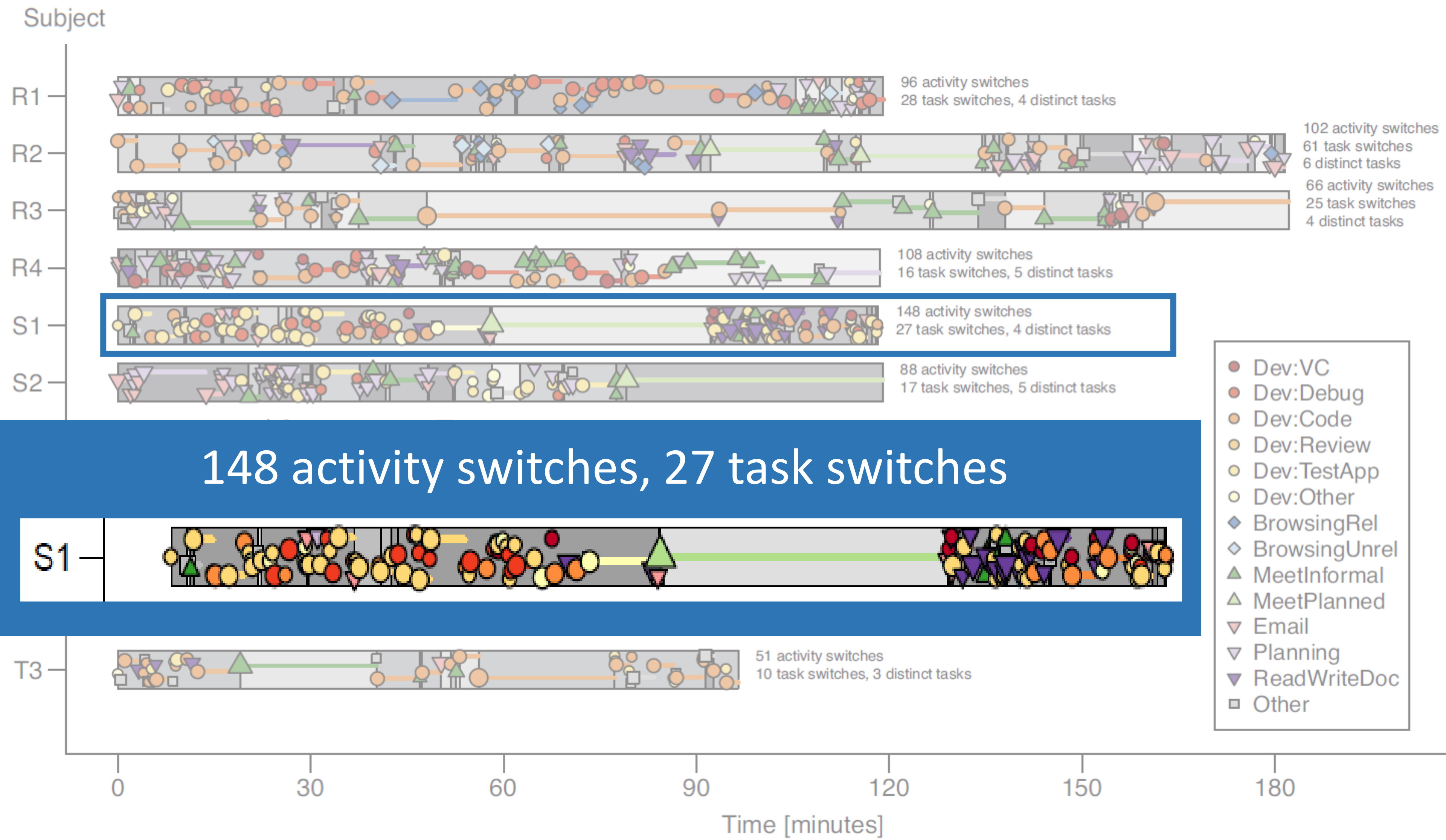
Which measures are helpful to assess your productivity?



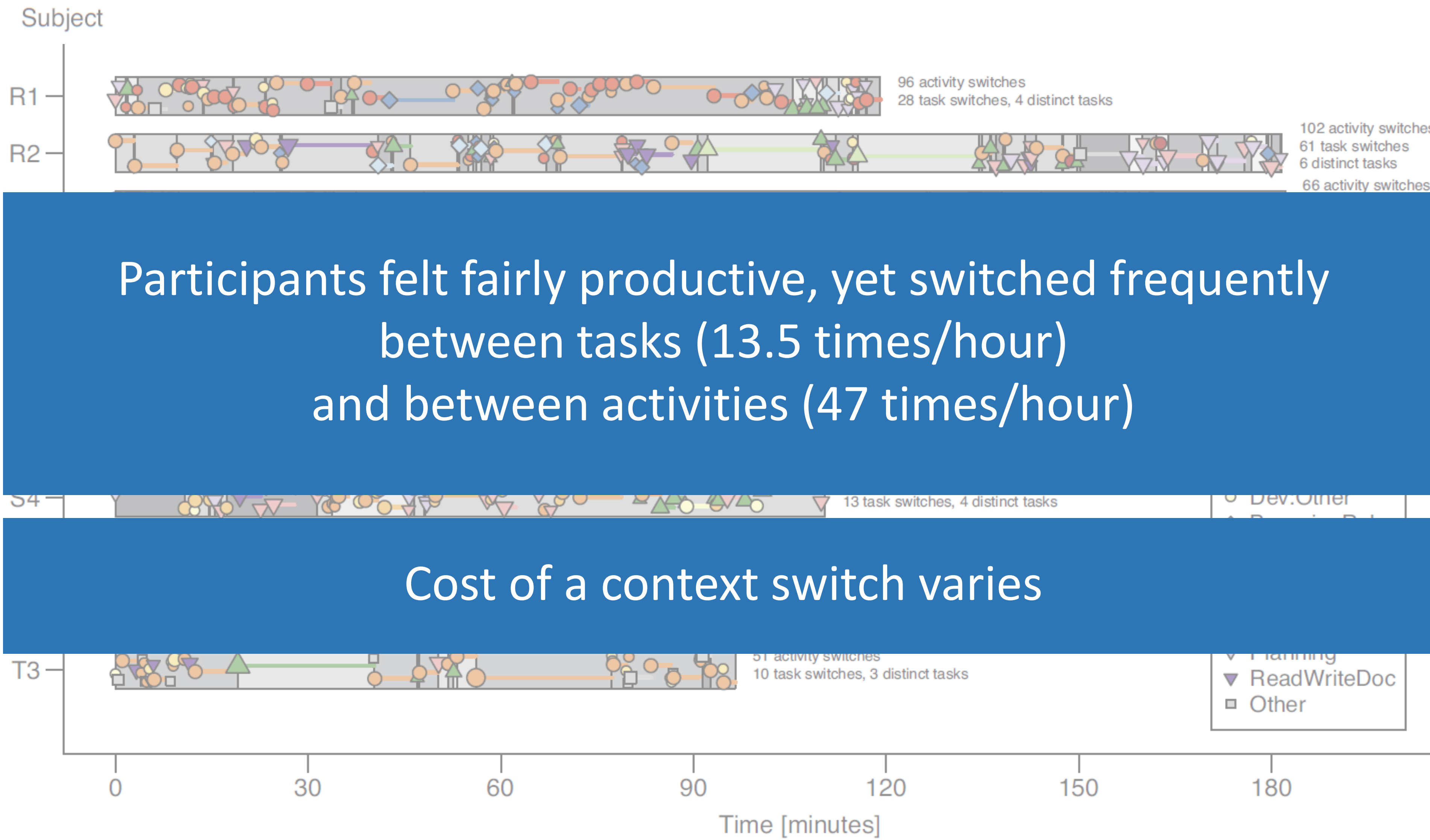
Observing developers



Observing developers



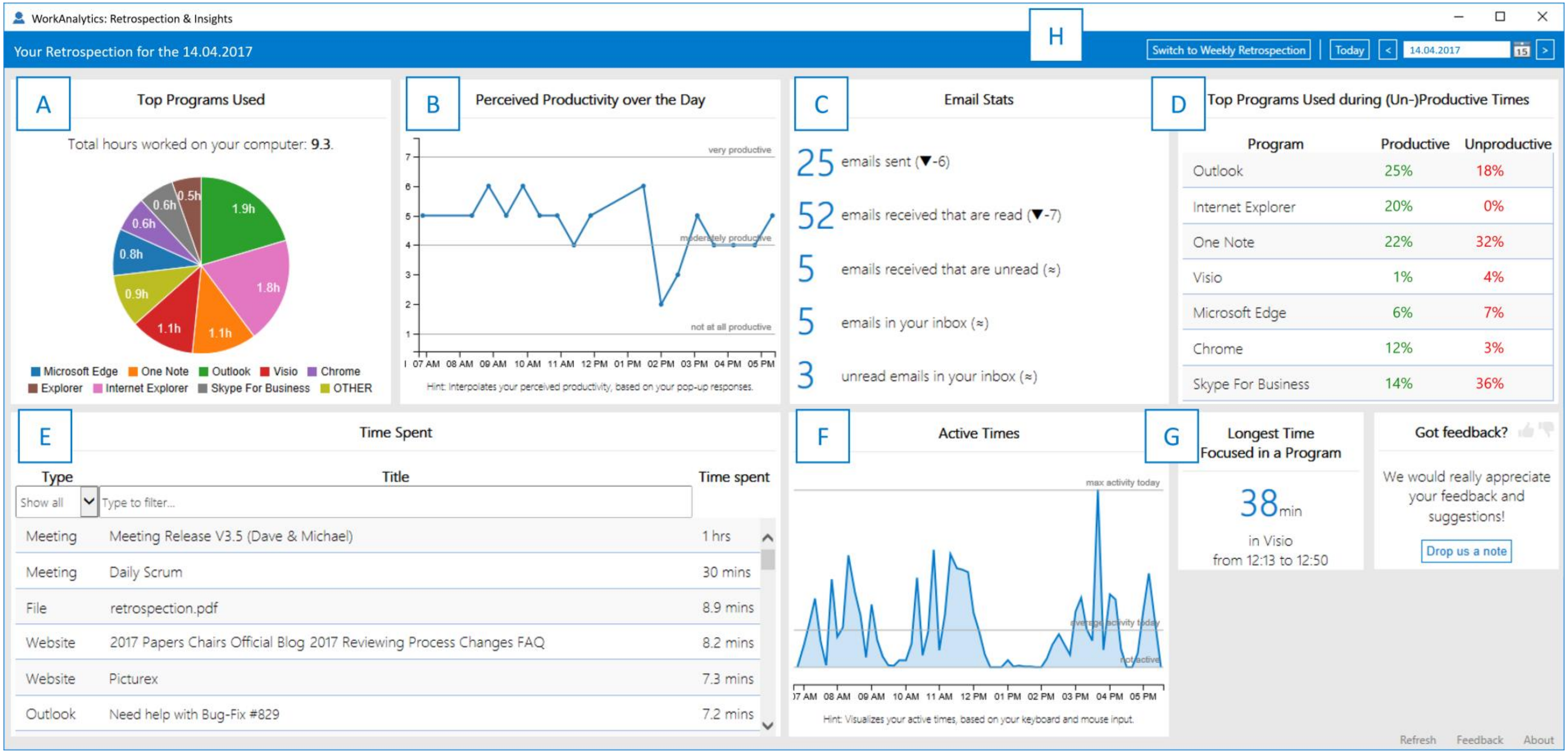
Observing developers



Participants felt fairly productive, yet switched frequently between tasks (13.5 times/hour) and between activities (47 times/hour)

Cost of a context switch varies

Retrospection



The tool collects data about:

Activity – Application usage – Meeting information – Email statistics – Perceived Productivity

Additional findings

Based on data collected from 20 users of the retrospection tool

Developers only spend about **half their time active on their computer**.

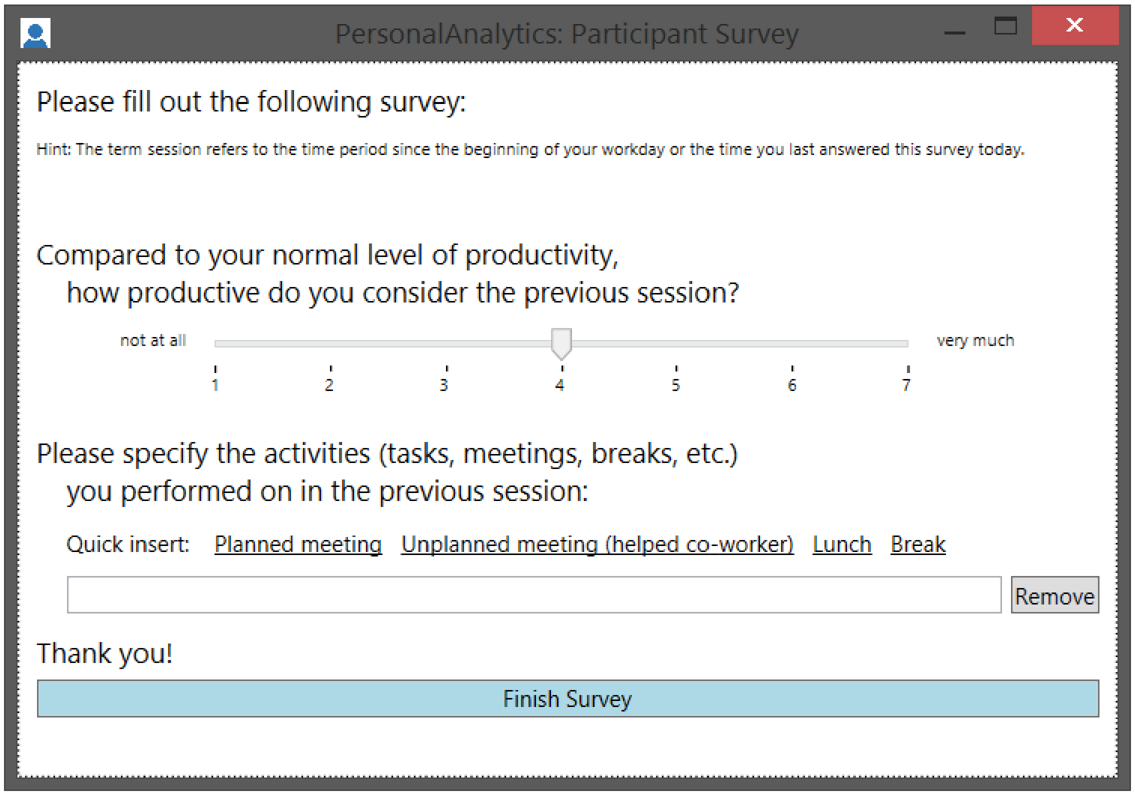
For every work hour, developers have an average of 2.5 short breaks, totaling 10.5 minutes of unplanned time away from their computer.

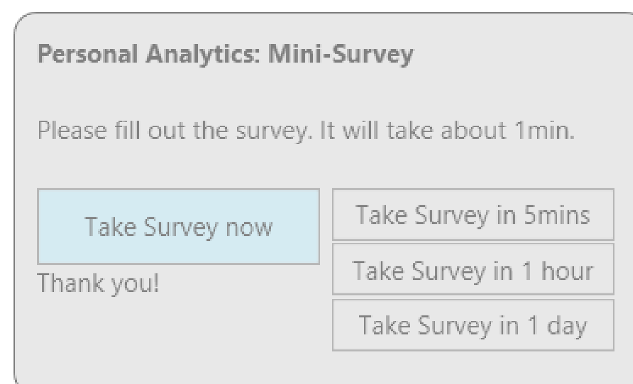
Developers spend about **a fourth of their time on coding** related activities and another fourth of their time on collaborative activities.

The range and time spent on **activities varies greatly** depending on the individual and company.

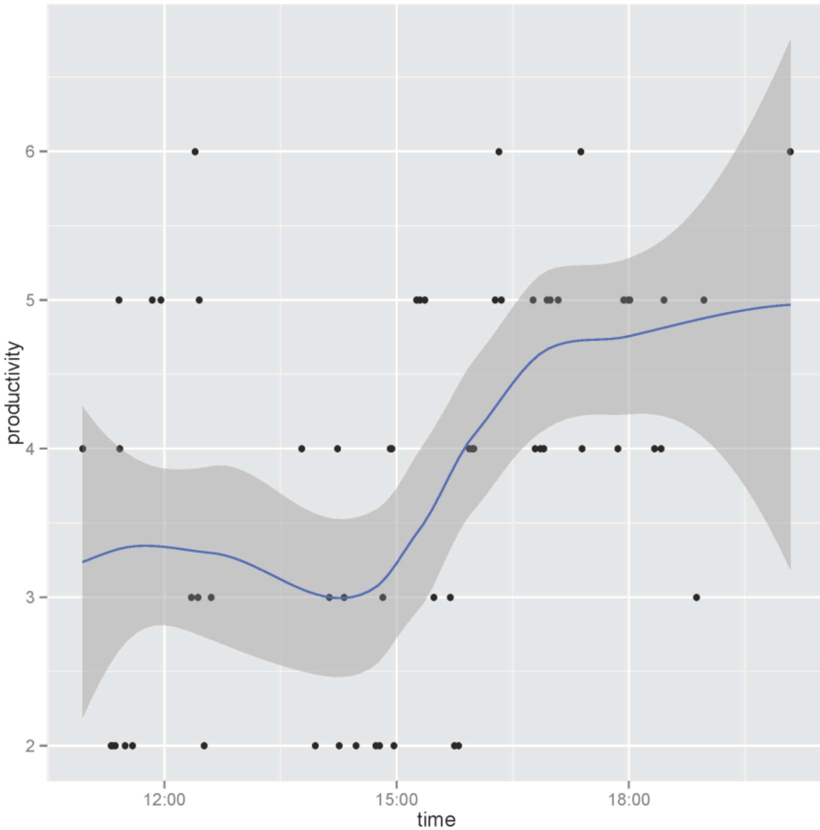
Developers’ **work is highly fragmented**, spending very short amounts of time (0.3 to 2 minutes) in one activity before switching to another one.

When are people productive?

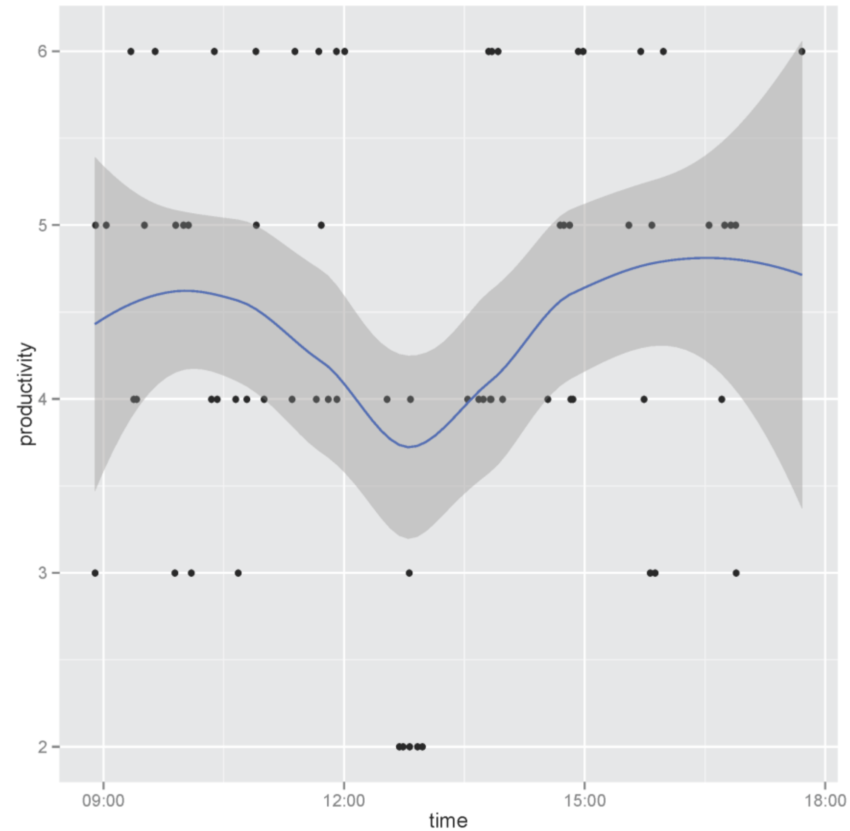




Productivity patterns



Higher productivity
in the afternoon



Lower productivity
during lunch

Productivity models

Neg	Pos	NA	Participant	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
			Ratings (total)	45	101	62	94	40	62	80	73	89	92	51	40	76	88	71	42	62	53	100	30
			Ratings (discarded)	0	29	3	4	0	0	0	7	10	10	1	0	7	0	0	0	4	8	11	4
			Ratings (included in model)	45	72	59	90	40	62	80	66	79	82	50	40	69	88	71	42	58	45	89	26
Neg	**Pos**	**NA**	Ratings (distribution)																				
3	3	0	**Session Duration (in hours)**		+	−	−		+		−					+							
2	1	0	**Percent of Session Before Noon**											−			−				+		
			Per Minute																				
2	2	0	# self-reported tasks	+								−		−		+							
2	4	0	# activity switches		+		−		+				+	−				+					
1	1	4	# meetings	NA		NA				+				NA				NA	−				
2	1	7	# instant messaging switches	NA	NA	NA	NA	+		NA	NA		NA								+		−
1	7	0	# keystrokes					+						+		+	−	+		+	+		+
2	7	0	# mouse clicks	+		+				+		+	+	+	−		+				−		
			Percent Activity																				
4	1	0	Dev. Coding						−				−			−					+		−
0	1	9	Dev. Debugger Use	NA	NA	NA				NA	NA	NA	NA	+							NA		NA
2	1	12	Dev. Code Reviews	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	−			NA	NA				−	+
2	0	3	Dev. Version Control		NA		−		NA				−							NA			
5	0	0	Email							−			−	−		−						−	
2	3	0	Planning		+		+							−	+							−	
4	3	0	Read / write documents	+		+			−				−	+		−				−			
6	0	5	Planned meeting	NA		NA		−	−	−		−		NA	NA	−		NA					−
3	2	9	Informal meeting	NA	+	NA	NA	NA					−	NA	NA	−	NA	NA		NA	+		−
2	0	8	Instant messaging	NA	NA	NA	NA	NA	−	NA	NA		NA								−		
1	2	0	Work related browsing							−			+								+		
5	0	2	Work unrelated browsing					−			NA		−			−	−	−		NA			
2	1	0	Other								+						−						−
2	1	4	Other RDP	NA		NA		−			+						NA	−					NA
6	2	2	Idle		−		−	−	−		+	−		NA		−	+	NA					

Productivity models

Participant	S1
Ratings (total)	45
Ratings (discarded)	0
Ratings (included in model)	45
Ratings (distribution)	
Session Duration (in hours)	
Percent of Session Before Noon	
Per Minute	
# self-reported tasks	+
# activity switches	
# meetings	NA
# instant messaging switches	NA
# keystrokes	
# mouse clicks	+
Percent Activity	
Dev. Coding	
Dev. Debugger Use	NA
Dev. Code Reviews	NA
Dev. Version Control	
Email	
Planning	
Read / write documents	+
Planned meeting	NA
Informal meeting	NA
Instant messaging	NA
Work related browsing	
Work unrelated browsing	
Other	
Other RDP	NA
Idle	

Productivity models

Participant	S2
Ratings (total)	101
Ratings (discarded)	29
Ratings (included in model)	72
Ratings (distribution)	
Session Duration (in hours)	+
Percent of Session Before Noon	
Per Minute	
# self-reported tasks	
# activity switches	+
# meetings	
# instant messaging switches	NA
# keystrokes	
# mouse clicks	
Percent Activity	
Dev. Coding	
Dev. Debugger Use	NA
Dev. Code Reviews	NA
Dev. Version Control	NA
Email	
Planning	+
Read / write documents	
Planned meeting	
Informal meeting	+
Instant messaging	NA
Work related browsing	
Work unrelated browsing	
Other	
Other RDP	
Idle	–

Productivity models

Participant	S3
Ratings (total)	62
Ratings (discarded)	3
Ratings (included in model)	59
Ratings (distribution)	
Session Duration (in hours)	−
Percent of Session Before Noon	
Per Minute	
# self-reported tasks	
# activity switches	
# meetings	NA
# instant messaging switches	NA
# keystrokes	
# mouse clicks	+
Percent Activity	
Dev. Coding	
Dev. Debugger Use	NA
Dev. Code Reviews	NA
Dev. Version Control	
Email	
Planning	
Read / write documents	+
Planned meeting	NA
Informal meeting	NA
Instant messaging	NA
Work related browsing	
Work unrelated browsing	
Other	
Other RDP	NA
Idle	

Productivity models

Neg	Pos	NA	Participant	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
			Ratings (total)	45	101	62	94	40	62	80	73	89	92	51	40	76	88	71	42	62	53	100	30
			Ratings (discarded)	0	29	3	4	0	0	0	7	10	10	1	0	7	0	0	0	4	8	11	4
			Ratings (included in model)	45	72	59	90	40	62	80	66	79	82	50	40	69	88	71	42	58	45	89	26
Neg	**Pos**	**NA**	Ratings (distribution)																				
3	3	0	**Session Duration (in hours)**		+	−	−		+		−					+							
2	1	0	**Percent of Session Before Noon**											−			−				+		
			Per Minute																				
2	2	0	# self-reported tasks	+																			
2	4	0	# activity switches		+		−																
1	1	4	# meetings	NA		NA																	
2	1	7	# instant messaging switches	NA	NA	NA	NA																
1	7	0	# keystrokes																				
2	7	0	# mouse clicks	+		+																	
			Percent Activity																				
4	1	0	Dev. Coding																				
0	1	9	Dev. Debugger Use	NA	NA	NA																	
2	1	12	Dev. Code Reviews	NA	NA	NA	NA																
2	0	3	Dev. Version Control		NA		−																
5	0	0	Email																				
2	3	0	Planning		+		+																
4	3	0	Read / write documents	+		+																	
6	0	5	Planned meeting	NA		NA																	
3	2	9	Informal meeting	NA	+	NA	NA																
2	0	8	Instant messaging	NA	NA	NA	NA																
1	2	0	Work related browsing							−			+								+		
5	0	2	Work unrelated browsing					−			NA		−			−	−	−		NA			
2	1	0	Other								+						−						−
2	1	4	Other RDP	NA		NA		−			+						NA	−					NA
6	2	2	Idle		−		−	−	−		+	−		NA		−	+	NA					

No two explanatory models are the same.

Perceived productivity depends on the individual

Productivity models

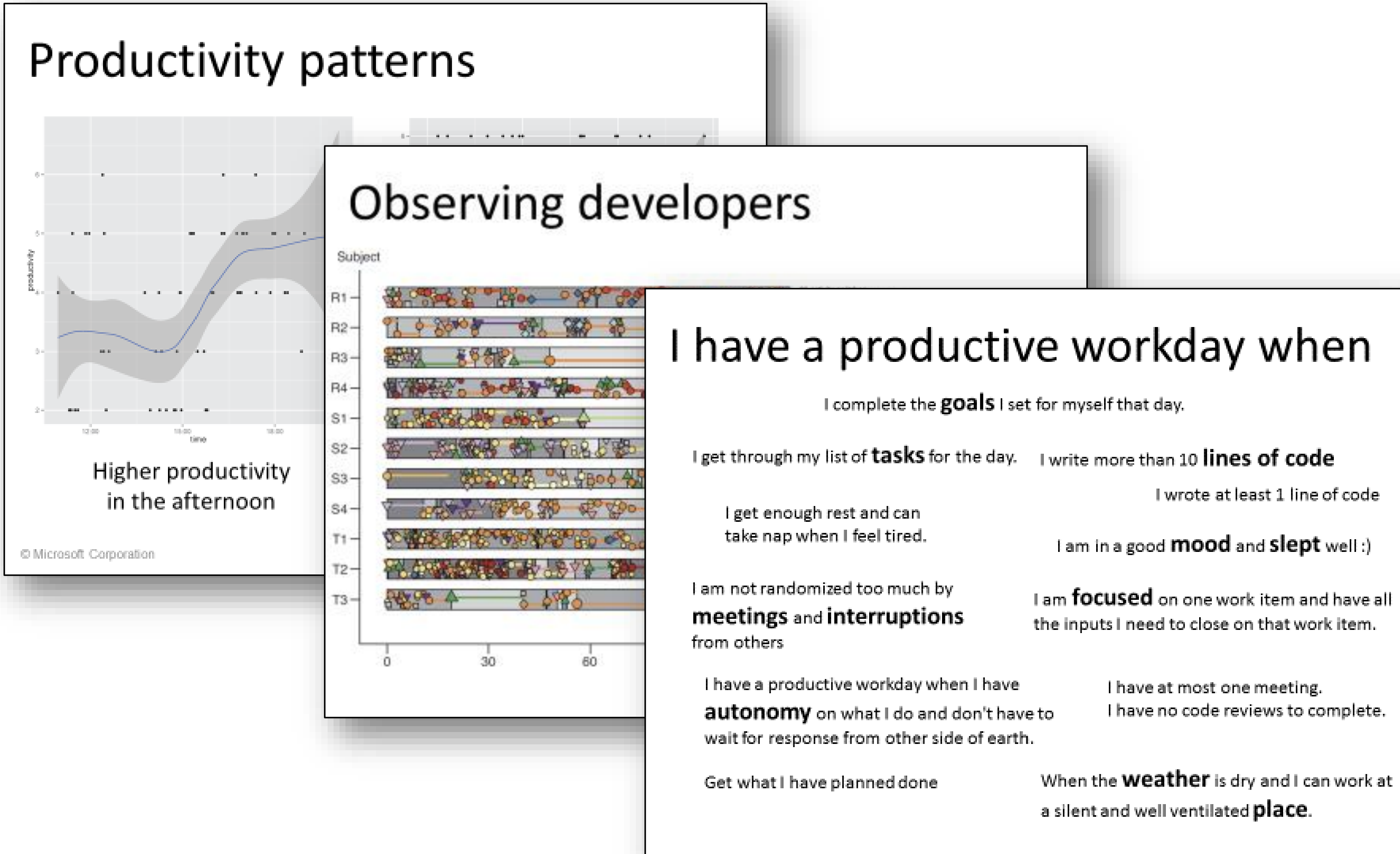
Neg	Pos	NA	**Participant**
			Ratings (total)
			Ratings (discarded)
			Ratings (included in model)
			Ratings (distribution)
3	3	0	**Session Duration (in hours)**
2	1	0	**Percent of Session Before Noon**
			Per Minute
2	2	0	# self-reported tasks
2	4	0	# activity switches
1	1	4	# meetings
2	1	7	# instant messaging switches
1	**7**	0	# keystrokes
2	**7**	0	# mouse clicks
			Percent Activity
4	1	0	Dev. Coding
0	1	9	Dev. Debugger Use
2	1	12	Dev. Code Reviews
2	0	3	Dev. Version Control
5	0	0	Email
2	3	0	Planning
4	3	0	Read / write documents
6	0	5	Planned meeting
3	2	9	Informal meeting
2	0	8	Instant messaging
1	2	0	Work related browsing
5	0	2	Work unrelated browsing
2	1	0	Other
2	1	4	Other RDP
6	2	2	Idle

No single factor provides explanatory power across all participants.

The Number of Keystrokes and the Number of Mouse Clicks (7× each) have more often positive influence.

The percent of activities Email (5×), Planned Meeting (6×), Work Unrelated Browsing (5×), and Idle (6×) have more often negative influence.

No two developers are the same



How are developers similar or different to each other?

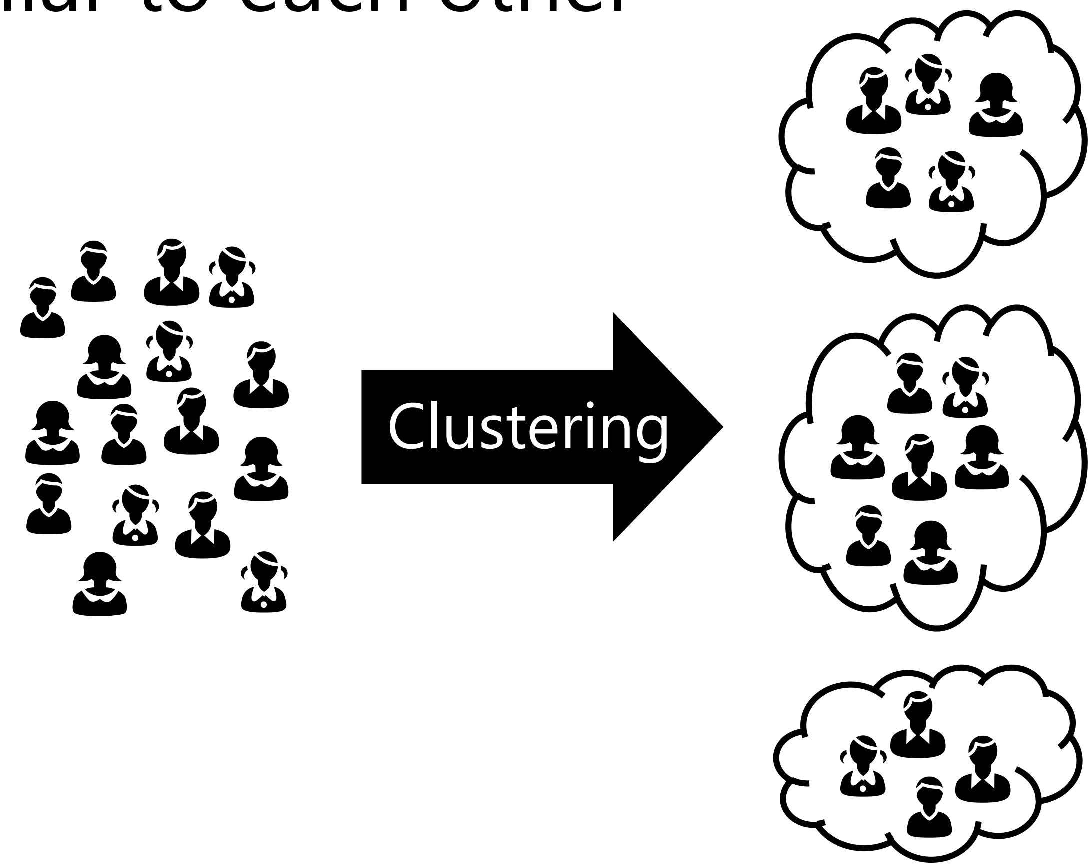
Empirical studies: differentiate between the types of software developers.

Tools: cater the tools for the different types of developers.

Inclusiveness: be inclusive of the many types of developers.

Towards personas of developers

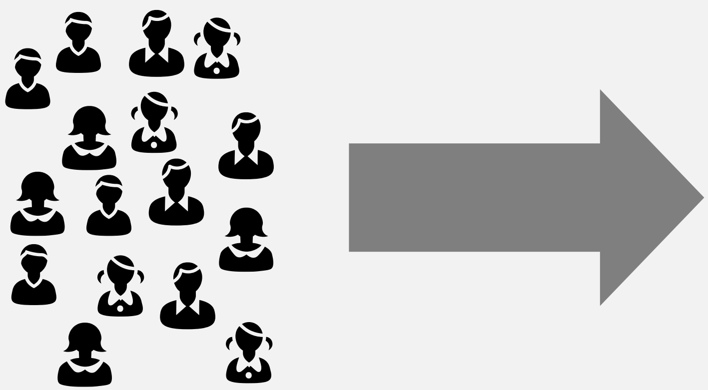
No two developers are the same
but some are similar to each other

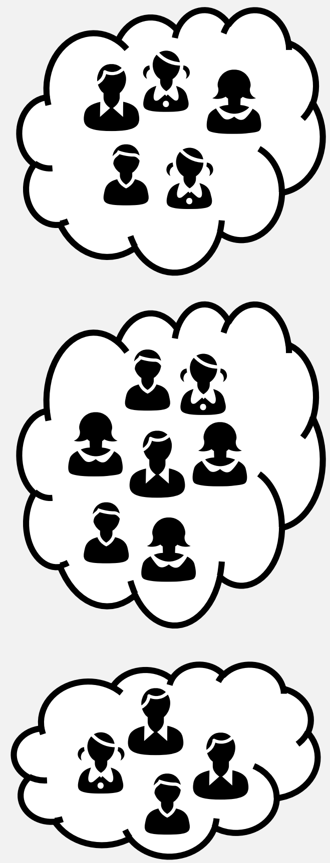


Personas: productivity perceptions

Clustering

413 developers
at Microsoft



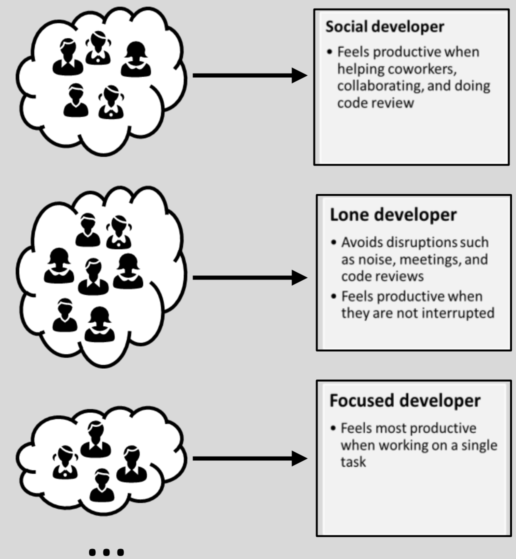


...

6 Clusters

Clusters are based on
Q3 Productivity perceptions

Cluster Descriptions



Descriptions are based on
Q1 Productive workday
Q2 Unproductive workday
Q3 Productivity perceptions
Q4 Productivity measures

A. N. Meyer, T. Zimmermann, T. Fritz: Characterizing Software Developers by Perceptions of Productivity. ESEM 2017

Perceptions of productivity

I feel more productive in the morning than in the afternoon.

I feel more productive in the afternoon than in the morning.

I feel productive on a day with little to no meetings.

I feel productive when I do code reviews.

I feel productive when I write code.

I feel productive when I listen to music.

I feel productive when I take a quick break, e.g., on Facebook.

I feel productive when I help my co-workers.

I feel productive when I read fewer emails than usual.

... (nine additional statements on productivity)

Six types of developers

Based on perceptions of productivity in a survey of 413 developers at Microsoft.

Social developer

- Feels productive when helping coworkers, collaborating, and doing code review

Lone developer

- Avoids disruptions such as noise, meetings, and code reviews
- Feels productive when they are not interrupted

Focused developer

- Feels most productive when working on a single task

Balanced developer

- Less affected by disruptions
- Feels unproductive when unfamiliar with tasks or when the tasks are unclear

Leading developer

- More comfortable with meetings
- Places less importance on coding activities than other developers

Goal-oriented developer

- Feels productive when completing or making progress on tasks
- Feels less productive when multi-tasking, without goals, or stuck
- More positive about meetings

Shared perceptions across the types

HIGHER SCORES

I feel productive when I write code (5 of the 6 types)

I feel productive on a day with little to no meetings (4 of the 6 types)

I feel productive when I am happy (4 of the 6 types)

I feel productive when I have fewer interruptions (4 of the 6 types)

🔨 The time I spent coding (5 of the 6)

🔨 The longest period focused on a task without an interruption

LOWER SCORES

I feel productive when I send more emails than usual (5 of the 6 types)

I feel I had a productive work day when my email inbox is emptier in the evening than in the morning (4 of the 6 types)

I feel productive when I visit social networks or news websites to do a quick break (4 of the 6 types)

If I have many program windows open on my screen, it decreases my perceived productivity

I feel productive on a particular day of the week, e.g., on Wednesdays (5 of the 6)

I feel more productive in the morning than in the afternoon (5 of the 6 types)

I feel less productive after lunch compared to the rest of the day (4 of the 6 types)

Personas: time spent

Acantha, the Autonomist Acumen
Lilo, the Continuous Learner
Isabelle, the Investigator
Cameron, the Communicator
Iman, the Interactive
Ava, the Advisor
Ciara, the Team Coder

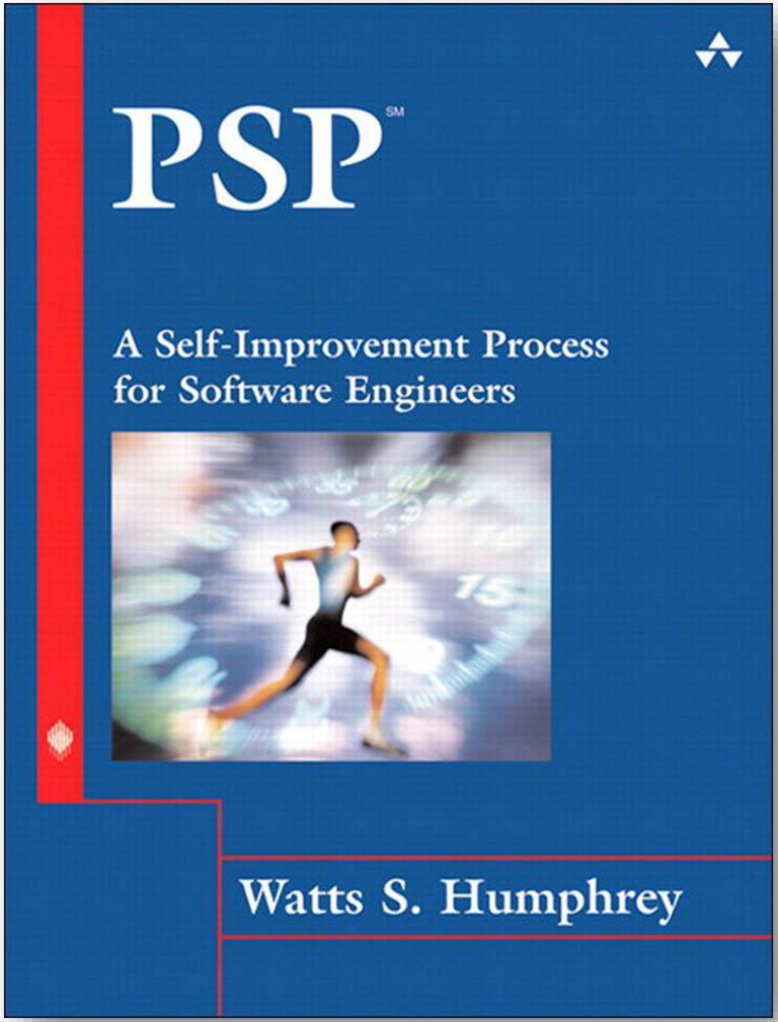
	Learning	Analyzing	Authoring	Co-authoring	Dissemination	Expert Search	Feedback	Info Organization	Info Search	Monitoring	Networking	Service Search
ALL 868 people	0.105 4.4	0.214 9.1	0.102 4.7	0.159 7.5	0.03 1.4	0.059 2.6	0.091 3.8	0.056 2.5	0.072 3.1	0.043 2	0.045 2.1	0.025 1.2
Cluster 1 118 people	0.077 3.5	0.148 6.9	0.384 18.2	0.081 4	0.028 1.3	0.04 1.8	0.067 3.2	0.039 1.8	0.049 2.4	0.034 1.7	0.033 1.6	0.019 0.9
Cluster 2 50 people	0.39 12.8	0.23 8.4	0.061 2	0.051 2.1	0.022 0.7	0.058 2.4	0.031 1.2	0.038 1.6	0.064 2.3	0.016 0.6	0.02 0.8	0.019 0.8
Cluster 3 125 people	0.093 3.6	0.461 18.1	0.058 2.3	0.039 1.6	0.023 1	0.062 2.4	0.073 2.8	0.044 1.7	0.07 2.9	0.03 1.2	0.03 1.3	0.018 0.8
Cluster 4 241 people	0.116 5.2	0.17 7.7	0.08 3.8	0.067 3.5	0.044 2.2	0.078 3.5	0.081 3.7	0.076 3.6	0.098 4.3	0.07 3.3	0.076 3.7	0.043 2.1
Cluster 5 175 people	0.078 3.7	0.196 9.3	0.044 2.2	0.302 14.6	0.026 1.2	0.051 2.5	0.091 4.2	0.046 2.3	0.066 3.2	0.037 1.9	0.039 2	0.022 1.1
Cluster 6 66 people	0.054 1.8	0.177 6.1	0.035 1.4	0.037 1.6	0.037 1.3	0.076 2.6	0.276 8.9	0.114 3.7	0.082 2.7	0.056 2	0.043 1.6	0.015 0.5
Cluster 7 93 people	0.059 2.8	0.127 5.8	0.036 1.7	0.534 24.3	0.014 0.6	0.035 1.6	0.071 3.2	0.027 1.3	0.043 2	0.023 1.1	0.021 1	0.009 0.4

Denae Ford, Thomas Zimmermann, Christian Bird and Nachiappan Nagappan
Characterizing Software Engineering Work with Personas Based on Knowledge Worker Actions
In International Symposium on Empirical Software Engineering and Measurement (ESEM 2017)

Opportunities for research

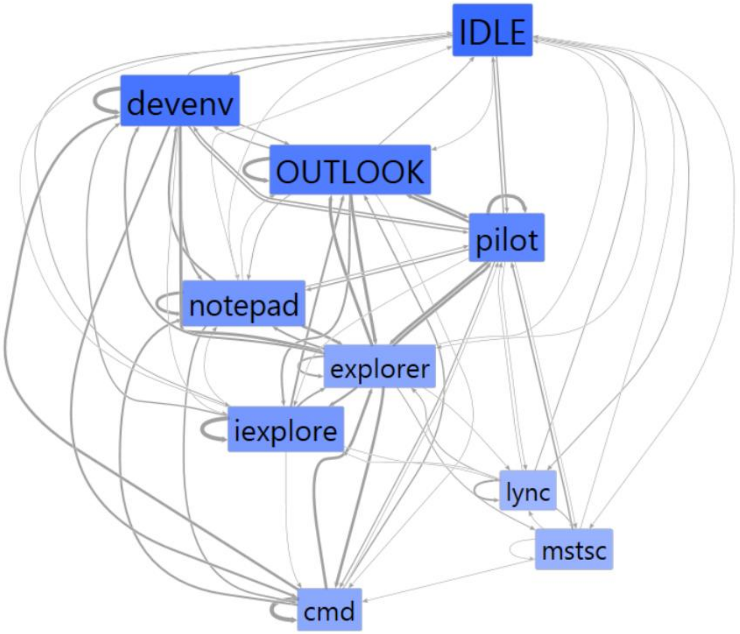
PSP on Steroids



Opportunities for research

PSP on Steroids

“Personal” Process Mining

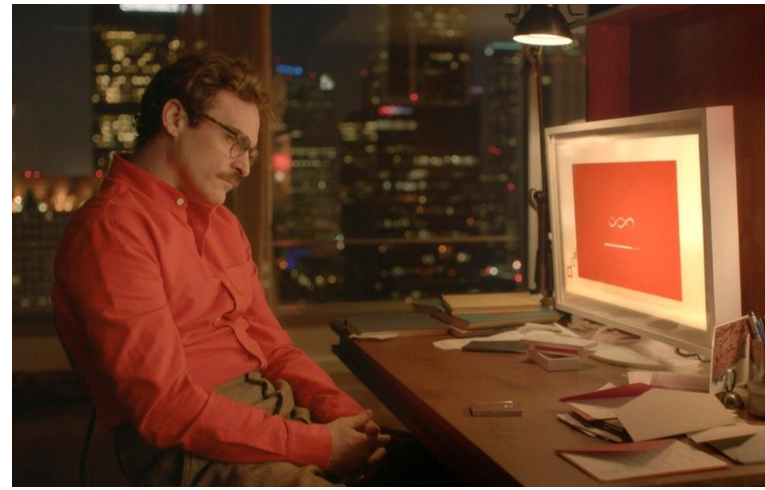


Opportunities for research

PSP on Steroids

"Personal" Process Mining

Personal Tutors



Working Styles of Data Scientists

Insight Provider

Specialists

Platform Builder

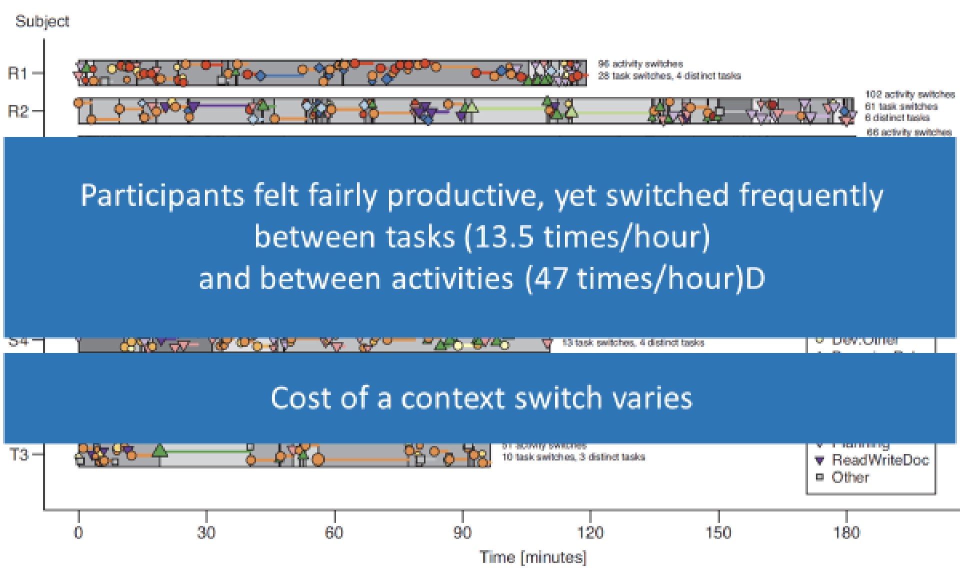
Polymath

Team Leader

Microsoft's Top 10 Questions

Microsoft's Top 10 Questions	Essential	Essential + Worthwhile
How do users typically use my application?	80.0%	99.2%
What parts of a software product are most used and/or loved by customers?	72.0%	98.5%
How effective are the quality gates we run at checkin?	62.4%	96.6%
How can we improve collaboration and sharing between teams?	54.5%	96.4%
What are the best key performance indicators (KPIs) for monitoring services?	53.2%	93.6%
What is the impact of a code change or requirements change to the project and its tests?	52.1%	94.0%
What is the impact of tools on productivity?	50.5%	97.2%
How do I avoid reinventing the wheel by sharing and/or searching for code?	50.0%	90.9%
What are the common patterns of execution in my application?	48.7%	96.6%
How well does test coverage correspond to actual code usage by our customers?	48.7%	92.0%

Observing developers



Participants felt fairly productive, yet switched frequently between tasks (13.5 times/hour) and between activities (47 times/hour)D

Cost of a context switch varies

The stereotypical software developer

100% of the time in front of a computer

YOU HAVEN'T SEEN ANYTHING
UNTIL YOU'VE SEEN
EVERYTHING*

Thank you!