Selecting Research Methods for Studying a Participatory Culture in Software Development

Margaret-Anne (Peggy) Storey

mstorey@uvic.ca

Collaborators: Alexey Zagalsky, Leif Singer, Daniel German

Talk goals…

Participatory development culture and why we should study it

Implications on *research methods*

Opportunities, challenges and *risks* ahead

Towards a *participatory research culture*

Talk goals…

Participatory development culture and
why we should study it

Implications on *research* methods

Opportunities, challenges and risks ahead

Towards a *participatory research culture*

*Three **game changing** trends…*

Projects:

stand alone programs -> ecosystems

Tools:

development -> socially enabled

Developers:

solo coding -> participatory culture

Properties of social tools

Architecture of participation
Transparency
Persistence
Emergence of behaviours


"After just seven years on the net, GitHub now boasts almost 9 million registered users. Each month, about 20 million others visit without registering... GitHub is now among the top 100 most popular sites on earth." [wired.com]

Social tools facilitate a **participatory development culture**, with support for the social **creation** and **sharing** of content, informal **mentorship**, and awareness that **contributions matter** to one other

Storey, M.-A., L. Singer, F. Figueira Filho, B. Cleary and A. Zagalsky, **The (R)evolutionary Role of Social Media in Software Engineering**, ICSE 2014 Future of Software Engineering.

*“The numbers are staggering. A project (Homebrew) that is just 5 years old has attracted **20.5k** --- the size of a small city! --- people to contribute to it. Ruby on Rails has been collaboratively developed by a community of **15k** people and still works! To compare these numbers with other software engineering projects is futile: most projects, even ones with a very long lifeline are very small in comparison.”*

[Georgious Gougios Blog: The Triumph of Online Collaboration]

Three **game changing** trends…

Projects:

stand alone programs -> ecosystems

Tools:

c ially enabled

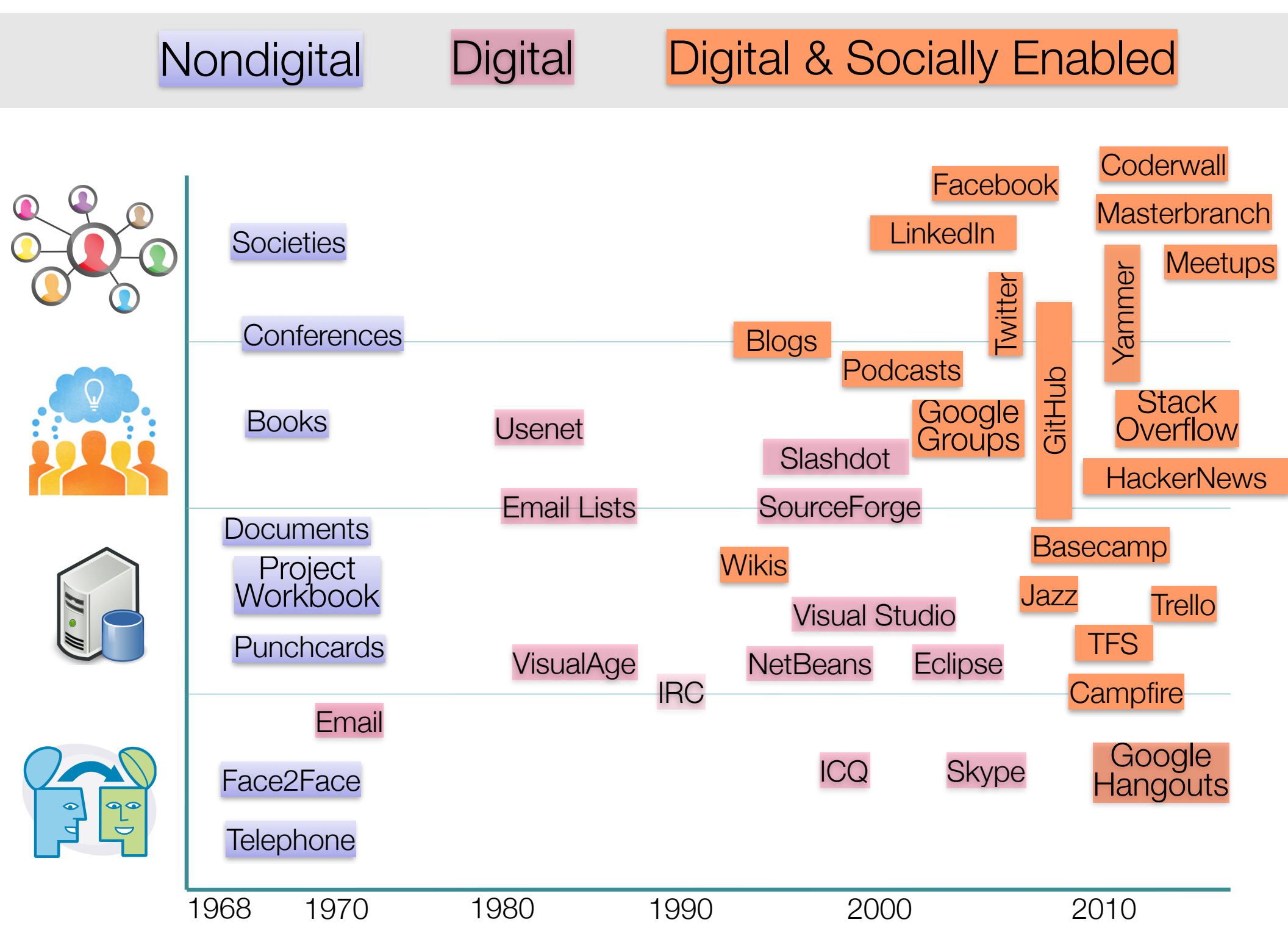
But why research this paradigm shift?

Developers:

solo coding -> participatory culture

Digital ecosystems:

Tools** shape and are shaped by participatory **communities** of practice facilitating the social production of **content



Storey, M.-A., L. Singer, F. Figueira Filho, B. Cleary and A. Zagalsky, The (R)evolutionary Role of Social Media in Software Engineering, ICSE 2014 Future of Software Engineering.

Survey: Which tools are used by developers and why?

Nondigital:

Face-to-face,
books,
magazines,
…

Digital:

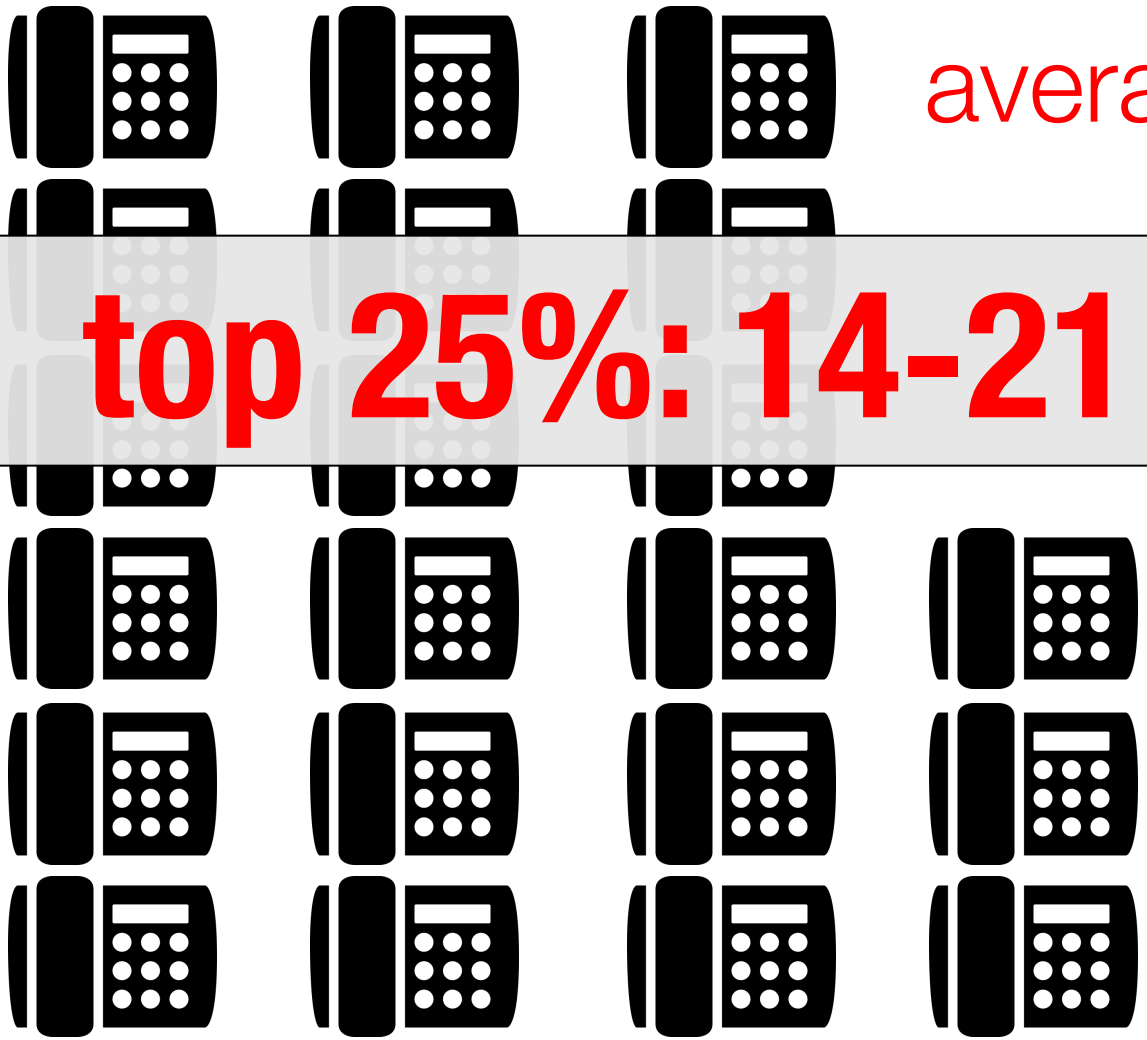
Web Search,
Public Chat,
Private Chat,
Discussion Groups,
…

Digital & social:

Feeds and Blogs,
Tagging,
Q&A,
SNS,
Code hosting,
…

Survey: http://leif.me/devsurvey/?source=icse

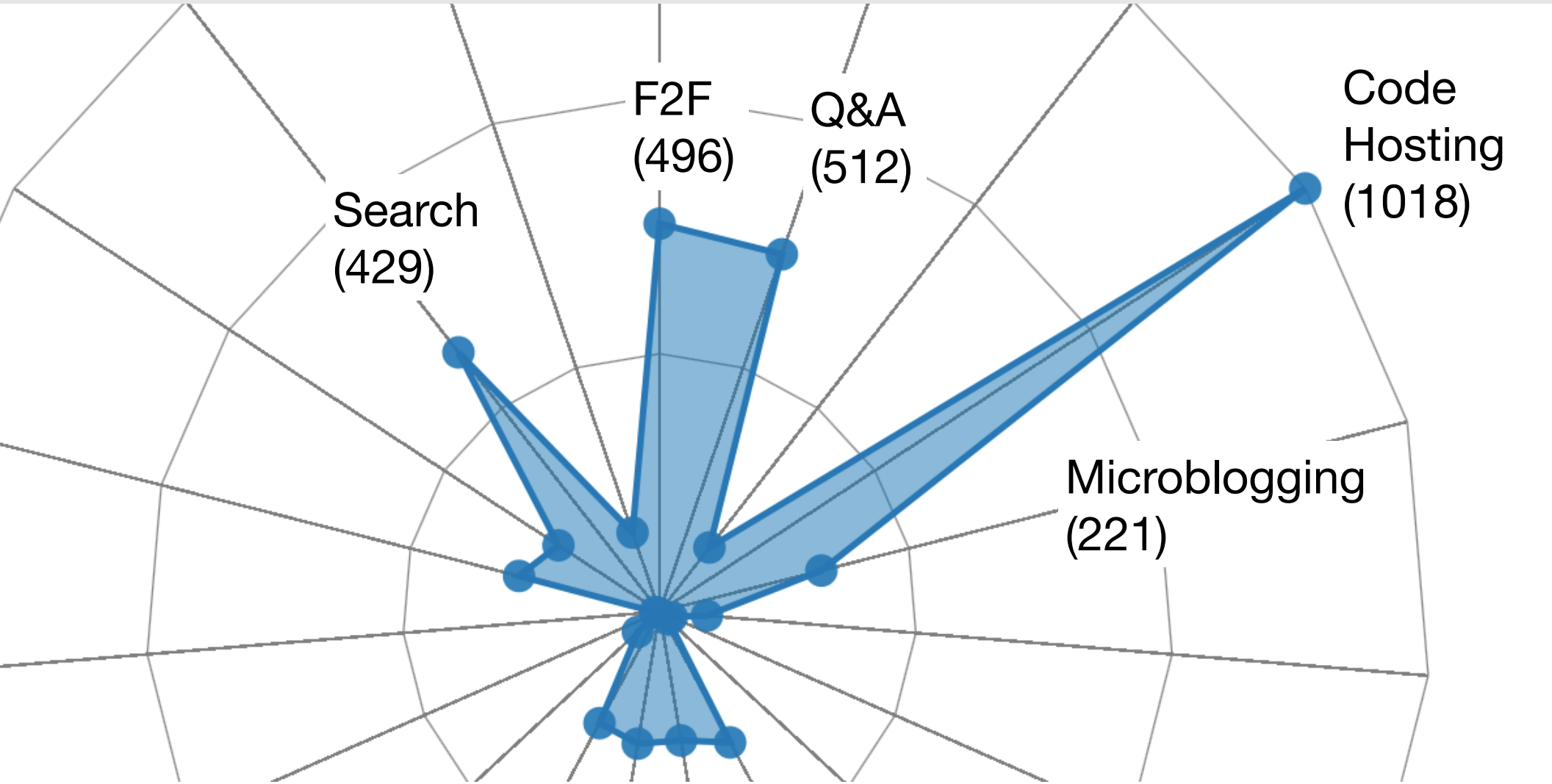
Selected Findings



average: 12 channels

top 25%: 14-21 channels

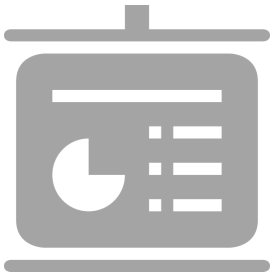
Most Important Channels



Interactive visualization: http://fose2014.thechiselgroup.org

Opportunities & Challenges

Social network to curate/find/learn relevant information but impact on productivity?



Community authored resources, but what about information fragmentation? What about quality?

Participatory culture but when does social become anti-social?

Rich social media ecosystems for developers but what about vendor lock-in?

Storey, M.-A., L. Singer, F. Figueira Filho, B. Cleary and A. Zagalsky, **The (R)evolutionary Role of Social Media in Software Engineering**, ICSE 2014 Future of Software Engineering.

Talk goals…

Participatory development culture, what we know about it and *why* we should study it

Implications on *research methods*

Opportunities, challenges and risks ahead

Towards a *participatory research culture*

Mining of participant trace data to support software analytics

Not about qualitative versus quantitative!

Generating participant research data to understand social and human aspects of development

Trace data

"Outcroppings of past human behaviour"

[McGrath]

Can be **mined** to improve:

quality of the software

experience of the users

developer productivity

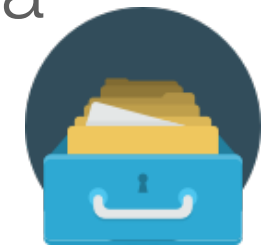
Dongmei Zhang & Tao Xie, http://research.microsoft.com/en-us/groups/sa/softwareanalyticsinpractice_minitutorial_icse2012.pdf

Prolific trace data sources…

Program data: runtime traces, program logs, system events, failure logs, performance logs, continuous deployment …

User data: usage logs, user surveys, user forums, A/B testing, twitter and blogs, …

Development data: source code versions, bug data, check-in information, test cases and results, communication between developers, social media



Techniques:

Association rules and frequent patterns

Classification

Clustering

Text mining/Natural Language processing

Searching and mining

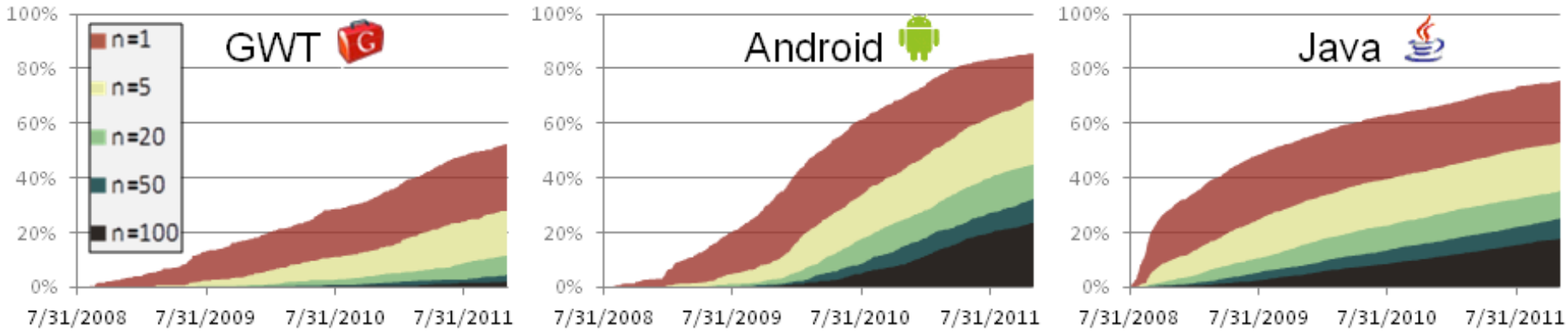
Qualitative analysis

See papers from the Mining Software Repositories Conference!

Stackoverflow as Crowd Documentation?

Coverage of **API documentation:** 77% of the Java API classes & 87% of Android API classes

Speed of coverage:



C. Parnin, C. Treude, L. Grammel and M.-A. Storey. **Crowd Documentation: Exploring the Coverage and the Dynamics of API Discussions on Stack Overflow".** at http://blog.ninlabs.com/2012/05/crowd-documentation/

On using trace data

Strengths: low interference, low reactivity, records made by the participants

Weaknesses: loosely linked or not linked to the concepts you measure, costly to analyze, treats humans as “rational” animals, trace data is only part of the picture

Researcher generated data

Surveys, interviews, observations

Quantitative, qualitative data

Hypothesis testing, theory building

some examples…

Twitter study

Qualitative data

Exploratory Survey: get a feel for attitudes

emailed 1,160 active GitHub users – 271 responded

Interviews: deeper insights about value, challenges

27 – from exploratory survey

Analysis: Grounded Theory

codes, categories, themes, memos, …

Validation Survey: do developers agree?

emailed 10,000 active GitHub users – 1,413 responded

Quantitative data

Keeping up at speed of light

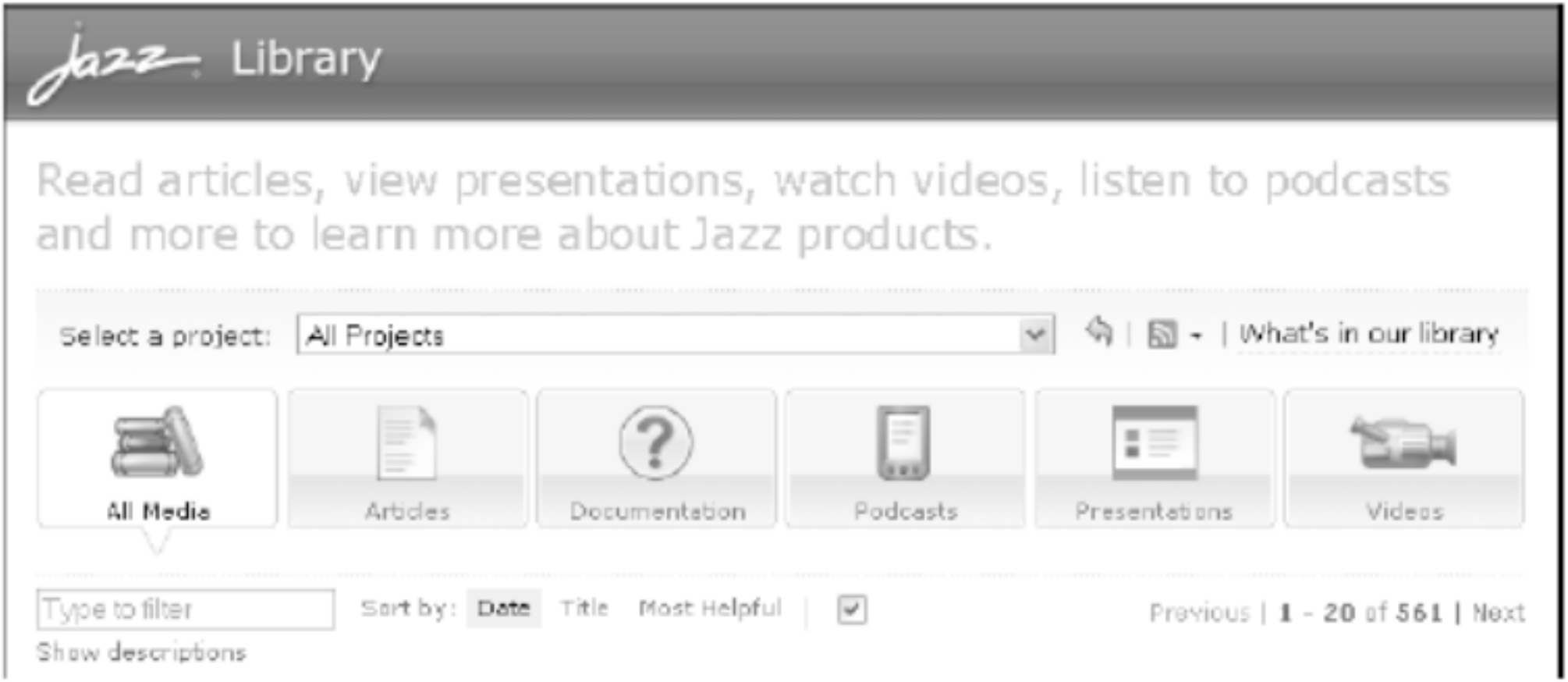
Findings:

- Awareness
- Learning
- Relationships
- Strategies
- **Why non-adoption**

"It was evolving way faster than I was able to keep up with it. And the only way to keep up was to follow some Node.js people on Twitter."

Leif Singer, Fernando Figueira Filho, Margaret-Anne Storey. *Software Engineering at the Speed of Light: How Developers Stay Current Using Twitter* ICSE 2014.

Ethnography at field site…



C. Treude and M.-A. Storey. Effective Communication of Software Development Knowledge Through Community Portals. ESEC/FSE '11.

Theory building: dimensions that characterize software artifacts in a community portal:

content
audience
trigger
collaboration
review
feedback
fanfare
time
sensitivity

C. Treude and M.-A. Storey. Effective Communication of Software Development Knowledge Through Community Portals. ESEC/FSE '11.

Limitations with researcher generated data…

Self reporting

Respondent and researcher bias

Ambiguity in instruments and collected data

Which methods should be used?

All methods are inherently flawed

Generalizability of the evidence over the populations of actors

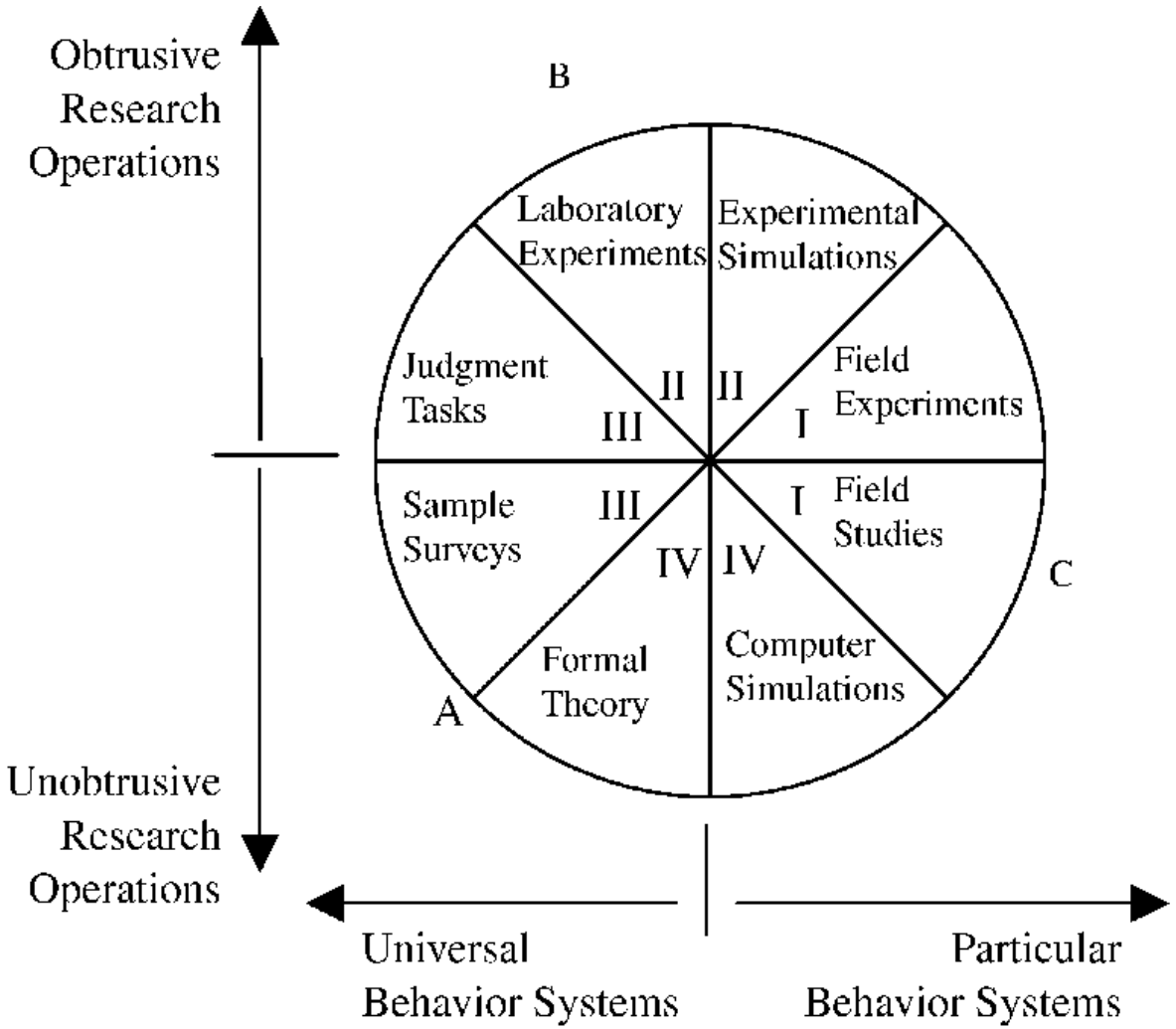
Precision of measurement of the behaviours being studied

Realism of the situation or context where the evidence is gathered

Although goal is to maximize the above three things – we cannot!

E. McGrath. Methodology matters: Doing research in the behavioral and social sciences.

Strategy Circumflex (McGrath)



Source: McGrath(1981)

I. Settings in natural systems

II. Contrived and created settings

III. Behavior not setting dependent

IV. No observations of behavior required

A. Point of maximum concern with generality over actors

B. Point of maximum concern with precision of measurement behavior

C. Point of maximum concern with system character of context

Choice of method depends on the research question being asked (exploratory, confirmatory, relationship) as well as the researcher's philosophical perspective

S Easterbrook, J Singer, MA Storey, D Damian.
Selecting empirical methods for software engineering research.

Power of Mixed Methods

Sequential explanatory strategy: e.g., quantitative analysis of trace data followed by qualitative analysis of interview data *(latter helps explain the former)*

Sequential exploratory strategy: e.g., analysis of qualitative data from surveys followed by analysis of quantitative trace data (*for testing emerging theory, explain early exploratory findings)*

Concurrent triangulation strategy: different methods used concurrently, improve validity

J. W. Creswell, Research Design: Qualitative, Quantitative, and Mixed Methods Approaches. Sage, 2013.

Stackoverflow study:
Mined trace data followed by interviews using visualization of the mined data…

http://latest-print.crowd-documentation.appspot.com/?api=android

Talk goals…

Participatory ***development*** *culture*, what we know about it and *why* we should study it

Implications on *research* **methods**

Opportunities, challenges and *risks* ahead

Towards a *participatory* ***research*** *culture*

Research opportunities

Makes **visible** the invisible

New **populations** to explore

Experimentation in **real time** (with **large N**)

May not need to **sample**

Longitudinal data

New kinds of data, **novel questions**

Community curated **resources** (e.g., GHTorrent)

Access to **participants**

Research challenges

Big data! (not just of trace data!)

Rapid pace of **change**

Studying **unstable** objects [Rogers]

Poor boundaries of study contexts

Need for new **tools**

Research risks

Opportunistic research studies!

Power over data resources

Data representativeness

Inaccuracies in profiles, exaggerations, skewed opinions

Perils from using GitHub data:

A repository is not necessarily a (development) project

Most projects are inactive or have few commits

Most projects are for personal use only

Only 10% of projects use pull requests

History can be rewritten on GitHub

A lot happens outside of GitHub

The Promises and Perils of Mining GitHub, Eirini Kalliamvakou et al, MSR 20134.

Risks for participants

Developers are humans!

- Private, public, **blurred spaces**

Intent of the research? [Tufecki, ACM CSCW 2015 Keynote]

- **Surveillance** at the level of the individual
- Use of **opaque algorithms** (biases unclear)
- **Social engineering**

Ethics!

During recruitment, data collection, reporting…

- Informed **consent**?
- Reporting **raw quotes** from “public” data?

Talk goals…

Participatory development culture, what we know about it and *why* we should study it

Implications on *research* methods

Opportunities, challenges and risks ahead

Towards a *participatory* ***research*** *culture*

Social media as a participatory platform for software engineering research

"Good to see a survey on this topic. It is wonderful to be part of a global developer movement and have the entire world of developers helping each other."
[Developer survey respondent]

Opportunities for researchers…

Collaboration and **sharing** of data and research instruments (mixed methods!)

Faster **dissemination** and **feedback** (from researchers and study informants)

Transparency of approach (peer auditing, replication)

Keeping up with research at the speed of light!

Researcher challenges….

Fragmentation of research findings

Information/data overload and **judging quality**

Social media **literacy**

Intimidation, **barriers**

Need to build or foster such a **community**

Researcher risks….

Vendor **lock-in**

Blurred boundaries between researchers and participants

Researcher identity and potential for **“mudslinging”** [Filipinova, CSCW 2015 paper]

TL;DR

Summary

Participatory development culture and *why* we should study it

Implications on *research methods*

Opportunities, challenges and *risks* ahead

Towards a *participatory research culture*

Concluding remarks…

Mixed methods help **triangulate** findings but also address different stages of theory development

Industry is moving fast, we have to **keep up**, adapt

Focus on **important questions**, give back, be **ethical**

Digital ecosystems:

?

Tools** shape and are shaped by participatory **communities** of practice facilitating the social production of **content

mstorey@uvic.ca

References

The Evolution of the Social Programmer
Mining Software Repositories (MSR) 2012 Keynote
http://www.slideshare.net/mastorey/msr-2012-keynote-storey-slideshare

The (R)evolution of Social Media in Software Engineering
ICSE Future of Software Engineering 2014
http://www.slideshare.net/mastorey/icse2014-fose-social-media
http://chiselgroup.files.wordpress.com/2014/01/fose14main-storey-submitted.pdf

H. Jenkins, K. Clinton, R. Purushotma, A. J. Robison, and M. Weigel. Confronting the challenges of participatory culture: Media education for the 21st century, 2006. http://digitallearning.macfound.org/atf/cf/%7B7E45C7E0-A3E0-4B89-AC9C-E807E1B0AE4E%7D/JENKINS_WHITE_PAPER.PDF

M. M. Wasko and S. Faraj. “It is what one does": why people participate and help others in electronic communities of practice. The Journal of Strategic Inform. Systems, 9(2-3):155-173, 2000. http://www.sciencedirect.com/science/article/pii/S0963868700000457

S Easterbrook, J Singer, MA Storey, D Damian. Selecting empirical methods for software engineering research. Guide to advanced empirical software engineering, 2008
http://link.springer.com/chapter/10.1007/978-1-84800-044-5_11

E. Mcgrath. Methodology matters: Doing research in the behavioral and social sciences. In *Readings in Human-Computer Interaction: Toward the Year 2000 (2nd ed.)* 1995 http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.361.1234

References (cont'd.)

A. Tashakkori, C. Teddlie. Mixed methodology: Combining qualitative and quantitative approaches. Vol. 46. Sage, 1998. http://books.google.ca/books?hl=en&lr=&id=qtW04-pRJZ0C&oi=fnd&pg=PR9&dq=Tashakkori&ots=6f6psz78wM&sig=lu5MXcC2HH1ld6EfiEz7bPnRWaE

K. Charmaz. Constructing grounded theory. Sage, 2014. http://www.sagepub.com/booksProdDesc.nav?prodId=Book235960

C.B. Seaman. Using Qualitative Methods in Empirical Studies of Software Engineering http://ccsl.ime.usp.br/files/QualMethods%20minicourse%20USP-1.pdf

P. Runeson, et al. Case study research in software engineering: Guidelines and examples. John Wiley & Sons, 2012. http://www.amazon.com/Case-Study-Research-Software-Engineering/dp/1118104358