The software developer as the ***knowledge worker*** *of tomorrow*

Margaret-Anne (Peggy) Storey

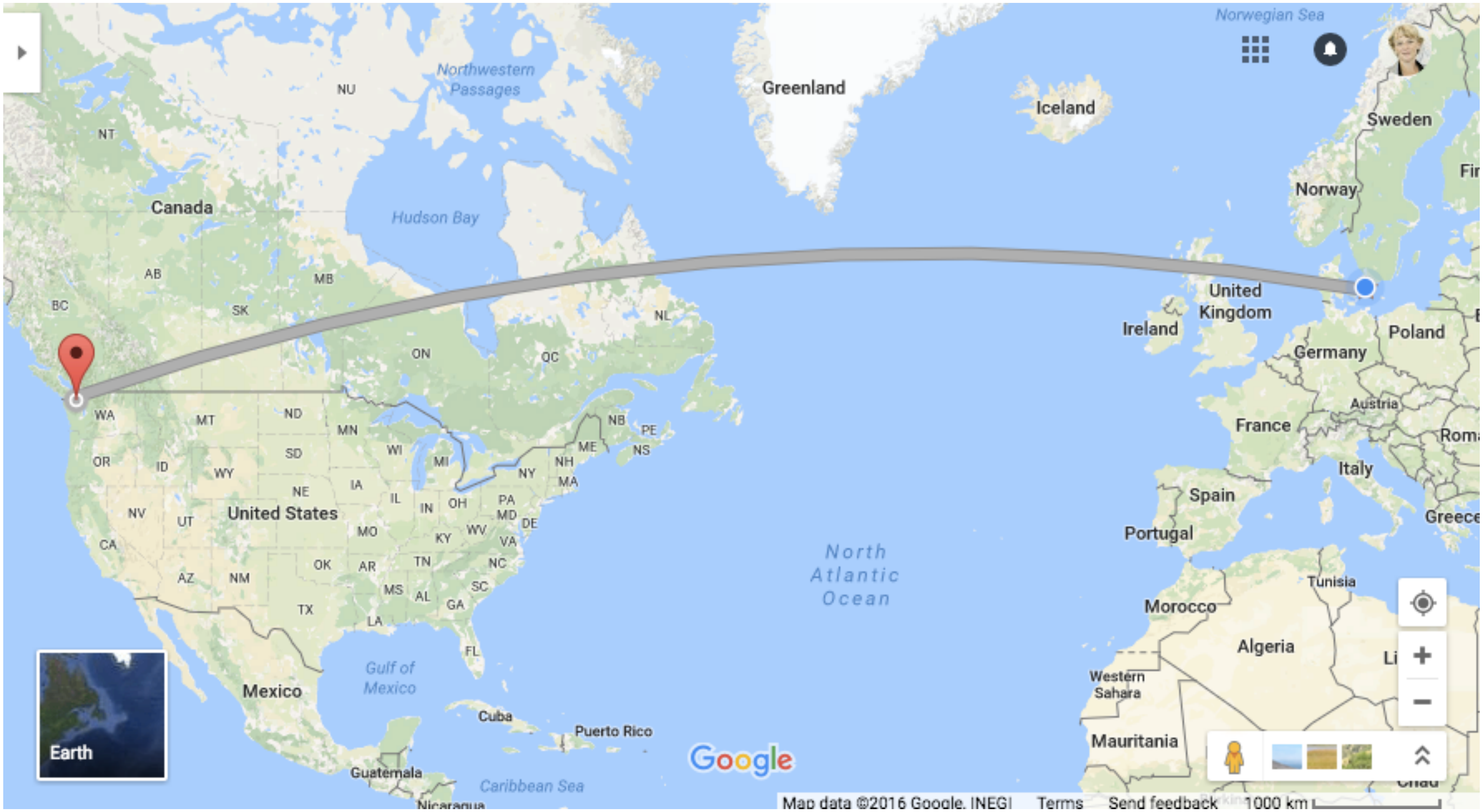
University of Victoria, Canada

Lise Meitner Guest Professor, Lund University 2016-2018

ELLIIT Workshop, Lund, Sweden— April 26, 2017

@margaretstorey

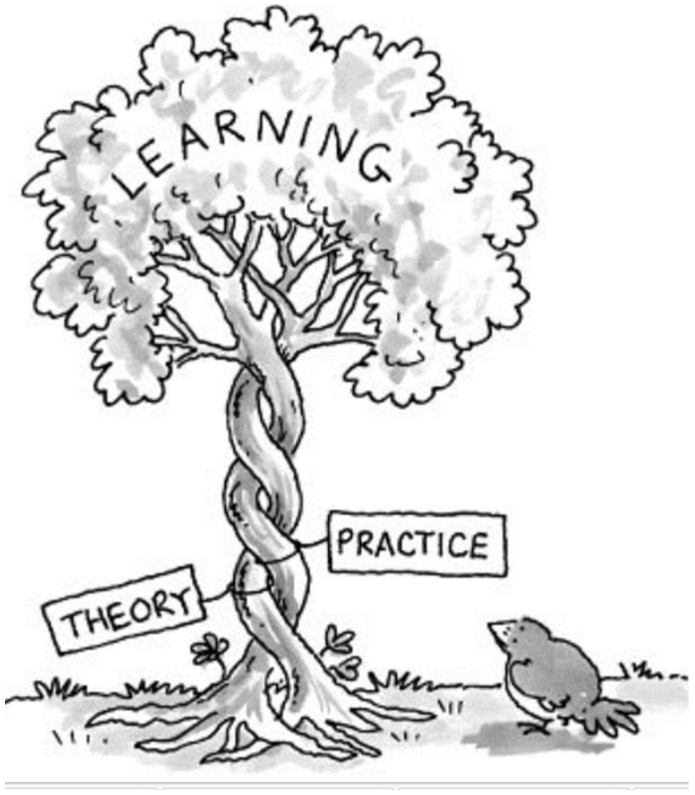
chisel


Greenland
Iceland
Sweden
Norway
Canada
Hudson Bay
Northwestern Passages
United Kingdom
Ireland
Poland
Germany
France
Austria
Italy
Spain
Portugal
Greece
Morocco
Algeria
Tunisia
Western Sahara
Mauritania
North Atlantic Ocean
United States
Mexico
Gulf of Mexico
Cuba
Puerto Rico
Guatemala
Caribbean Sea
Nicaragua
Earth
Google
Map data ©2016 Google, INEGI
Terms
Send feedback
1000 km

Research goals

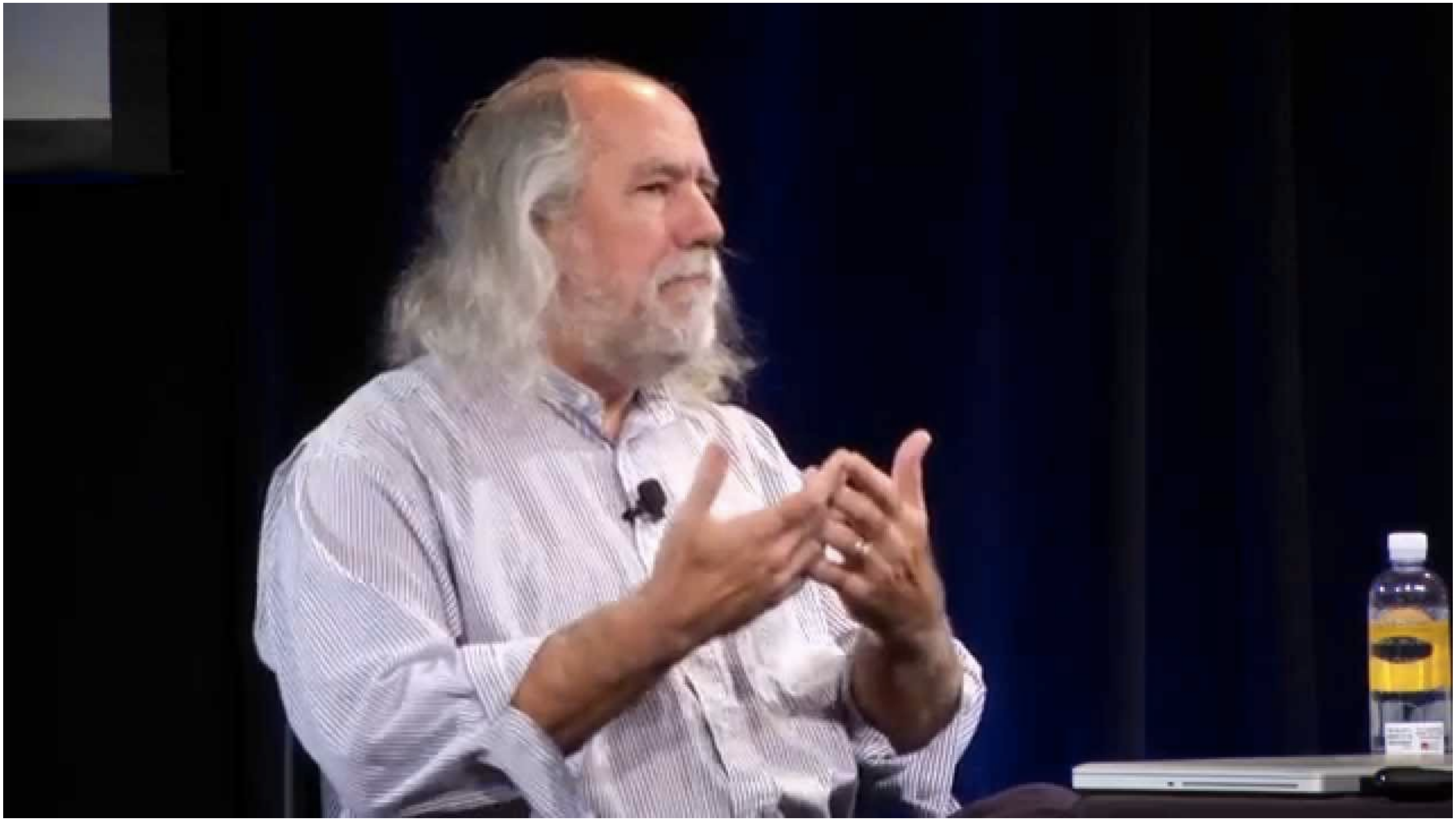
Gain an understanding of how developers collaborate to create complex systems

Improve development and communication tools developers rely on



"software is eating the world"

--- Marc Andreesen



"software brings hardware to life"
---Grady Booch

This Car Runs on Code

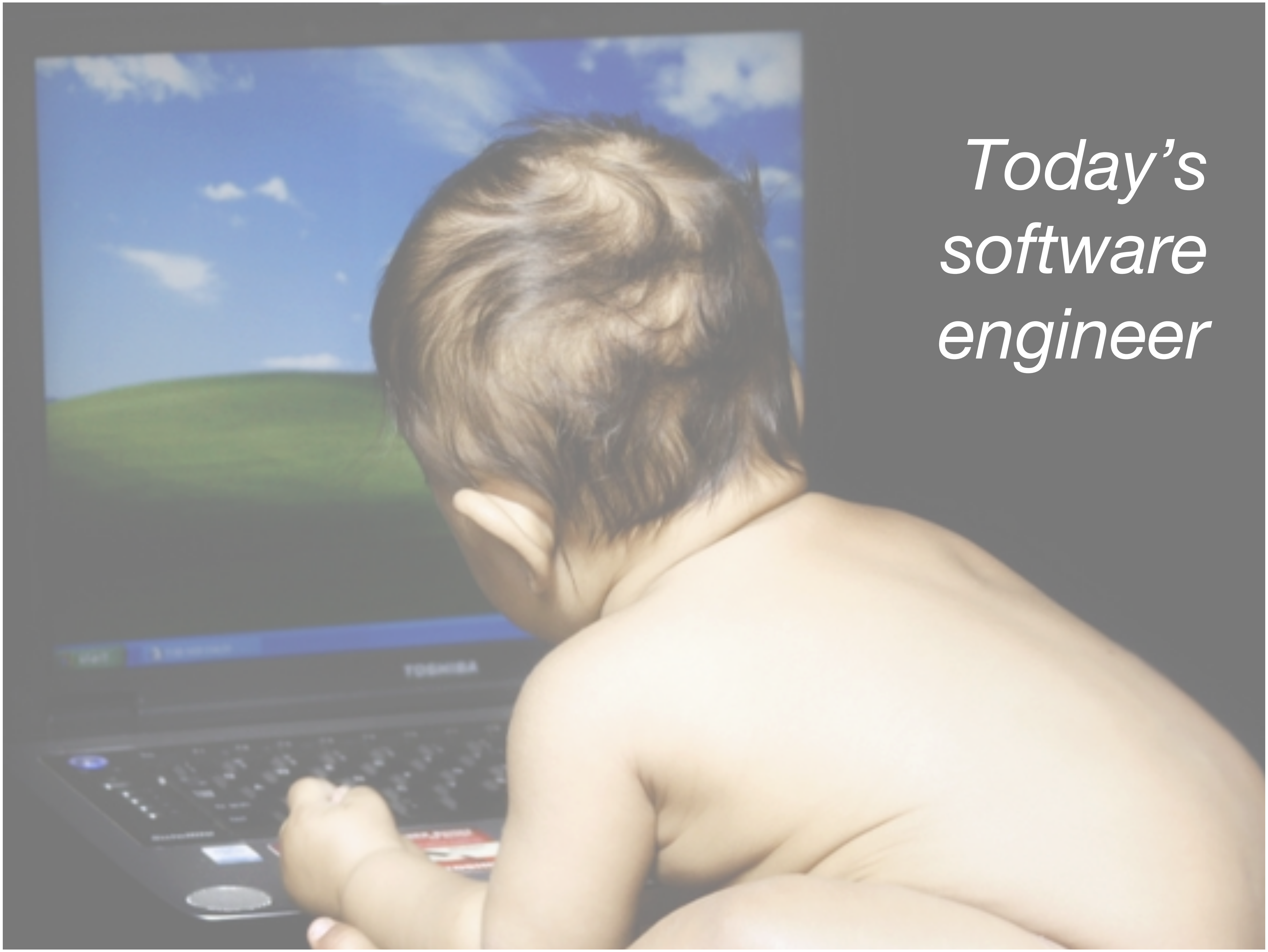
100-200 million lines of code ~100 processors
70% effort spent on software

> 50% warranty costs due to software

“I know how this was done because I did it”

“I need complete understanding”

Peter Norvig, Coders at Work

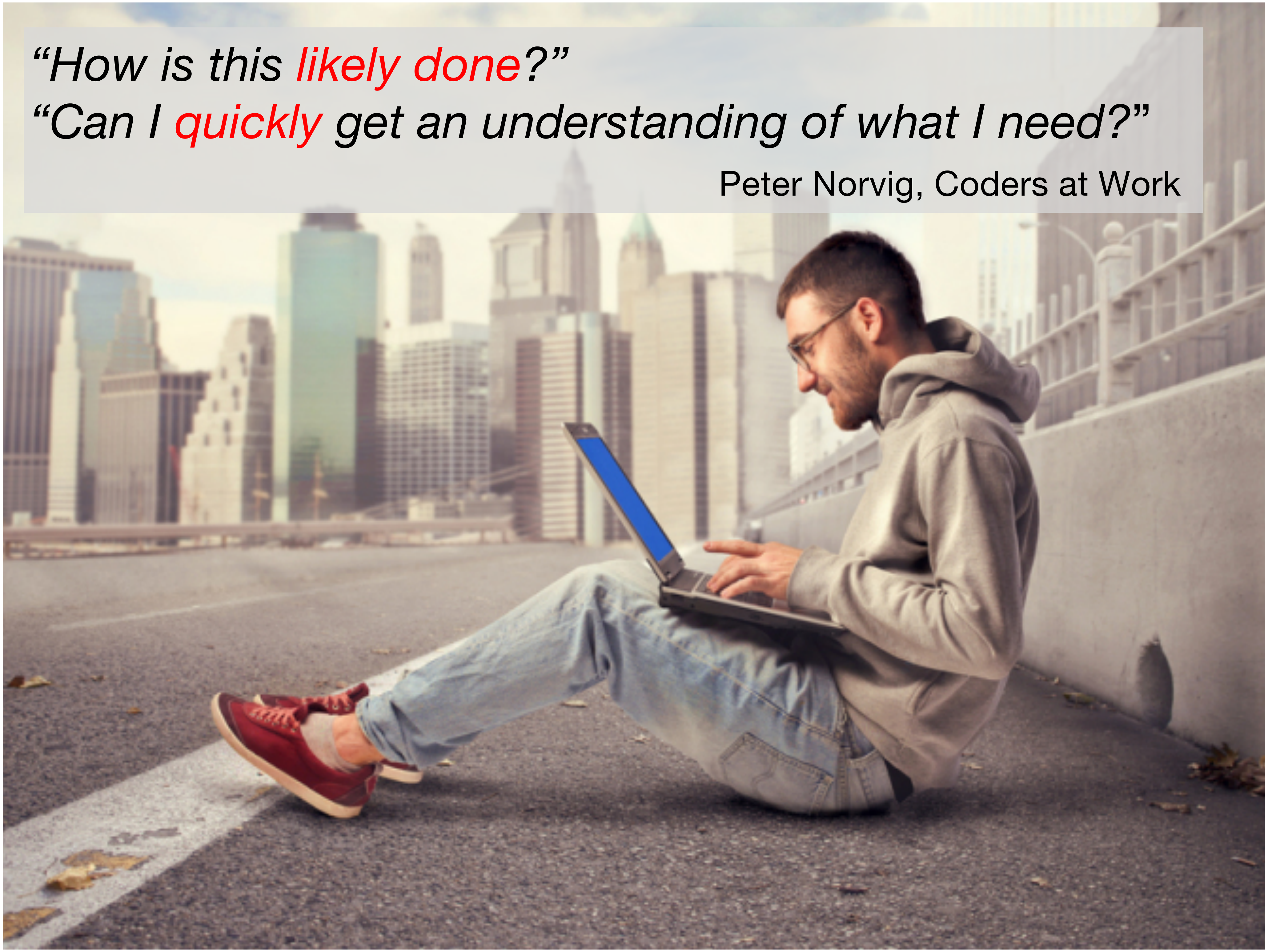


Today's software engineer

“How is this likely done?”

“Can I quickly get an understanding of what I need?”

Peter Norvig, Coders at Work



Engineering at Google

30,000 developers in 40 offices

2 Billion LOC

1 Billion files, 9 million source files

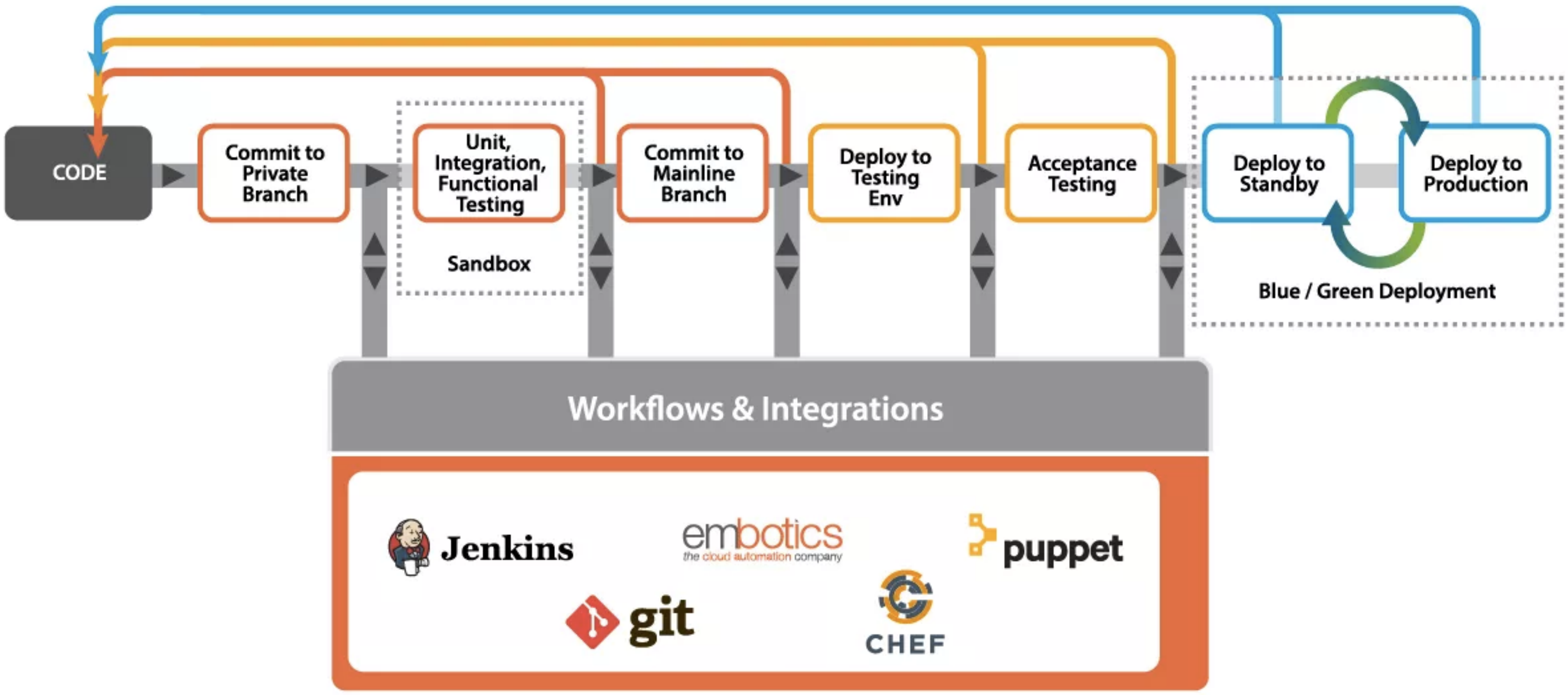
35 million commits, 40,000 commits per day

One monolithic repository!

*“Scaling to **1000s of developers** — **automation** is required!”* [Jacek Czerwonka]

Modern Developer's Toolbox

IDE
Source control
Continuous integration, deployment
Code review
Testing frameworks
Issue tracking
Experimentation platforms
Project management
Communication tools -> Social media…



http://talkingtechwithshd.com/devops-modern-software-development-process/

Programming as a "theory building process"

Software is built using tacit knowledge captured in developers' head(s), and from externalized knowledge embodied in their development tools, channels, and project artifacts [Naur, 1985].

Discussion topics

Theories of media and how media shape software development

How social/communication channels have evolved over time in software engineering

From software development to knowledge work

Discussion topics

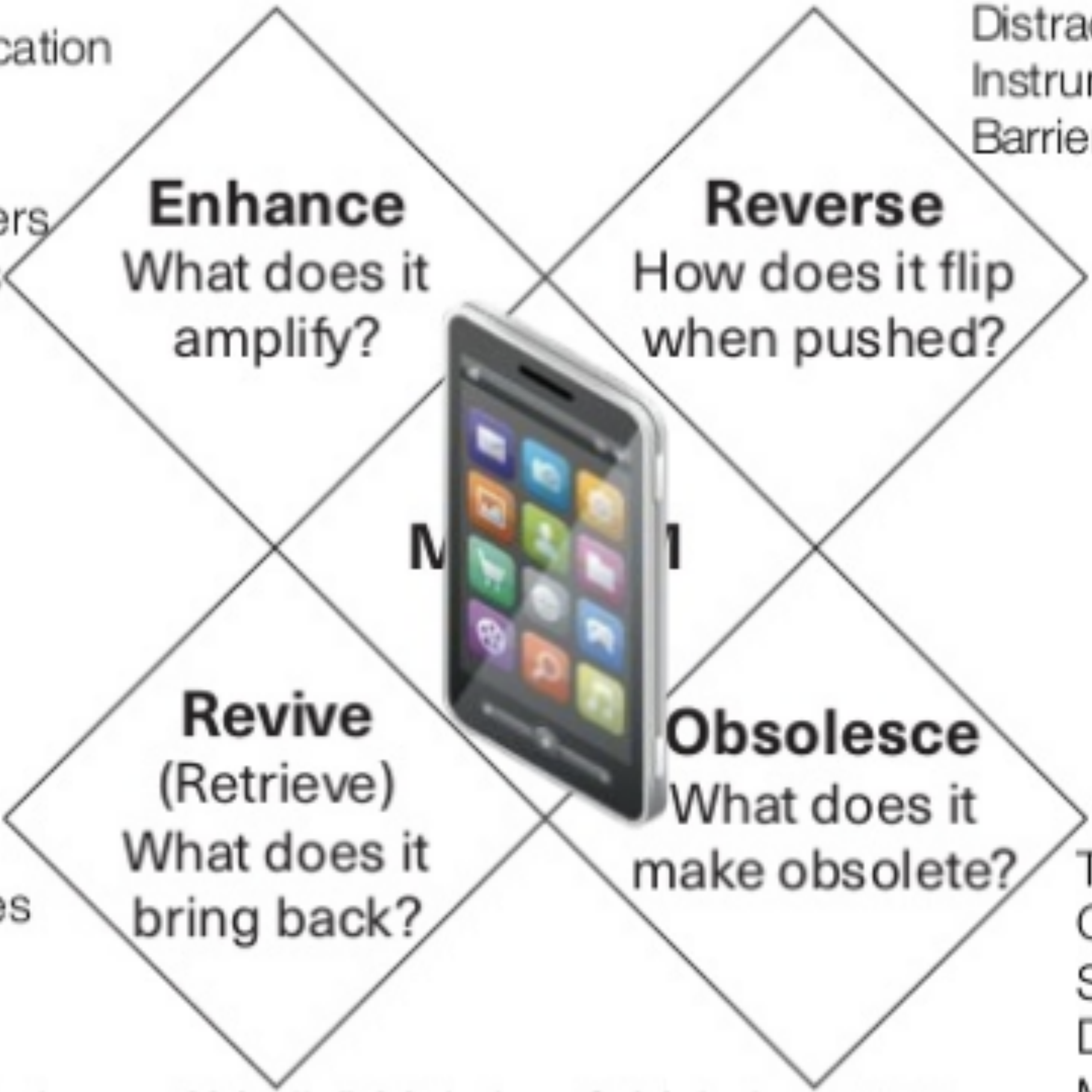
- Theories of media and how media shape software development
- How social/communication channels have evolved over time in software engineering
- From software development to knowledge work

The McLuhan Tetrad

The smart phone

Immediate gratification
App for that
Connectivity
Immediate answers
(Weak) social ties

Avoid phone calls
Distraction
Instrument of doom
Barrier to relationships



Rotary ring tones
Low-fi games

Telephone
Camera
Scanner
Desktop software
MP3 Player

Adapted from “The Laws of Media”, McLuhan & McLuhan, 1988

“If we understand the revolutionary transformations caused by new media, we can anticipate and control them; but if we continue in our self-induced subliminal trance, we will be their slaves.”
[Marshall McLuhan, 1974]

Communities of practice emerge when people share a passion and learn together [Wenger]

Social Media and *Participatory Cultures* *[Jenkins]*

Low barriers to artistic expression and engagement

Strong support for **sharing** one's creations

Informal **mentorship** for novices

Members believe their **contributions matter**

Members care about **social connections** and what others think about their creations

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.

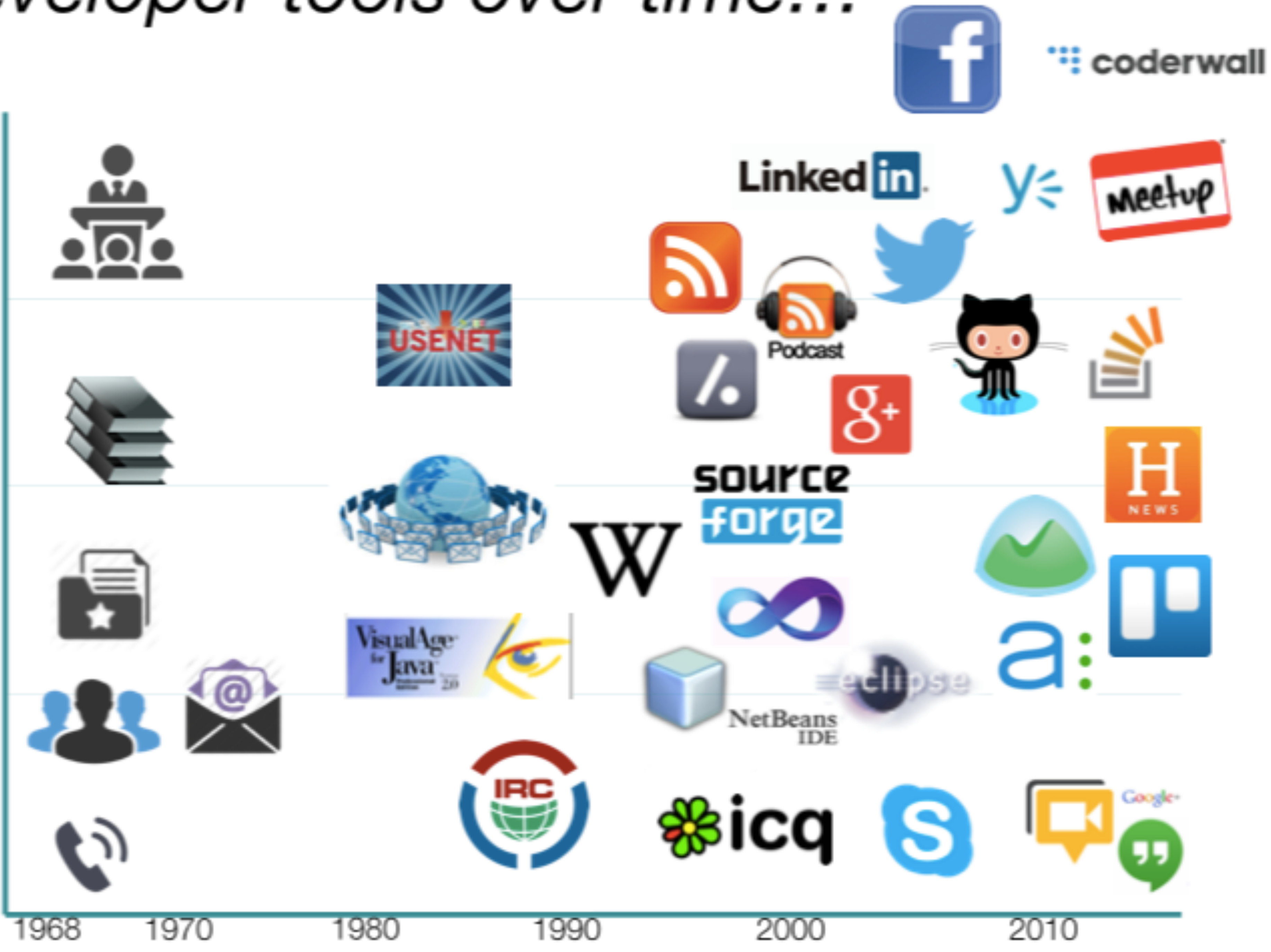
The Participatory Culture in Software Engineering is not new

THE CATHEDRAL
AND THE
BAZAAR
ERIC S. RAYMOND
FSF
FREE AS IN FREEDOM
NO BETA SHALL PASS BEFORE IT'S TIME!
Splendid isolation
RELEASE EARLY
RELEASE OFTEN
EGOLESSS
HARNESS BRAIN POWER
IT'S A CULTURE IDEOLOGY METHODOLOGY
OPEN
LISTEN
PROBLEM FINDING
Too many programmers work on projects they neither NEED nor WANT
REUSE WHEN POSSIBLE
EVERY GOOD PIECE OF WORK STARTS BY SCRATCHING A PERSONAL ITCH
WITH ENOUGH EYEBALLS ALL BUGS ARE SHALLOW
@digitalforsythe #oped12

Discussion topics

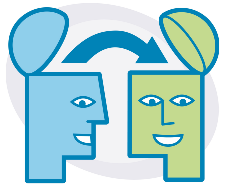
- Theories of media and how media shape software development
- How social/communication channels have evolved over time in software engineering
- From software development to knowledge work

Developer tools over time...



Understanding affordances of channels for communicating knowledge [Wasko et al.]

Communicating knowledge that is embedded in people (F2F, email, chat…)

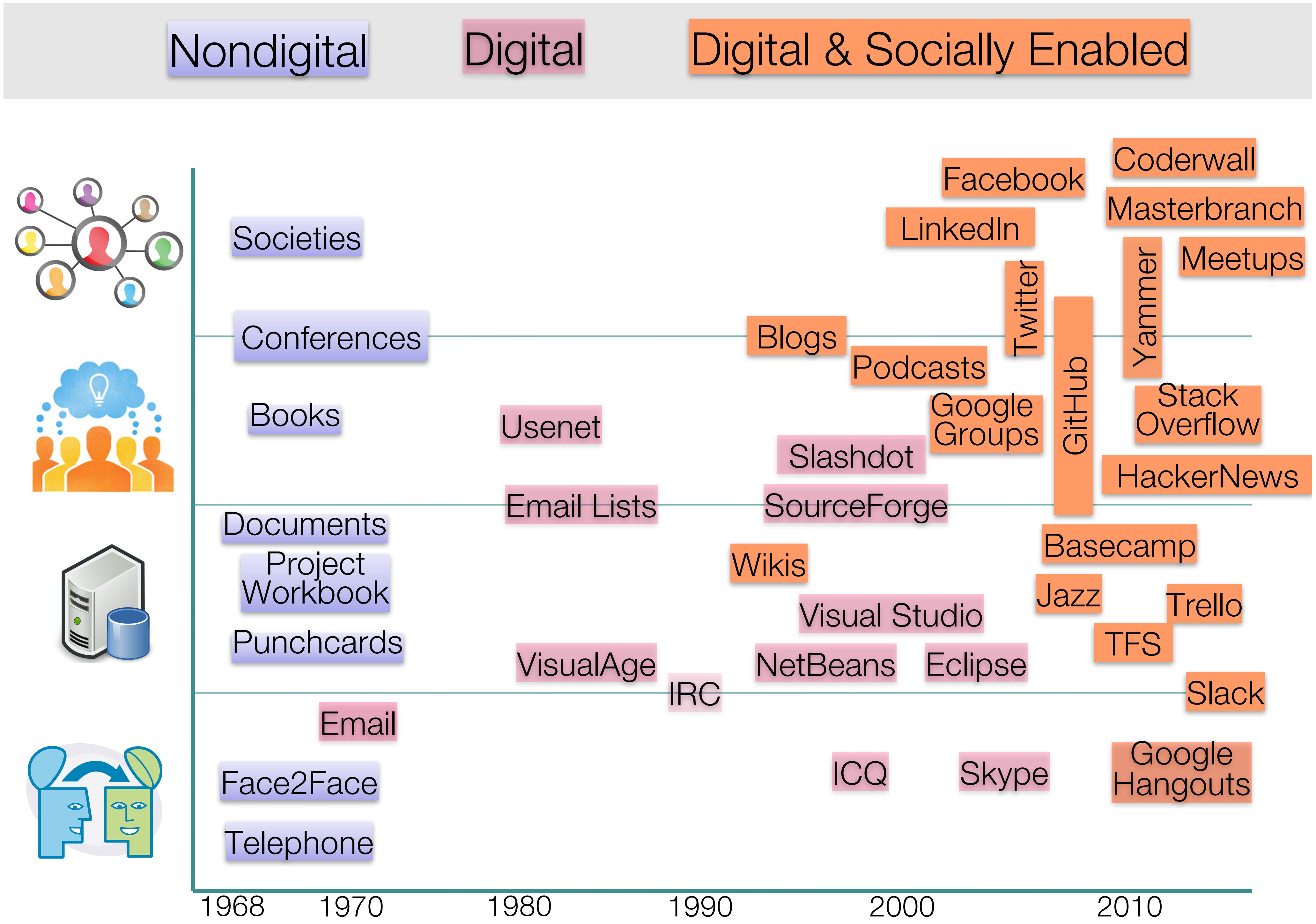


Communicating knowledge embedded in project artifacts (GitHub, Visual Studio…)

Communicating knowledge embedded in community resources (Books, Usenet)

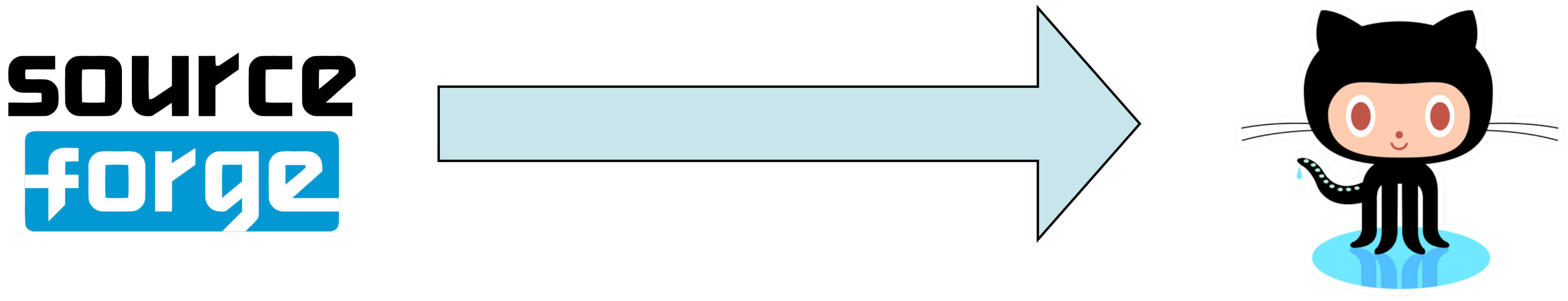
(new!) Communicating knowledge about social networks (Facebook, Coderwall, Twitter…)

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.


Nondigital
Digital
Digital & Socially Enabled
Societies
Conferences
Books
Documents
Project Workbook
Punchcards
Face2Face
Telephone
Email
Usenet
Email Lists
VisualAge
IRC
Blogs
Podcasts
Wikis
Slashdot
SourceForge
Visual Studio
NetBeans
Eclipse
ICQ
Skype
Google Groups
Facebook
LinkedIn
Twitter
GitHub
Yammer
Coderwall
Masterbranch
Meetups
Stack Overflow
HackerNews
Basecamp
Jazz
Trello
TFS
Slack
Google Hangouts
1968
1970
1980
1990
2000
2010

Properties of social dev tools

Architecture of participation
Transparency
Persistence
Emergence of behaviours



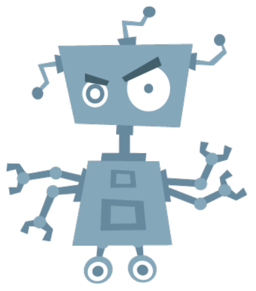
How developers **stay up to date** using Twitter

How developers **crowdsource** software **documentation**

How developers share **tacit knowledge**

How thousands of developers **coordinate which code** is committed and accepted through GitHub

How **bots integrate** collaboration and automation



How **gamification** can enhance **productivity**

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."

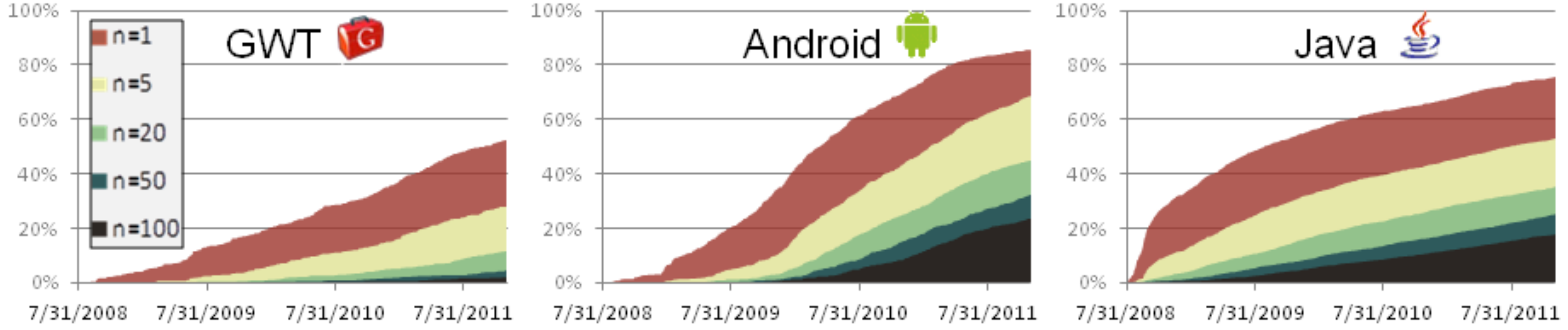
Leif Singer, Fernando Figueira Filho, Margaret-Anne Storey. *Software Engineering at the Speed of Light: How Developers Stay Current Using Twitter* *ICSE 2014.*

stackoverflow

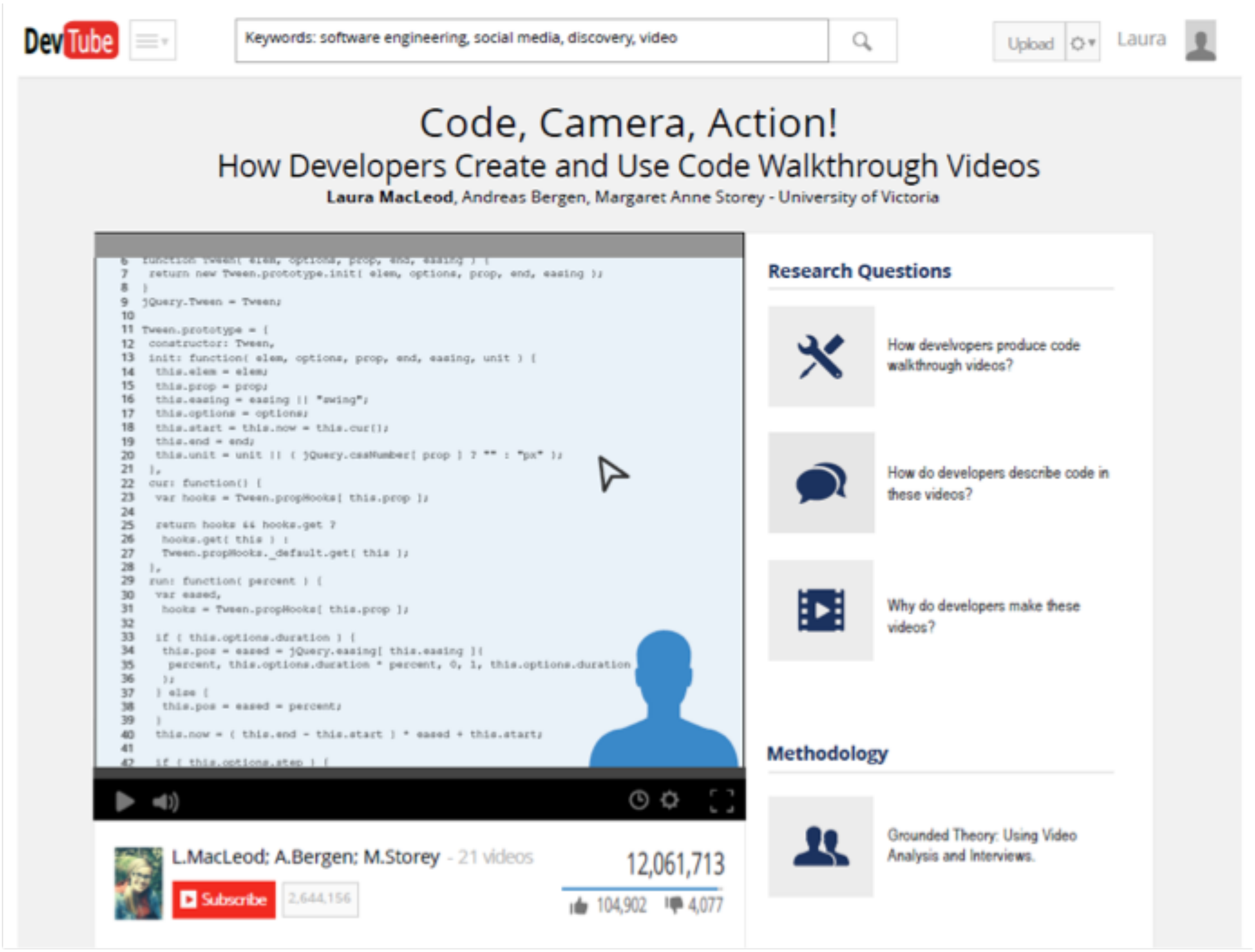
Crowd Documentation?

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/


DevTube
Keywords: software engineering, social media, discovery, video
Upload
Laura

Code, Camera, Action!

How Developers Create and Use Code Walkthrough Videos

Laura MacLeod, Andreas Bergen, Margaret Anne Storey - University of Victoria

```
function Tween( elem, options, prop, end, easing ) {
 return new Tween.prototype.init( elem, options, prop, end, easing );
}
jQuery.Tween = Tween;

Tween.prototype = {
 constructor: Tween,
 init: function( elem, options, prop, end, easing, unit ) {
  this.elem = elem;
  this.prop = prop;
  this.easing = easing || "swing";
  this.options = options;
  this.start = this.now = this.cur();
  this.end = end;
  this.unit = unit || ( jQuery.cssNumber[ prop ] ? "" : "px" );
 },
 cur: function() {
  var hooks = Tween.propHooks[ this.prop ];

  return hooks && hooks.get ?
   hooks.get( this ) :
   Tween.propHooks._default.get( this );
 },
 run: function( percent ) {
  var eased,
   hooks = Tween.propHooks[ this.prop ];

  if ( this.options.duration ) {
   this.pos = eased = jQuery.easing[ this.easing ](
    percent, this.options.duration * percent, 0, 1, this.options.duration
   );
  } else {
   this.pos = eased = percent;
  }
  this.now = ( this.end - this.start ) * eased + this.start;

  if ( this.options.step ) {
```

Research Questions

- How develvopers produce code walkthrough videos?
- How do developers describe code in these videos?
- Why do developers make these videos?

Methodology

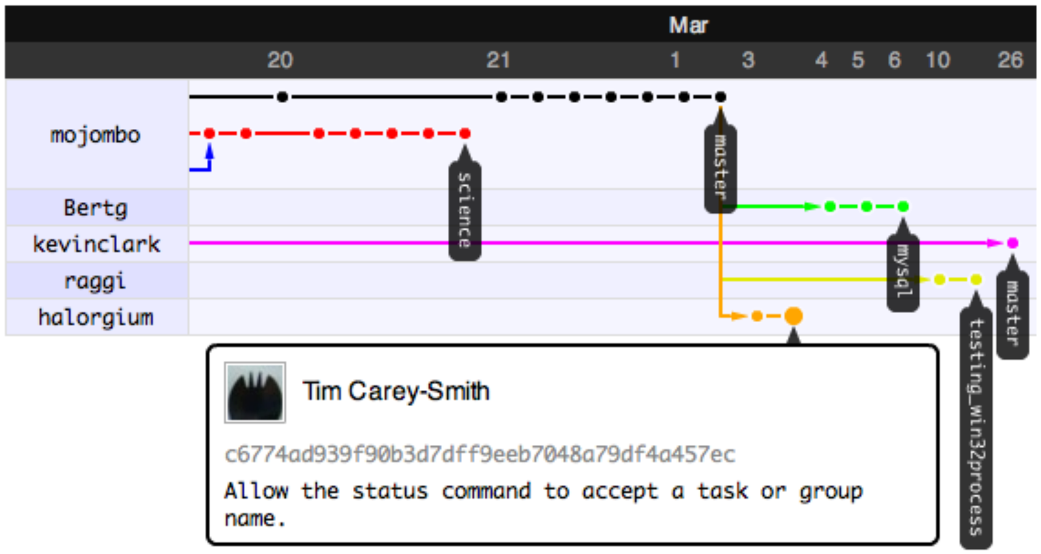
- Grounded Theory: Using Video Analysis and Interviews.

Code, camera action: How software developers **document and share program knowledge** using YouTube, L MacLeod, A Bergen, MA Storey - 2015 23rd IEEE Int. Conf. on Program Comprehension, 2015

"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]

How project owners use pull requests on GitHub for code review

Code quality, style, fit… influence decision to accept



Tim Carey-Smith

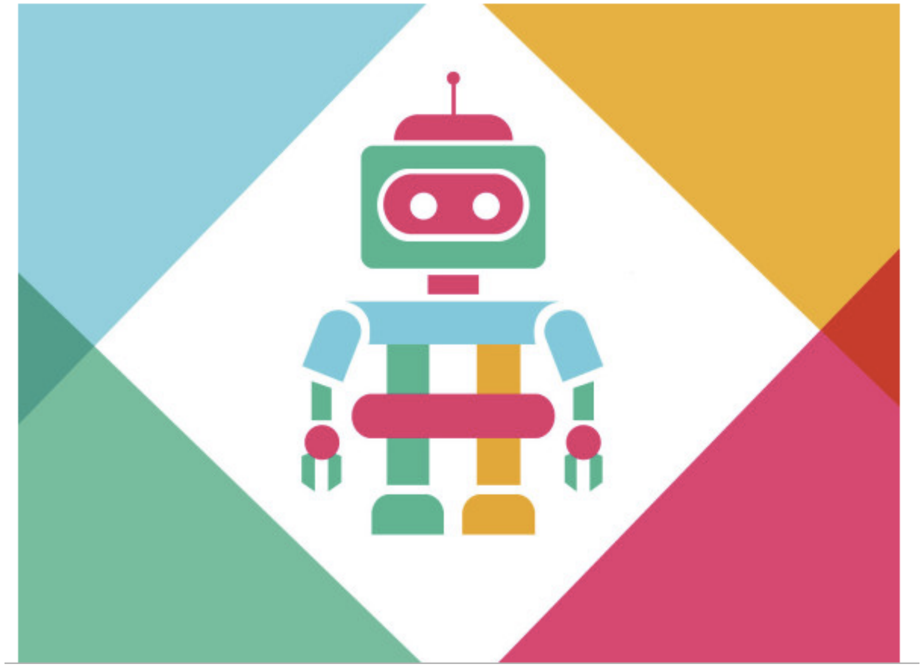
c6774ad939f90b3d7dff9eeb7048a79df4a457ec

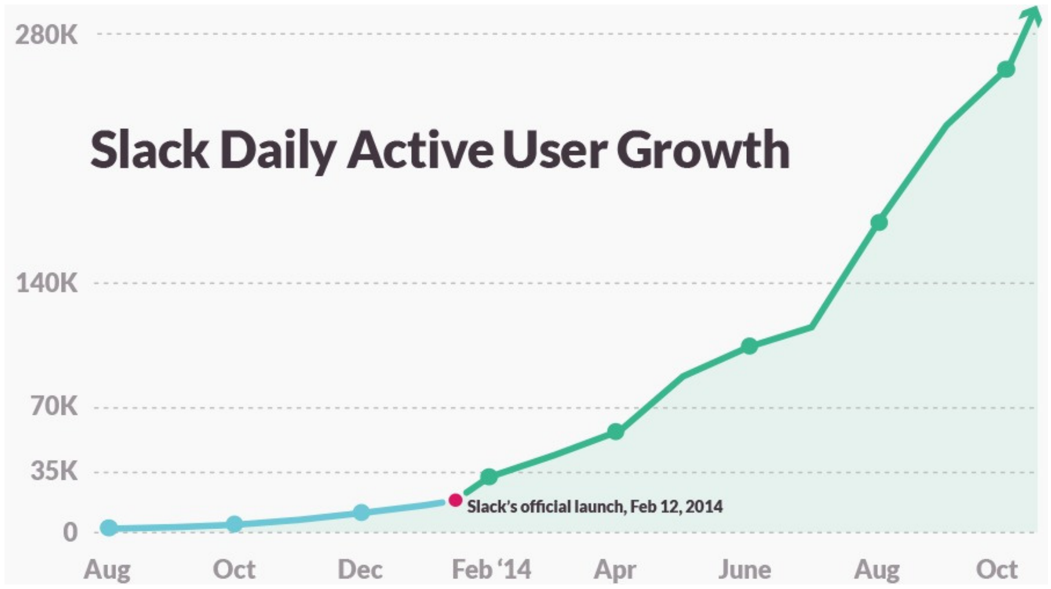
Allow the status command to accept a task or group name.

http://www.gousios.gr/blog/How-do-project-owners-use-pull-requests-on-Github

Why developers are slacking off?

SLACK IS OVERRUN WITH BOTS. FRIENDLY, WONDERFUL BOTS




Slack Daily Active User Growth
280K
140K
70K
35K
0
Slack's official launch, Feb 12, 2014
Aug
Oct
Dec
Feb '14
Apr
June
Aug
Oct

Kevin Kelly, Futurist: *"You'll be paid in the future based on how well you work with robots."*

What is a bot?

A bot is an application that performs automated, repetitive, pre-defined tasks

Conduit between users and services through a conversational UI

From setting an alarm, to telling you today's weather forecast, to gathering information

*Software development Bot **roles***

Code Bots

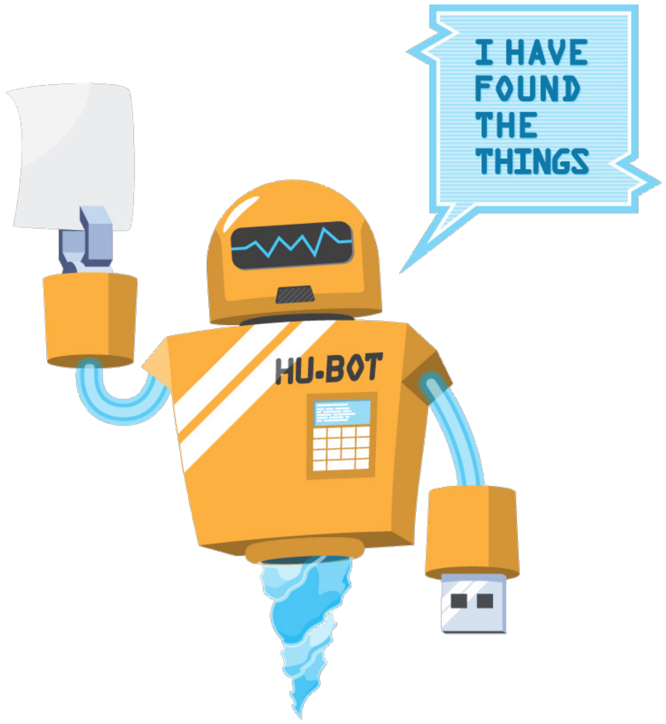
Test Bots

DevOps Bots

Support Bots

Documentation Bots

Entertainment Bots



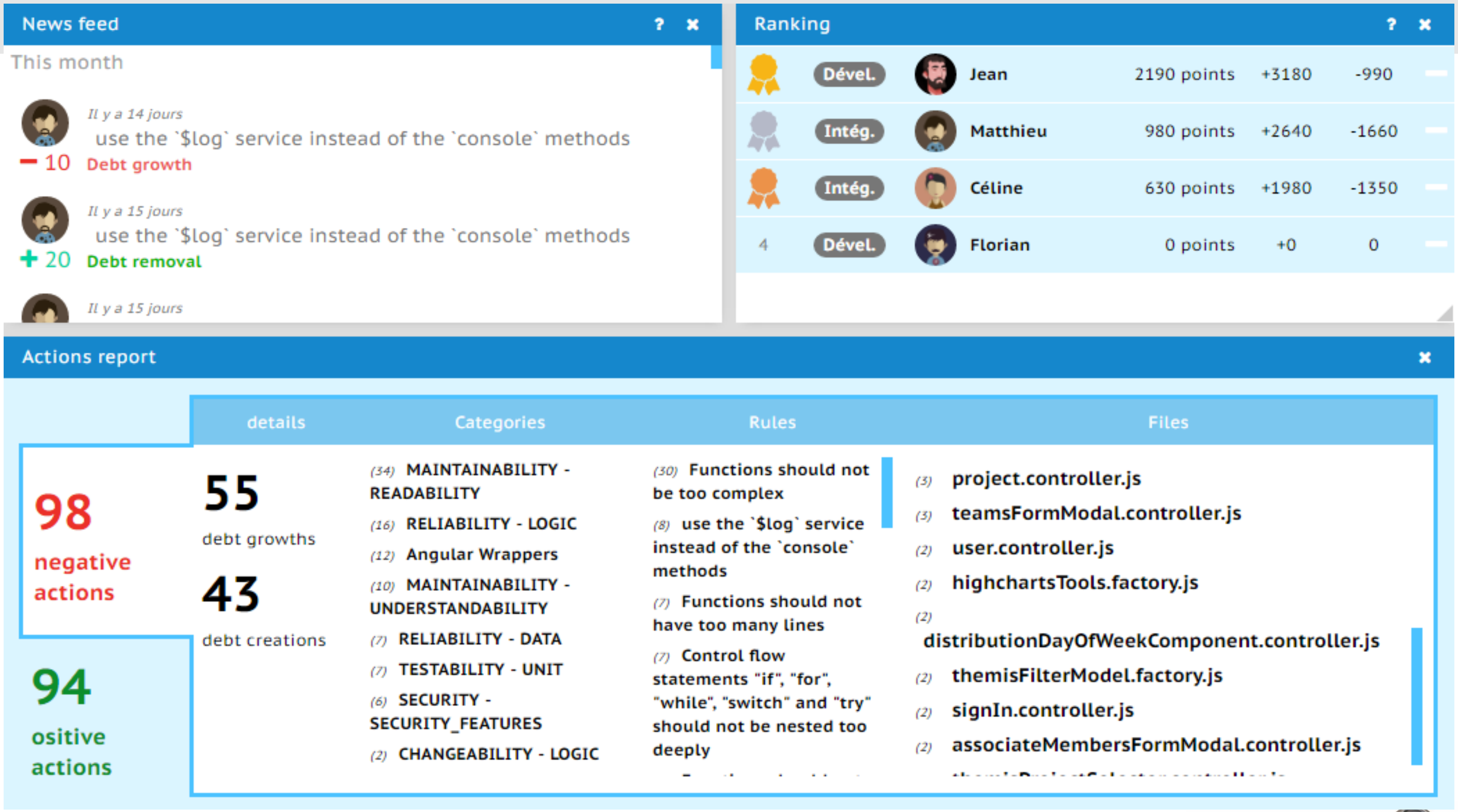
ChatOps are "putting tools right in the middle of the conversation"

- Jesse Newland, GitHub

B. Lin, A. Zagalsky, M.-A. Storey, and A. Serebrenik. Why developers are slacking off: Understanding how software teams use slack. CSCW 2016 (poster paper).

Categories also inspired by Sven Peter: https://svenpet.com/talks/rise-of-the-machines-automate-your-development/

Gamification: a Game Changer for Managing Technical Debt? A Design Study



News feed

This month

Il y a 14 jours
use the `$log` service instead of the `console` methods
– 10 Debt growth

Il y a 15 jours
use the `$log` service instead of the `console` methods
+ 20 Debt removal

Il y a 15 jours

Ranking

	Dével.	Jean	2190 points	+3180	-990
	Intég.	Matthieu	980 points	+2640	-1660
	Intég.	Céline	630 points	+1980	-1350
4	Dével.	Florian	0 points	+0	0

Actions report

98 negative actions

94 ositive actions

details

55 debt growths

43 debt creations

Categories

- (34) MAINTAINABILITY - READABILITY
- (16) RELIABILITY - LOGIC
- (12) Angular Wrappers
- (10) MAINTAINABILITY - UNDERSTANDABILITY
- (7) RELIABILITY - DATA
- (7) TESTABILITY - UNIT
- (6) SECURITY - SECURITY_FEATURES
- (2) CHANGEABILITY - LOGIC

Rules

- (30) Functions should not be too complex
- (8) use the `$log` service instead of the `console` methods
- (7) Functions should not have too many lines
- (7) Control flow statements "if", "for", "while", "switch" and "try" should not be nested too deeply

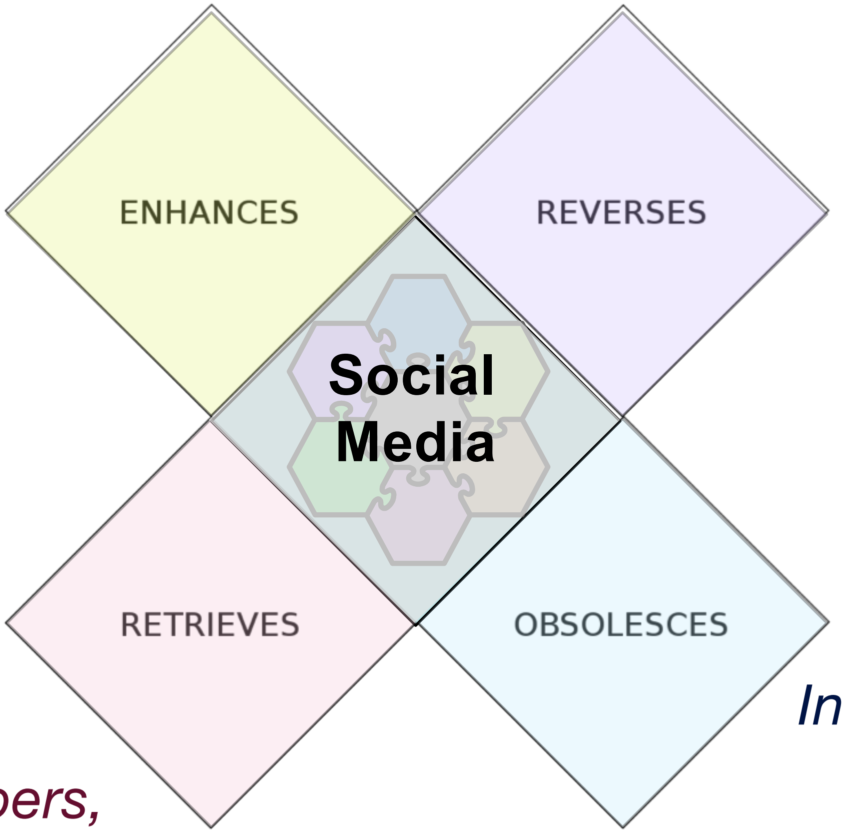
Files

- (3) project.controller.js
- (3) teamsFormModal.controller.js
- (2) user.controller.js
- (2) highchartsTools.factory.js
- (2) distributionDayOfWeekComponent.controller.js
- (2) themisFilterModel.factory.js
- (2) signIn.controller.js
- (2) associateMembersFormModal.controller.js

Matthieu Foucault & Margaret-Anne Storey, UVic
Xavier Blanc & Jean Remy Falleri, Universite de Bordeaux, France
Cedric Teyton, ProMyze, Bordeaux, France (under review)

(Distributed) Community formation, awareness, transparency, knowledge curation, motivation, learning, reuse, reputation

Informal processes, geek culture, reliance on search, privacy concerns, fragmentation. interruptions, Vendor lock-in



Programming gurus, end users as developers, verbal discussions, portfolios,competition, communities of practice

In-house expertise/jobs, formal documentation, classroom education, CVs, email lists, need for co-location

Discussion topics

Theories of media and how media shape software development

How social/communication channels have evolved over time in software engineering

From software development to knowledge work

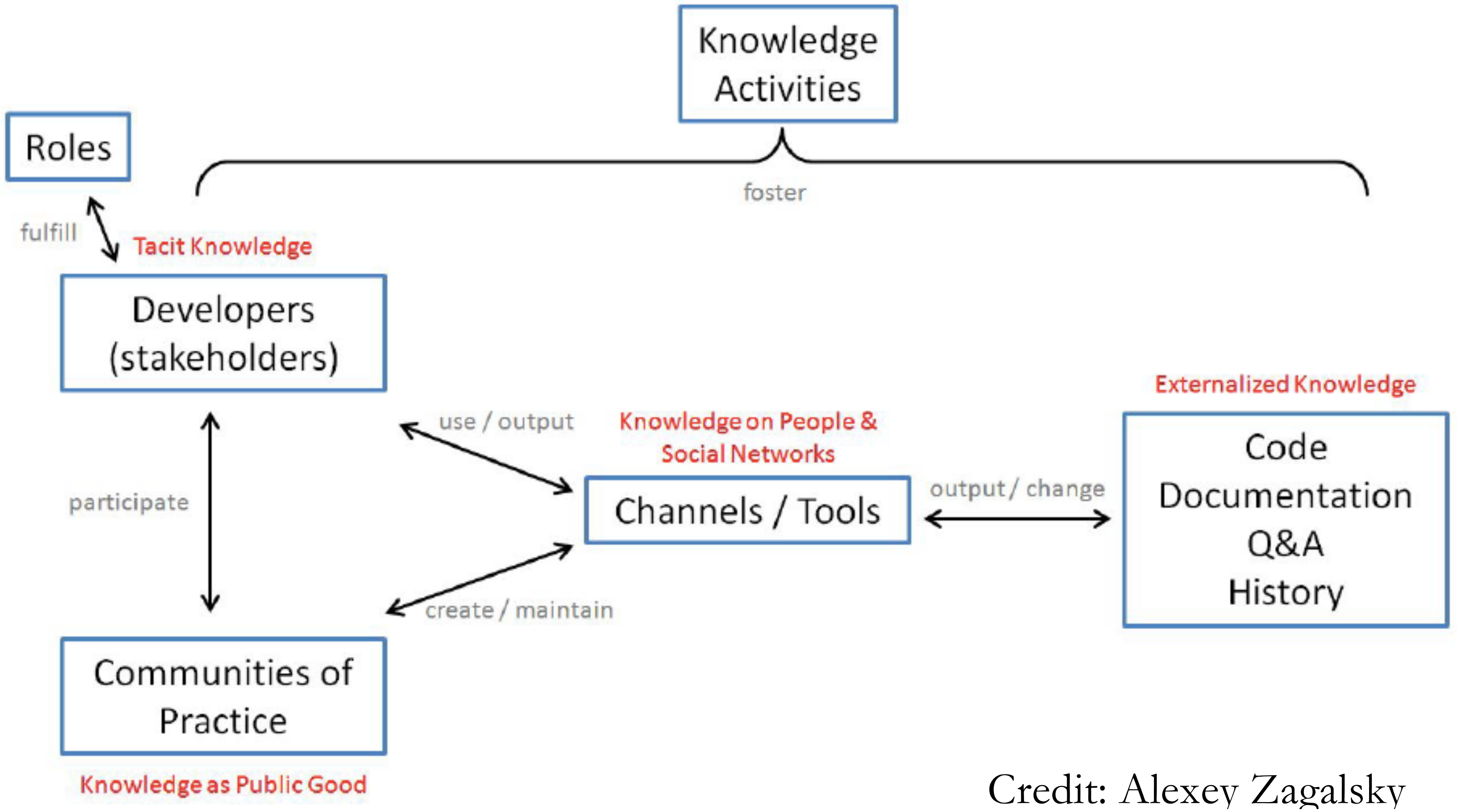
Knowledge workers

Primary job is thinking or **creating**, often **collaboratively**

Learning is key

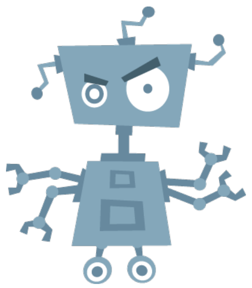
"*the most valuable asset of a 21st-century institution, whether business or non-business, will be its knowledge workers and their* ***productivity***" --- Peter Drucker, 1957

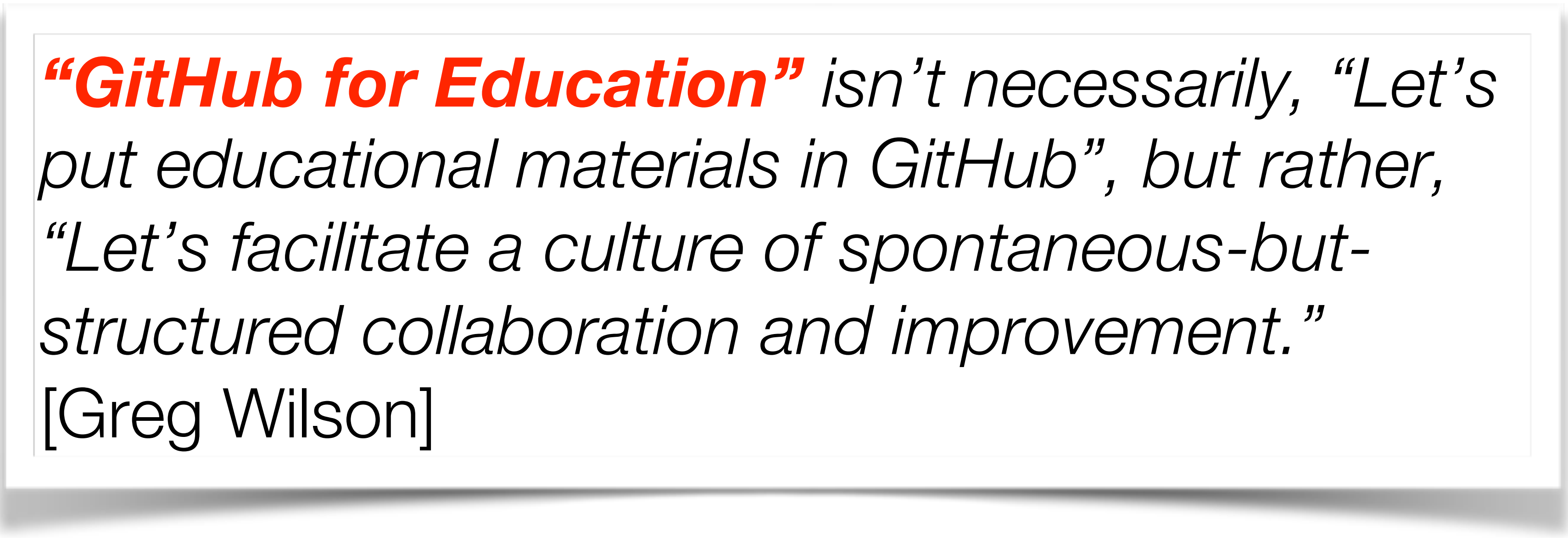
Knowledge Theory in Software Development



Credit: Alexey Zagalsky

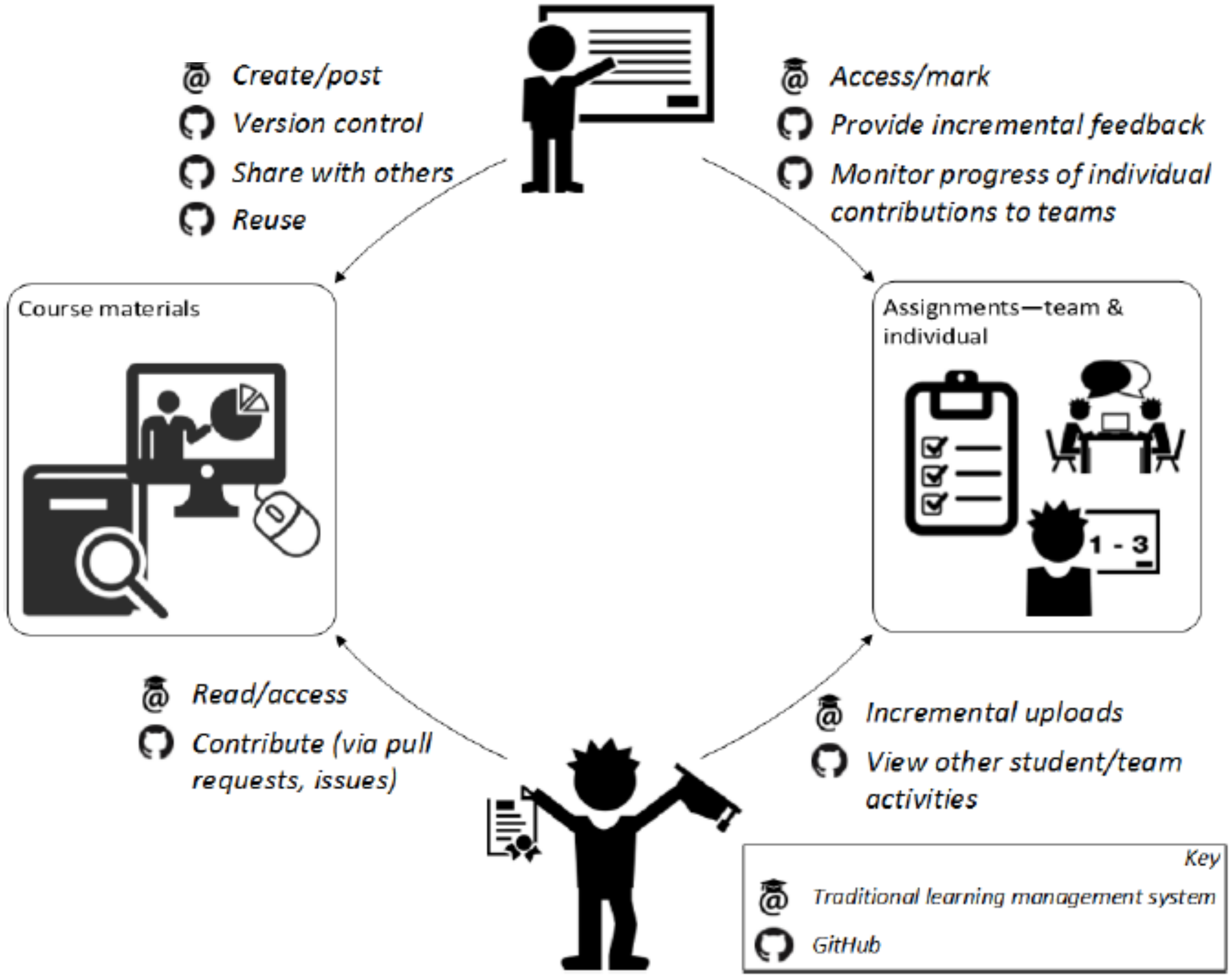
Developer tools from today are the knowledge worker tools of tomorrow:



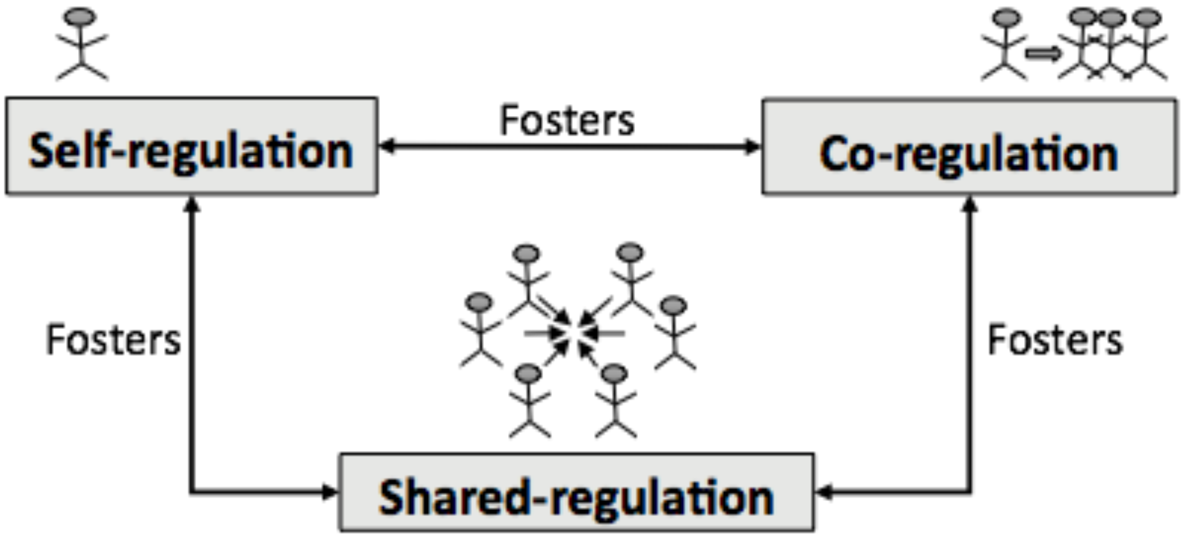


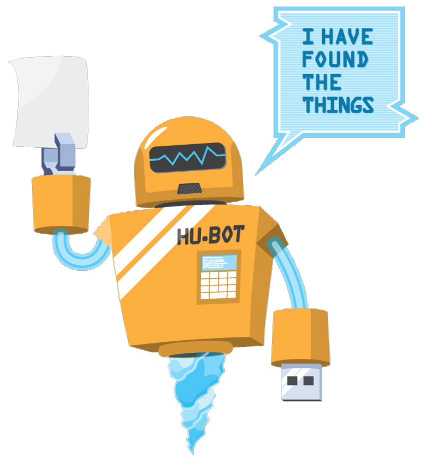
“GitHub for Education” *isn’t necessarily, “Let’s put educational materials in GitHub”, but rather, “Let’s facilitate a culture of spontaneous-but-structured collaboration and improvement.”*
[Greg Wilson]

Alexey Zagalsky, Joseph Feliciano, Margaret-Anne Storey, Yiyun Zhao, Weiliang Wang. The Emergence of GitHub as a Collaborative Platform for Education, CSCW 2015, Vancouver, Canada.
http://thechiselgroup.org/2014/10/27/the-emergence-of-github-as-a-collaborative-platform-for-education/


Create/post
Version control
Share with others
Reuse
Access/mark
Provide incremental feedback
Monitor progress of individual contributions to teams
Course materials
Assignments—team & individual
1 - 3
Read/access
Contribute (via pull requests, issues)
Incremental uploads
View other student/team activities
Key
Traditional learning management system
GitHub

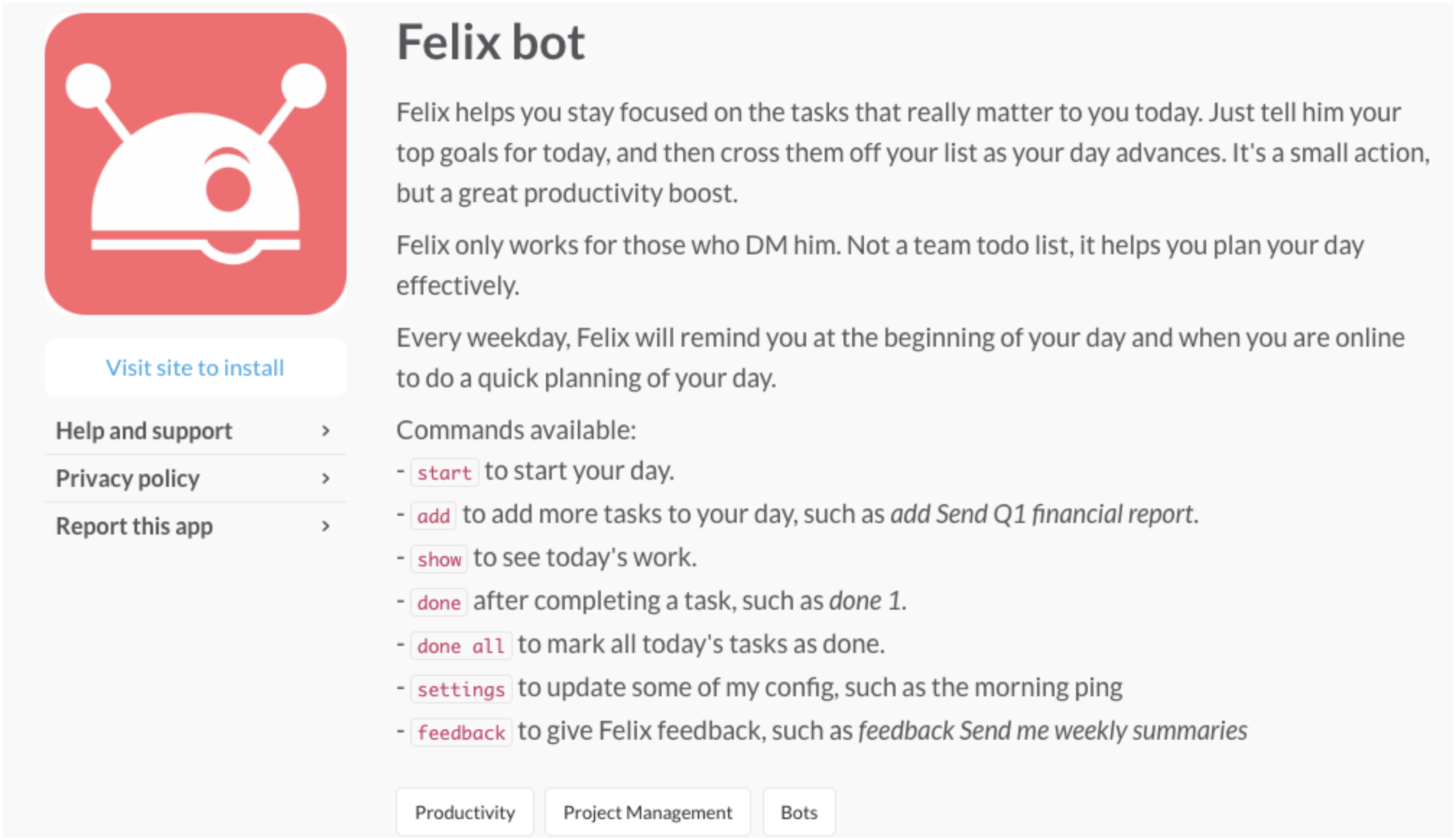
Understanding Bots through a Model of Regulation (CSCW 2017)





Maryi Arciniegas-Mendez
maryia@uvic.ca
@maryiarciniegas

Self regulation bots



Felix bot

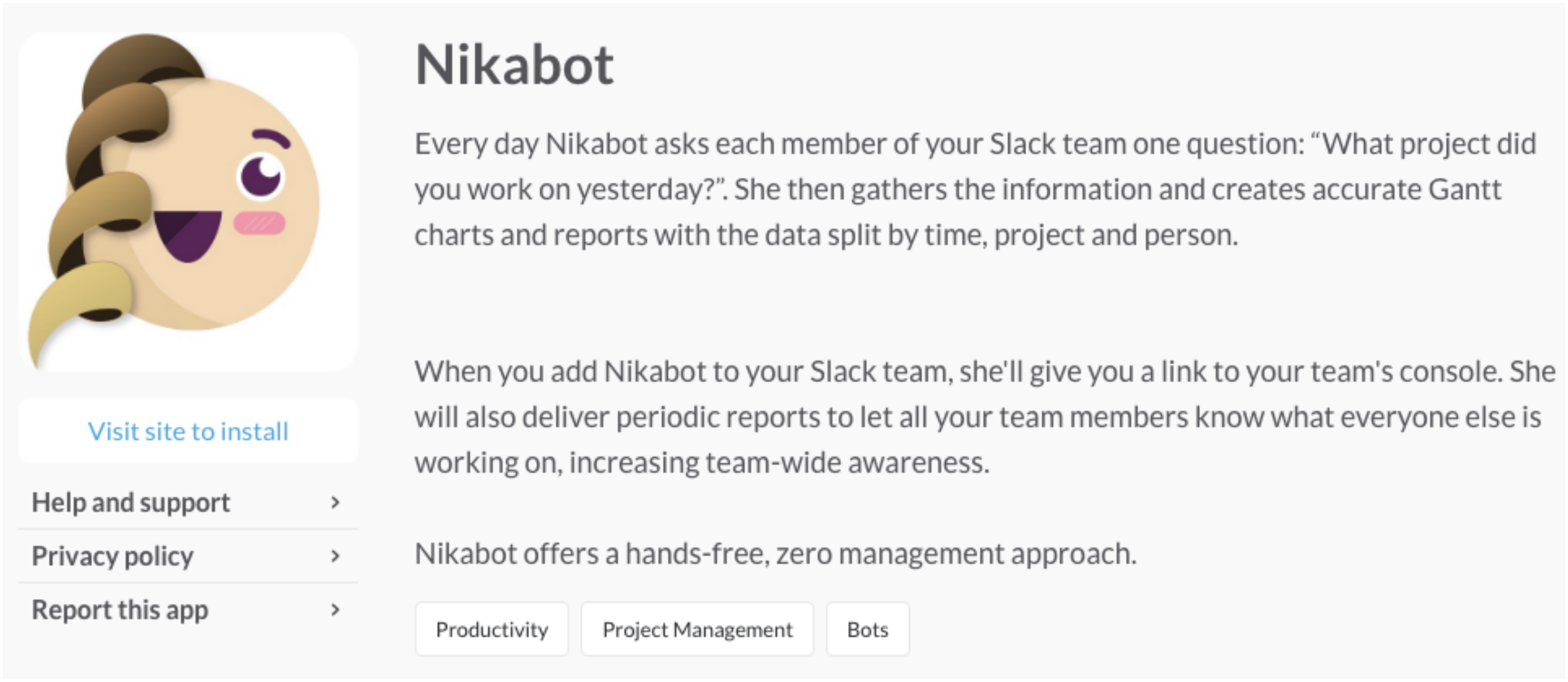
Felix helps you stay focused on the tasks that really matter to you today. Just tell him your top goals for today, and then cross them off your list as your day advances. It's a small action, but a great productivity boost.

Felix only works for those who DM him. Not a team todo list, it helps you plan your day effectively.

Every weekday, Felix will remind you at the beginning of your day and when you are online to do a quick planning of your day.

Commands available:
- `start` to start your day.
- `add` to add more tasks to your day, such as *add Send Q1 financial report*.
- `show` to see today's work.
- `done` after completing a task, such as *done 1*.
- `done all` to mark all today's tasks as done.
- `settings` to update some of my config, such as the morning ping
- `feedback` to give Felix feedback, such as *feedback Send me weekly summaries*

Co-regulation



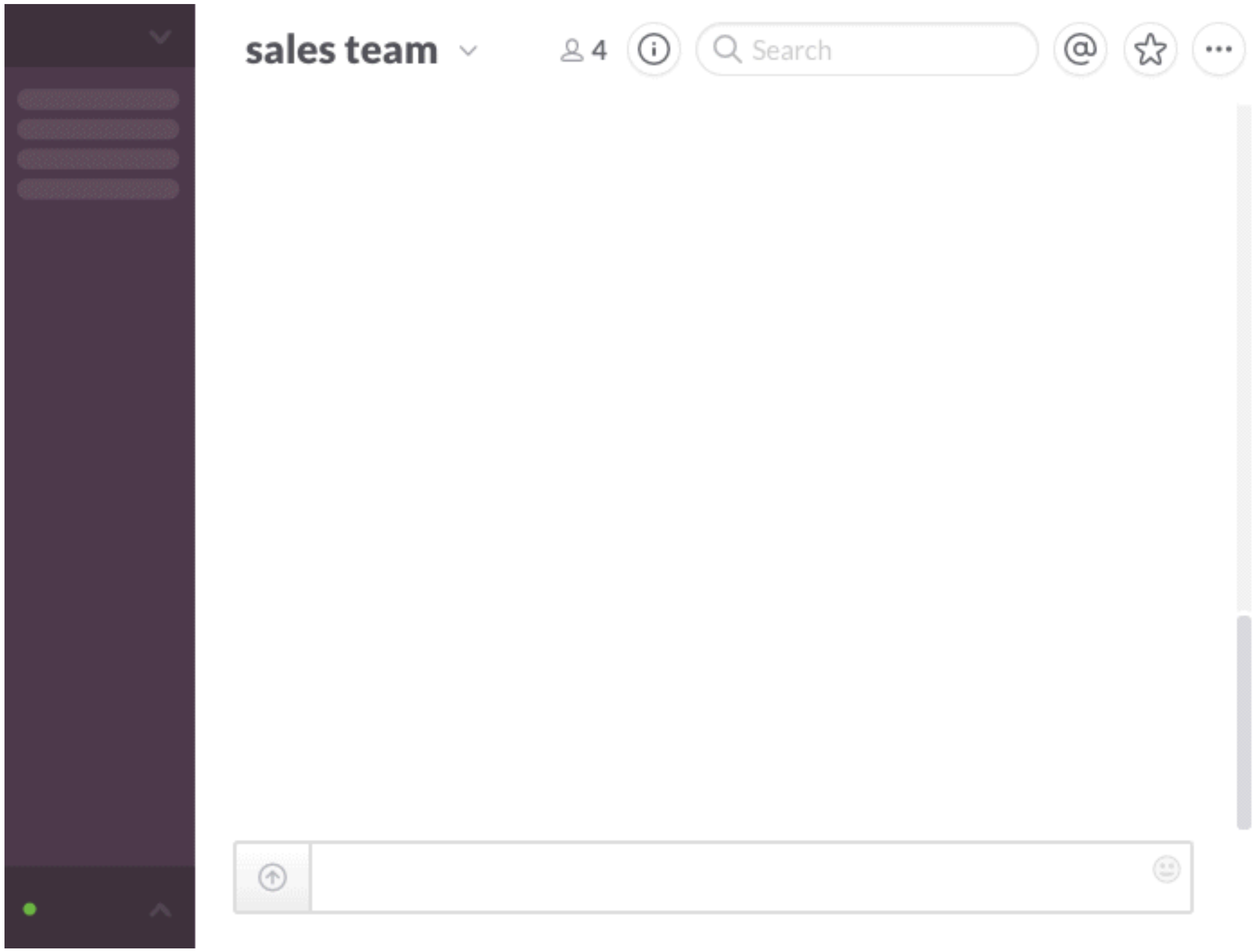
Nikabot

Every day Nikabot asks each member of your Slack team one question: “What project did you work on yesterday?”. She then gathers the information and creates accurate Gantt charts and reports with the data split by time, project and person.

When you add Nikabot to your Slack team, she'll give you a link to your team's console. She will also deliver periodic reports to let all your team members know what everyone else is working on, increasing team-wide awareness.

Nikabot offers a hands-free, zero management approach.

Shared regulation bots



http://meekan.com/slack/

Learning from existing research

Concerns?

Fragmentation of content and communication

Barriers to community building

Big data needs **“thick data”**

Opportunities?

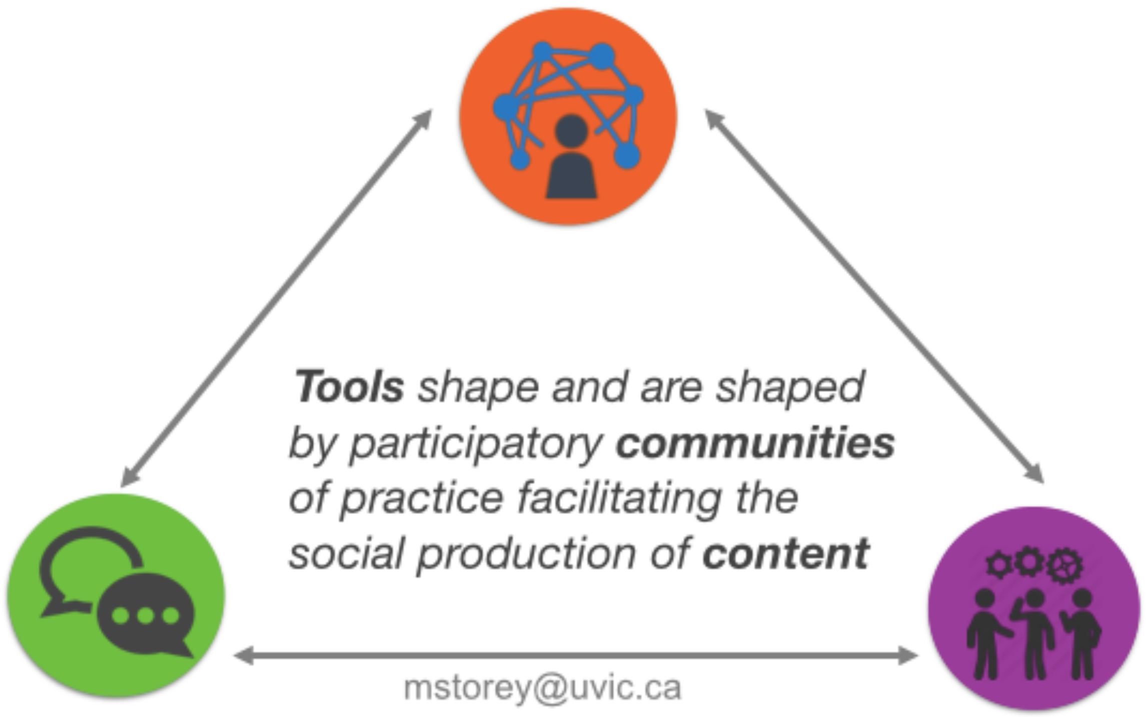
Learning is essential to productivity

Automation and AI show great potential to improve knowledge work productivity

Software developers are the **knowledge workers** of tomorrow….

"Program or be programmed"
Douglas Rushkoff

@margaretstorey


Tools** shape and are shaped by participatory **communities** of practice facilitating the social production of **content
mstorey@uvic.ca

Additional References (see slides for others)

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

M. Storey et al., **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

M. Arciniegas-Mendez, A. Zagalsky, M.-A. Storey, and A. F. Hadwin. **Regulation as an enabler for collaborative software development.** In Proceedings of the Eighth International Workshop on Cooperative and Human Aspects of Software Engineering, pages 97-100. IEEE Press, 2015.

B. Lin, A. Zagalsky, M.-A. Storey, and A. Serebrenik. **Why developers are slacking off: Understanding how software teams use slack.** In Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing Companion, pages 333{336. ACM, 2016.

M. Storey and A. Zagalsky, **Disrupting Developer Productivity One Bot at a Time**, FSE 2016 Visions Track, 2016,.

M. Storey, L. Singer, F. Figueira Filho, A. Zagalsky, and D. German, **How Social and Communication Channels Shape and Challenge a Participatory Culture in Software Development**, Transactions on Software Engineering, to appear.

G. Gousios, M.-A. Storey, and A. Bacchelli, **Work Practices and Challenges in Pull-Based Development: The Contributor's Perspective,** in Proceedings of the 38th International Conference on Software Engineering, 2016, pp. 285–296.

. Storey, L. Singer, F. Figueira Filho, A. Zagalsky, and D. German, **How Social and Communication Channels Shape and Challenge a Participatory Culture in Software Development**, Transactions on Software Engineering, to appear.

Acknowledgements

Alexey Zagalsky,
Georgious Gougios,
Chris Parnin,
Fernando Figueria Filho,
Daniel German,
Maryi Arciniegas.