Octocat, here and elsewhere, by GitHub https://octodex.github.com

Lessons in Social Coding

Software Analytics in the Age of GitHub

Bogdan Vasilescu
CMU

@b_vasilescu http://bvasiles.github.io

SSE 2016, Seattle

THE EVOLUTION OF THE "SOCIAL PROGRAMMER"

github SOCIAL CODING

ashley williams
ashleygwilliams

npm, inc
ridgewood, queens, NYC
http://ashleygwilliams.github.io/
Joined on Oct 31, 2011

776 Followers **38** Starred **15** Following

Organizations

Contributions | Repositories | Public activity

Follow

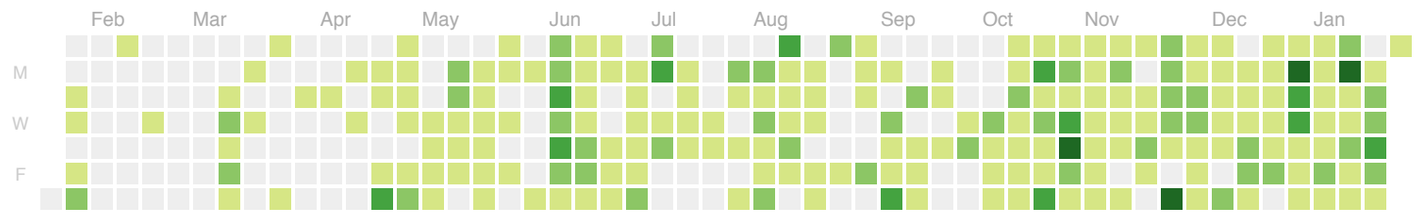
Popular repositories

- **breakfast-repo** — a collection of videos, recordings, and podcast… — 208 ★
- **x86-kernel** — a simple x86 kernel, extended with Rust — 48 ★
- **ashleygwilliams.github.io** — hi, i'm ashley. nice to meet you. — 37 ★
- **jsconf-2015-deck** — deck for jsconf2015 talk, "if you wish to learn e… — 32 ★
- **ratpack** — sinatra boilerplate using activerecord, sqlite, a… — 32 ★

Repositories contributed to

- npm/**docs** — The place where all the npm docs live. — 44 ★
- mozilla/**publish.webmaker.org** — The teach.org publishing service for goggles a… — 2 ★
- npm/**marky-markdown** — npm's markdown parser — 104 ★
- artisan-tattoo/**assistant-frontend** — ember client for assistant-API — 5 ★
- npm/**npm-camp** — a community conference for all things npm — 1 ★

Public contributions



Summary of pull requests, issues opened, and commits. Learn how we count contributions.

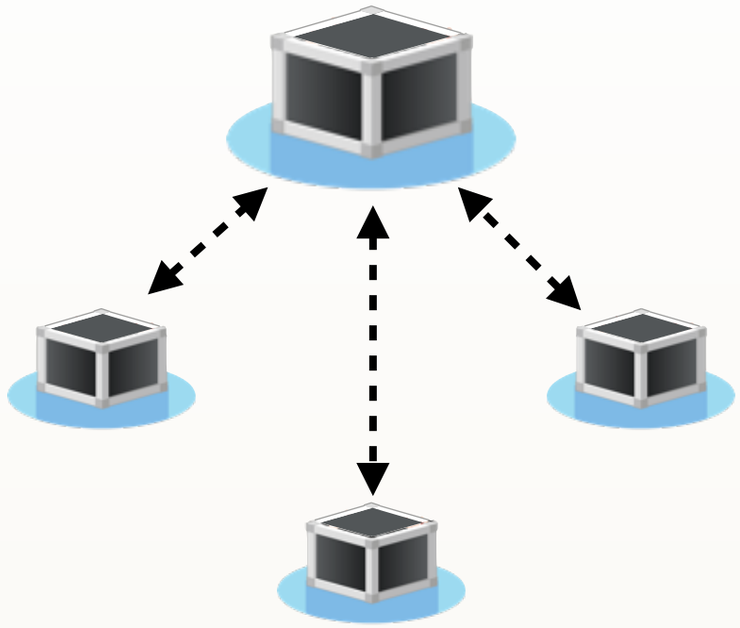
Less More

Contributions in the last year	Longest streak	Current streak
1,886 total	37 days	7 days
Jan 24, 2015 – Jan 24, 2016	October 7 – November 12	January 18 – January 24

https://github.com/ashleygwilliams

- Programming in a socially networked world: the evolution of the social programmer
  C Treude, F Figueira Filho, B Cleary, MA Storey.
  FutureCSD-CSCW 2012
- Social coding in GitHub: transparency and collaboration in an open software repository
  L Dabbish, C Stuart, J Tsay, J Herbsleb.
  CSCW 2012
- Social networking meets software development: Perspectives from GitHub, MSDN, Stack Exchange, and TopCoder
  A Begel, J Bosch, MA Storey.
  IEEE Software 2013

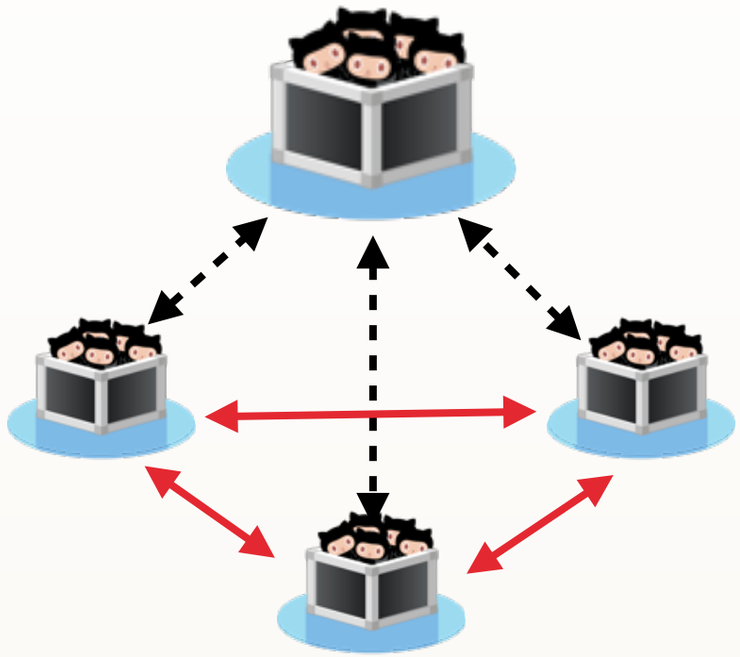
“SOCIAL CODING”: CODE IS MEANT TO BE SHARED



“SOCIAL CODING”: CODE IS MEANT TO BE SHARED

github SOCIAL CODING

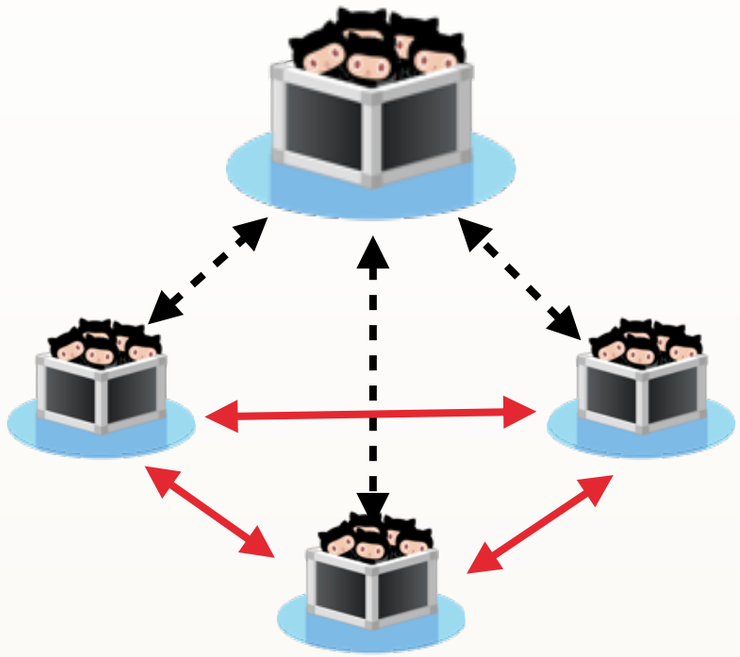
GIT



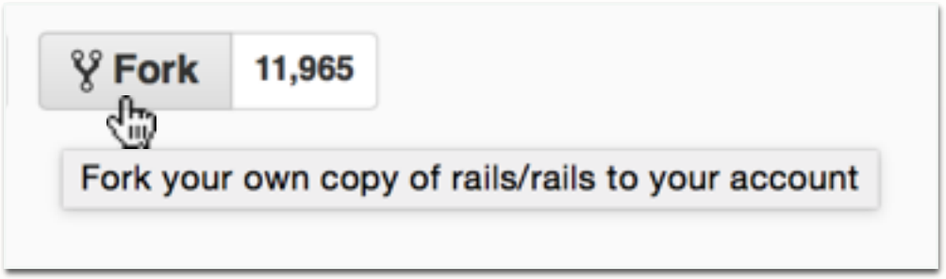
“SOCIAL CODING”: CODE IS MEANT TO BE SHARED

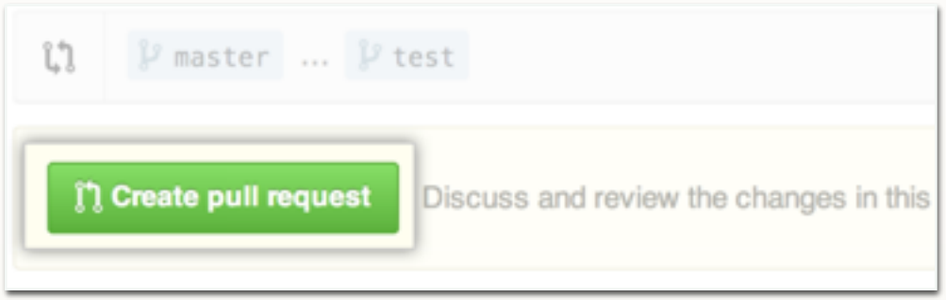
github SOCIAL CODING

GIT



GITHUB UI

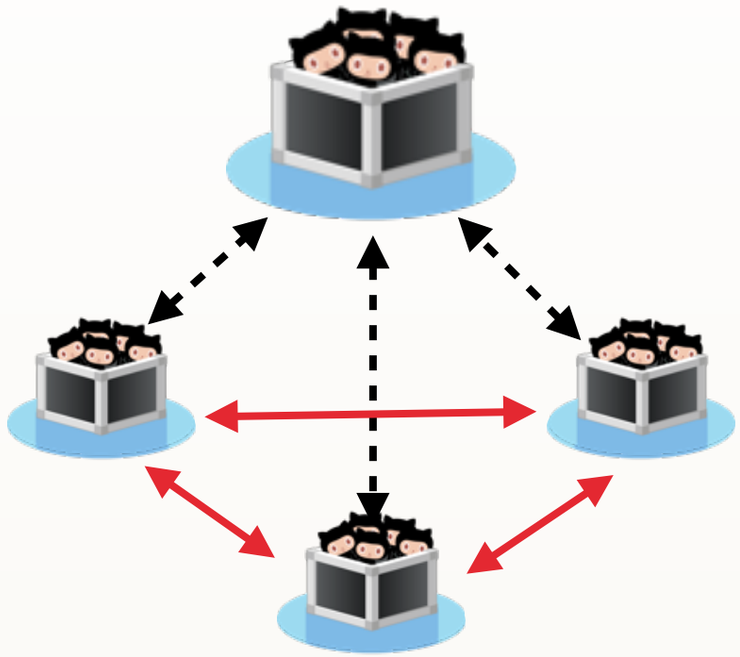




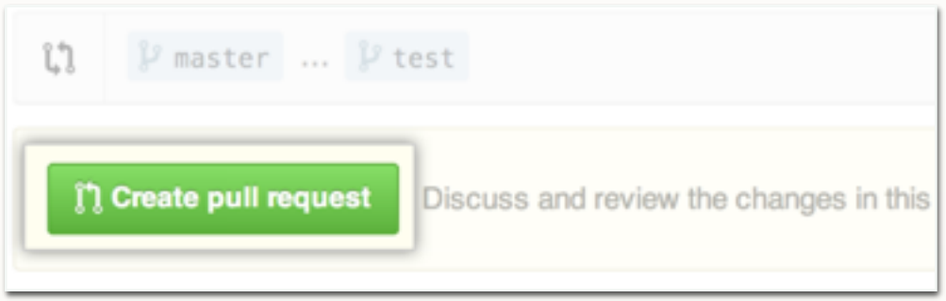
“SOCIAL CODING”: CODE IS MEANT TO BE SHARED

github SOCIAL CODING

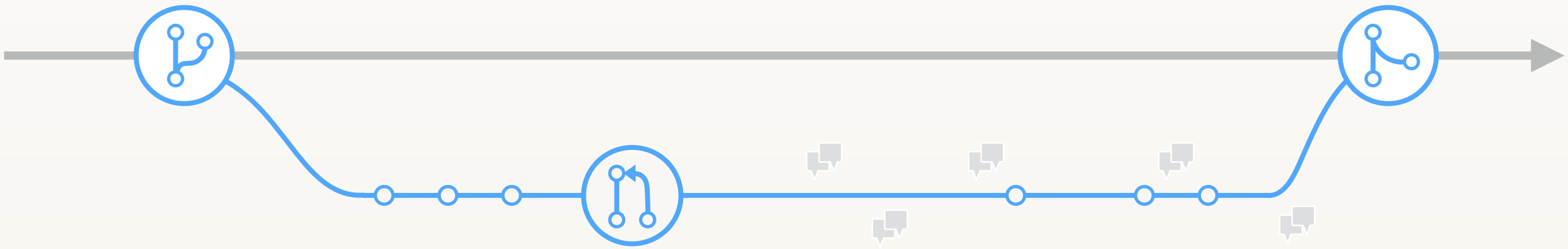
GIT



GITHUB UI



THE “PULL REQUEST” MODEL

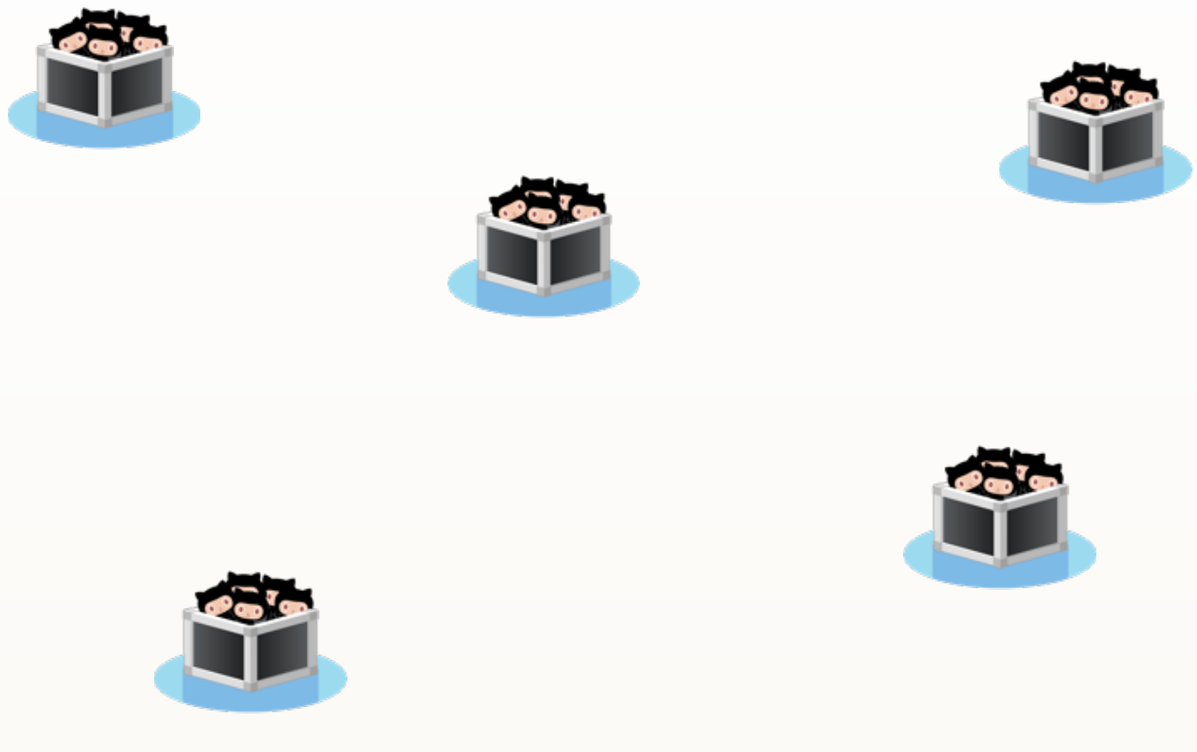


Unified development, testing, code review, integration → DEVOPS

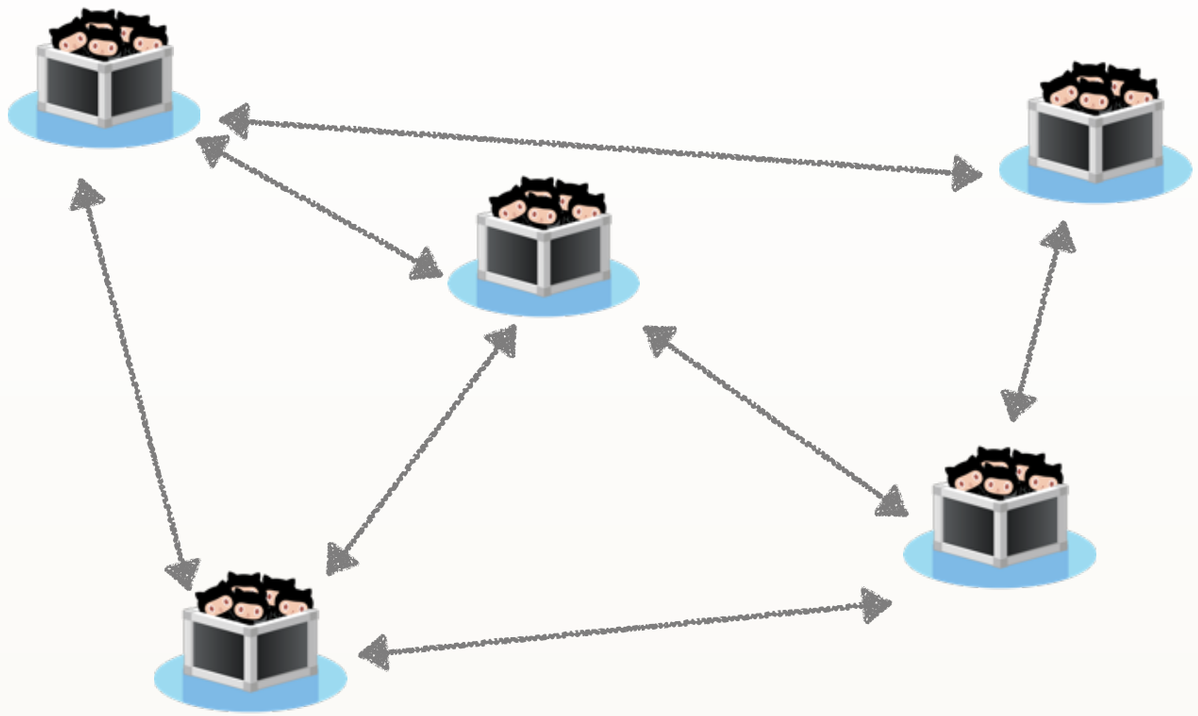
Lowest ever barrier to entry for newcomers

Democratic, open, social process

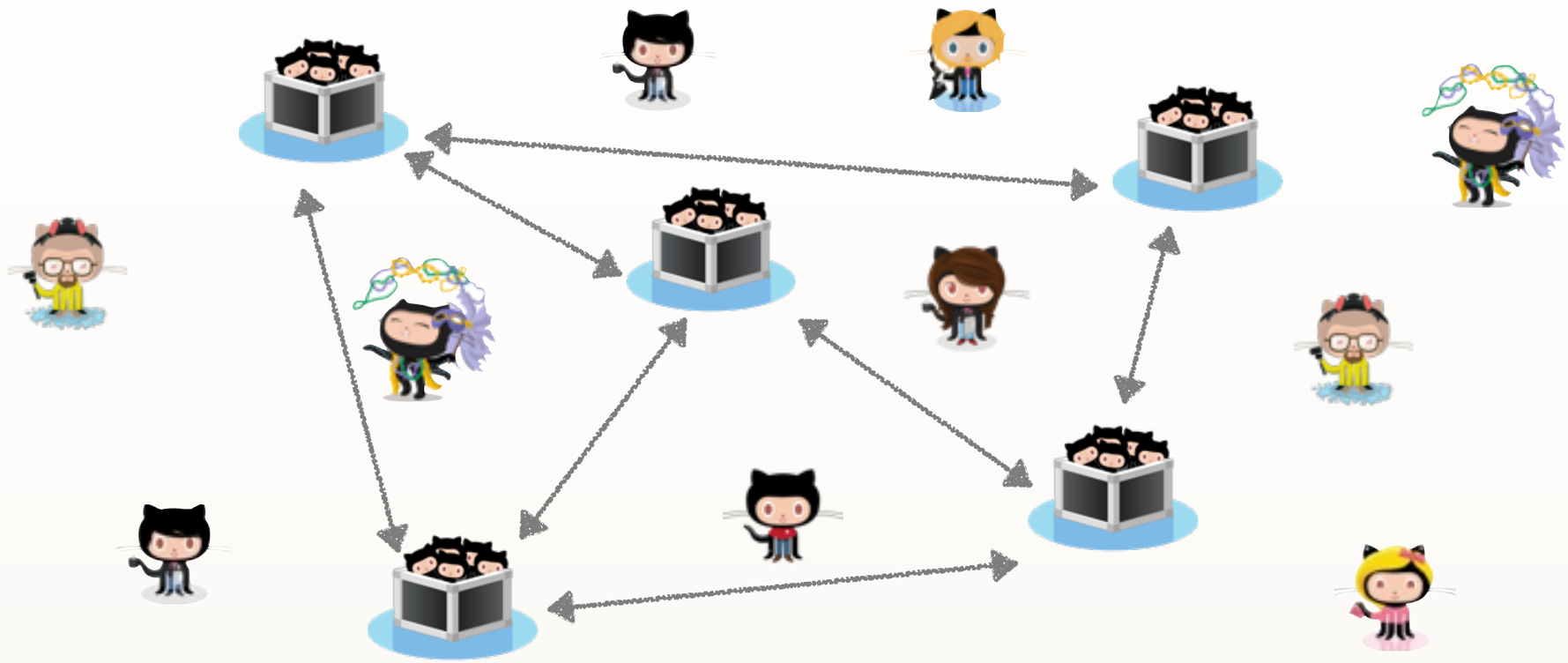
LARGE, DIVERSE, COMPLEX, SOCIAL ECOSYSTEM



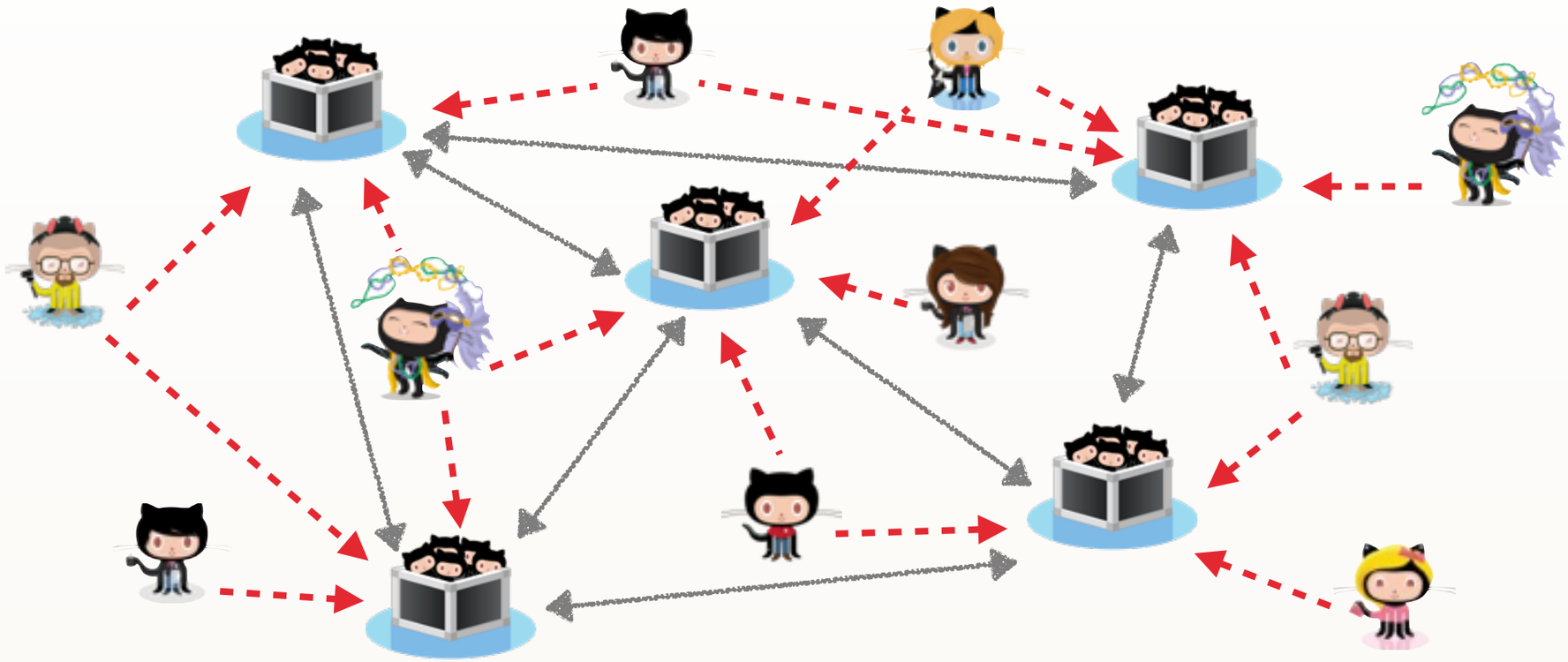
LARGE, DIVERSE, COMPLEX, SOCIAL ECOSYSTEM



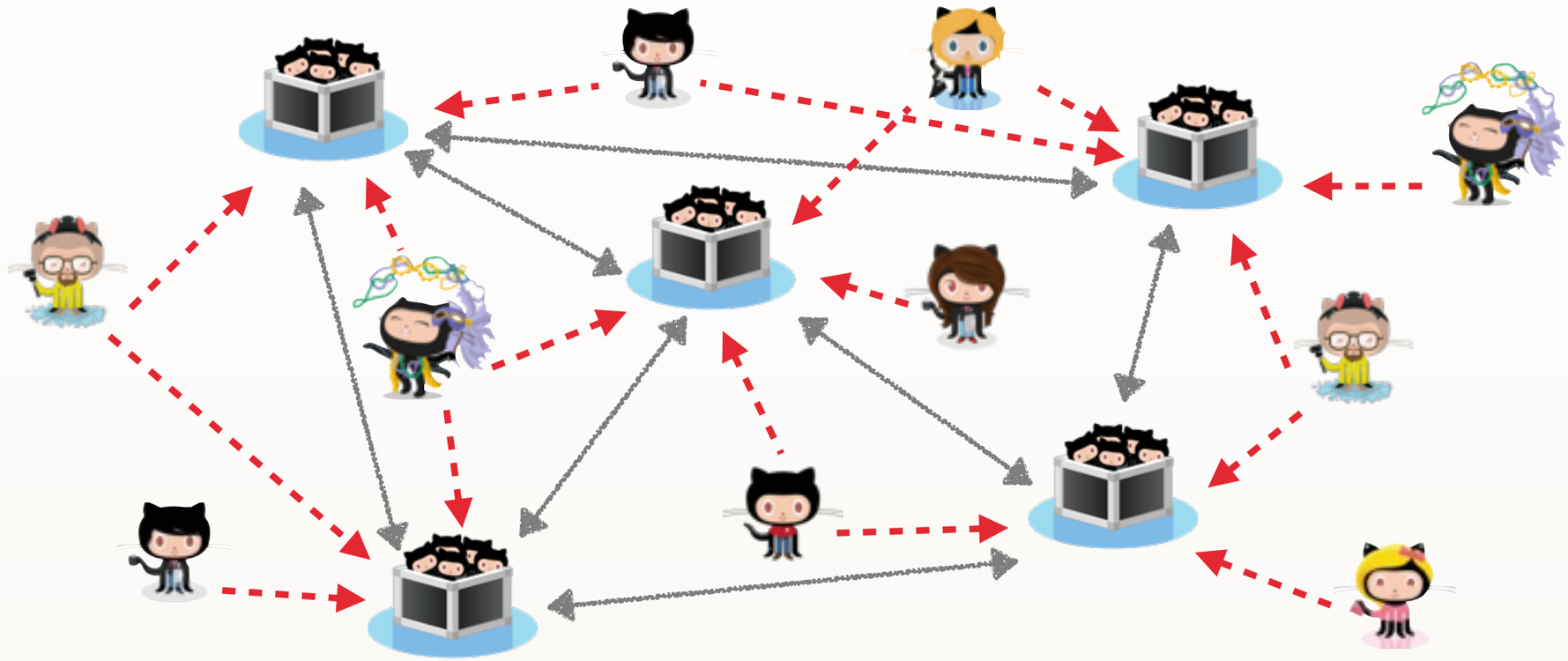
LARGE, DIVERSE, COMPLEX, SOCIAL ECOSYSTEM



LARGE, DIVERSE, COMPLEX, SOCIAL ECOSYSTEM



SOFTWARE DEVELOPMENT REVOLUTION



PRACTICE

- Large, distributed teams
- Process automation, DevOps
- Transparency, socialization, signaling

SOFTWARE DEVELOPMENT REVOLUTION

OPEN-SOURCE IS GROWING

Companies:

- 78% run OSS
- 66% build on top of OSS

BlackDuck 2015 survey https://goo.gl/Ltaqqs

SOFTWARE DEVELOPMENT REVOLUTION

BlackDuck 2015 survey https://goo.gl/Ltaqqs

OPEN-SOURCE IS GROWING

open source™

Companies:

- 78% run OSS
- 66% build on top of OSS

SOCIAL CODING IS GROWING

~~12~~ 18 M people

~~31~~ 47 M repositories

- GitHub stats from: https://github.com/about

SOFTWARE DEVELOPMENT REVOLUTION

BlackDuck 2015 survey https://goo.gl/Ltaqqs

OPEN-SOURCE IS GROWING

Companies:

- 78% run OSS
- 66% build on top of OSS

SOCIAL CODING IS GROWING

~~12~~ 18 M people

~~31~~ 47 M repositories

18.5 million software dev's

- GitHub stats from: https://github.com/about
- World estimates from: http://goo.gl/Htnni9

SOFTWARE DEVELOPMENT REVOLUTION

OPEN-SOURCE IS GROWING

open source™

Companies:

- 78% run OSS
- 66% build on top of OSS

BlackDuck 2015 survey https://goo.gl/Ltaqqs

SOCIAL CODING IS GROWING

~~12~~ 18 M people

~~31~~ 47 M repositories

18.5 million software dev's

15,000+ people

- GitHub stats from: https://github.com/about
- World estimates from: http://goo.gl/Htnni9

SOFTWARE DEVELOPMENT REVOLUTION

OPEN-SOURCE IS GROWING

open source™

Companies:

- 78% run OSS
- 66% build on top of OSS

SOCIAL CODING IS GROWING

~~12~~ 18 M people

~~31~~ 47 M repositories

18.5 million software dev's

15,000+ people

CULTURE CHANGE

"it's just so uncool not sharing the code in the age of social coding"

- GitHub stats from: https://github.com/about
- World estimates from: http://goo.gl/Htnni9

SOFTWARE DEVELOPMENT REVOLUTION

OPEN-SOURCE IS GROWING

open source™

Companies:

- 78% run OSS
- 66% build on top of OSS

SOCIAL CODING IS GROWING

~~12~~ 18 M people

~~31~~ 47 M repositories

18.5 million software dev's

15,000+ people

CULTURE CHANGE

"it's just so uncool not sharing the code in the age of social coding"

HIRING

- **$100+** /hour:
  - owns popular OSS products;
  - stack**overflow** score > 20K; …
- **$50+** /hour:
  - active OSS contributor;
  - stack**overflow** score > 5K; …

- GitHub stats from: https://github.com/about
- World estimates from: http://goo.gl/Htnni9
- **How Much Do You Cost?** Yegor Bugayenko http://goo.gl/N0mL3F
- Activity traces and signals in software developer recruitment and hiring J Marlow, L Dabbish. *CSCW 2013*

SOFTWARE DEVELOPMENT REVOLUTION

OPEN-SOURCE IS GROWING

open source™

Companies:

- 78% run OSS
- 66% build on top of OSS

SOCIAL CODING IS GROWING

~~12~~ 18 M people

~~31~~ 47 M repositories

18.5 million software dev's

15,000+ people

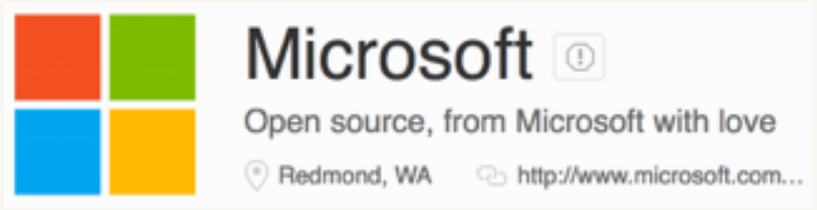
CULTURE CHANGE

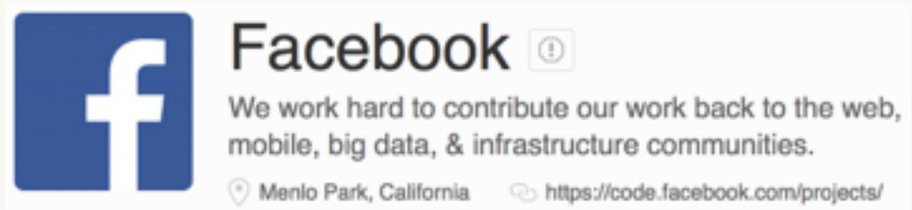
"it's just so uncool not sharing the code in the age of social coding"

HIRING

- **$100+** /hour:
  - owns popular OSS products;
  - stack**overflow** score > 20K; …
- **$50+** /hour:
  - active OSS contributor;
  - stack**overflow** score > 5K; …

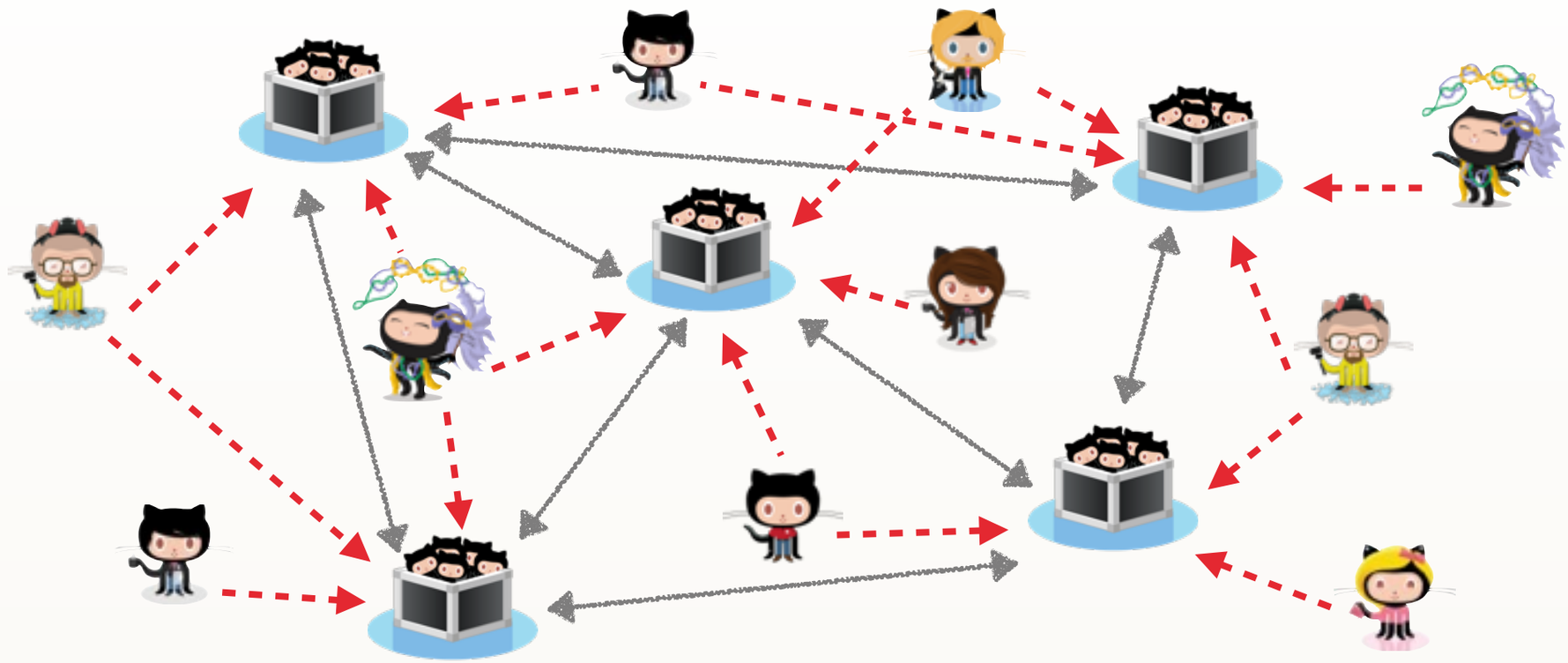
INDUSTRIAL INVOLVEMENT & ADOPTION





- GitHub stats from: https://github.com/about
- World estimates from: http://goo.gl/Htnni9
- Open source-style collaborative development practices in commercial projects using GitHub E Kalliamvakou, D Damian, K Blincoe, L Singer, DM German. *ICSE 2015*
- How Much Do You Cost? Yegor Bugayenko http://goo.gl/N0mL3F
- Activity traces and signals in software developer recruitment and hiring J Marlow, L Dabbish. *CSCW 2013*

EMPIRICAL RESEARCH REVOLUTION



PRACTICE

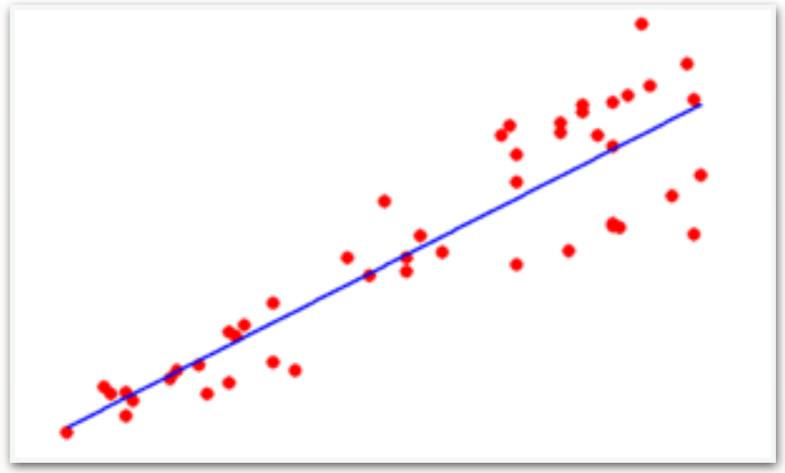
- Large, distributed teams
- Process automation, DevOps
- Transparency, socialization, signaling

RESEARCH

- Breadth of topics, from impression formation to programming languages and software quality
- “Big data”, mixed methods

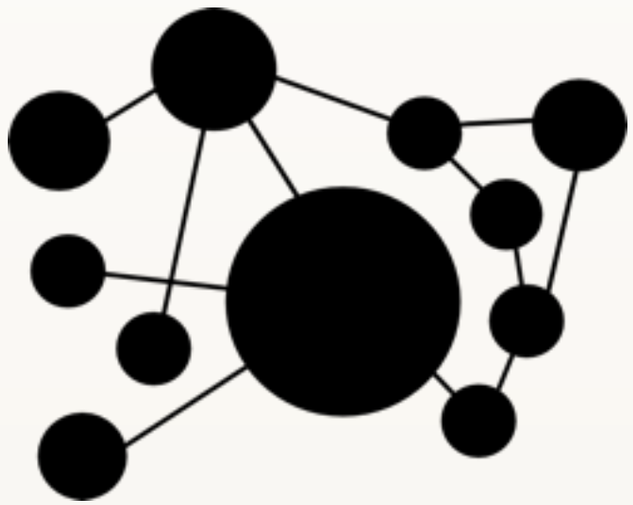
TOOLKIT FOR SOCIAL SOFTWARE ENGINEERING RESEARCHERS

THEORY



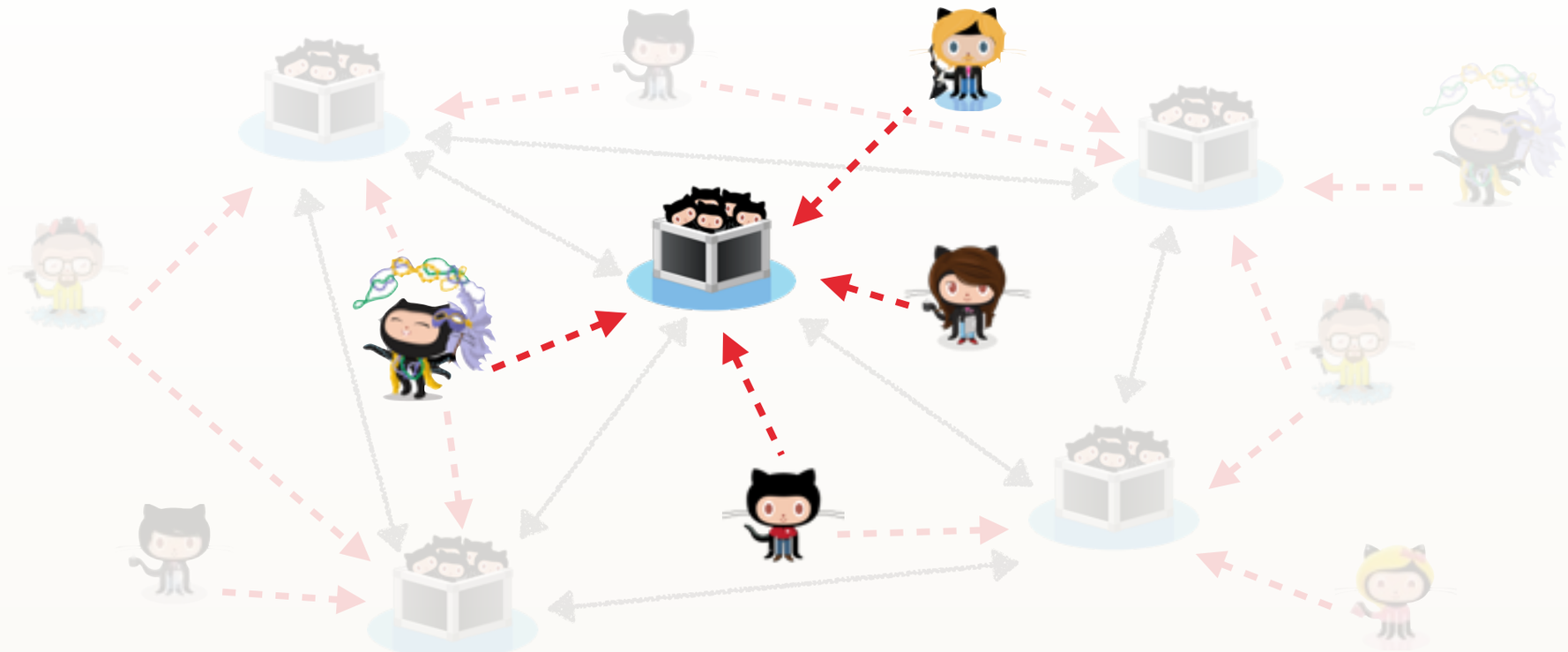
STATISTICS

QUALITATIVE METHODS



NETWORK SCIENCE

EXAMPLE 1: PULL REQUEST EVALUATION TIME



rails / rails

Watch 2,305 | Star 33,378 | Fork 13,615

Code | Issues 554 | Pull requests 667 | Projects 0 | Pulse | Graphs

Filters | is:pr is:open | Labels | Milestones | New pull request

667 Open | 16,951 Closed | Author | Labels | Milestones | Assignee | Sort

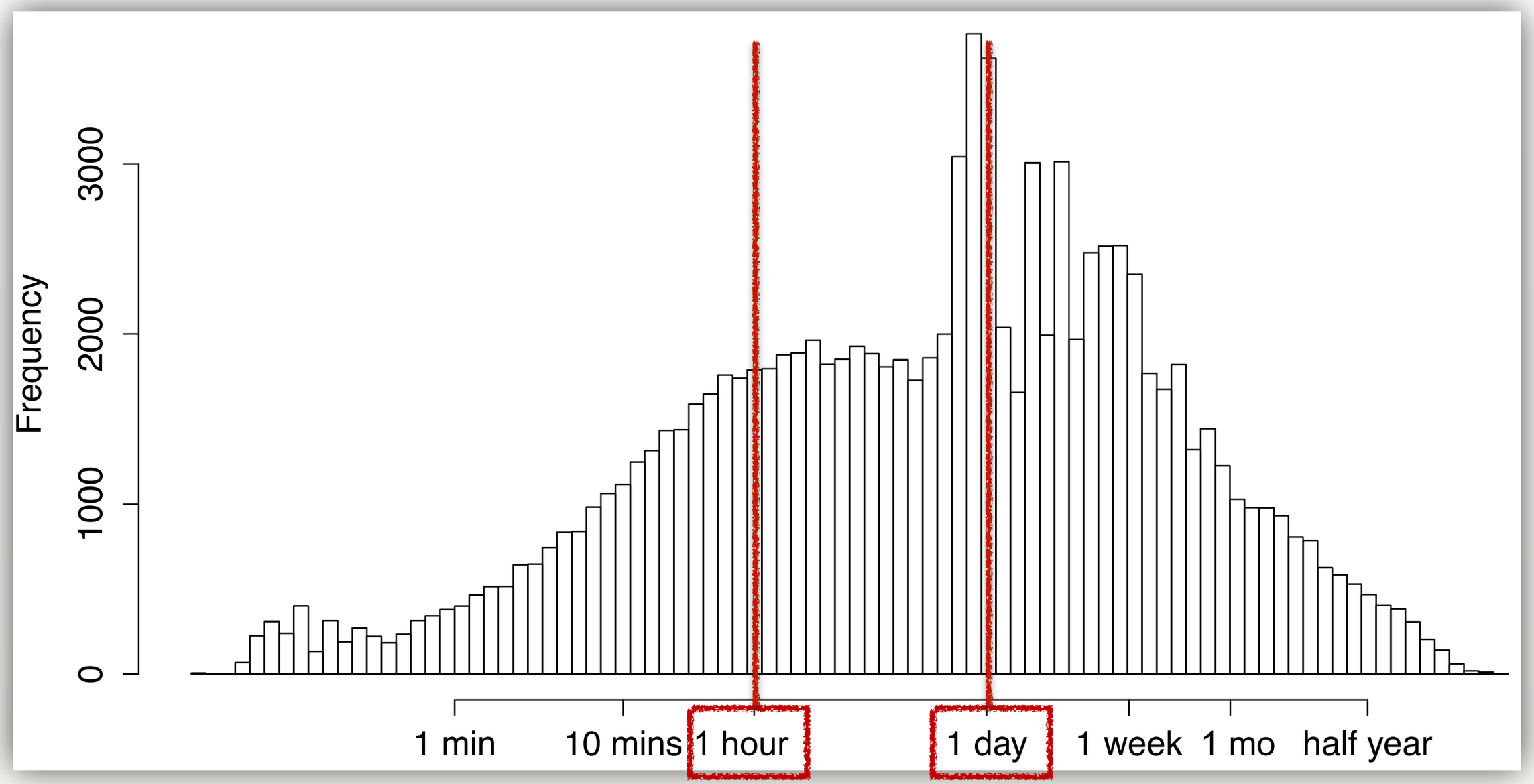
Schema cache in YAML
#27042 opened 23 minutes ago by kirs — 1

Add `:skip_sprockets` to `Rails::PluginBuilder::PASSTHROUGH_OPTIONS` railties
#27040 opened 4 hours ago by ttanimichi — 1

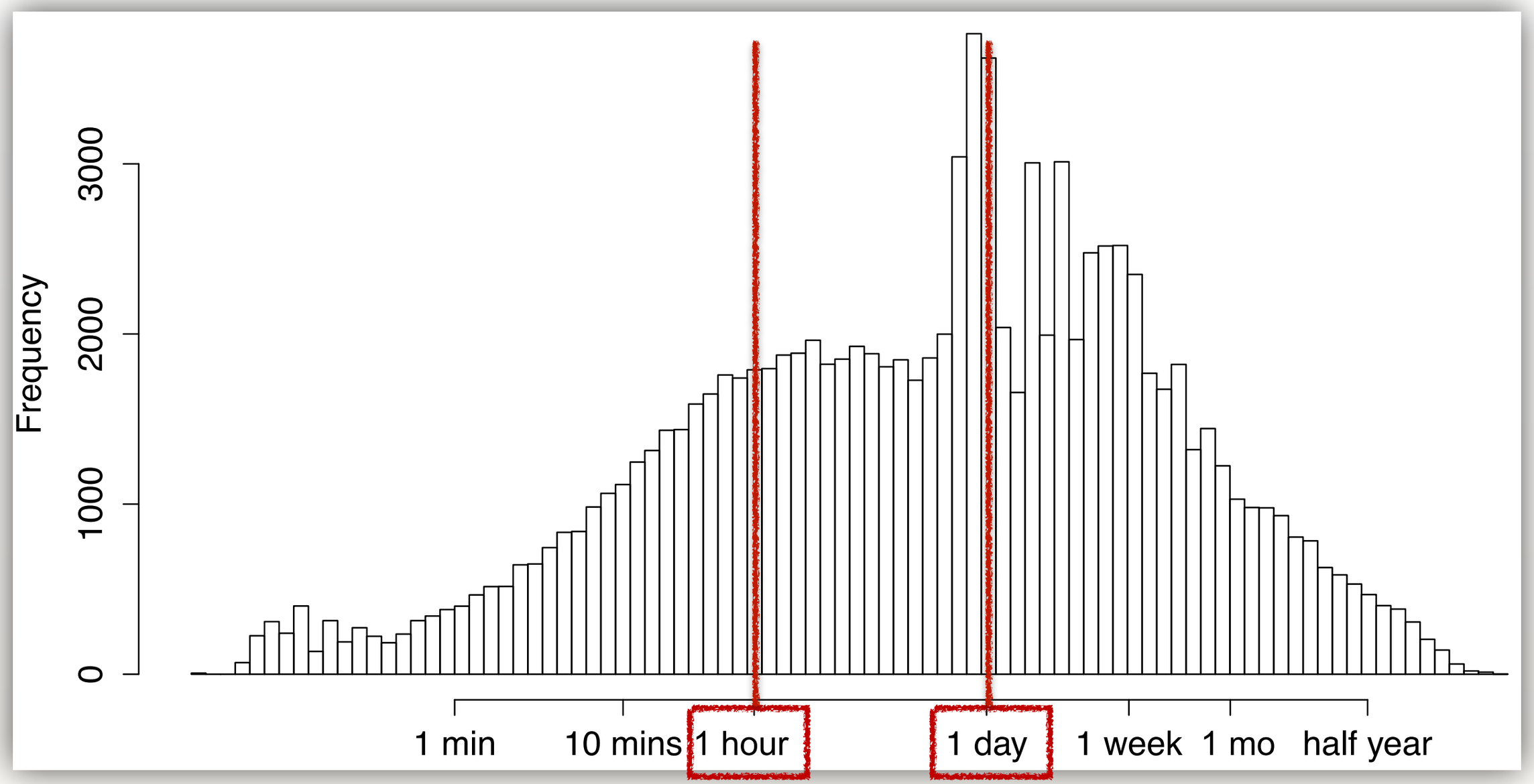
Remove Active Support deprecations activesupport needs feedback
#27035 opened 16 hours ago by pixeltrix — 2

EXAMPLE 1: PULL REQUEST EVALUATION TIME

HOW TO PREDICT?



EXAMPLE 1: PULL REQUEST EVALUATION TIME

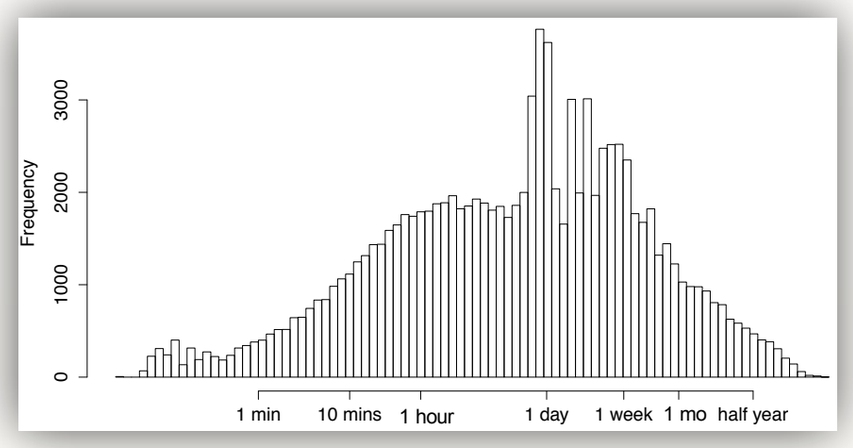


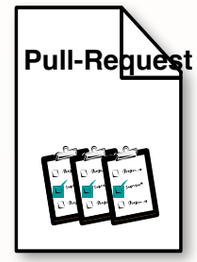
Hypothesis:

Technical attributes dominate: Size, Complexity, Having Tests

EXAMPLE 1: PULL REQUEST EVALUATION TIME

FACTORS





Size

- n_additions
- n_commits

Review

- n_comments

Experience & Social Connections

- merge_rate
- connection_strength
- n_followers

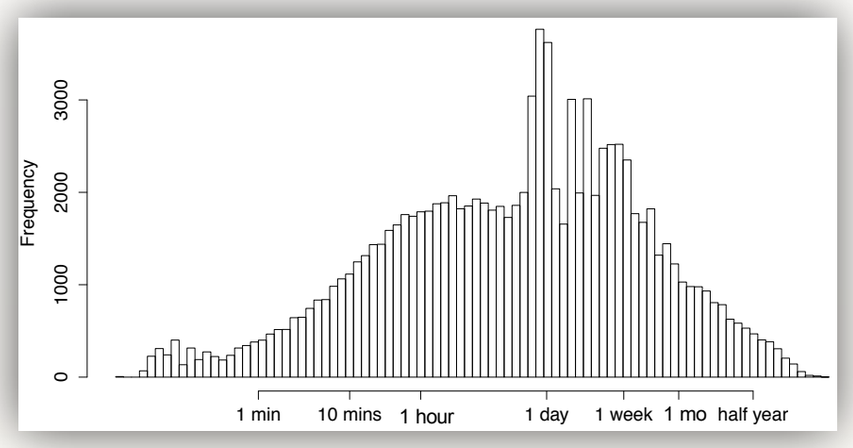
[Gousios et al, ICSE'14, ICSE'15]
[Tsay et al, ICSE'14, FSE'14]

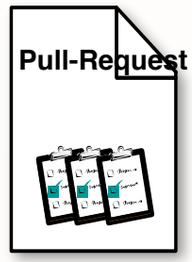
EXAMPLE 1: PULL REQUEST EVALUATION TIME

MODELS

M1: Previously-identified factors

✓ $R^2 = 36.2\%$



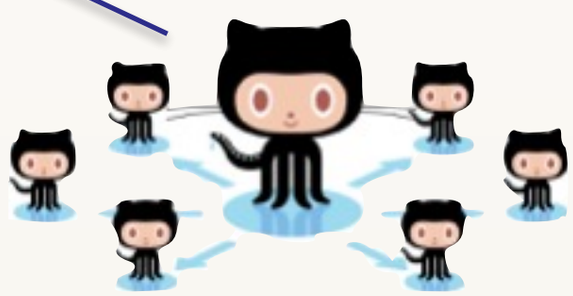


Size

- n_additions
- n_commits

Review

- n_comments



Experience & Social Connections

- merge_rate
- connection_strength
- n_followers

[Gousios et al, ICSE'14, ICSE'15]
[Tsay et al, ICSE'14, FSE'14]

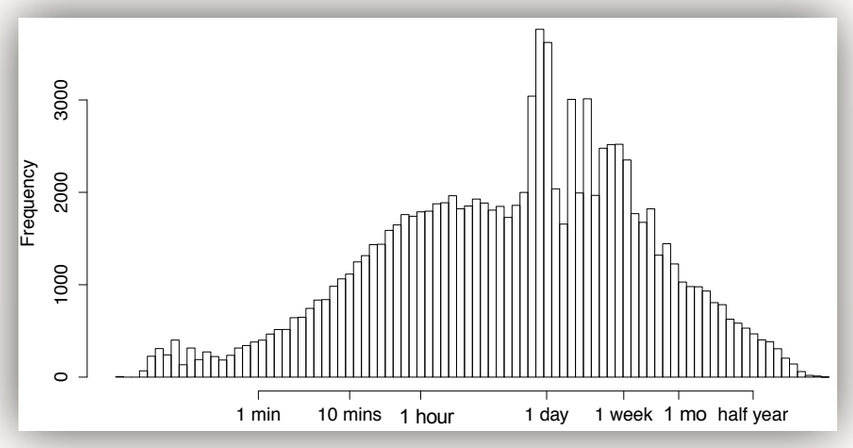
EXAMPLE 1: PULL REQUEST EVALUATION TIME

MODELS

M2: M1 + process-related factors + continuous integration

Title & description

- n_tokens

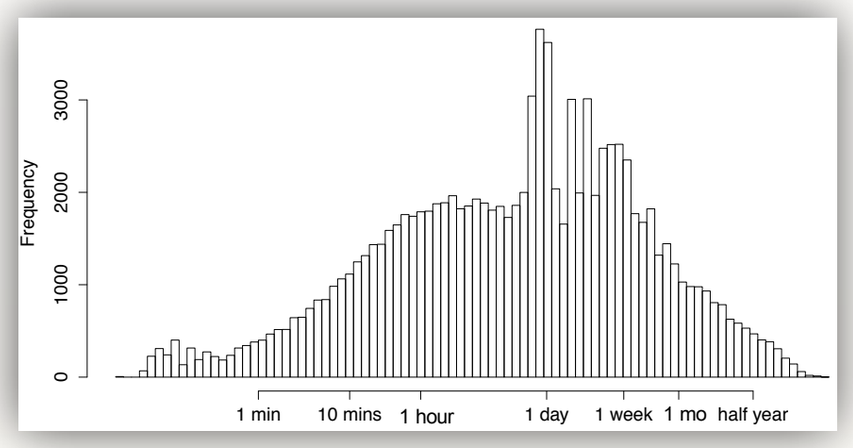


[MSR 2015]

EXAMPLE 1: PULL REQUEST EVALUATION TIME

MODELS

M2: M1 + process-related factors + continuous integration



Title & description

- n_tokens

Management

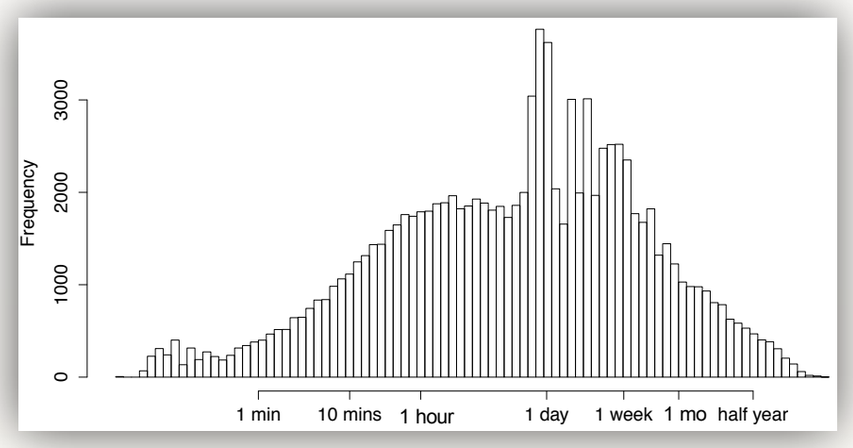
- workload
- availability

[MSR 2015]

EXAMPLE 1: PULL REQUEST EVALUATION TIME

MODELS

M2: M1 + process-related factors + continuous integration



Title & description

- n_tokens

Priority

- time_to_first _response

Continuous Integration

- response time

Management

- workload
- availability

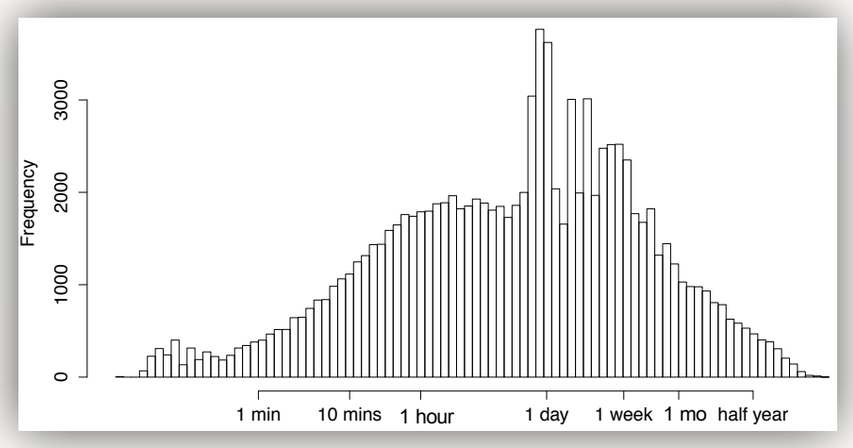
[MSR 2015]

EXAMPLE 1: PULL REQUEST EVALUATION TIME

MODELS

M2: M1 + process-related factors + continuous integration

✓ $R^2 = 58.7\%$



Title & description
- n_tokens

Priority
- time_to_first _response

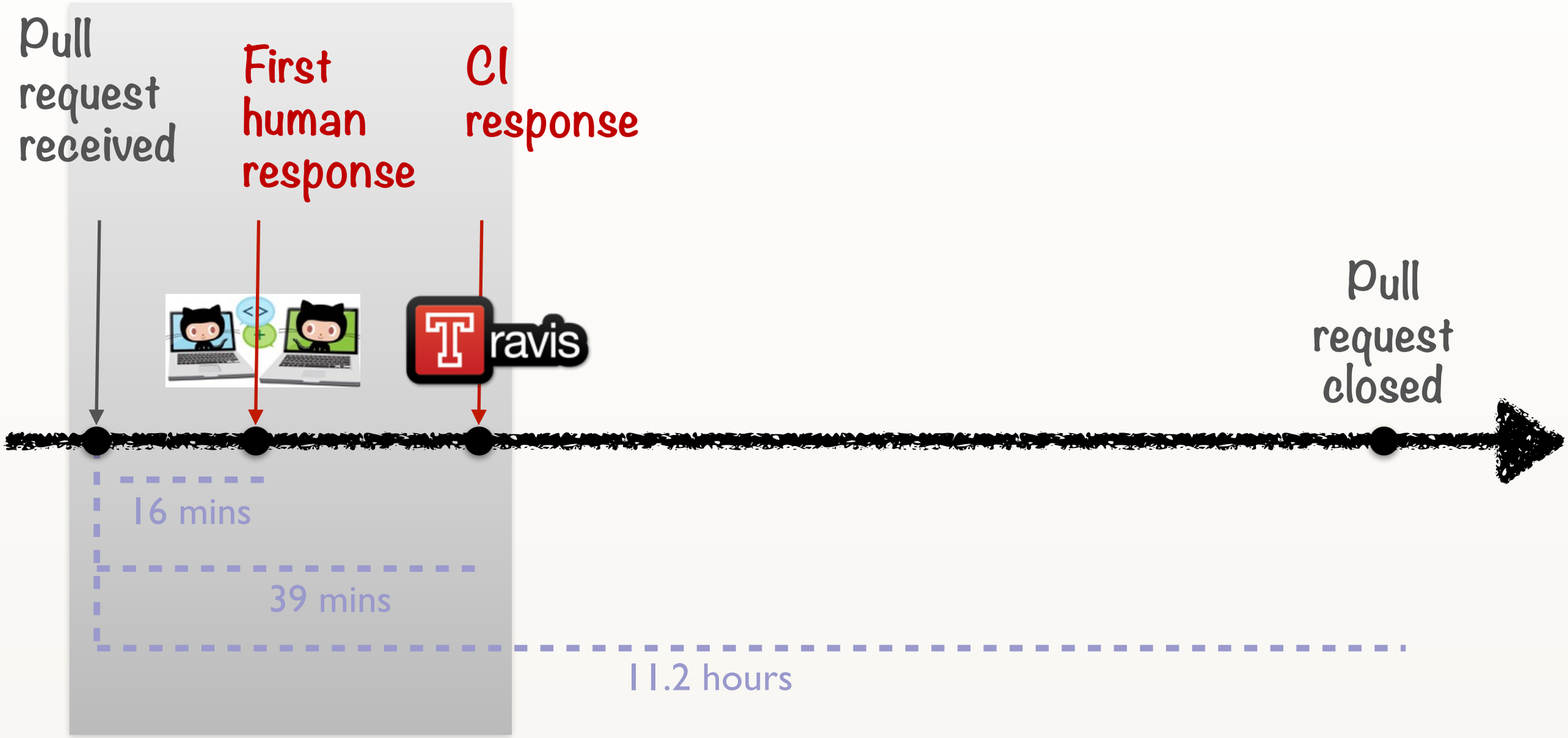
Continuous Integration
- response time

Management
- workload
- availability

Social tagging
- @mention
- #issue

[MSR 2015]

EXAMPLE 1: PULL REQUEST EVALUATION TIME IS PREDICTABLE

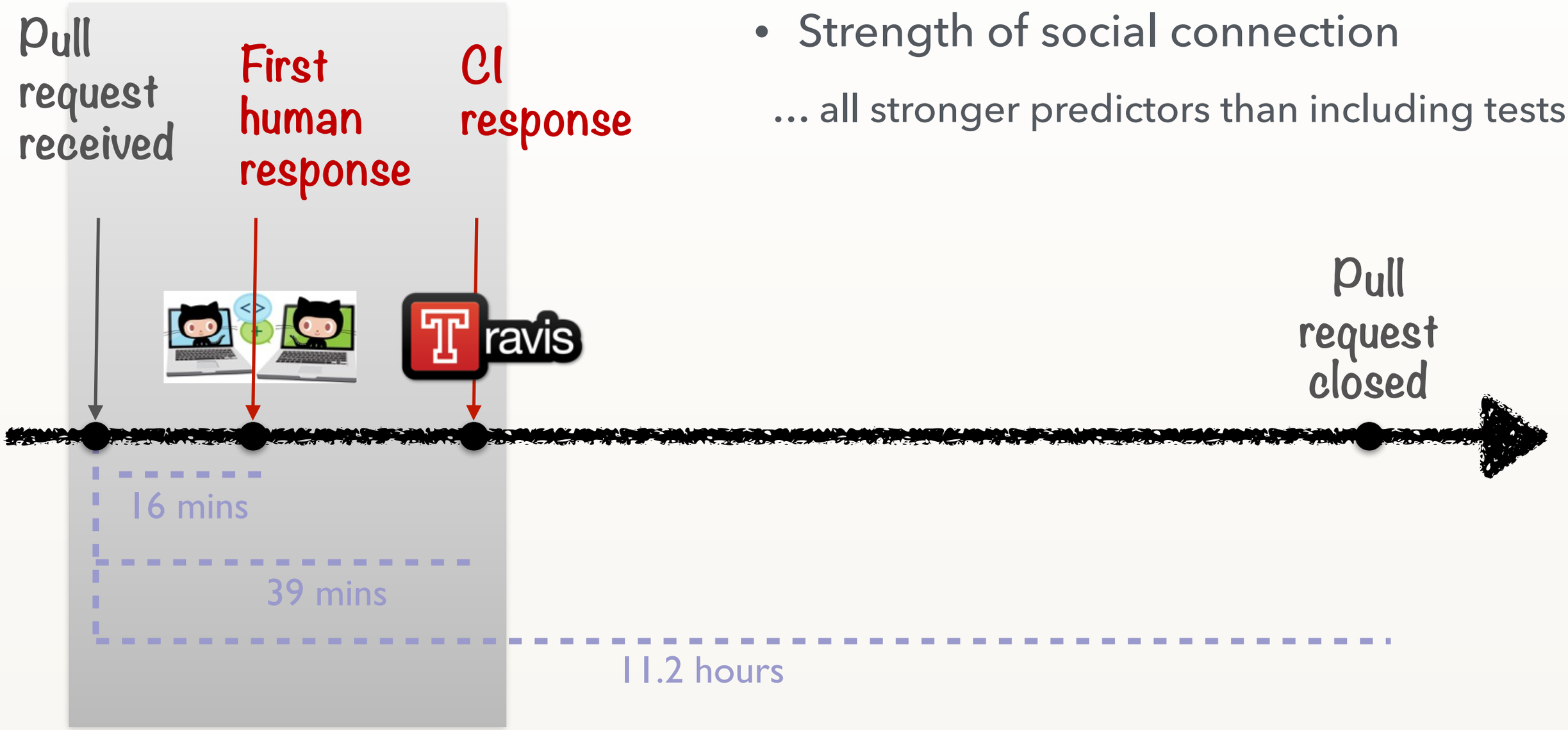


EXAMPLE 1: PULL REQUEST EVALUATION TIME

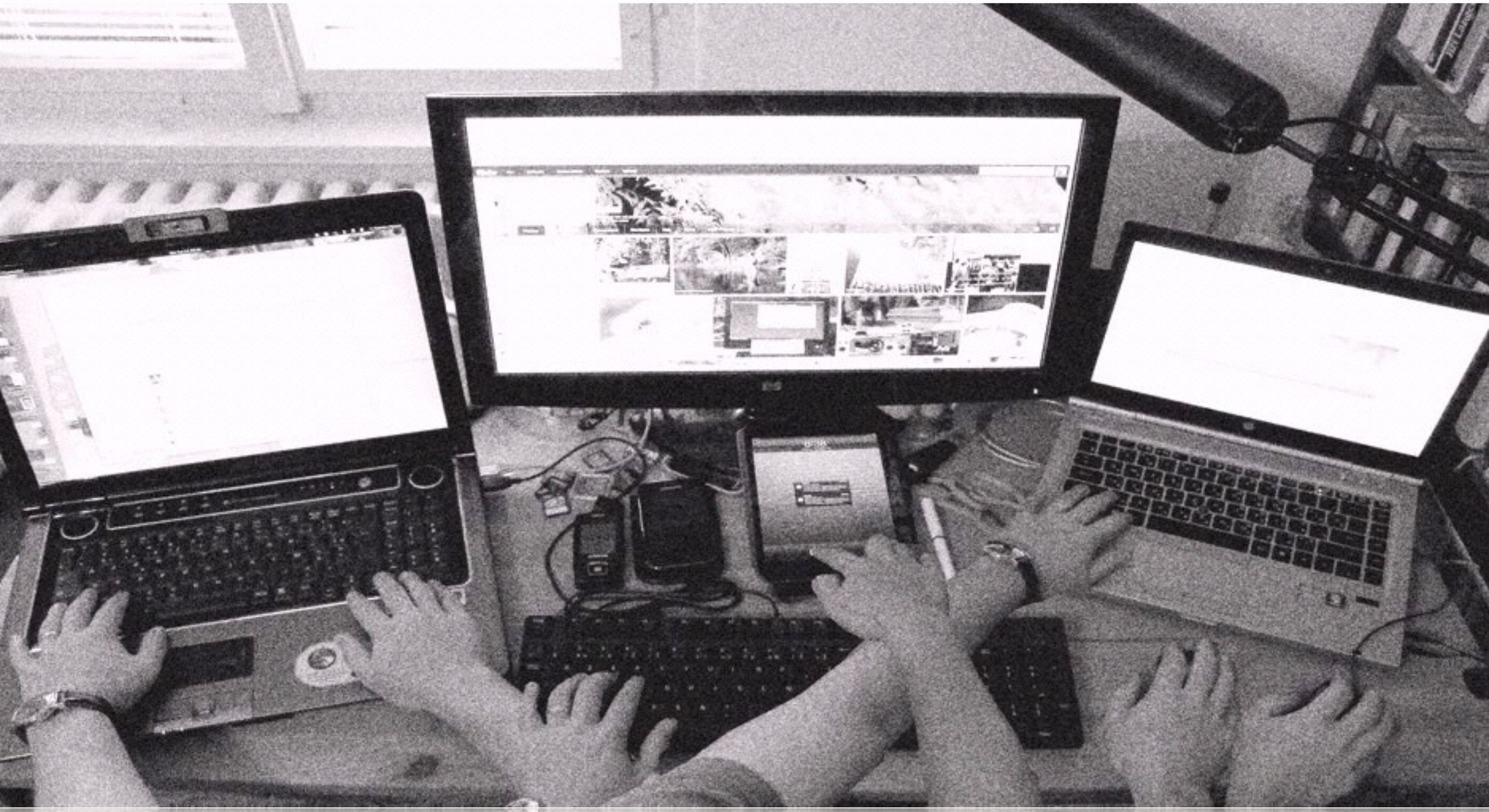
SOCIAL CODING!

- Submitter is core developer
- Number of followers
- Strength of social connection

... all stronger predictors than including tests

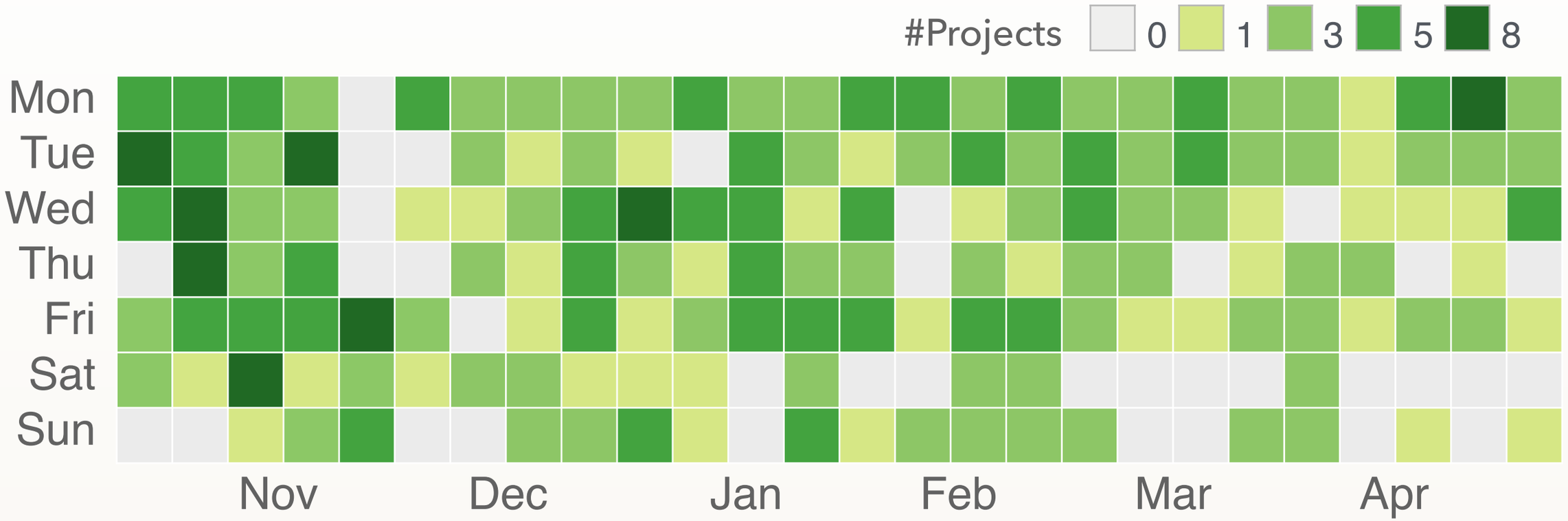


EXAMPLE 2: MULTITASKING



SOFTWARE DEVELOPERS MULTITASK TOO

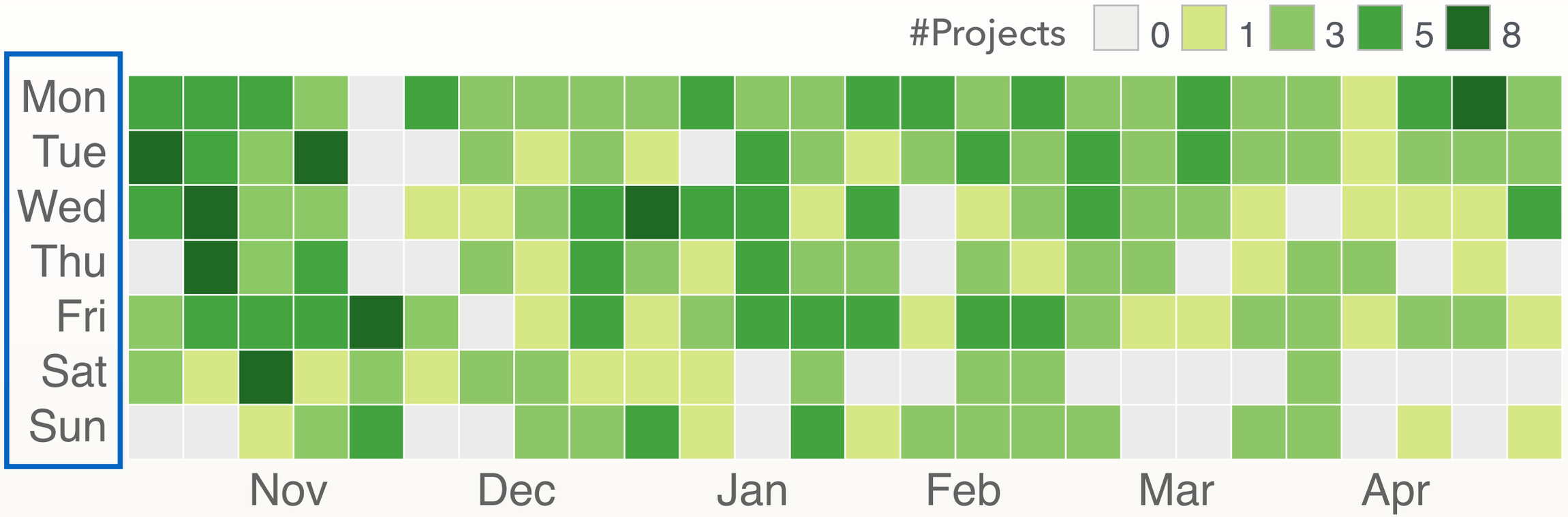
EXAMPLE: GitHub developer (25 Nov 2013 – 18 May 2014)



SOFTWARE DEVELOPERS MULTITASK TOO

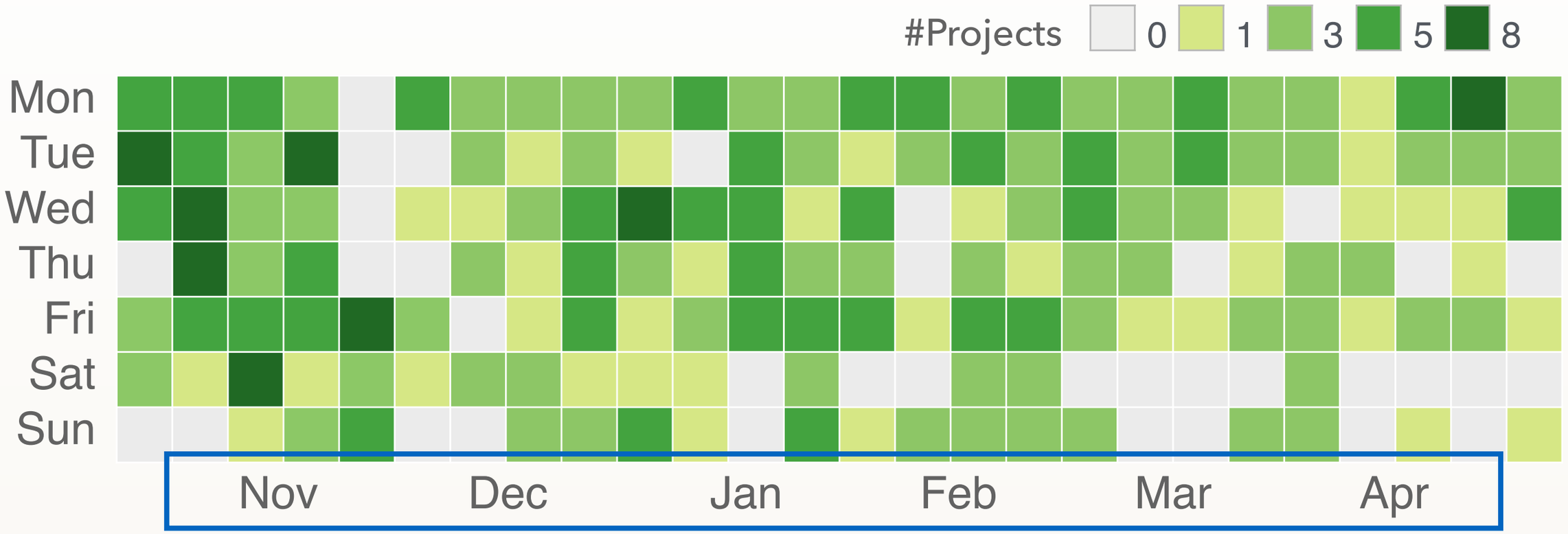
github SOCIAL CODING

EXAMPLE: GitHub developer (25 Nov 2013 – 18 May 2014)



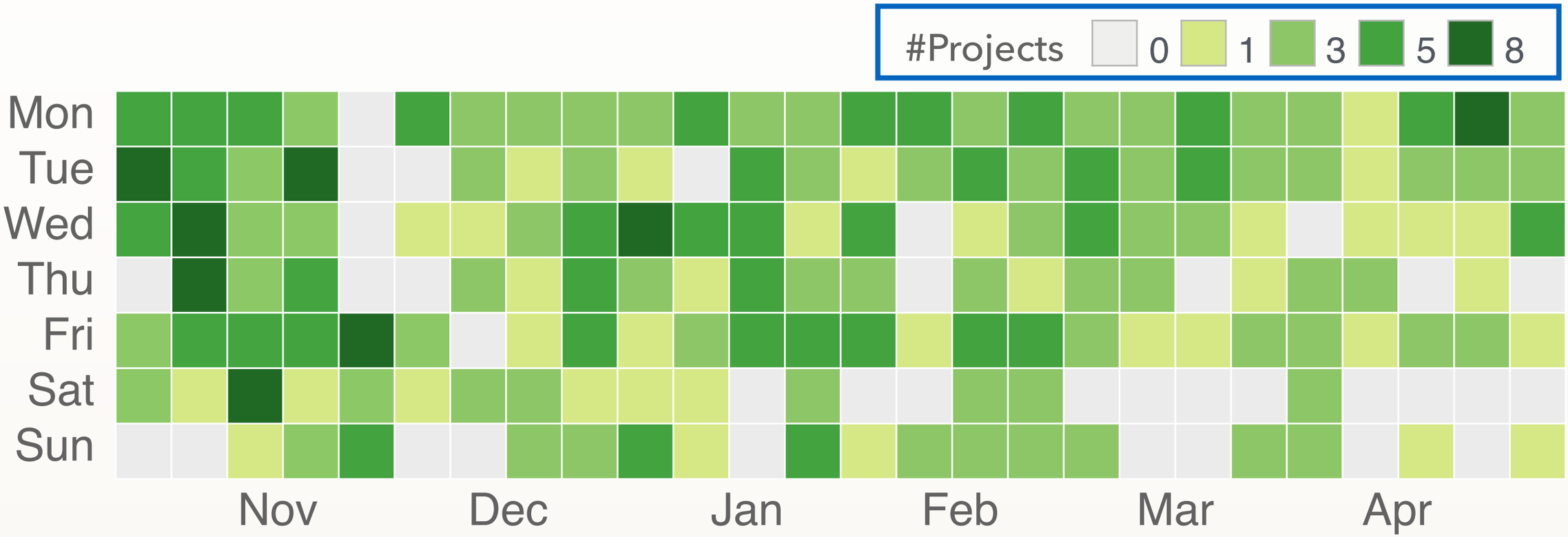
SOFTWARE DEVELOPERS MULTITASK TOO

EXAMPLE: GitHub developer (25 Nov 2013 – 18 May 2014)



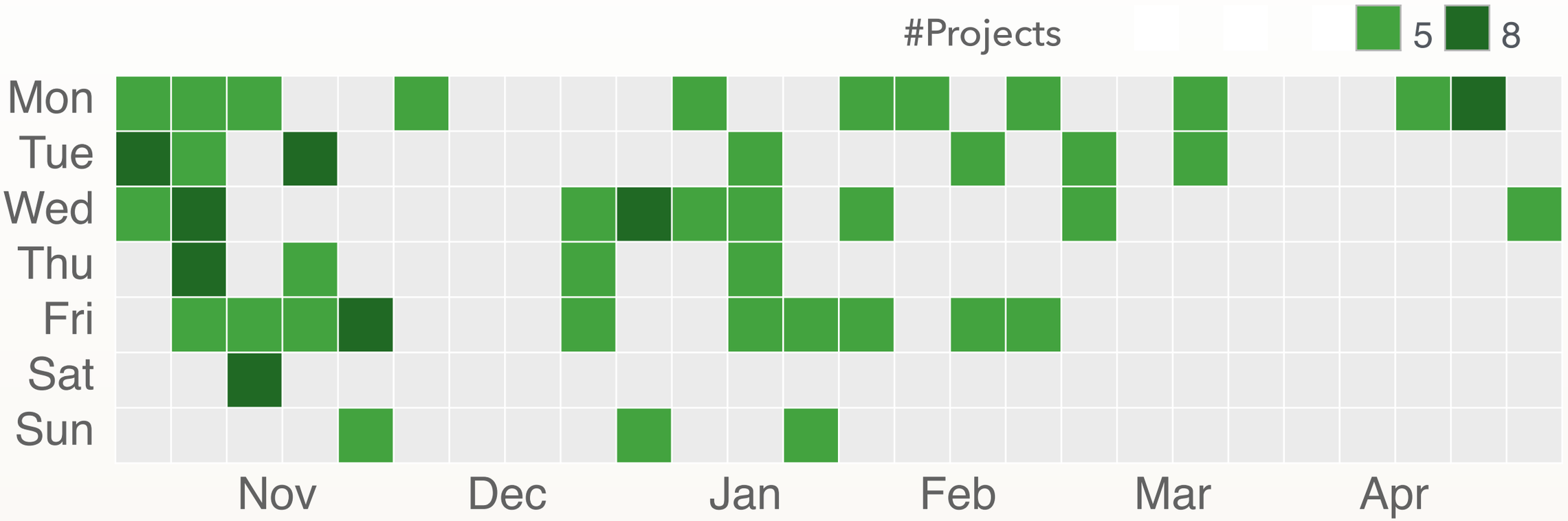
SOFTWARE DEVELOPERS MULTITASK TOO

EXAMPLE: GitHub developer (25 Nov 2013 – 18 May 2014)



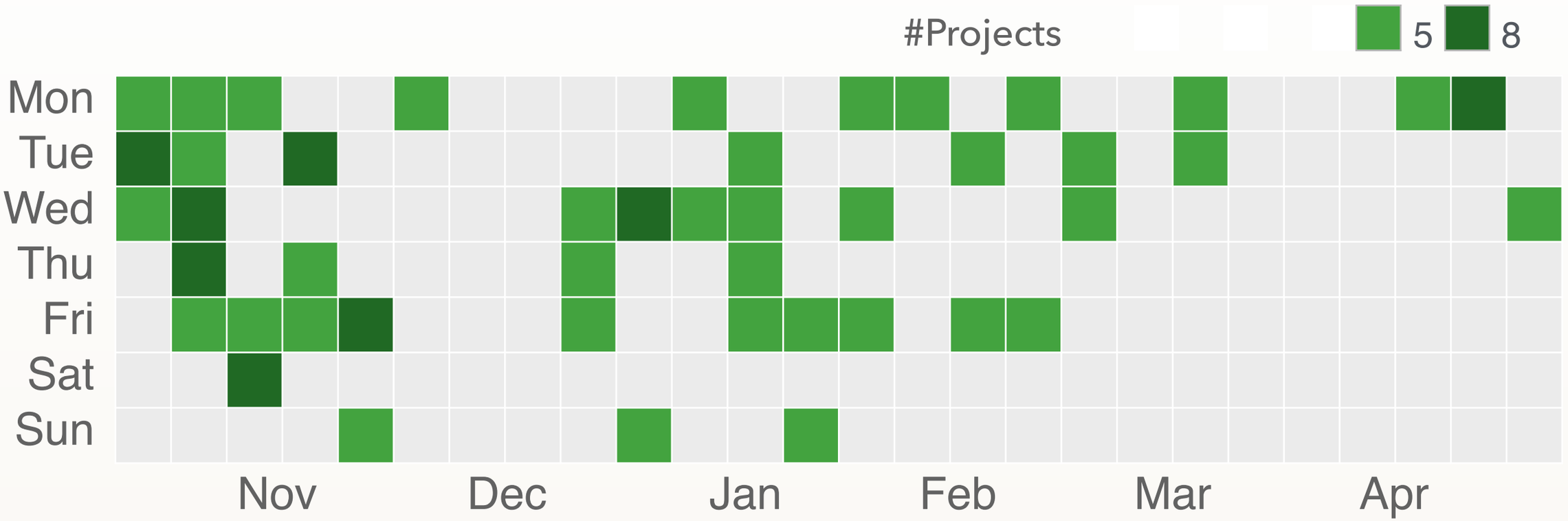
SOFTWARE DEVELOPERS MULTITASK TOO

EXAMPLE: GitHub developer (25 Nov 2013 – 18 May 2014)



SOFTWARE DEVELOPERS MULTITASK TOO

EXAMPLE: GitHub developer (25 Nov 2013 – 18 May 2014)

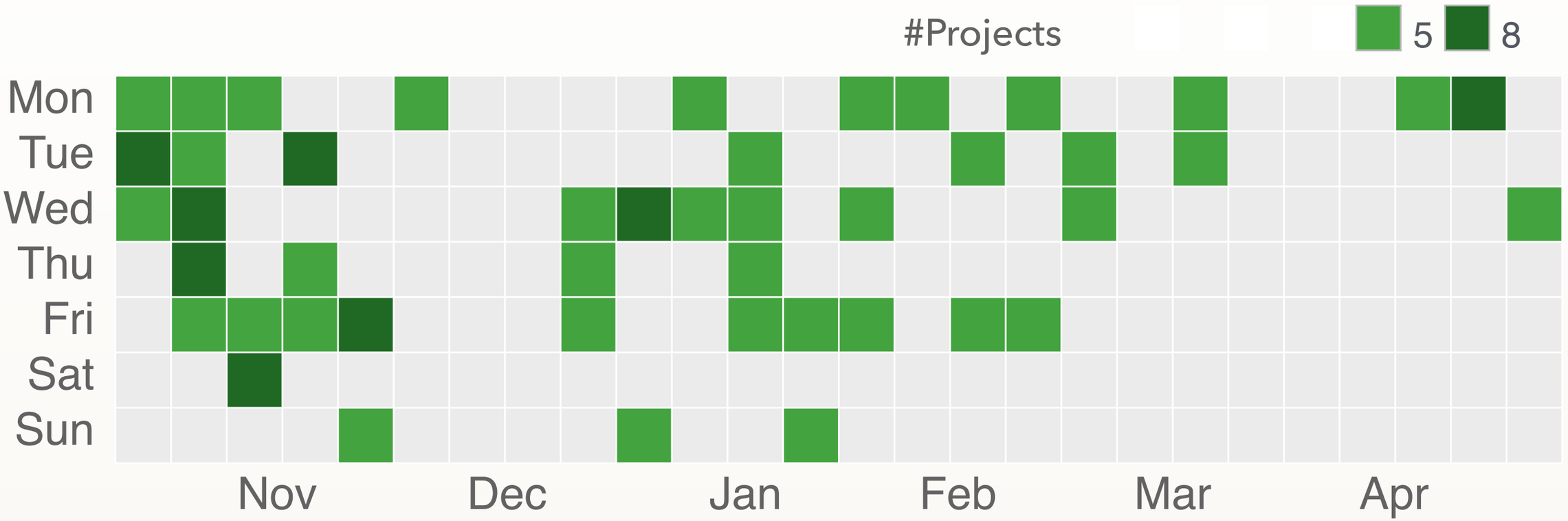


WHY?

- Request from other dev's / management

SOFTWARE DEVELOPERS MULTITASK TOO

EXAMPLE: GitHub developer (25 Nov 2013 – 18 May 2014)

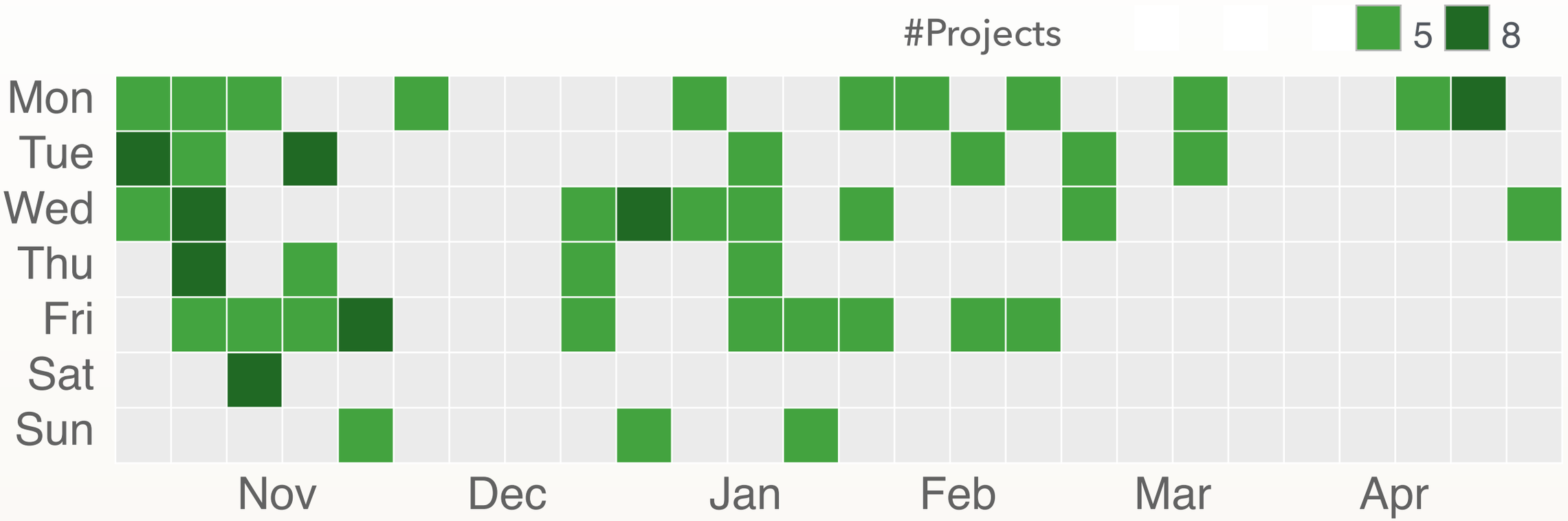


WHY?

- Request from other dev's / management
- Dependencies

SOFTWARE DEVELOPERS MULTITASK TOO

EXAMPLE: GitHub developer (25 Nov 2013 – 18 May 2014)

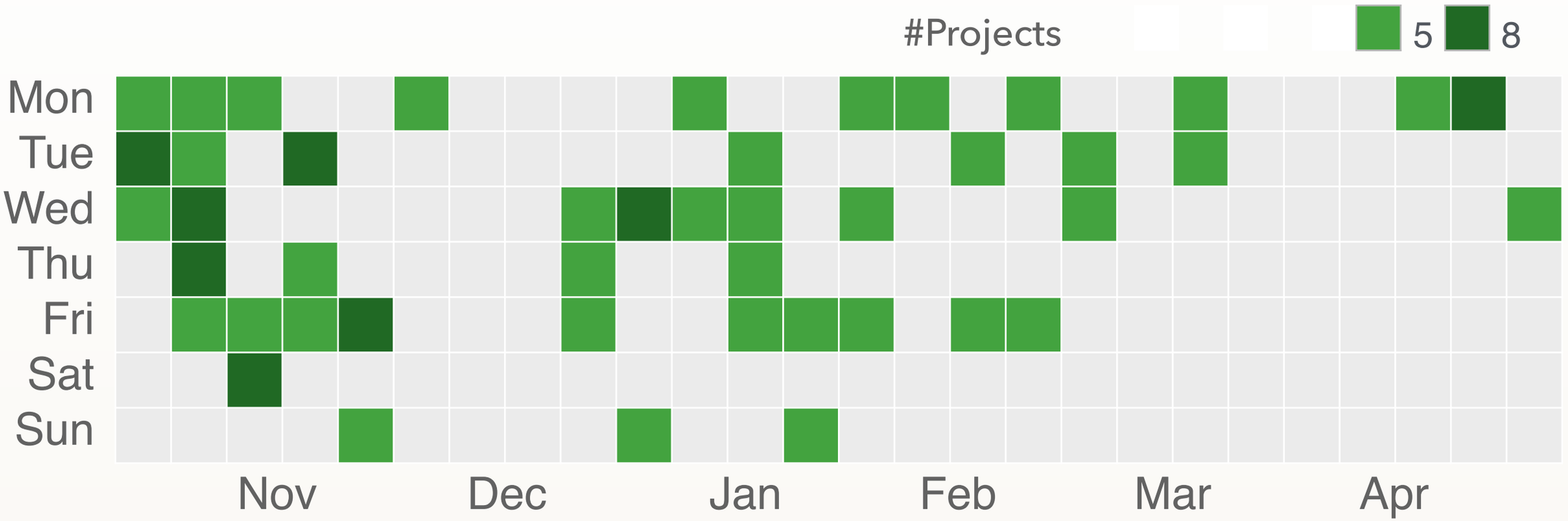


WHY?

- Request from other dev's / management
- Dependencies
- Being "stuck"
- Downtime

SOFTWARE DEVELOPERS MULTITASK TOO

EXAMPLE: GitHub developer (25 Nov 2013 – 18 May 2014)

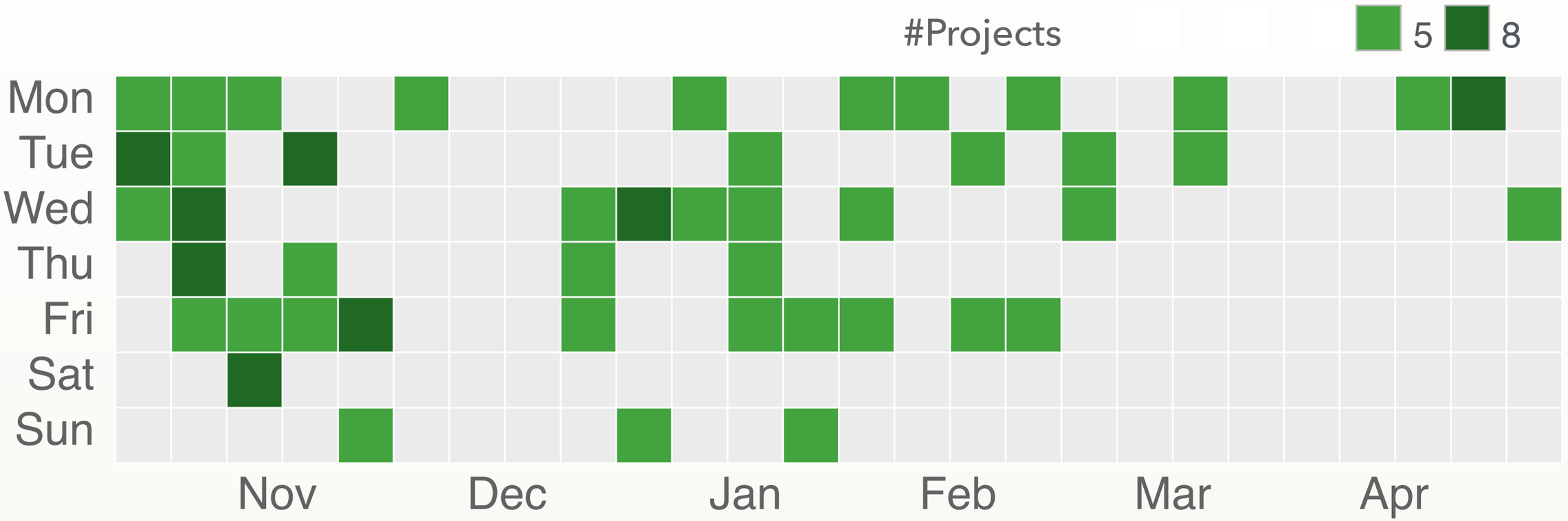


WHY?

- Request from other dev's / management
- Dependencies
- Being "stuck"
- Downtime
- Personal interest

SOFTWARE DEVELOPERS MULTITASK TOO

EXAMPLE: GitHub developer (25 Nov 2013 – 18 May 2014)



WHY?

- Request from other dev's / management
- Dependencies
- Being "stuck"
- Downtime
- Personal interest
- Signaling

THEORY: HOW DOES MULTITASKING AFFECT PERFORMANCE?

PROS

CONS

THEORY: HOW DOES MULTITASKING AFFECT PERFORMANCE?

PROS

- **Fill downtime**

Switch focus between projects to utilize time more efficiently

(Adler and Benbunan-Fich, 2012)

CONS

THEORY: HOW DOES MULTITASKING AFFECT PERFORMANCE?

PROS

Fill downtime

Switch focus between projects to utilize time more efficiently

(Adler and Benbunan-Fich, 2012)

Cross-fertilisation

Easier to work on other projects if knowledge is transferrable

(Lindbeck and Snower, 2000)

CONS

THEORY: HOW DOES MULTITASKING AFFECT PERFORMANCE?

PROS

- **Fill downtime**

  Switch focus between projects to utilize time more efficiently

  (Adler and Benbunan-Fich, 2012)

- **Cross-fertilisation**

  Easier to work on other projects if knowledge is transferrable

  (Lindbeck and Snower, 2000)

CONS

- **Cognitive switching cost**

  Depends on interruption duration, complexity, moment

  (Altmann and Trafton, 2002)
  (Borst, Taatgen, van Rijn, 2015)

THEORY: HOW DOES MULTITASKING AFFECT PERFORMANCE?

PROS

- **Fill downtime**

  Switch focus between projects to utilize time more efficiently

  (Adler and Benbunan-Fich, 2012)

- **Cross-fertilisation**

  Easier to work on other projects if knowledge is transferrable

  (Lindbeck and Snower, 2000)

CONS

- **Cognitive switching cost**

  Depends on interruption duration, complexity, moment

  (Altmann and Trafton, 2002)
  (Borst, Taatgen, van Rijn, 2015)

- **"Project overload"**

  Mental congestion when too much multitasking

  (Zika-Viktorsson, Sundstrom, Engwall, 2006)

THEORY: HOW DOES MULTITASKING AFFECT PERFORMANCE?

PROS

- **Fill downtime**

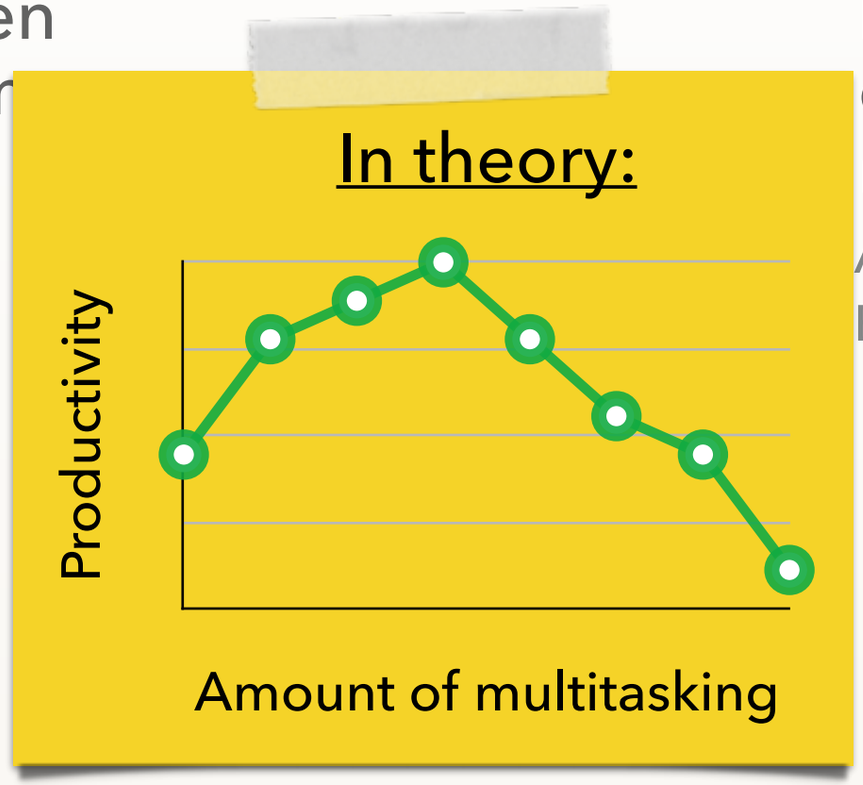
  Switch focus between projects to utilize tim more efficiently

  (Adler and Benbunan-Fich, 2012)

- **Cross-fertilisation**

  Easier to work on other projects if knowledge is transferrable

  (Lindbeck and Snower, 2000)



CONS

- **Cognitive switching cost**

  Depends on interruption duration, complexity, moment

  Altmann and Trafton, 2002)
  Borst, Taatgen, van Rijn, 2015)

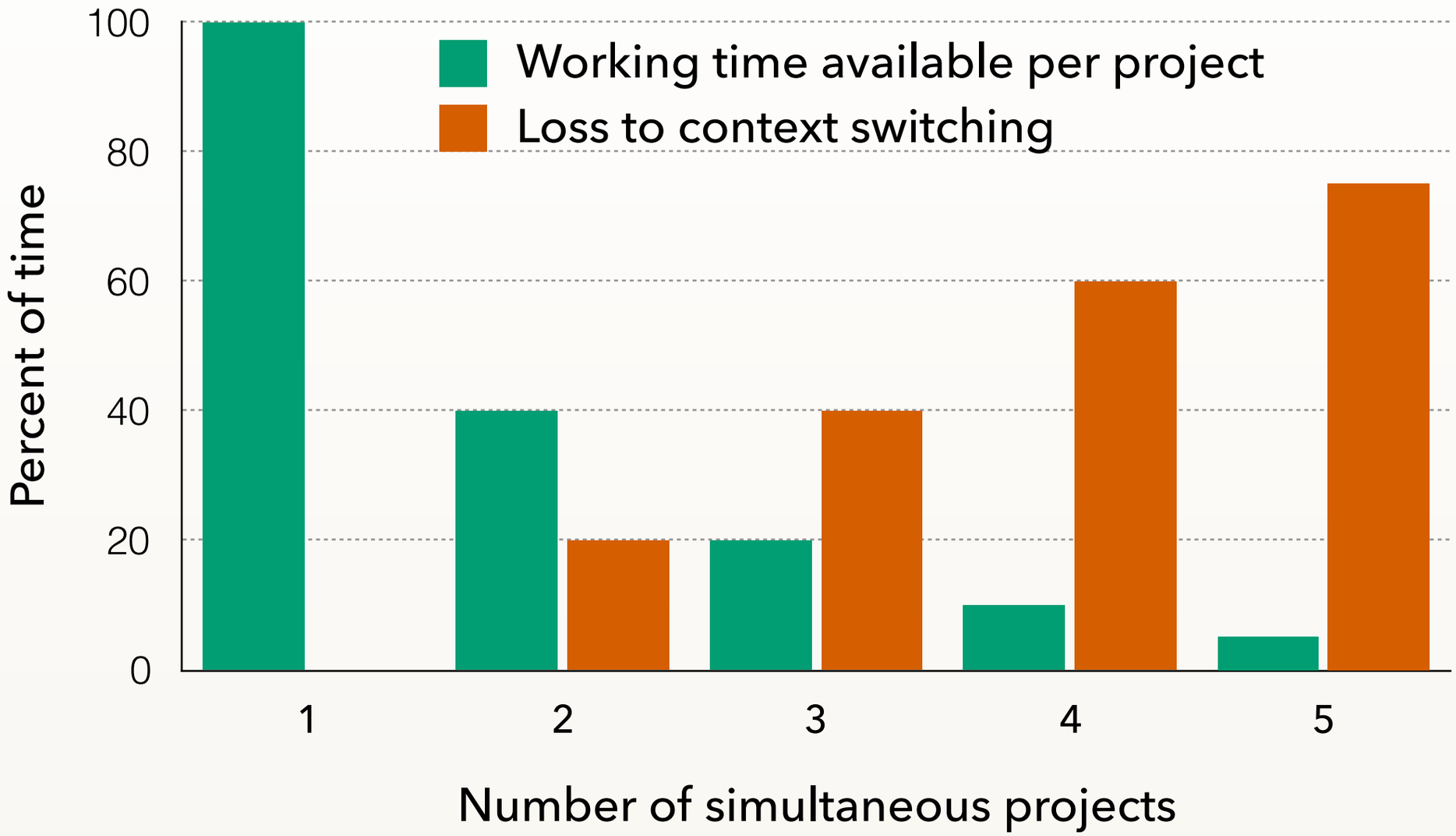
- **"Project overload"**

  Mental congestion when too much multitasking

  (Zika-Viktorsson, Sundstrom, Engwall, 2006)

HARDLY ANY EMPIRICAL EVIDENCE

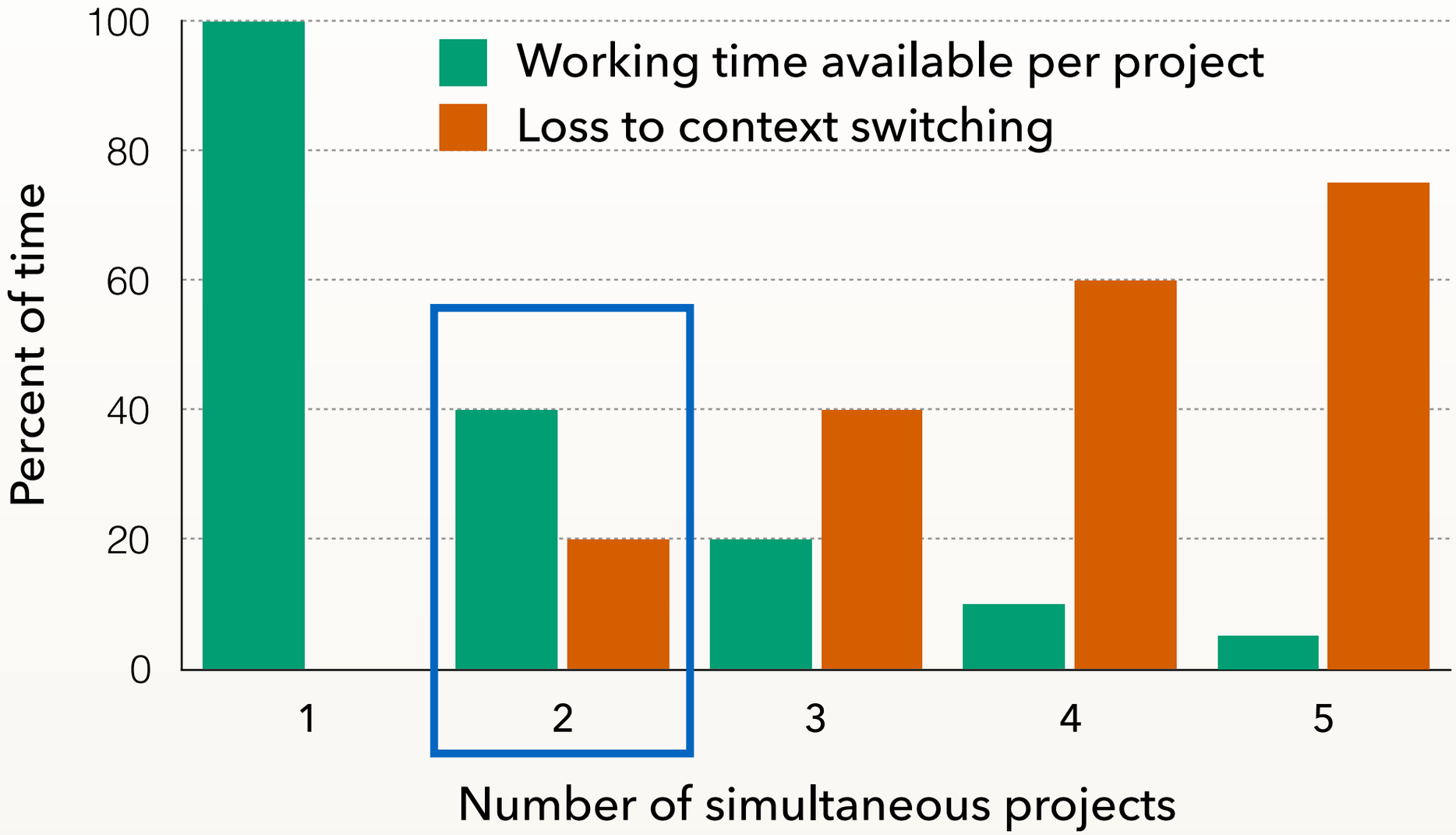
Rule of thumb (Weinberg, 1992) - not based on data



From: http://blog.codinghorror.com/the-multi-tasking-myth/

HARDLY ANY EMPIRICAL EVIDENCE

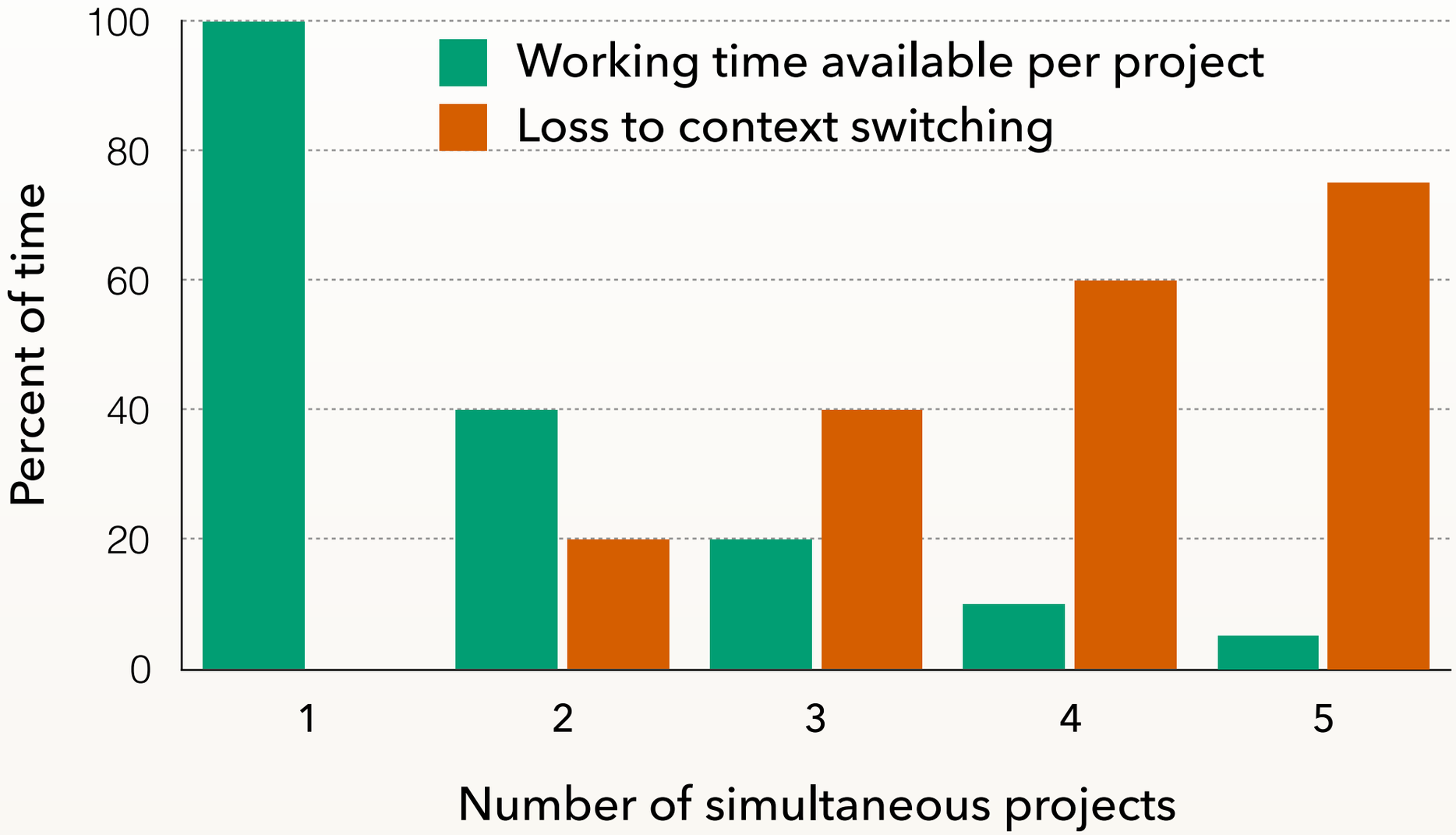
Rule of thumb (Weinberg, 1992) - not based on data



From: http://blog.codinghorror.com/the-multi-tasking-myth/

HARDLY ANY EMPIRICAL EVIDENCE

Rule of thumb (Weinberg, 1992) - not based on data



From: http://blog.codinghorror.com/the-multi-tasking-myth/

Recent work:

- Resuming interrupted tasks
  (Parnin and DeLine, 2010)
- Work fragmentation
  (Sanchez, Robbes, and Gonzalez, 2015)

THIS WORK: LARGE-SCALE EMPIRICAL STUDY

WHAT?

Multitasking across projects

? Trends ? Reasons ? Effects ? Limits

HOW?

Sample:

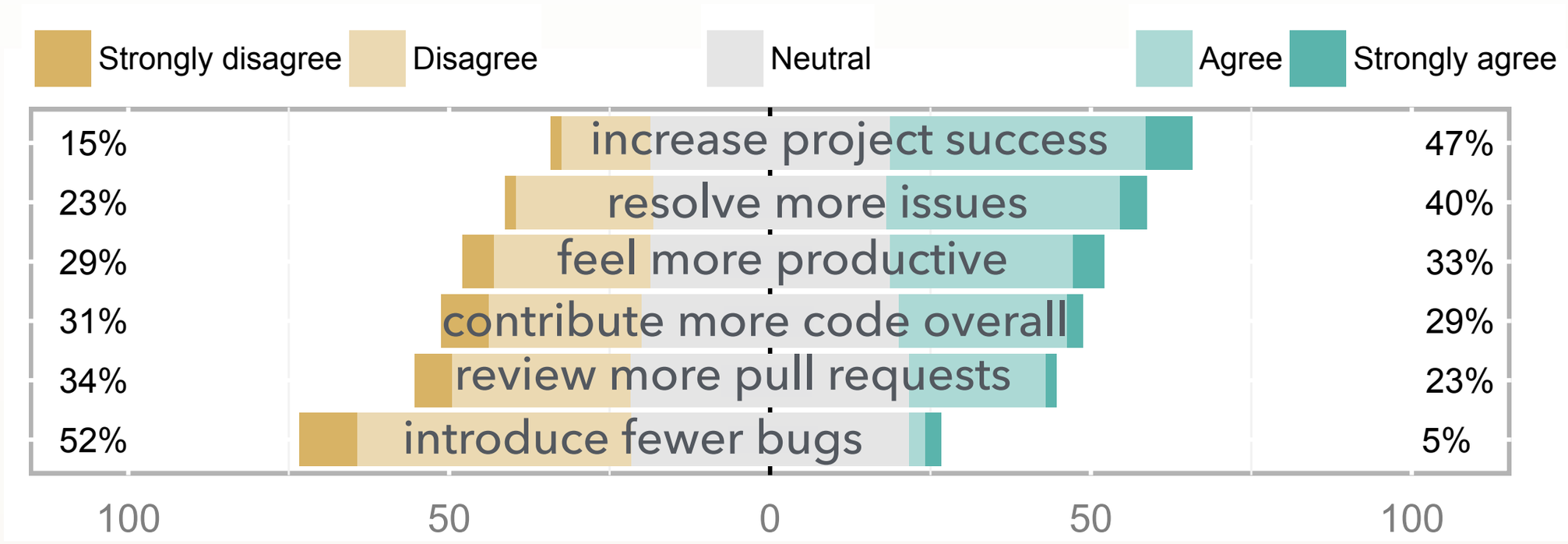
- 1,200 programmers
- 5+ years of activity
- 50,000+ projects total

Data mining + User survey
(15% resp. rate)

EFFECTS: PERCEPTION VS. DATA

PERCEPTION

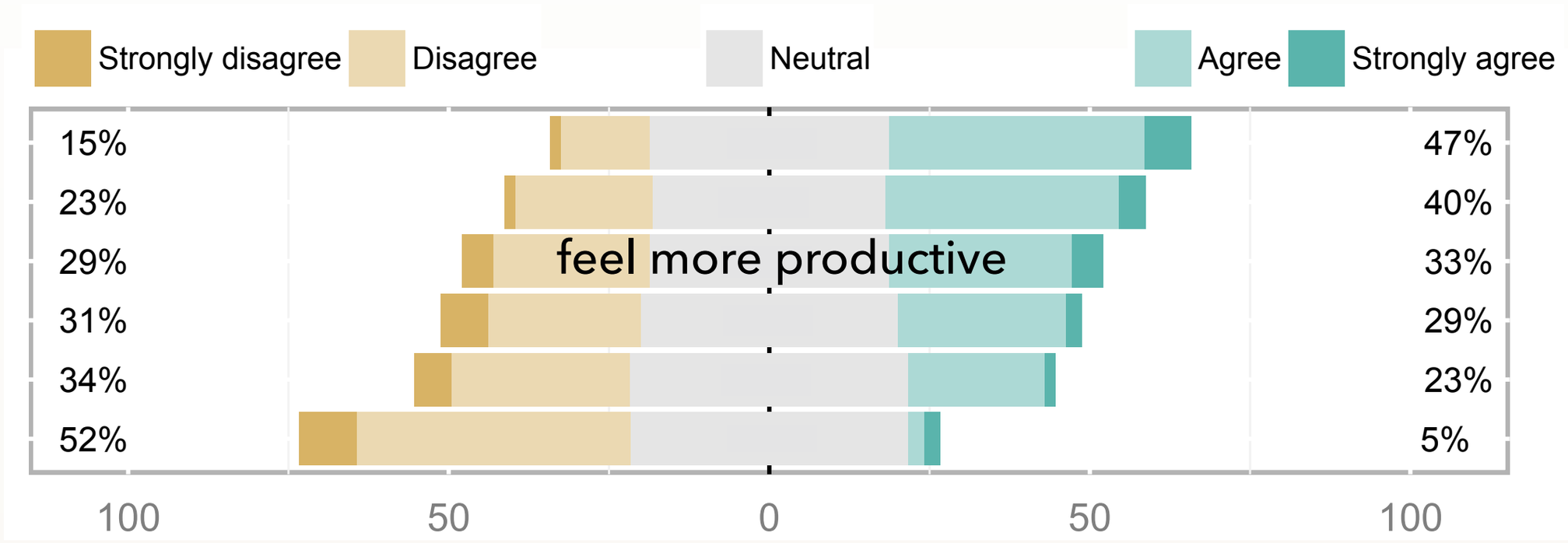
"When contributing to multiple projects in parallel, I:"



EFFECTS: PERCEPTION VS. DATA

PERCEPTION

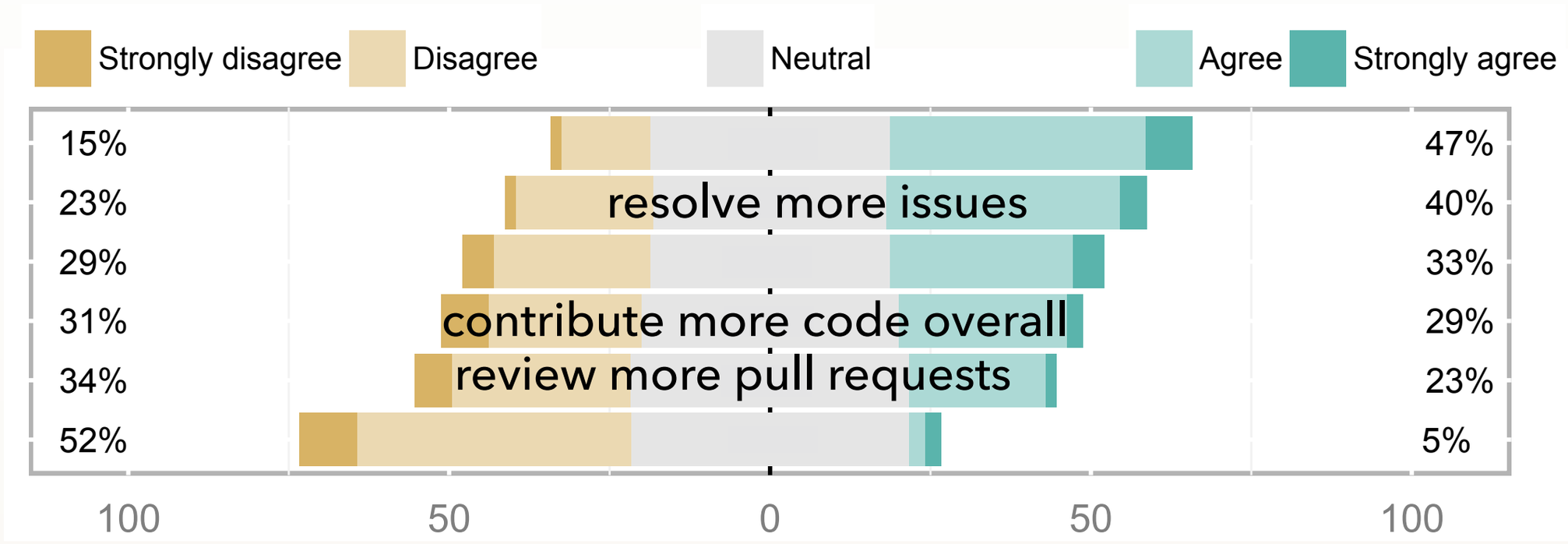
"When contributing to multiple projects in parallel, I:"



EFFECTS: PERCEPTION VS. DATA

PERCEPTION

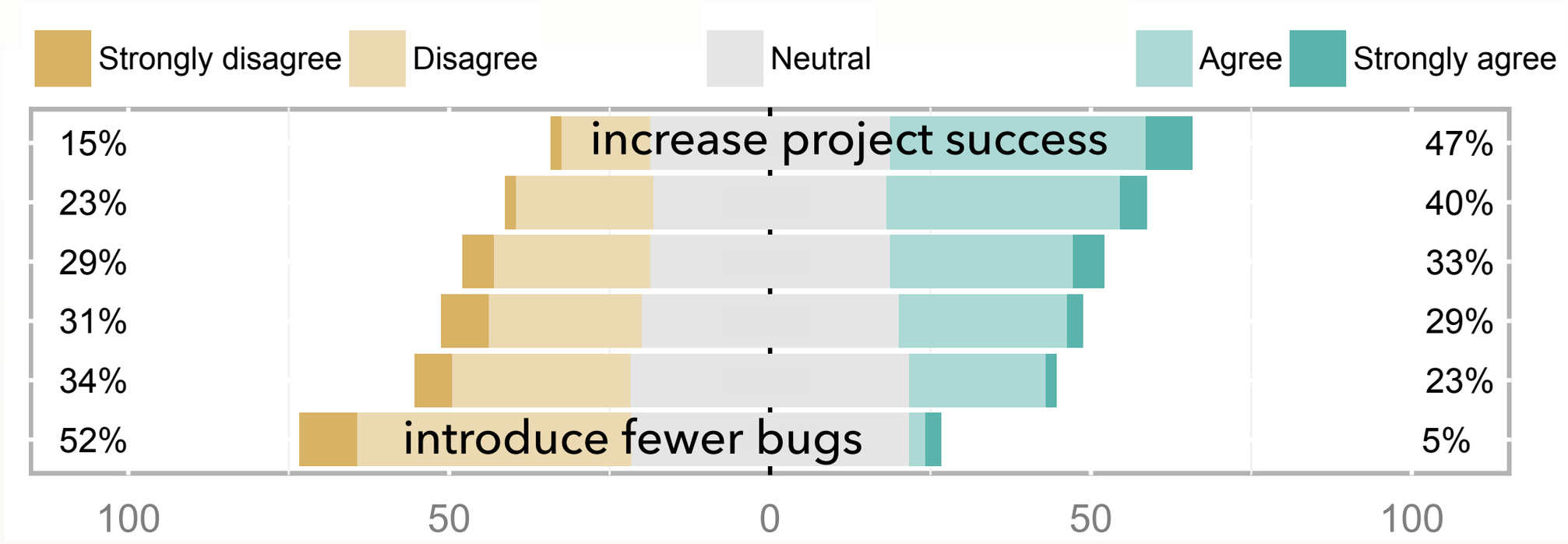
"When contributing to multiple projects in parallel, I:"



EFFECTS: PERCEPTION VS. DATA

PERCEPTION

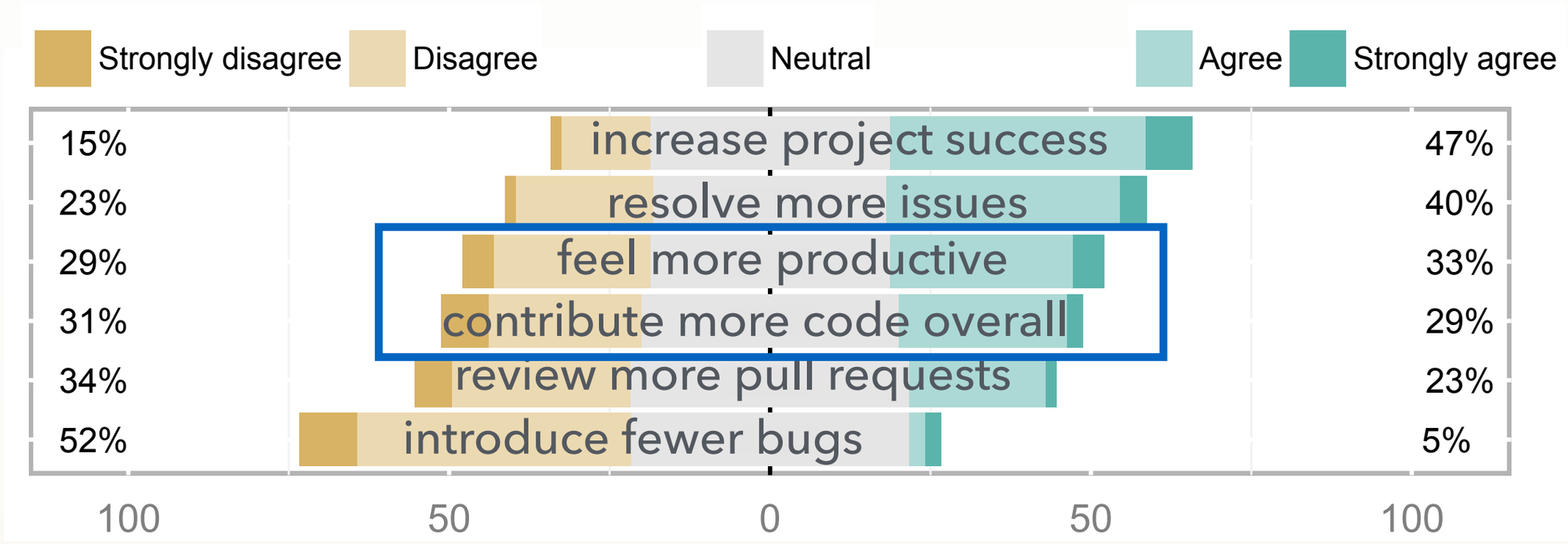
"When contributing to multiple projects in parallel, I:"



EFFECTS: PERCEPTION VS. DATA

PERCEPTION

"When contributing to multiple projects in parallel, I:"



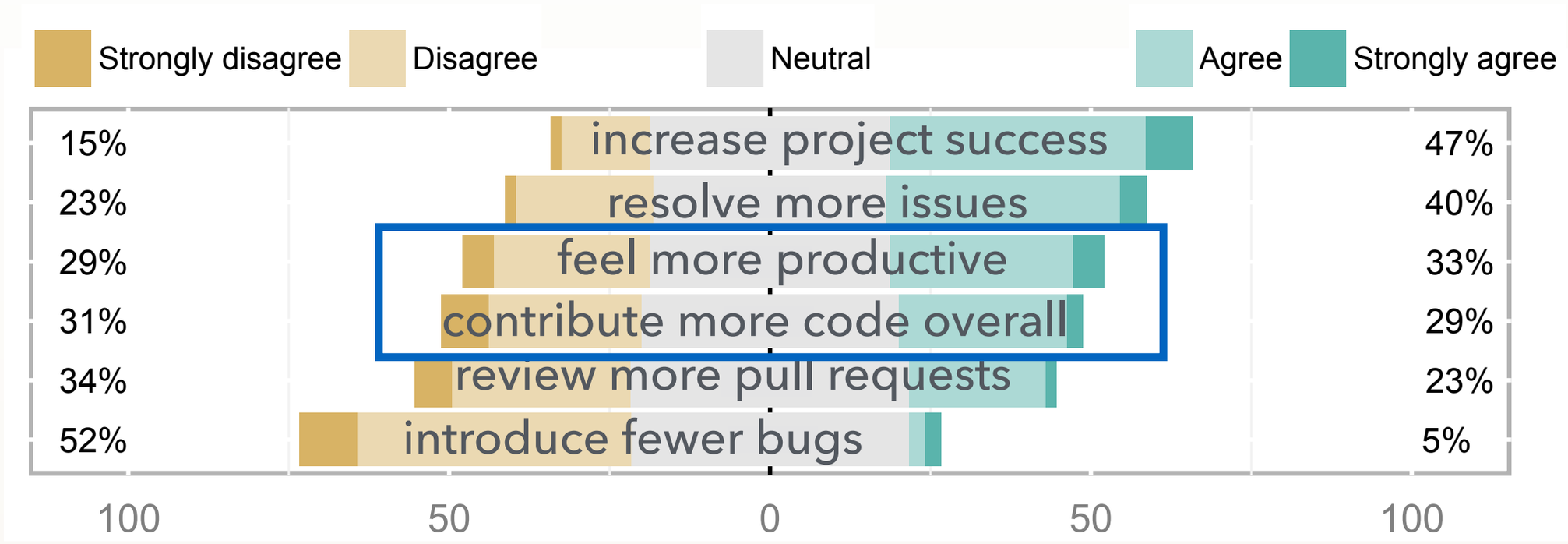
EMPIRICAL DATA

Multitasking vs. code production

EFFECTS: PERCEPTION VS. DATA

PERCEPTION

"When contributing to multiple projects in parallel, I:"



EMPIRICAL DATA

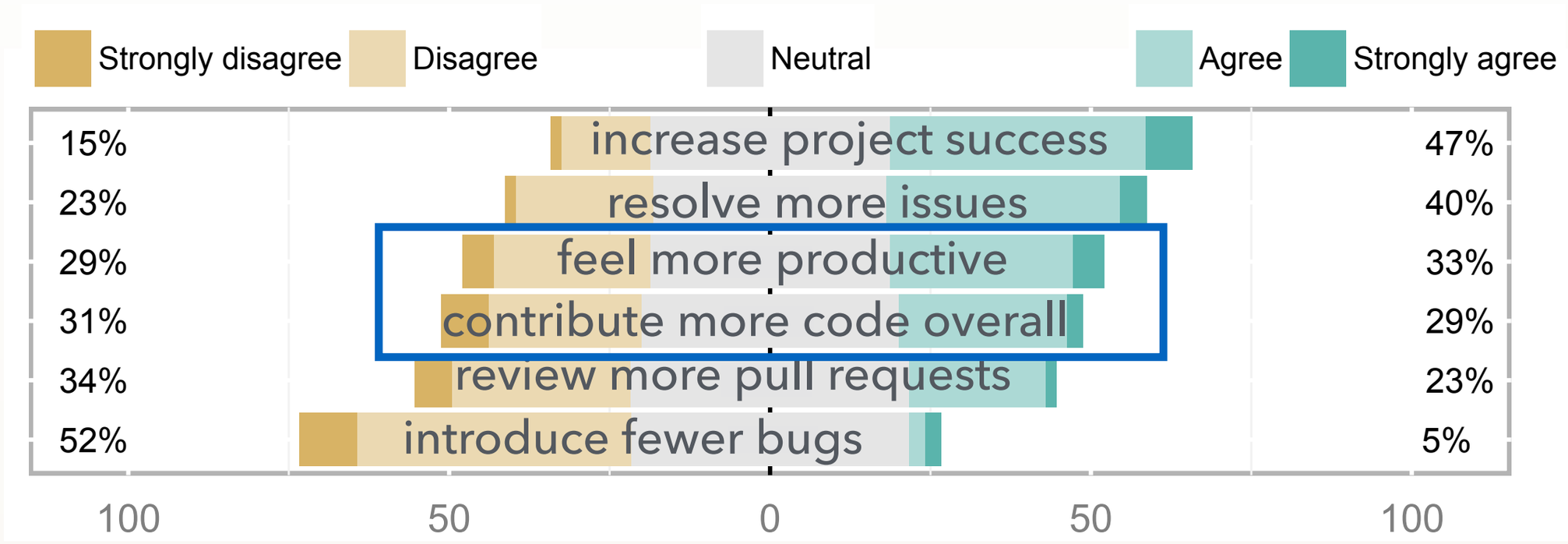
Multitasking vs. code production

Daily multitasking correlates to amount of code produced

EFFECTS: PERCEPTION VS. DATA

PERCEPTION

"When contributing to multiple projects in parallel, I:"



EMPIRICAL DATA

Multitasking vs. code production

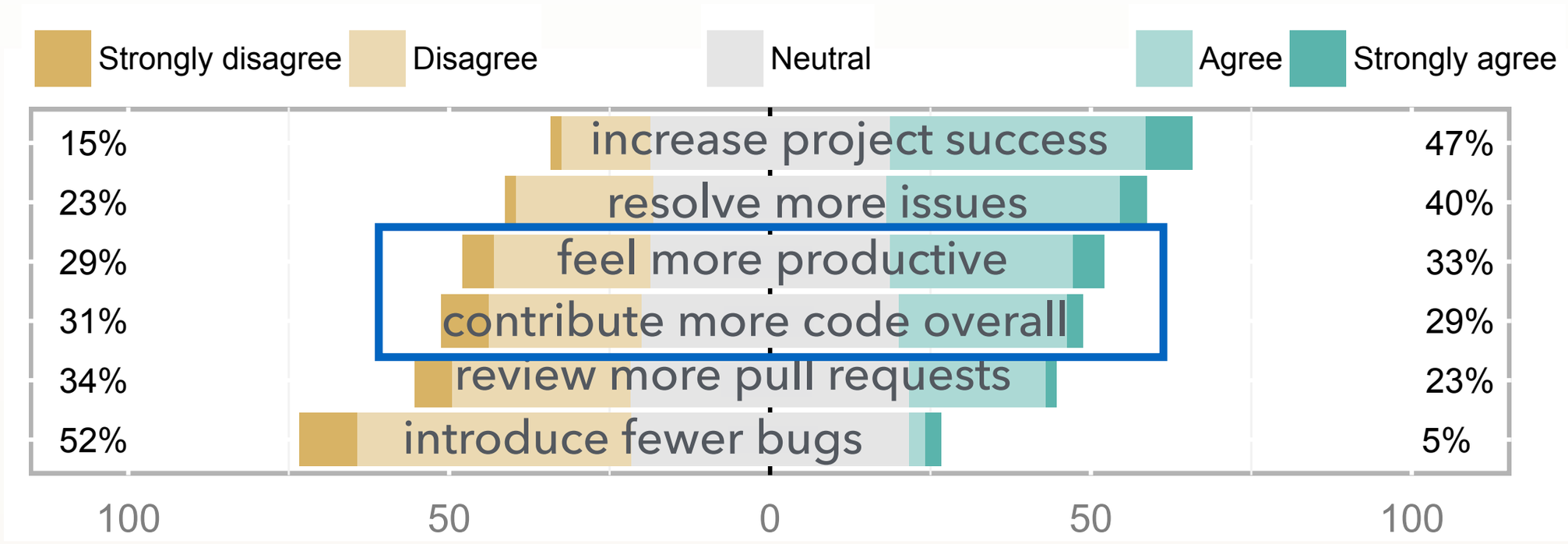
Daily multitasking correlates to amount of code produced

Weekly and day-to-day scheduling of work matters

EFFECTS: PERCEPTION VS. DATA

PERCEPTION

"When contributing to multiple projects in parallel, I:"



EMPIRICAL DATA

Multitasking vs. code production

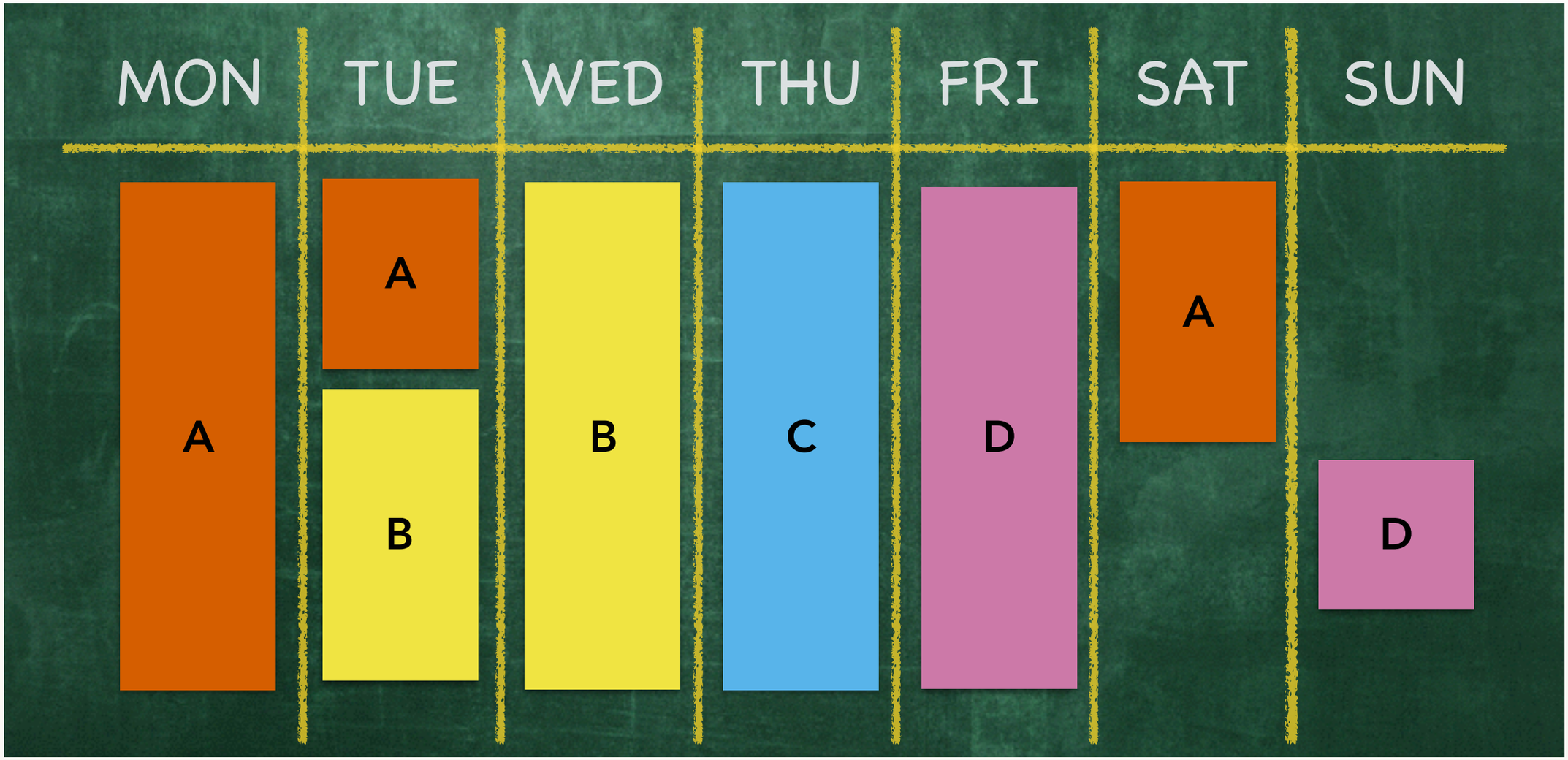
Daily multitasking correlates to amount of code produced

Weekly and day-to-day scheduling of work matters

No scheduling is productive beyond 5 projects/week

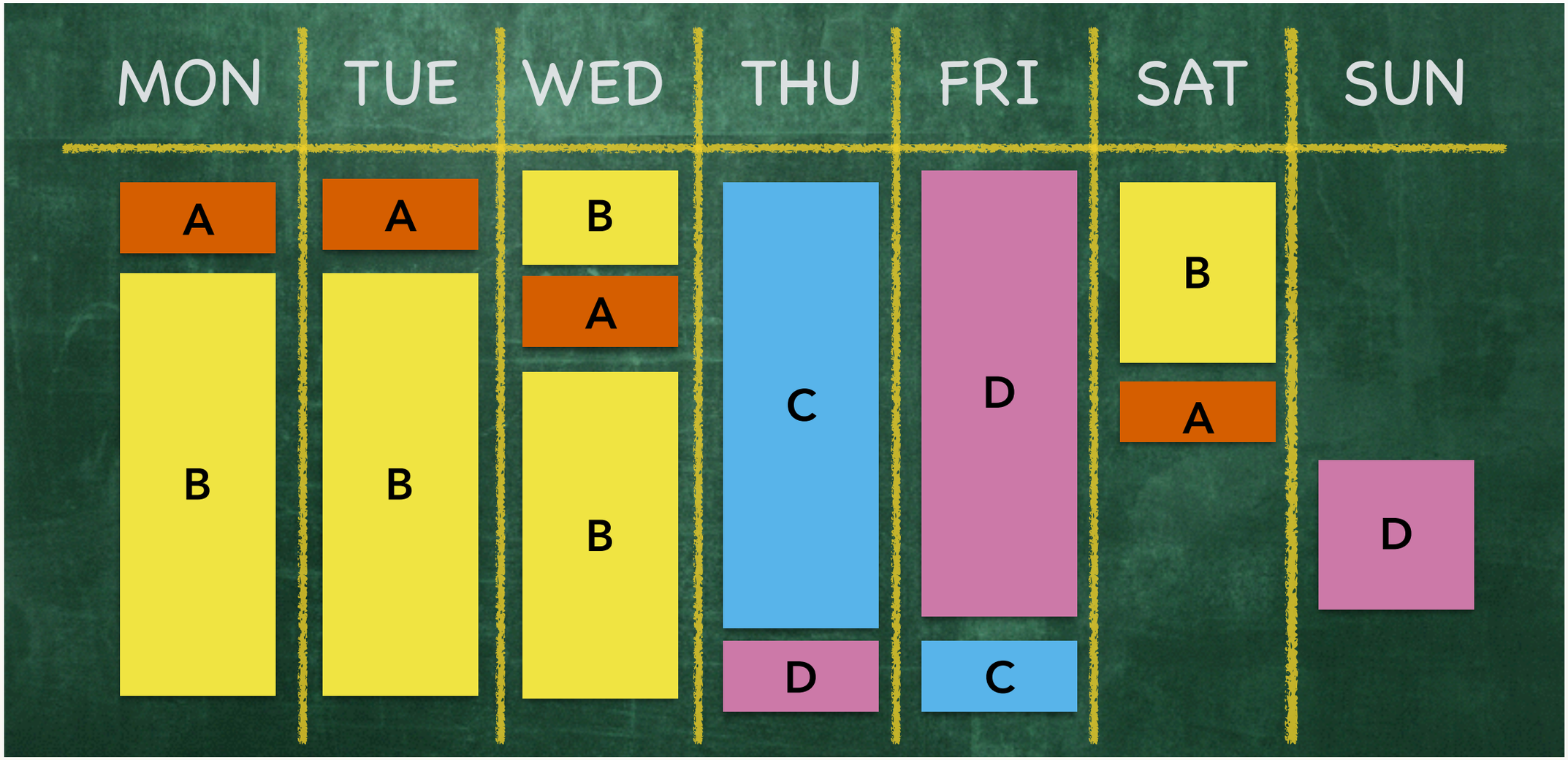
MODELING MULTITASKING

- Period matters



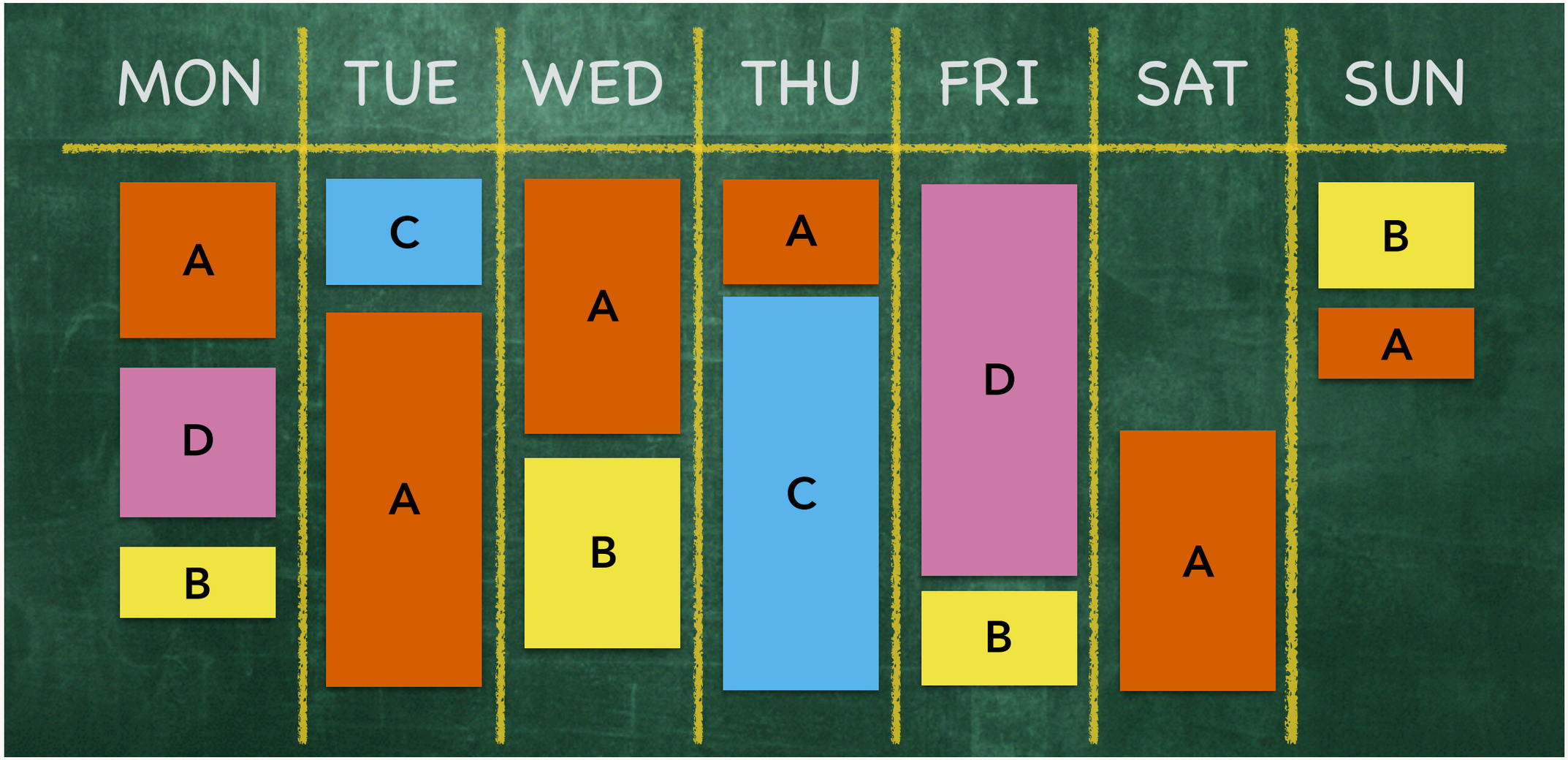
MODELING MULTITASKING

- Period matters
- Effort matters (A vs. B)



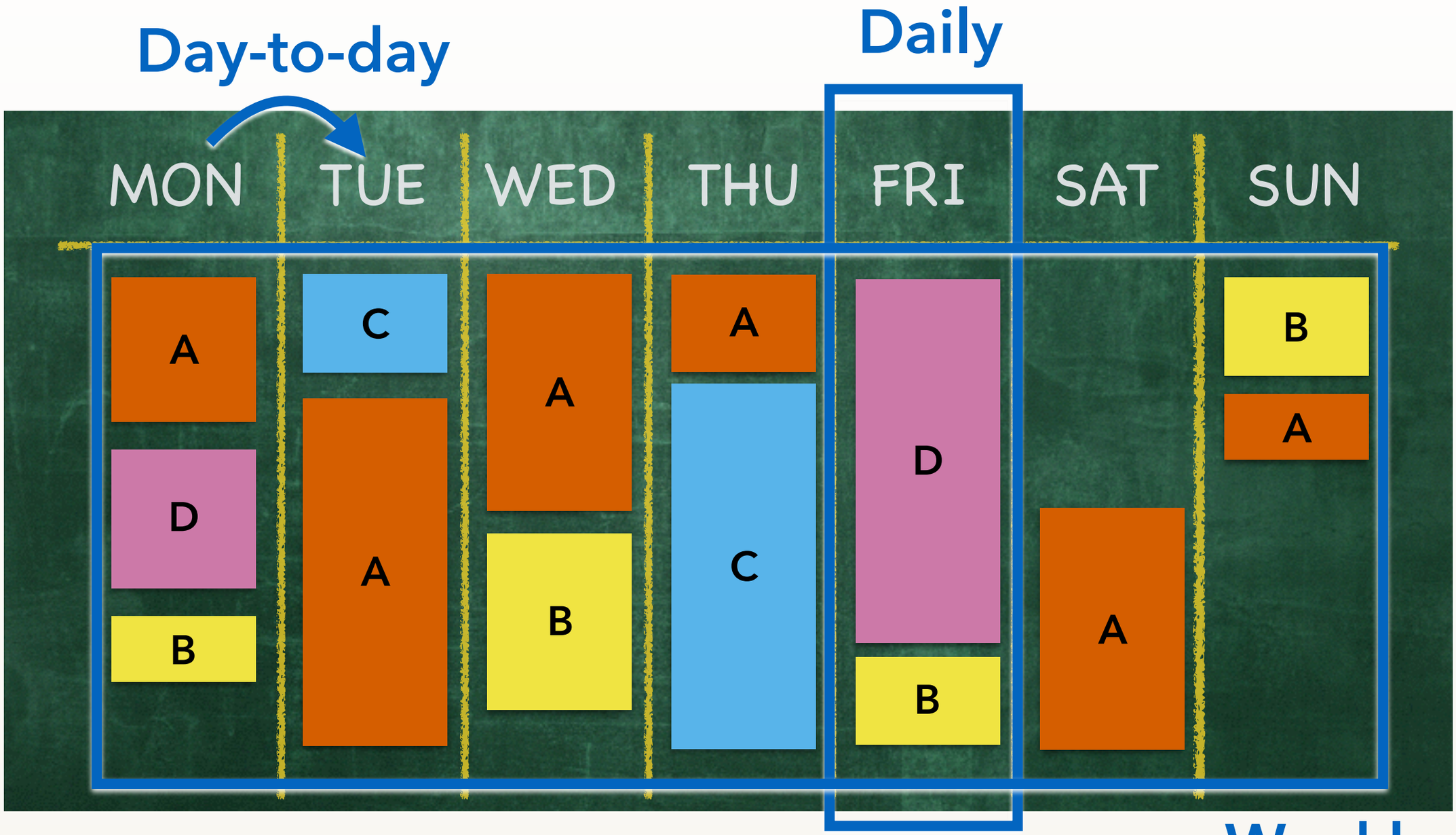
MODELING MULTITASKING

- Period matters
- Effort matters
- Break matters (A vs. D)
- …



MODELING MULTITASKING

- Period matters
- Effort matters
- Break matters
- …



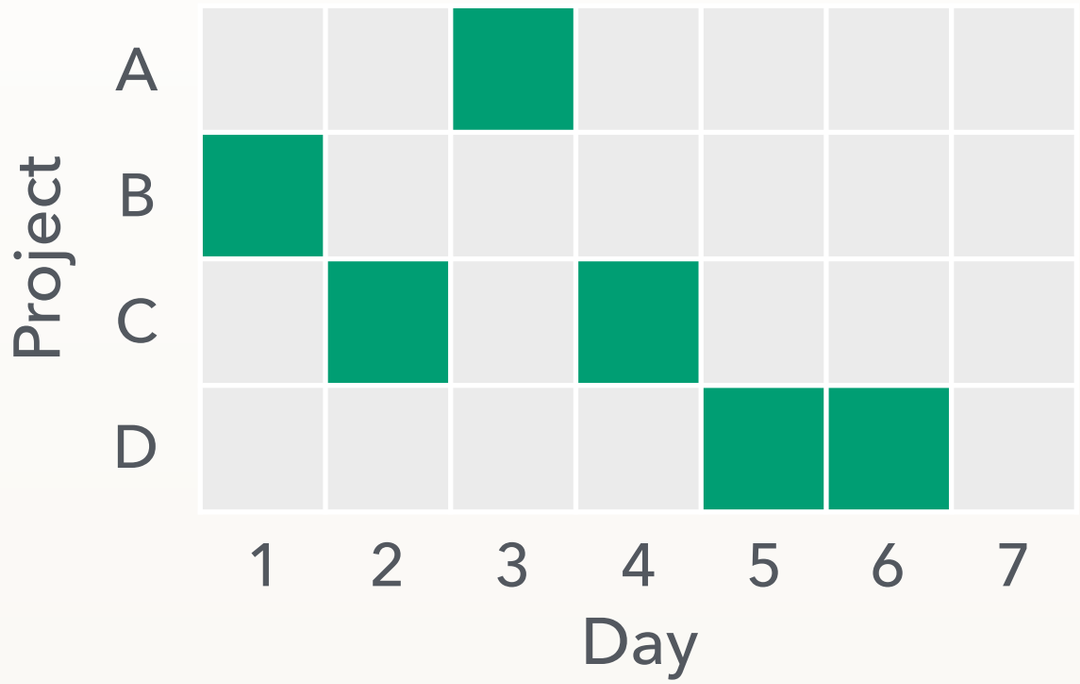
WE MODELED:

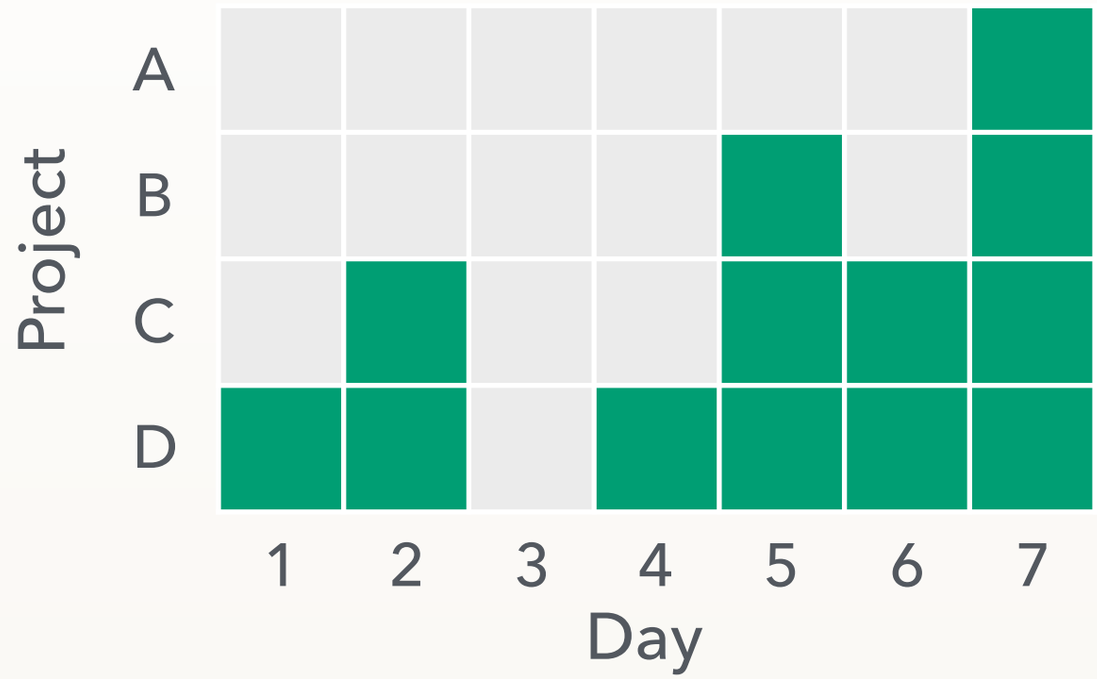
- One-week panels
- Three dimensions

MULTITASKING DIMENSIONS

1. PROJECTS PER DAY

Working sequentially vs. Within-day multitasking

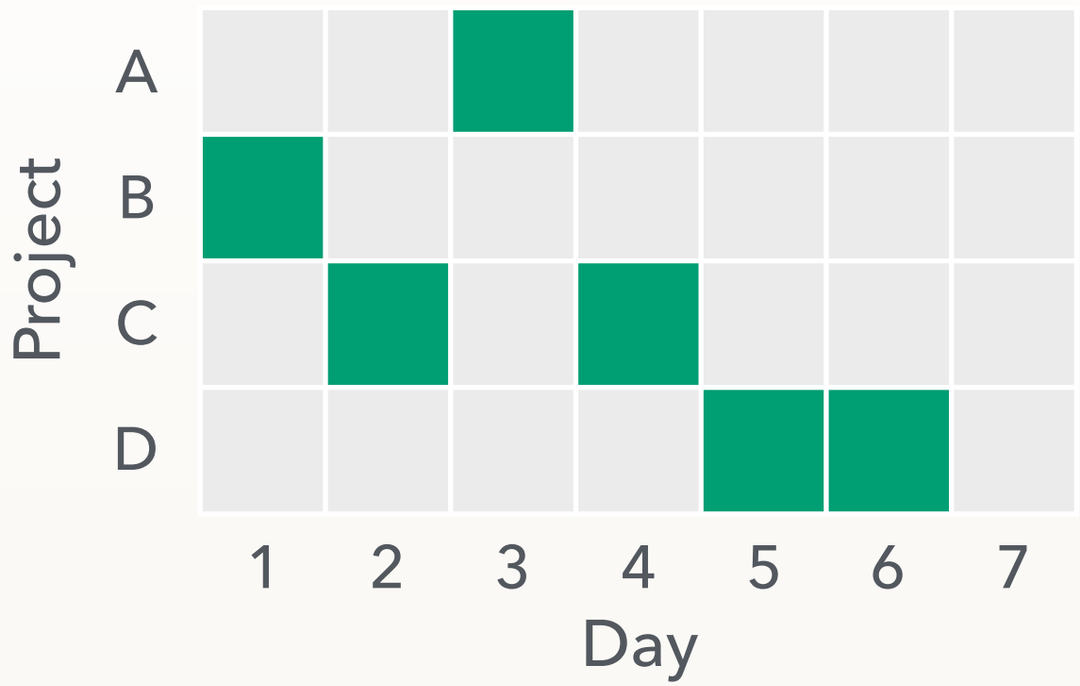




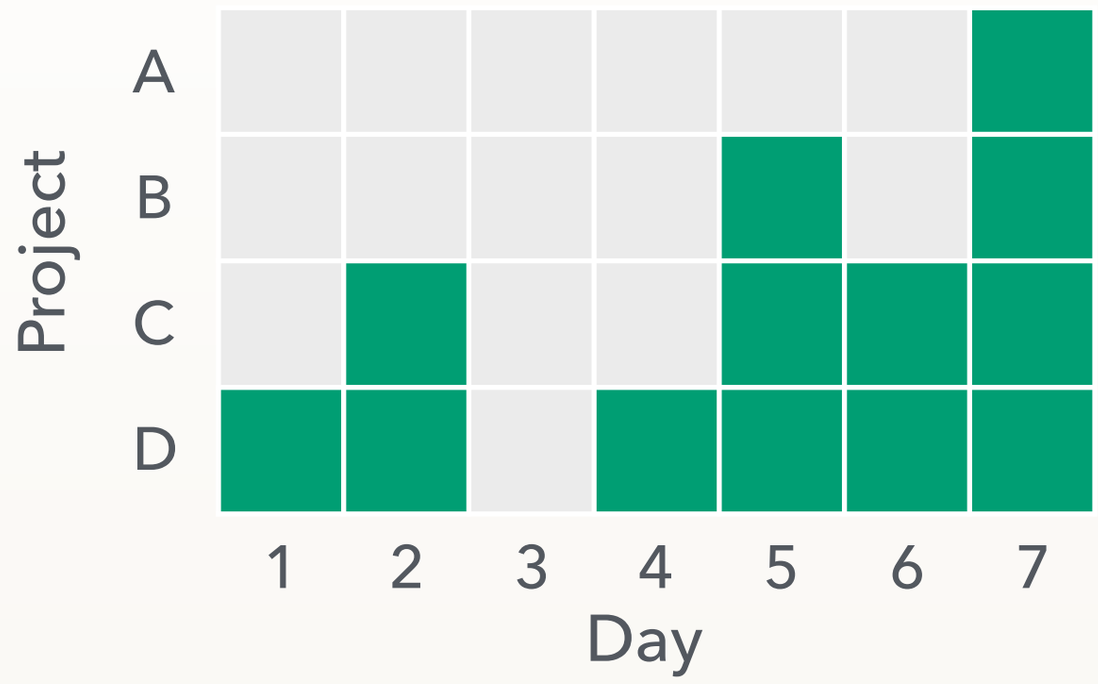
MULTITASKING DIMENSIONS

1. PROJECTS PER DAY

Working sequentially vs. Within-day multitasking



$$\text{AvgProjectsPerDay} = 1$$

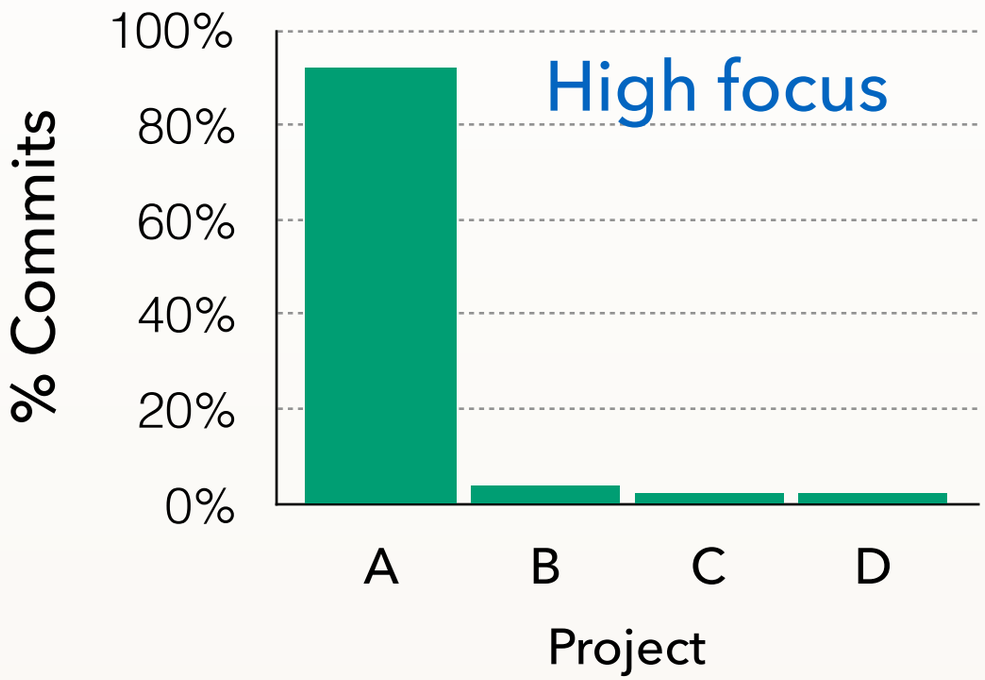


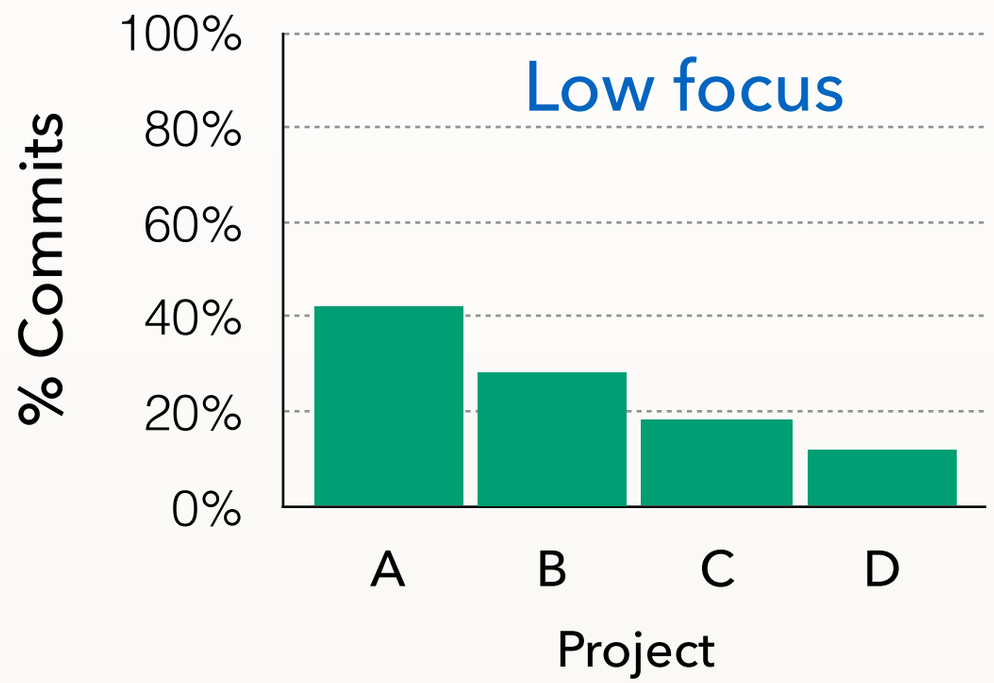
$$\text{AvgProjectsPerDay} = 2.2$$

MULTITASKING DIMENSIONS

2. WEEKLY FOCUS

Focusing on one project vs. Contributing evenly to all

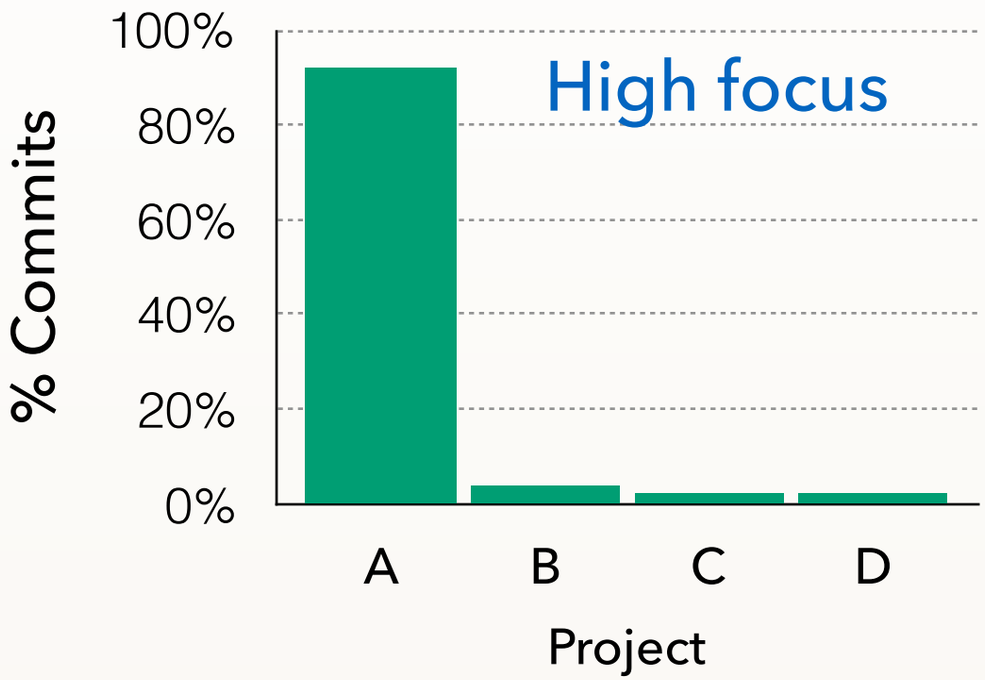




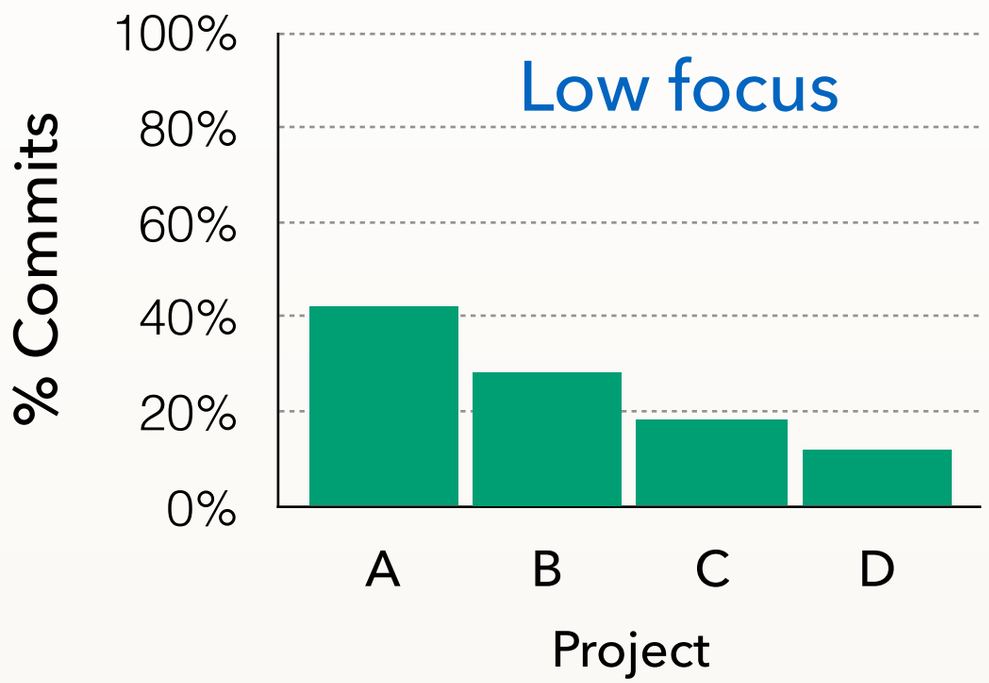
MULTITASKING DIMENSIONS

2. WEEKLY FOCUS

Focusing on one project vs. Contributing evenly to all



$$S_{\text{Focus}} = 0.25$$



$$S_{\text{Focus}} = 1.85$$

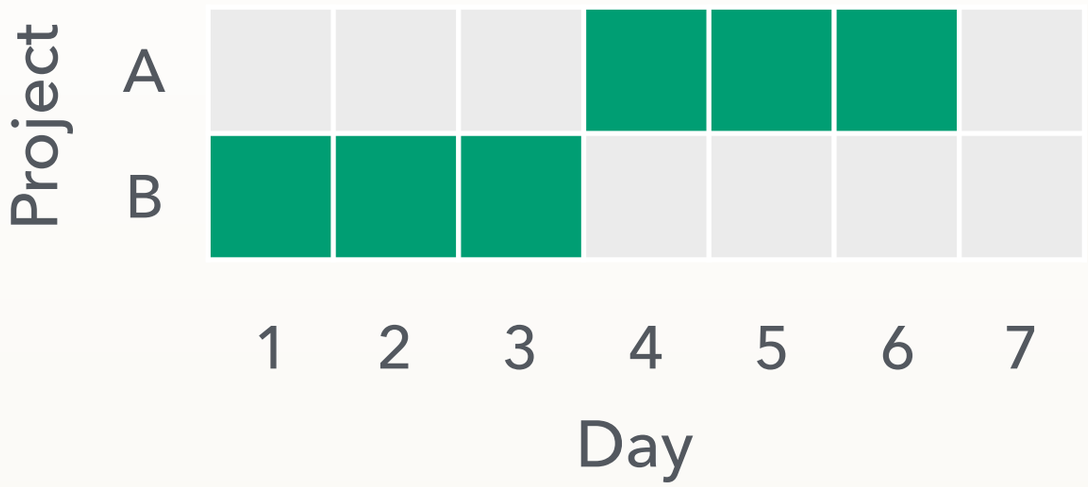
Shannon entropy: $S_{\text{Focus}} = -\sum_{i=1}^{N} p_i \log_2 p_i$

Fraction commits in project i

MULTITASKING DIMENSIONS

3. DAY-TO-DAY FOCUS

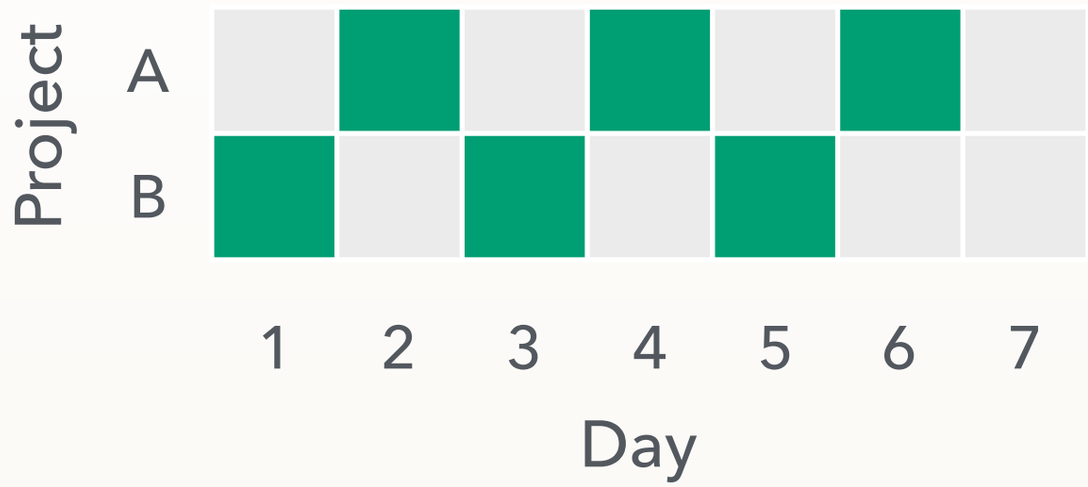
Repetitive day-to-day



$$\text{AvgProjectsPerDay} = 1$$
$$S_{\text{Focus}} = 1$$

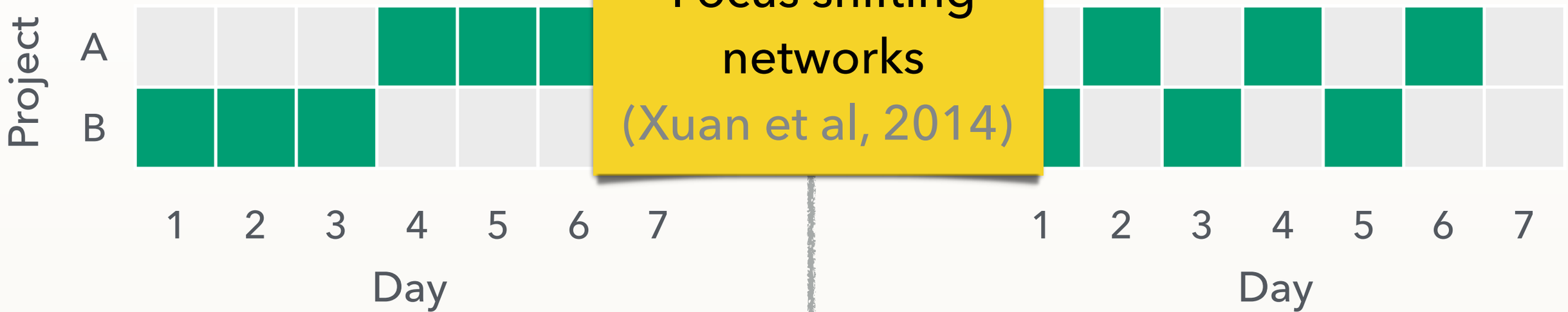
vs.

Switching focus



$$\text{AvgProjectsPerDay} = 1$$
$$S_{\text{Focus}} = 1$$

Repetitive day-to-day vs. Switching focus



Focus shifting networks (Xuan et al, 2014)

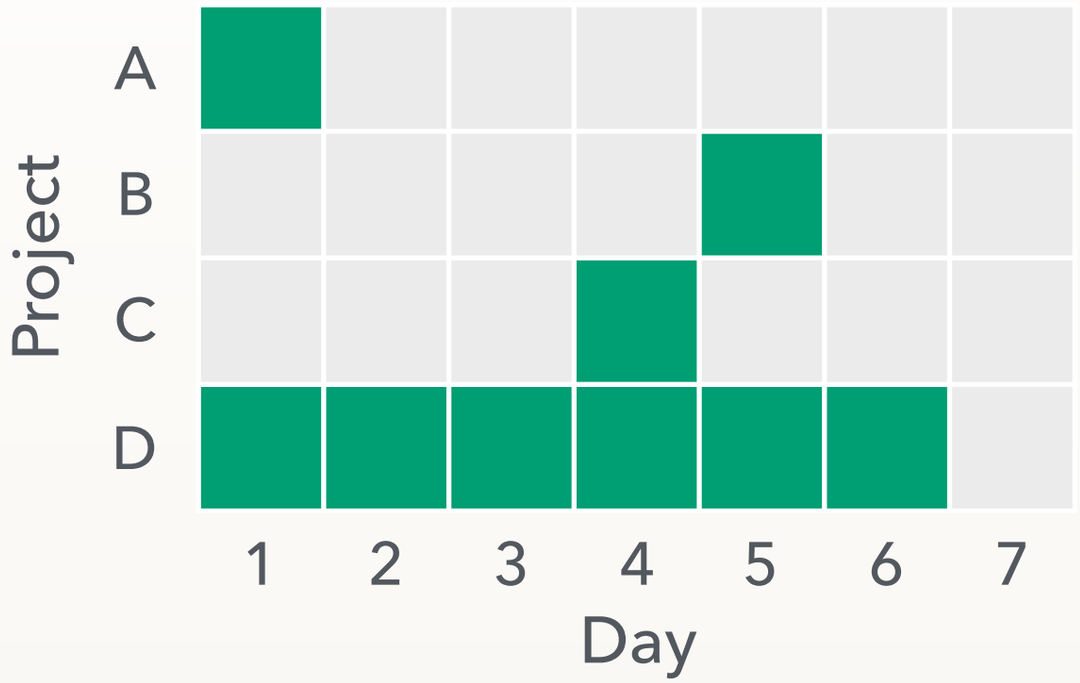
$\text{AvgProjectsPerDay} = 1$
$S_{\text{Focus}} = 1$

$\text{AvgProjectsPerDay} = 1$
$S_{\text{Focus}} = 1$

MULTITASKING DIMENSIONS

3. DAY-TO-DAY FOCUS

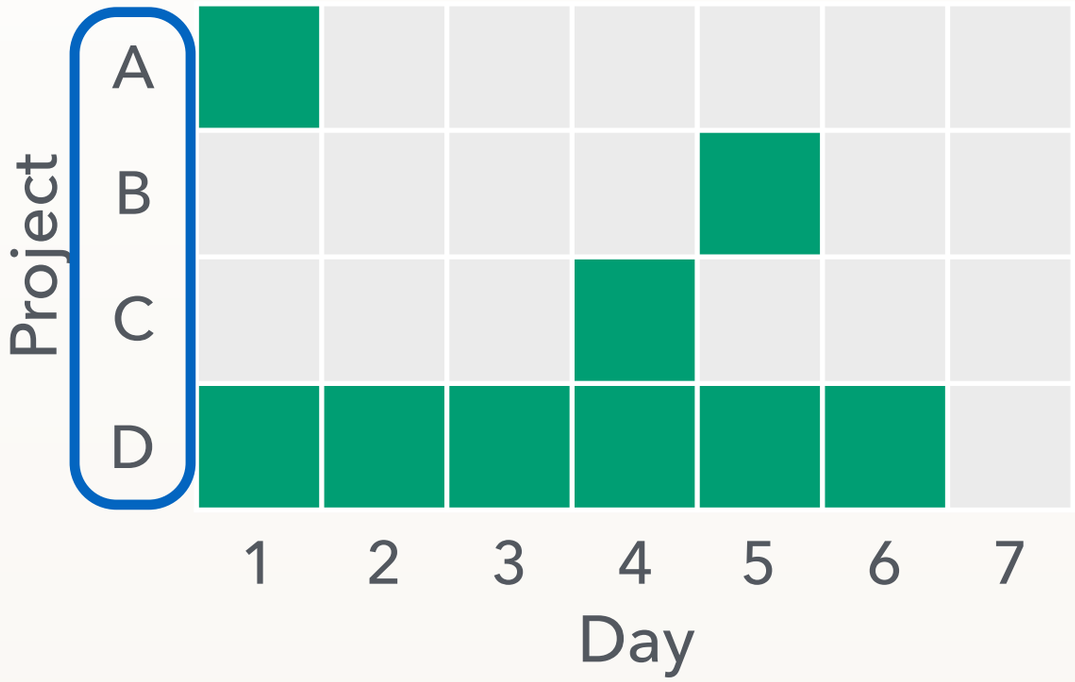
Repetitive day-to-day vs. Switching focus

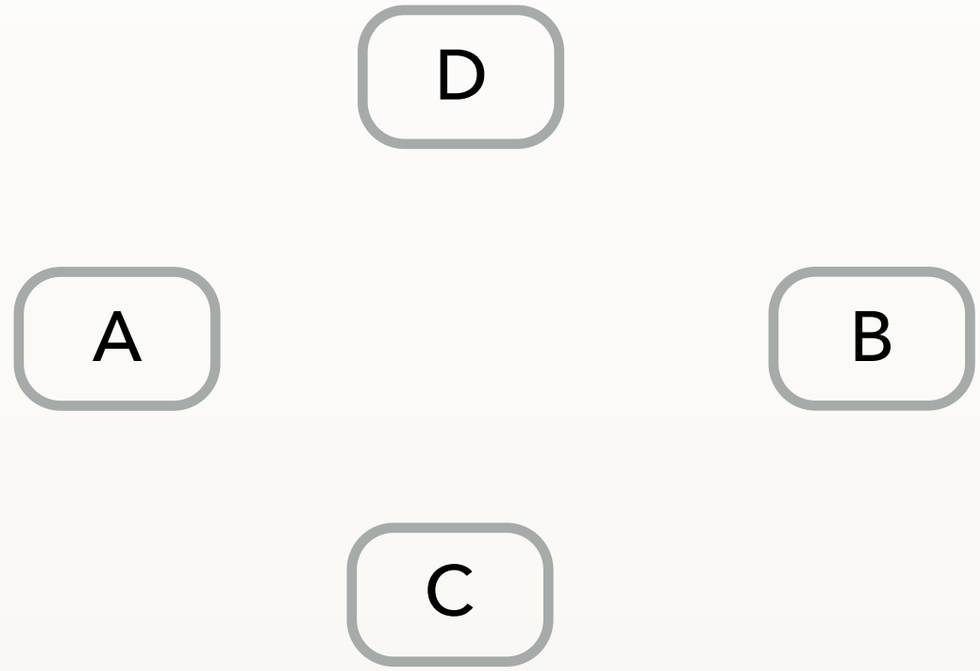


MULTITASKING DIMENSIONS

3. DAY-TO-DAY FOCUS

Repetitive day-to-day vs. Switching focus

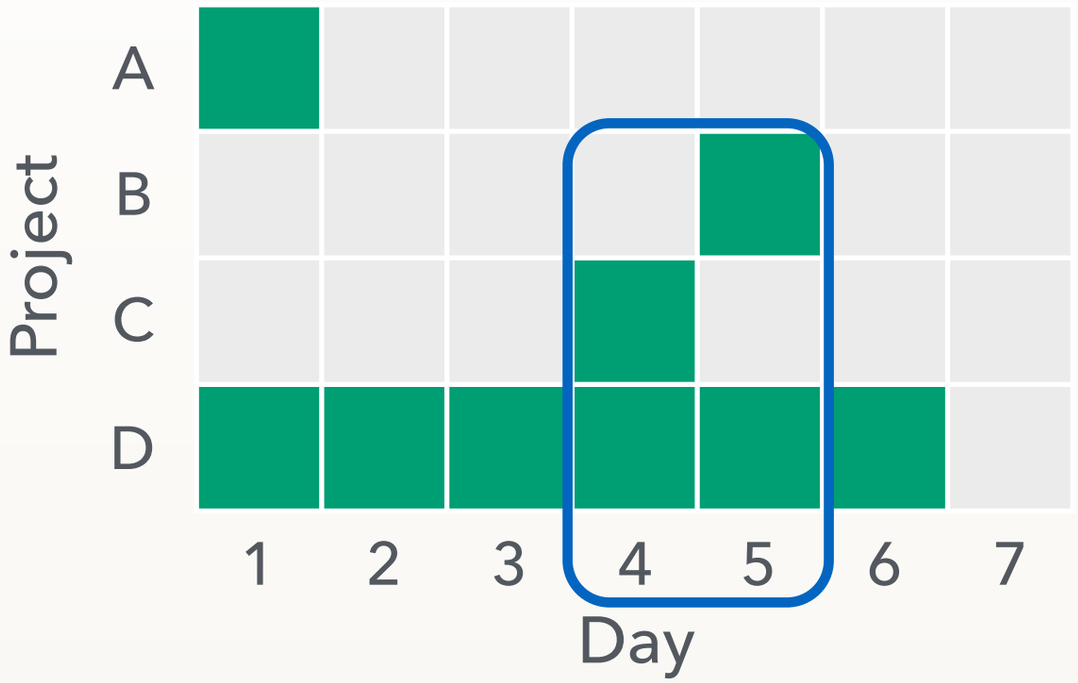


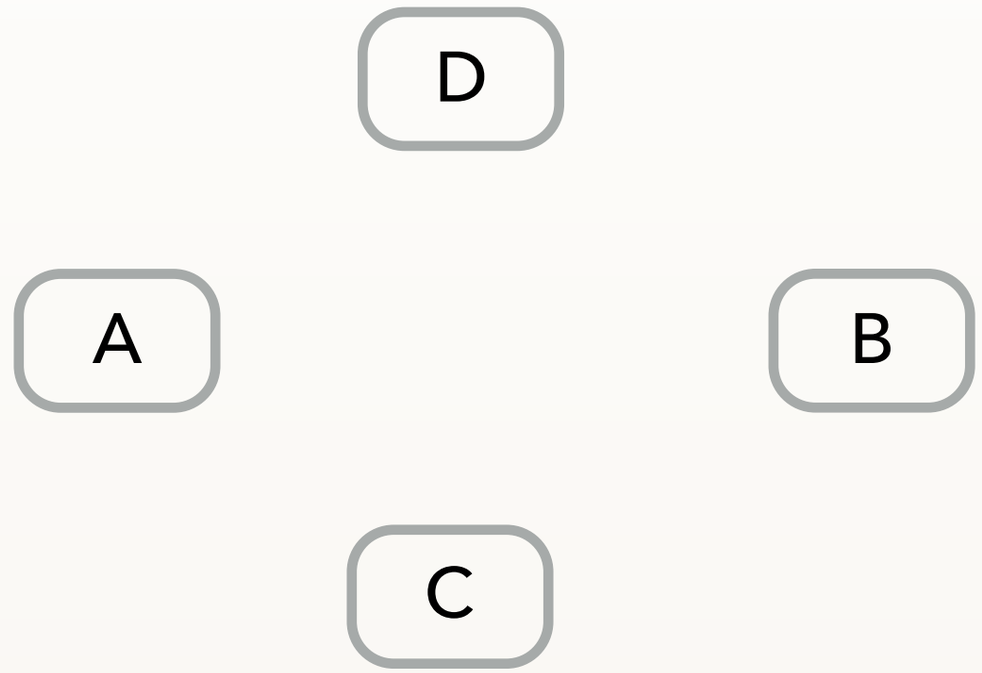


MULTITASKING DIMENSIONS

3. DAY-TO-DAY FOCUS

Repetitive day-to-day vs. Switching focus

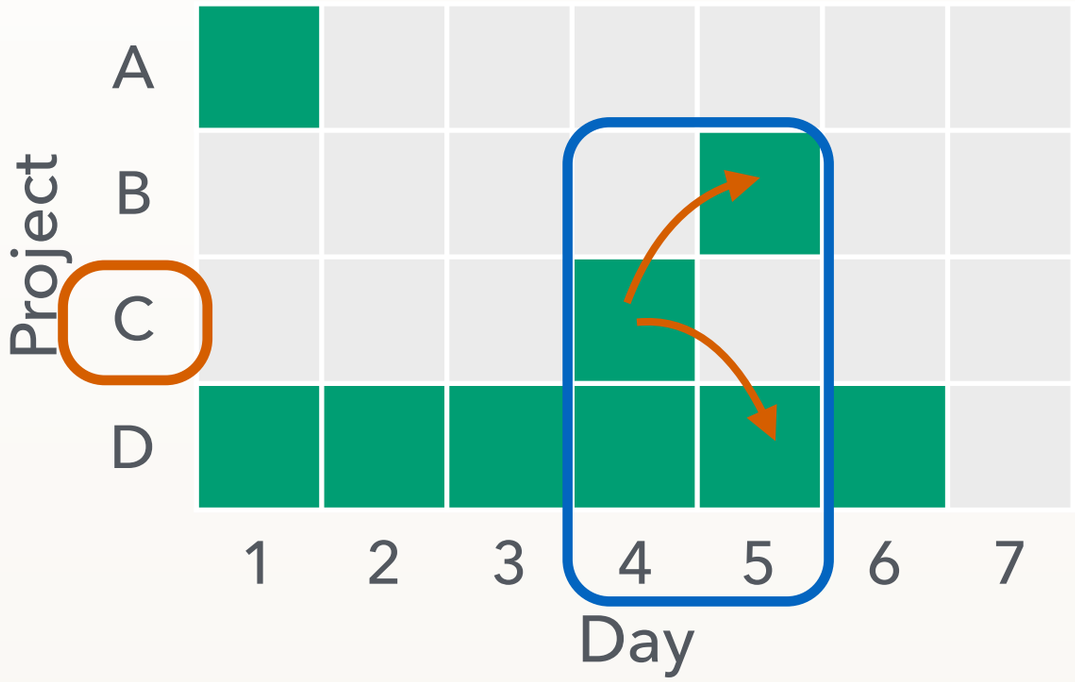


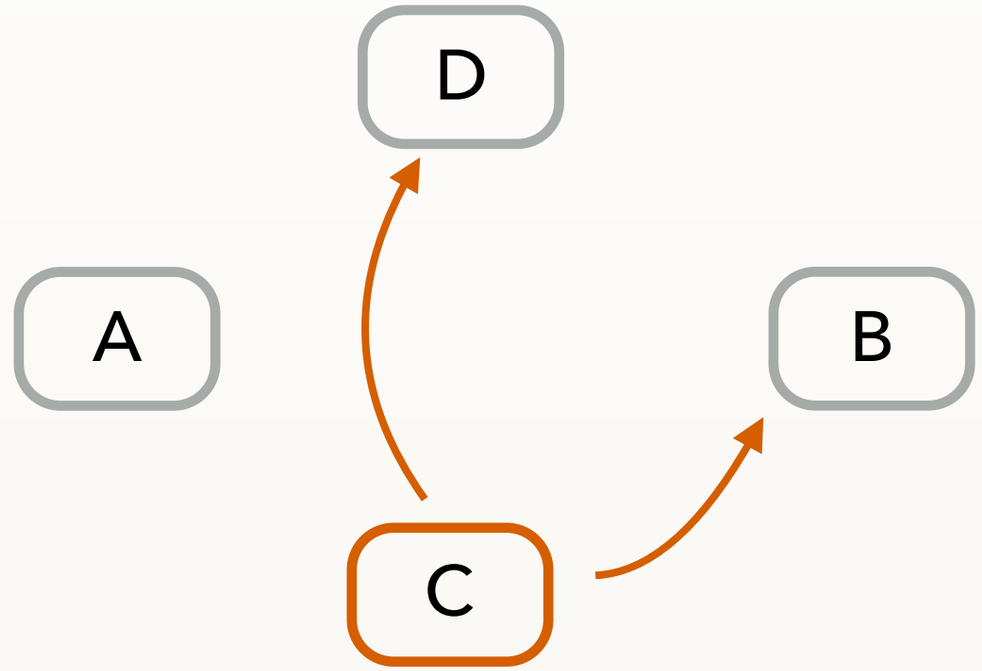


MULTITASKING DIMENSIONS

3. DAY-TO-DAY FOCUS

Repetitive day-to-day vs. Switching focus

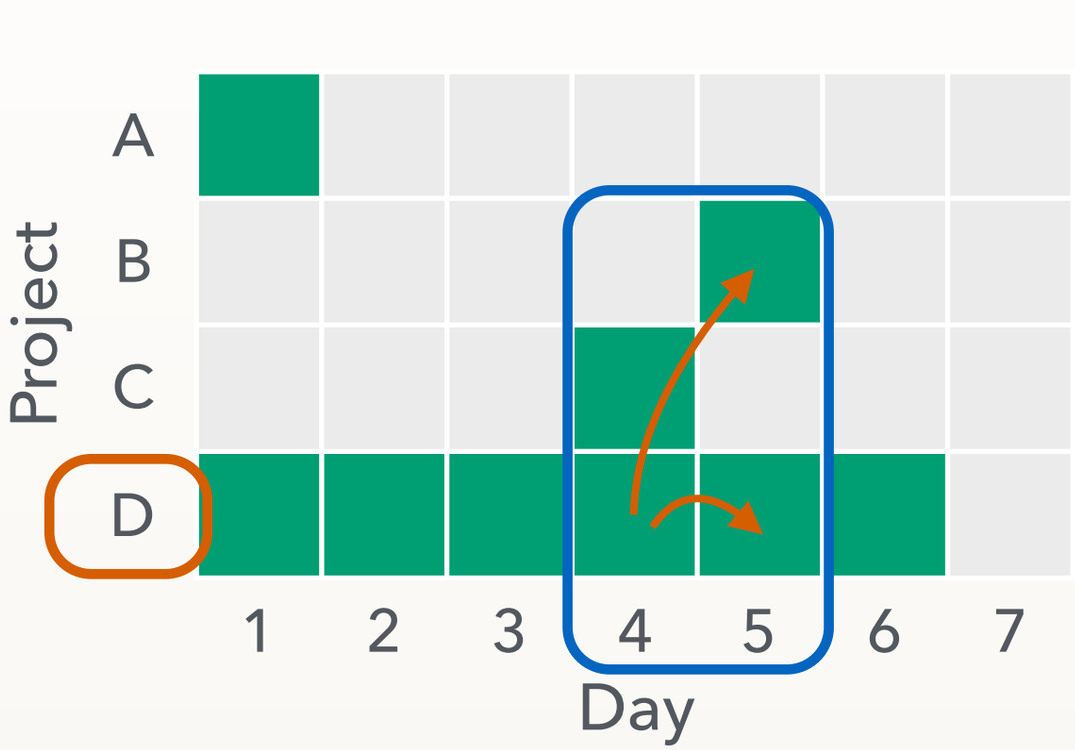


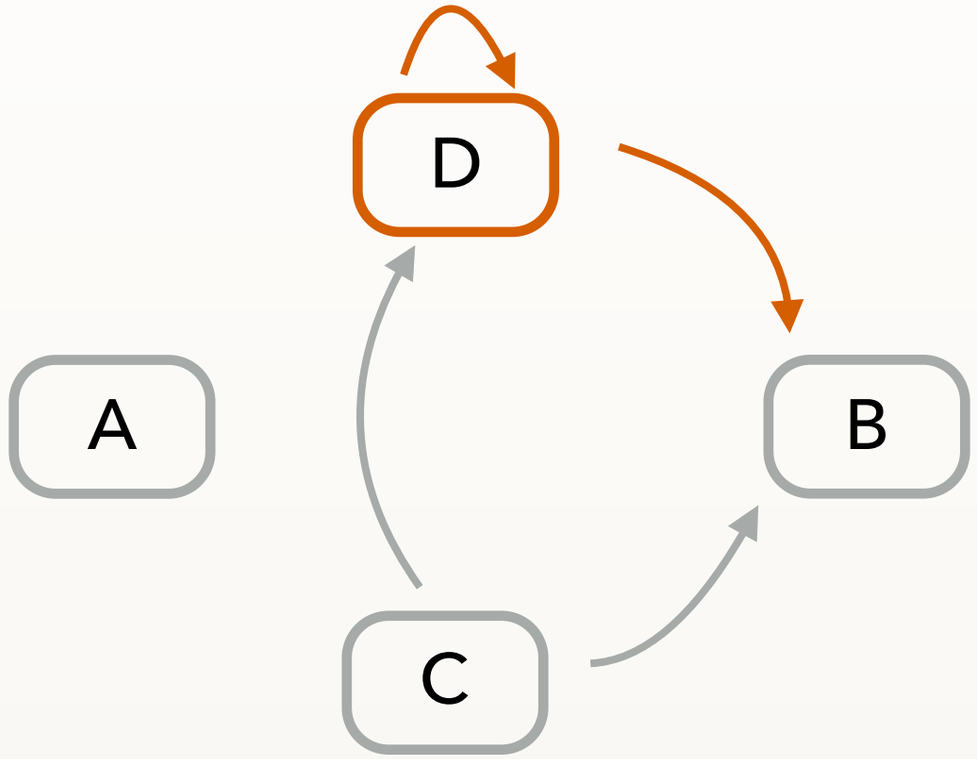


MULTITASKING DIMENSIONS

3. DAY-TO-DAY FOCUS

Repetitive day-to-day vs. Switching focus

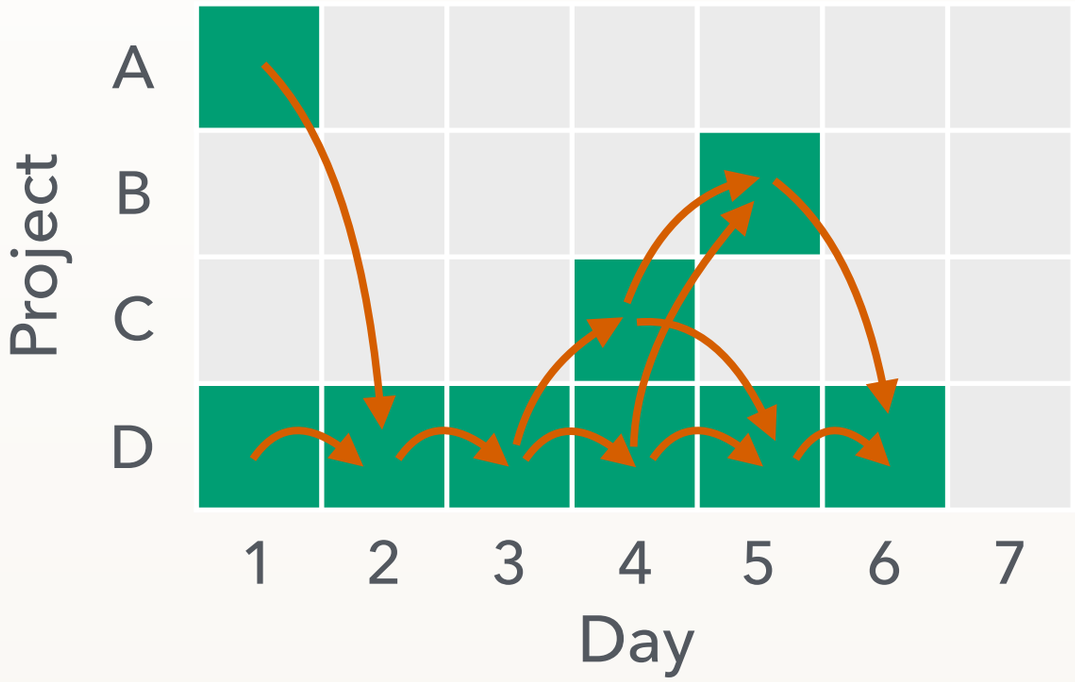


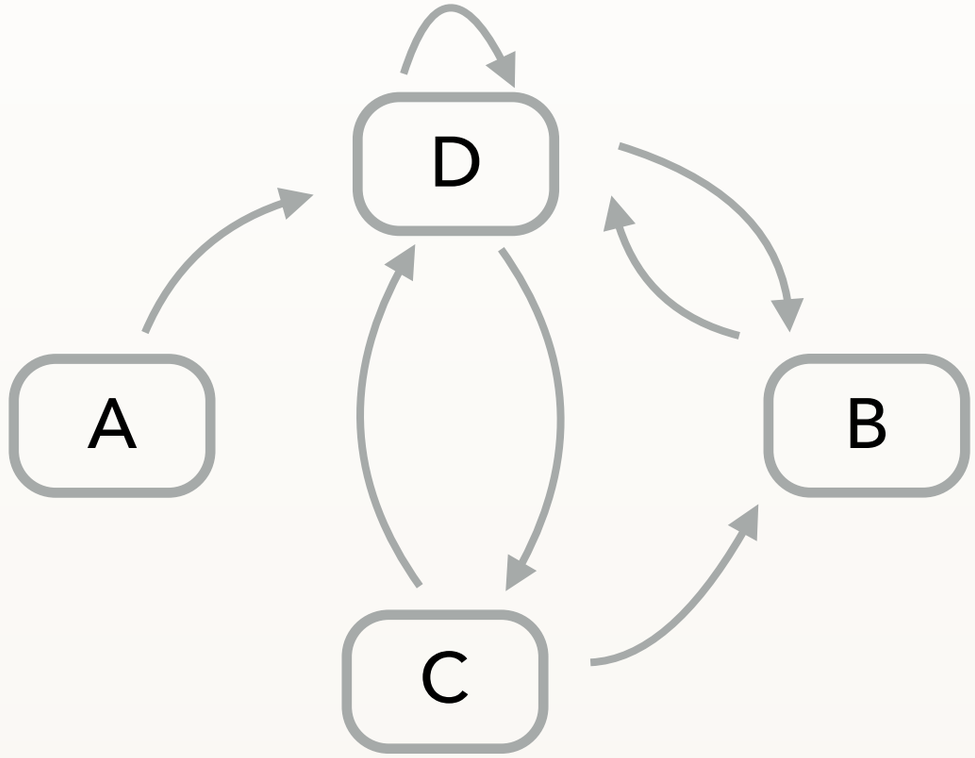


MULTITASKING DIMENSIONS

3. DAY-TO-DAY FOCUS

Repetitive day-to-day vs. Switching focus

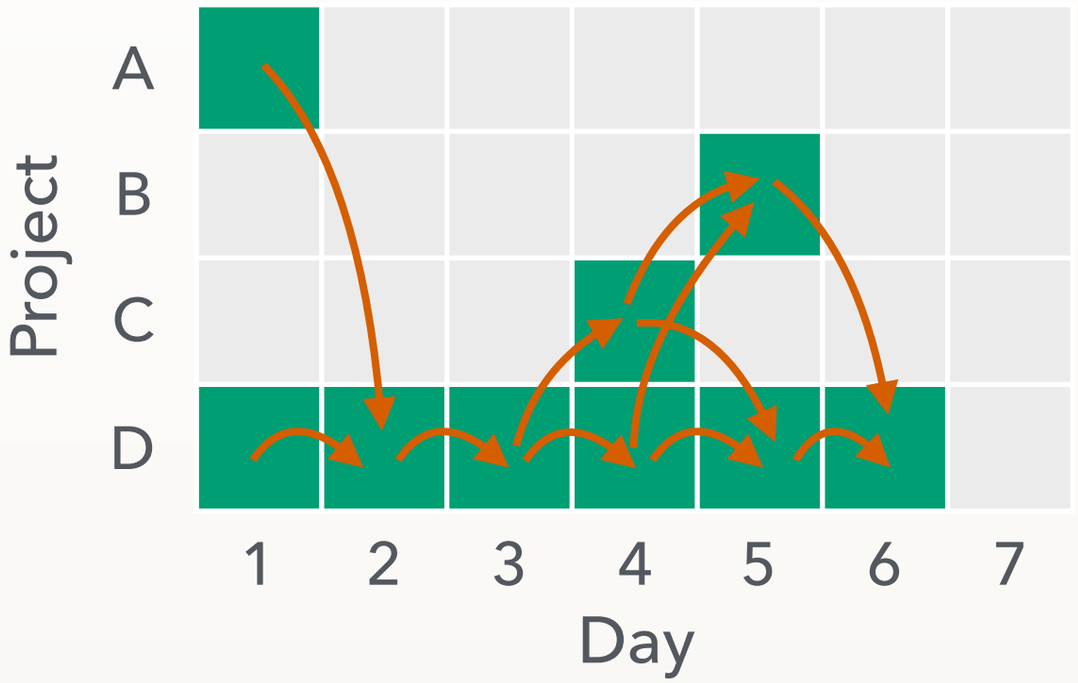


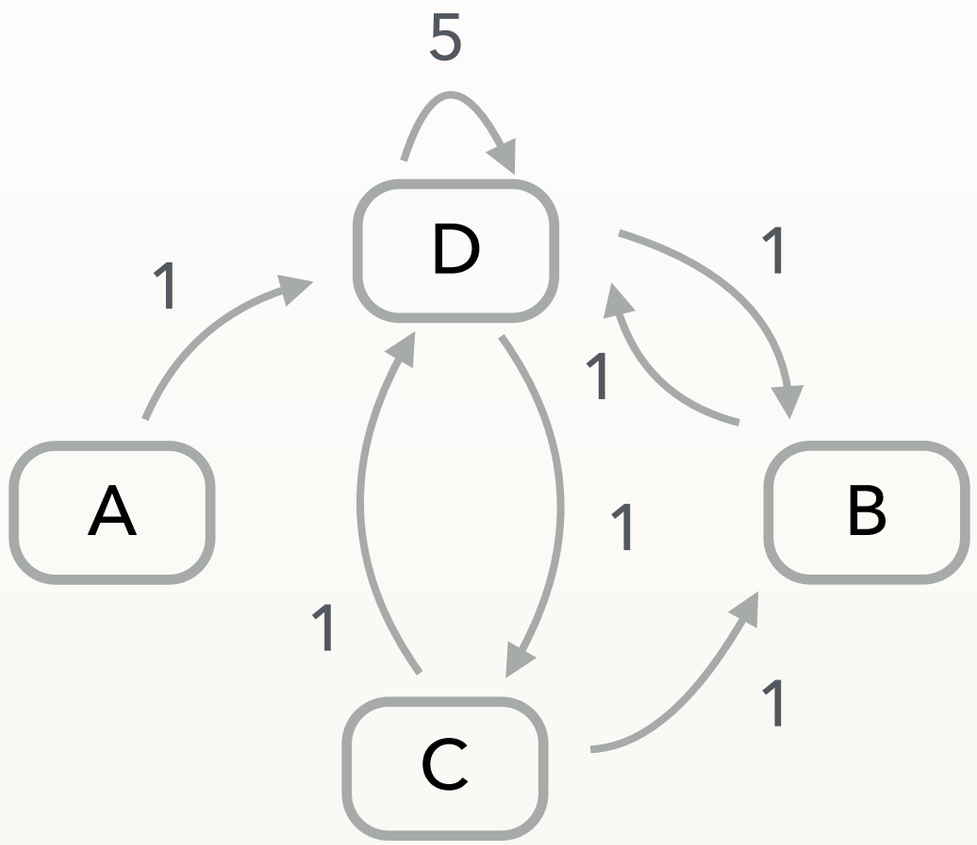


MULTITASKING DIMENSIONS

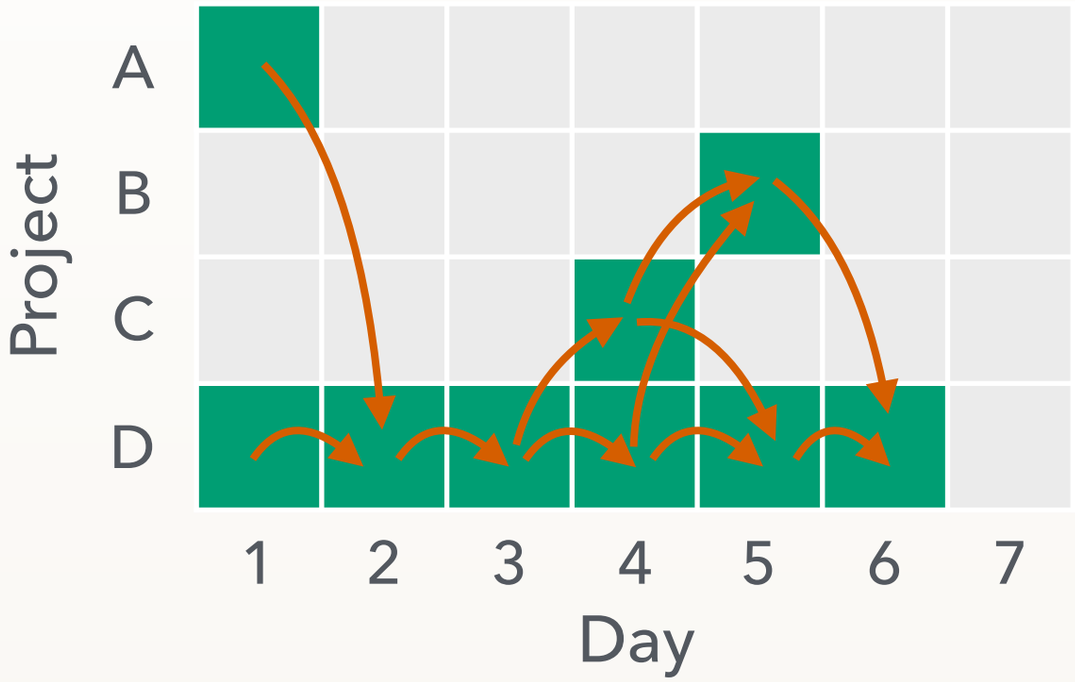
3. DAY-TO-DAY FOCUS

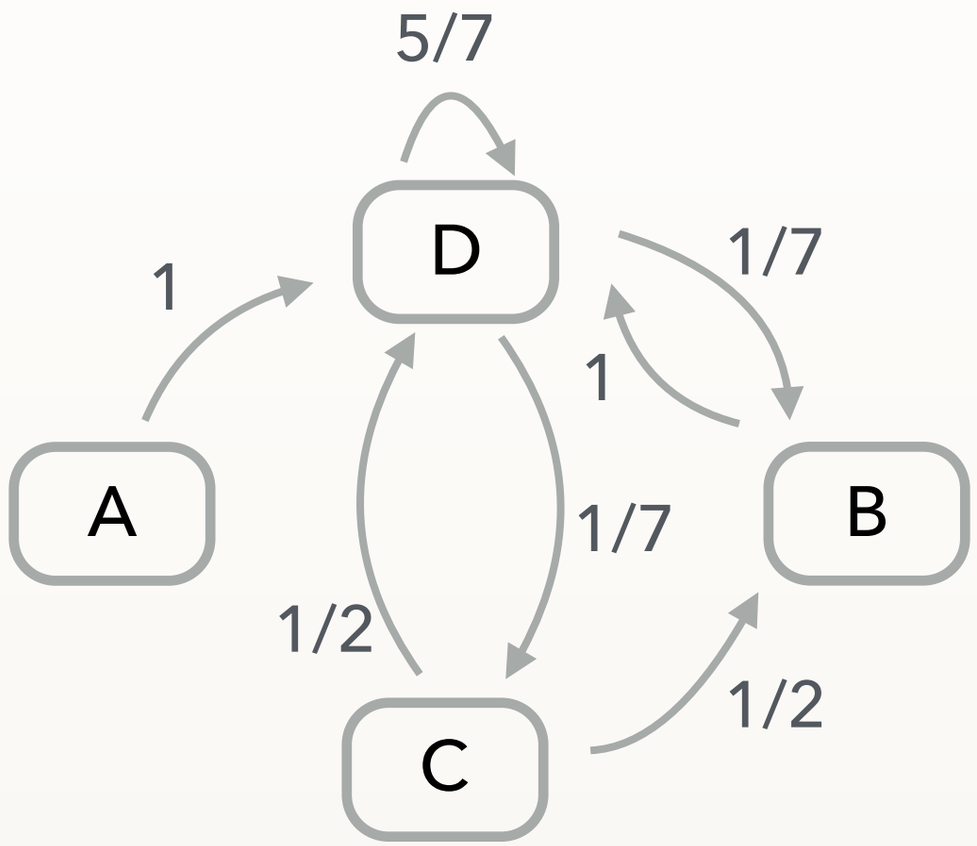
Repetitive day-to-day vs. Switching focus



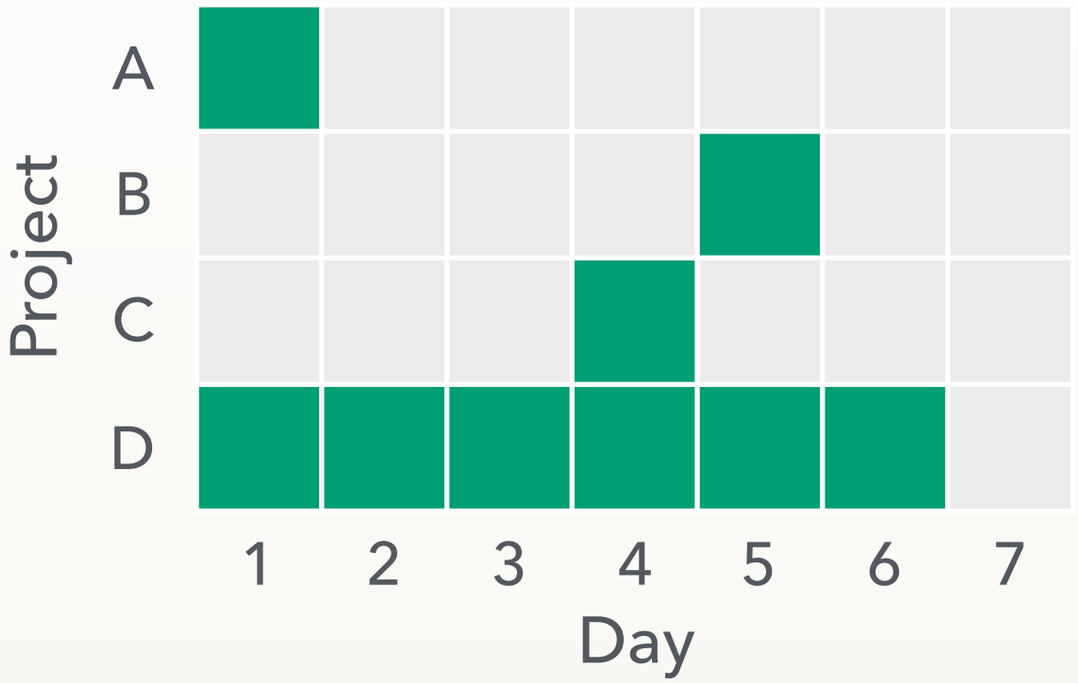


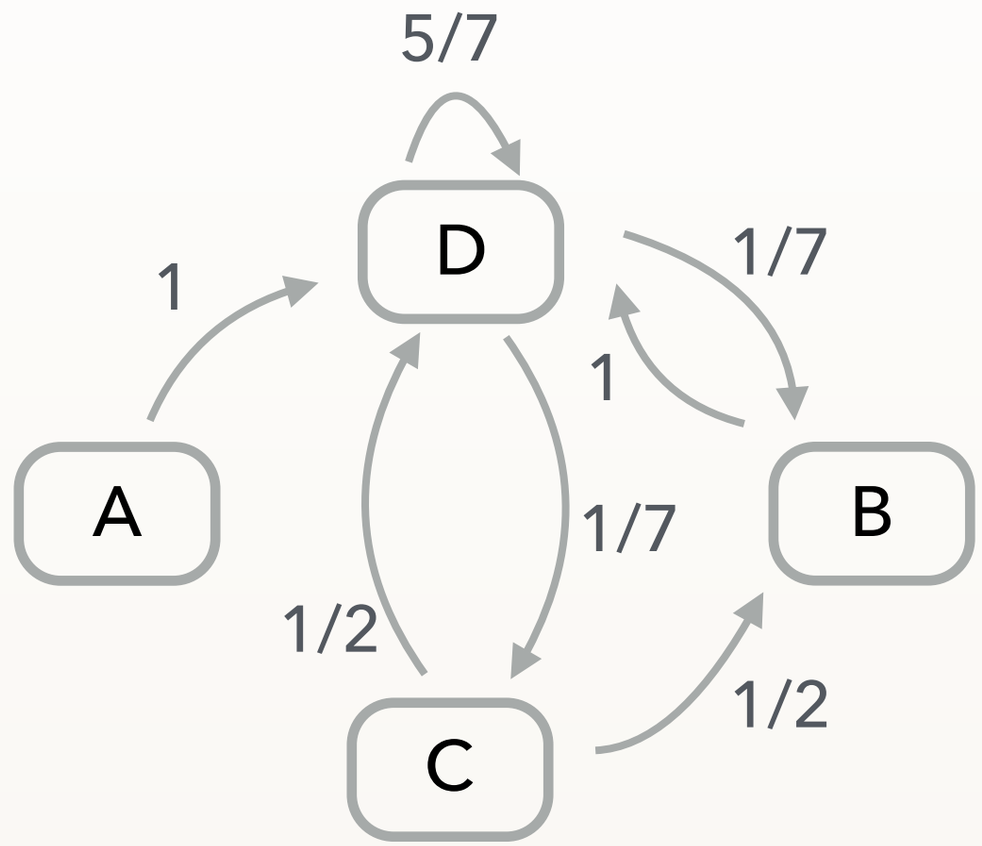
Repetitive day-to-day vs. Switching focus





Repetitive day-to-day vs. Switching focus





Markov entropy:

$$S_{\text{Switch}} = -\sum_{i=1}^{N}\left[p_i \sum_{j\in\pi_i} p(j|i)\log_2 p(j|i)\right]$$

How predictable is my focus tomorrow if today I work on project j?

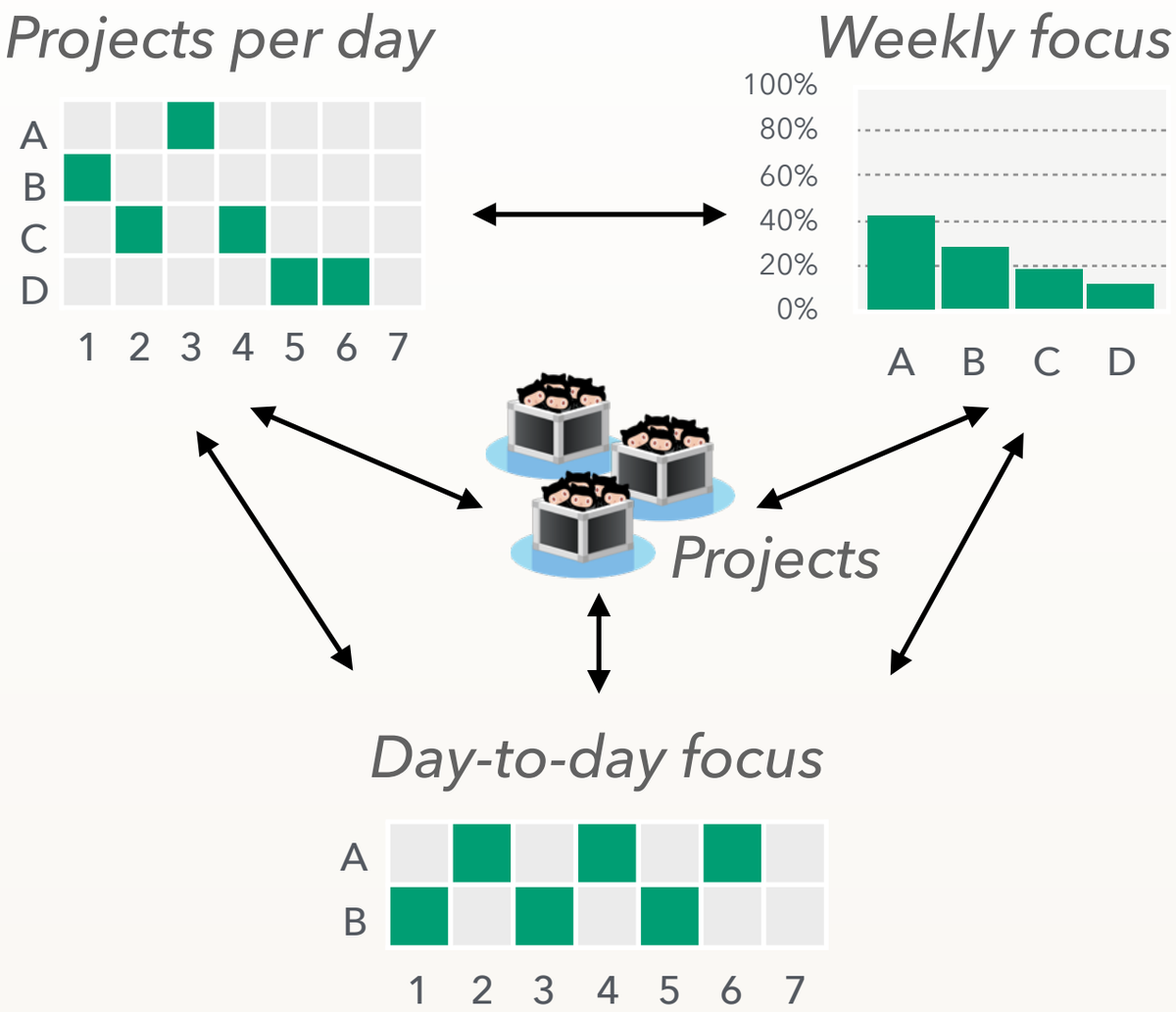
LINEAR MIXED-EFFECTS REGRESSION

Response:
LOC added / week

Controls:

- time
- total projects
- programming languages

Predictors:



Longitudinal data

- 1,200 developers
- 5+ years each: multiple weeks of observation

Random effect: developer

- developer-to-developer variability in the response

Random slope: time | developer

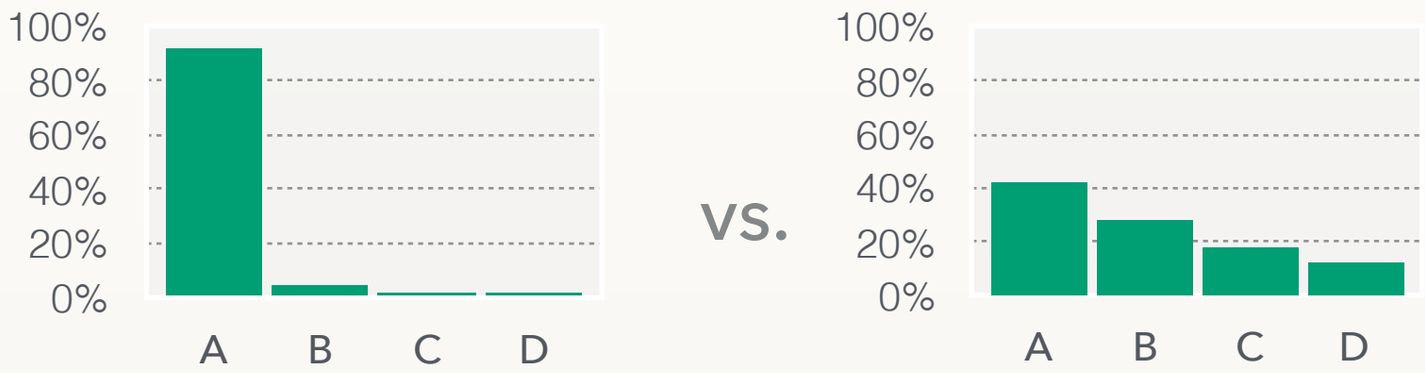
- developers more productive initially may be less strongly affected by time passing

MULTITASKERS DO MORE; SCHEDULING MATTERS

Higher LOC added

Projects per day

Weekly focus

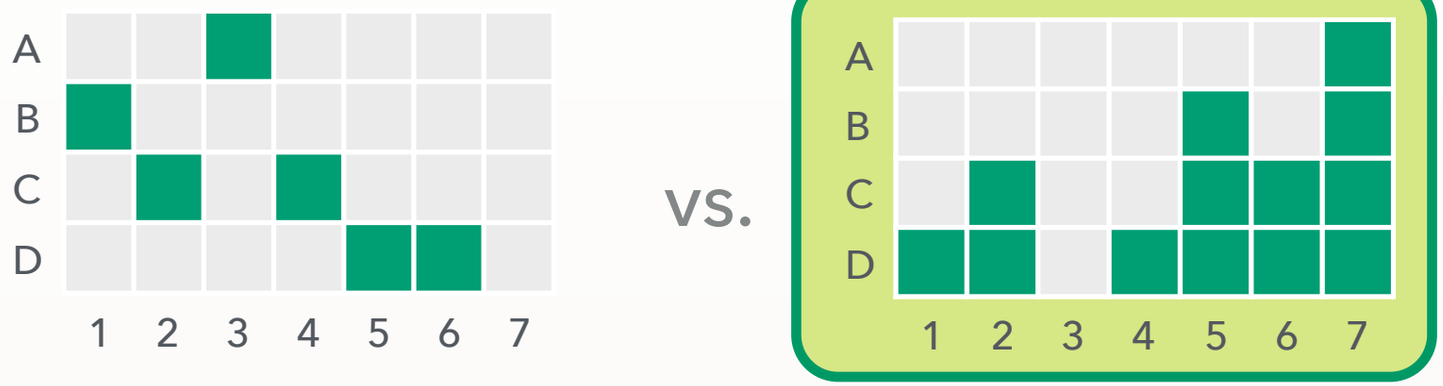


Day-to-day focus (repeatability)

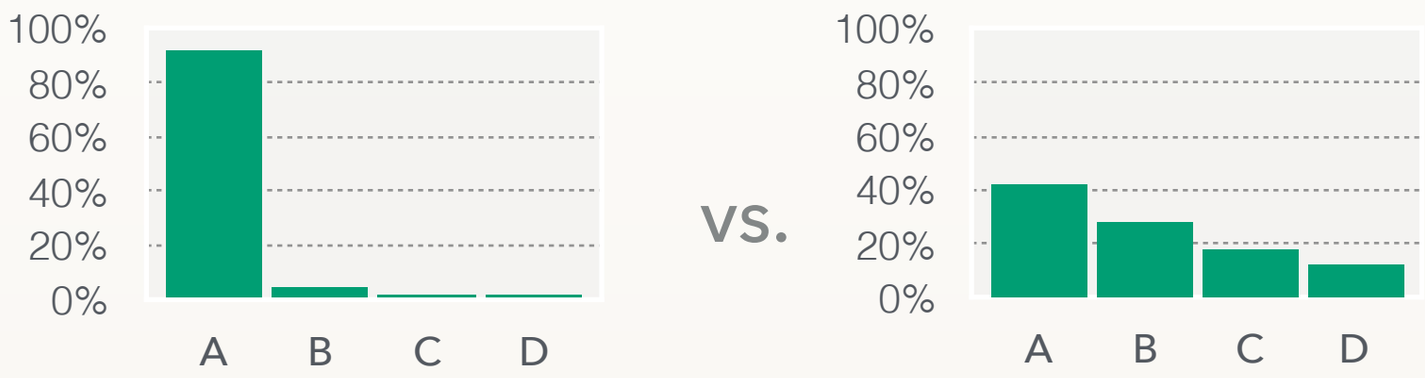
MULTITASKERS DO MORE; SCHEDULING MATTERS

Higher LOC added

Projects per day



Weekly focus

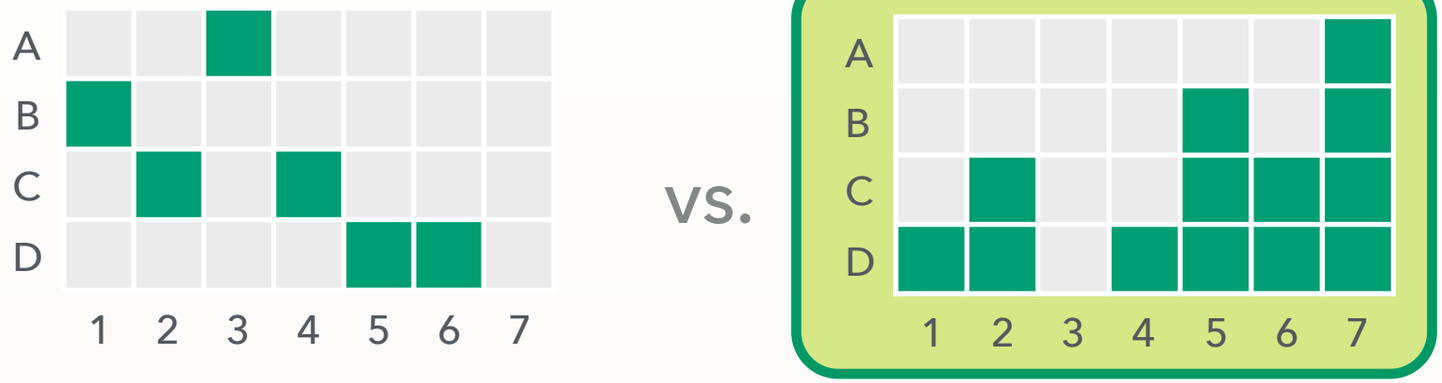


Day-to-day focus (repeatability)

MULTITASKERS DO MORE; SCHEDULING MATTERS

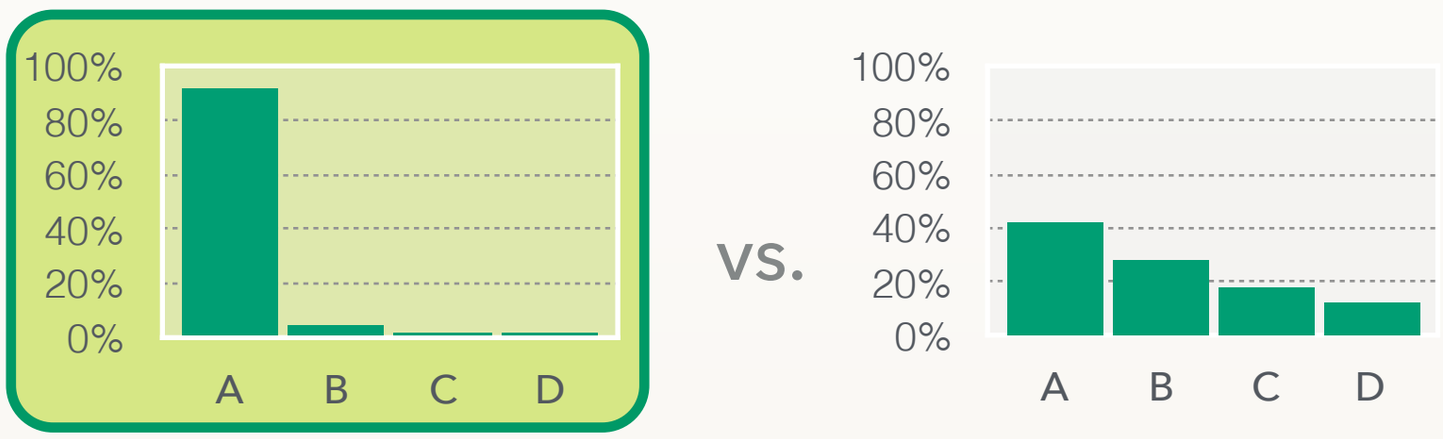
Higher LOC added

Projects per day



More within-day multitasking

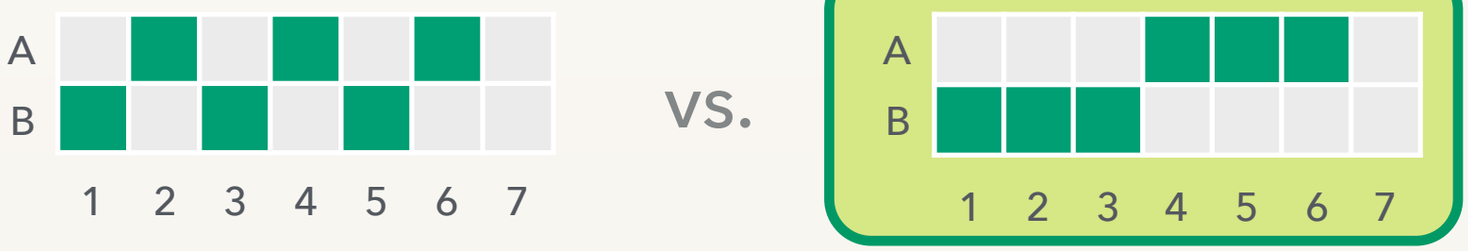
Weekly focus



Higher focus

More repetitive day-to-day work

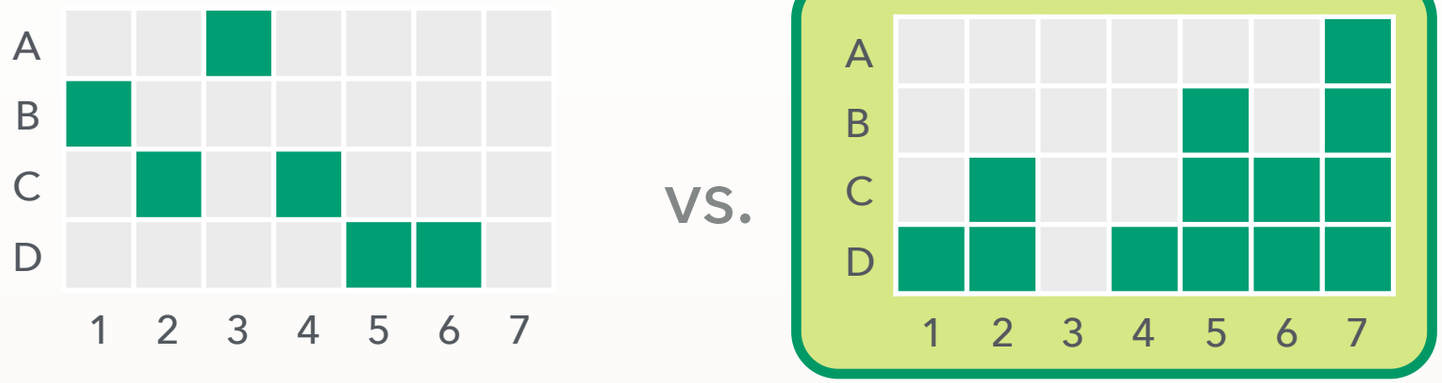
Day-to-day focus (repeatability)



MULTITASKERS DO MORE; SCHEDULING MATTERS

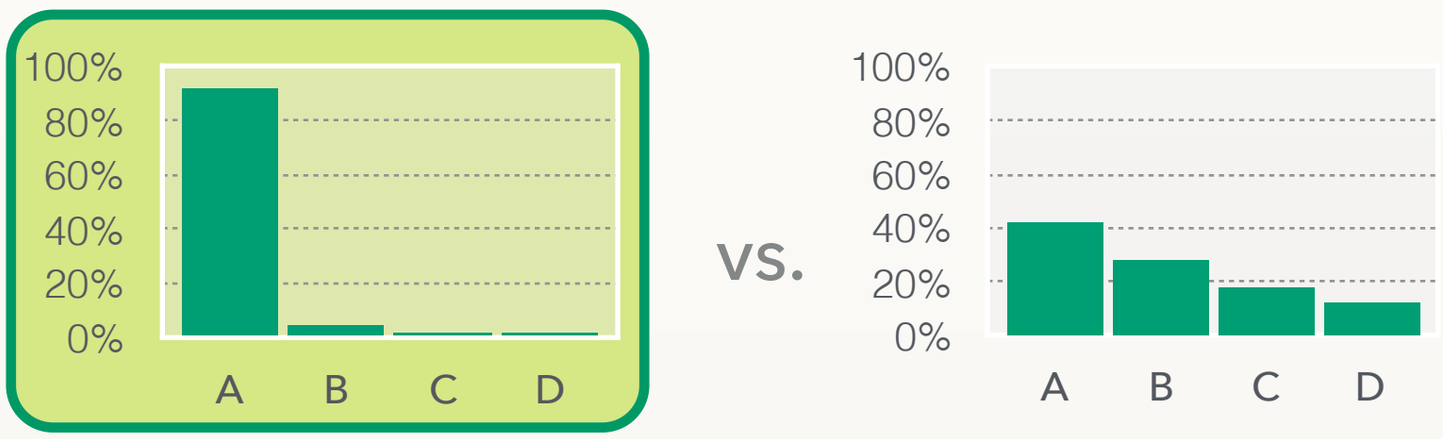
Higher LOC added

Projects per day



More within-day multitasking

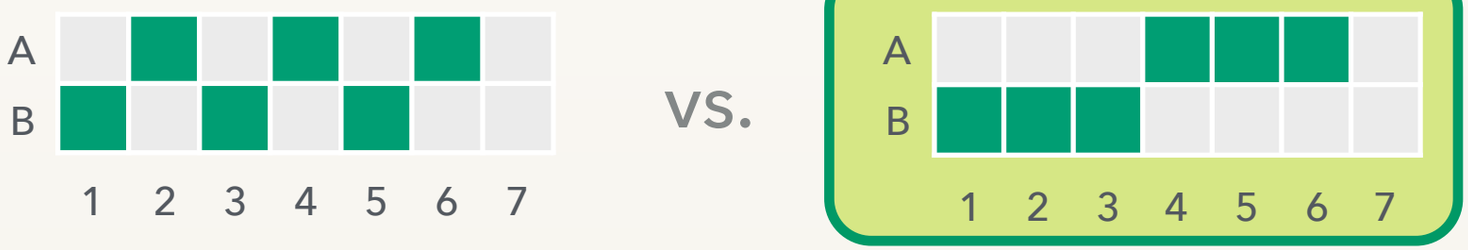
Weekly focus



Higher focus

More repetitive day-to-day work

Day-to-day focus (repeatability)



<u>Interaction effects</u>:

No scheduling is productive over 5 projects/week

MULTITASKERS DO MORE; SCHEDULING MATTERS

Higher LOC added

Theory: How does multitasking affect performance?

PROS

- **Fill downtime**

  Switch focus between projects to utilize tim more efficiently

  (Adler and Benbunan-Fich, 2012)

- **Cross-fertilisation**

  Easier to work on other projects if knowledge is transferrable

  (Lindbeck and Snower, 2000)

CONS

- **Cognitive switching cost**

  Depends on interruption duration, complexity, moment

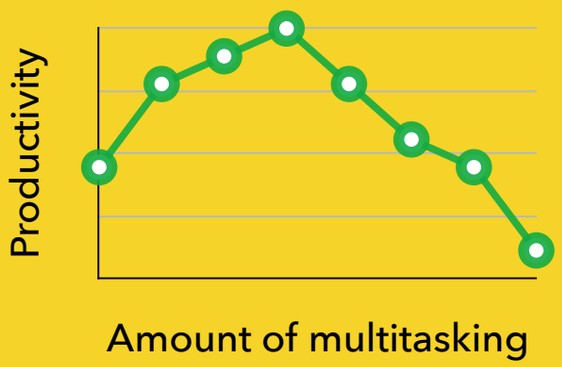
  (Altmann and Trafton, 2002) Borst, Taatgen, van Rijn, 2015)

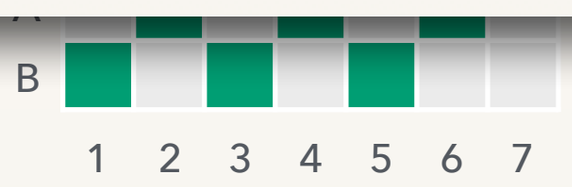
- **"Project overload"**

  Mental congestion when too much multitasking

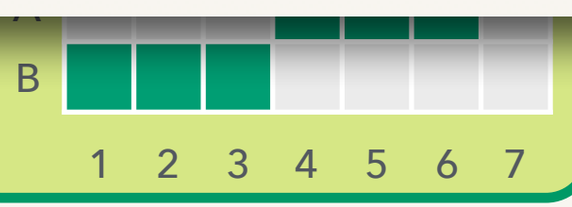
  (Zika-Viktorsson, Sundstrom, Engwall, 2006)

In theory:





VS.



More within-day multitasking

Higher focus

More repetitive day-to-day work

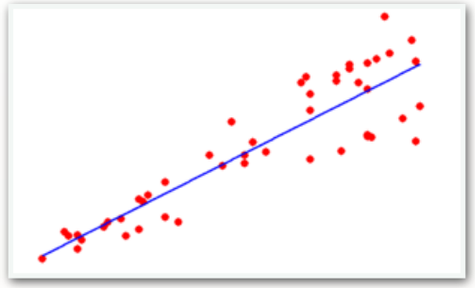
Interaction effects:

No scheduling is productive over 5 projects/week

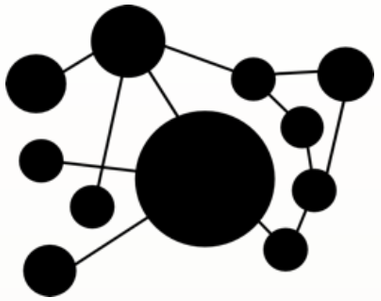
TOOLKIT FOR SOCIAL SOFTWARE ENGINEERING RESEARCHERS

THEORY

QUALITATIVE METHODS



STATISTICS



NETWORK SCIENCE

EXAMPLE 1: PULL REQUEST EVALUATION TIME

MODELS

M2: M1 + process-related factors + continuous integration

✓ R^2 = 58.7%

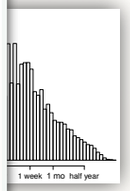
Title & description
- n_tokens

Priority
- time_to_first _response

Continuous Integration
- response time

Management
- workload
- availability

Social tagging
- @mention
- #issue



1 2 3 4 5 6 7
Day

THEORY: HOW DOES MULTITASKING AFFECT PERFORMANCE?

PROS

Fill downtime

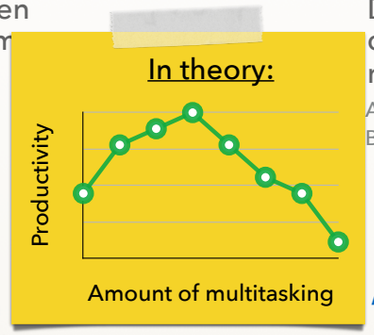
Switch focus between projects to utilize tim more efficiently

(Adler and Benbunan-Fich, 2012)

Cross-fertilisation

Easier to work on other projects if knowledge is transferrable

(Lindbeck and Snower, 2000)



CONS

Cognitive switching cost

Depends on interruption duration, complexity, moment

Altmann and Trafton, 2002)
Borst, Taatgen, van Rijn, 2015)

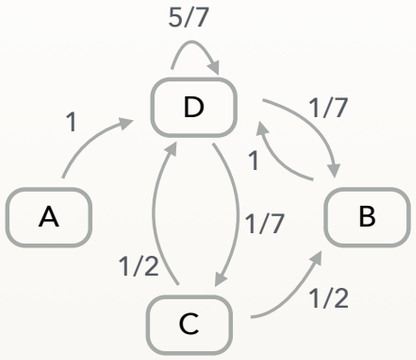
"Project overload"

Mental congestion when too much multitasking

(Zika-Viktorsson, Sundstrom, Engwall, 2006)

3. DAY-TO-DAY FOCUS

vs. Switching focus



Markov entropy: $S_{\text{Switch}} = -\sum_{i=1}^{N}\left[p_i \sum_{j\in\pi_i} p(j|i)\log_2 p(j|i)\right]$

How predictable is my focus tomorrow if today I work on project j?