Dajana Günther

daj@trifork.com

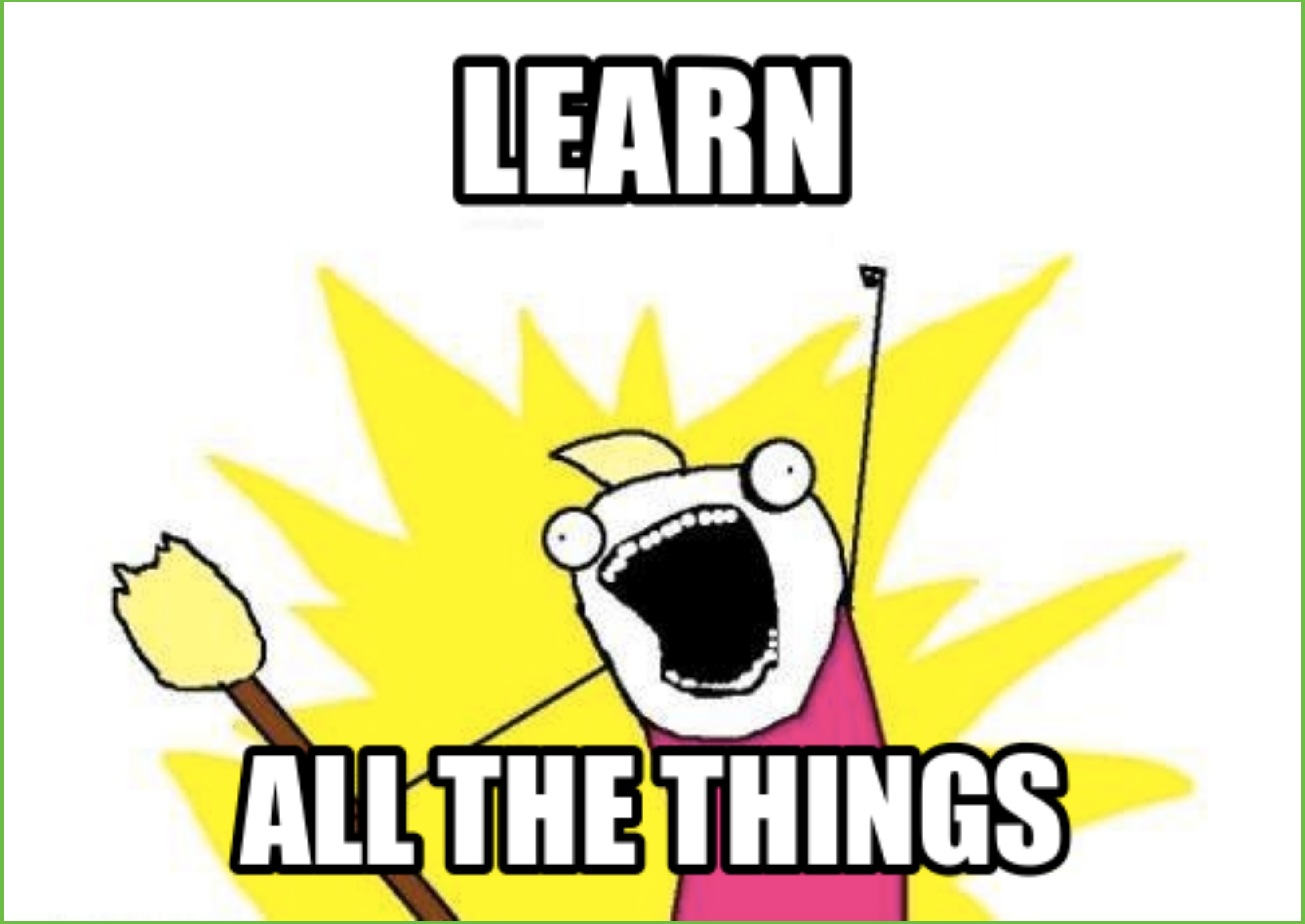
@dajanaguenther

Junior goto; conference

Why?

- old, dusty, **public** school system
- new literacy
- there’s a need!

What?



Junior goto; conference

How?

- weekend workshops
- several different topics
- working with mentors out of the communities
- max. 3 juniors to 1 mentor

Next steps

1. talk to local IT communities

2. set up a program

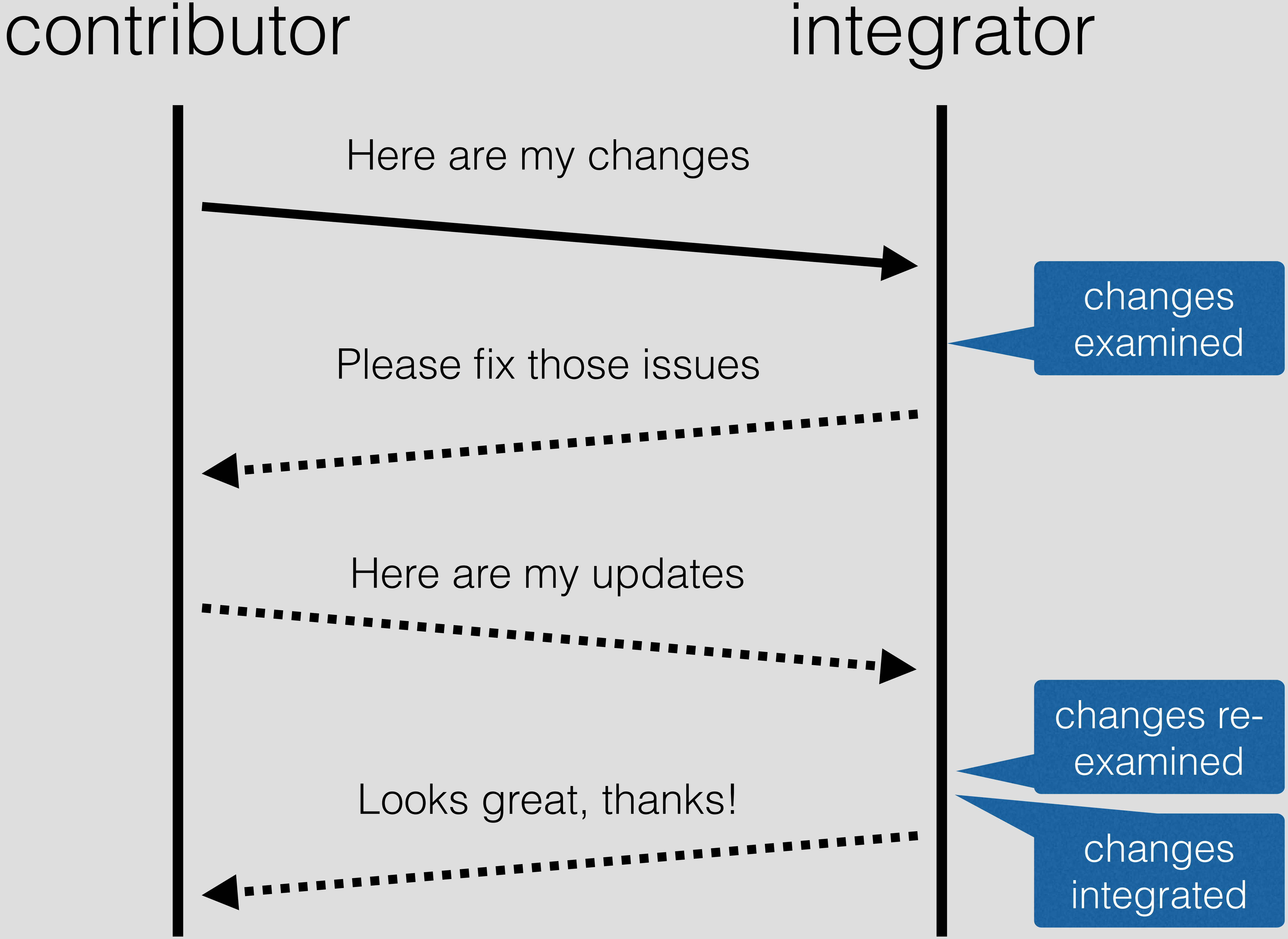
3. find
    - mentors
    - facilities
    - sponsors
    - juniors!

Questions?

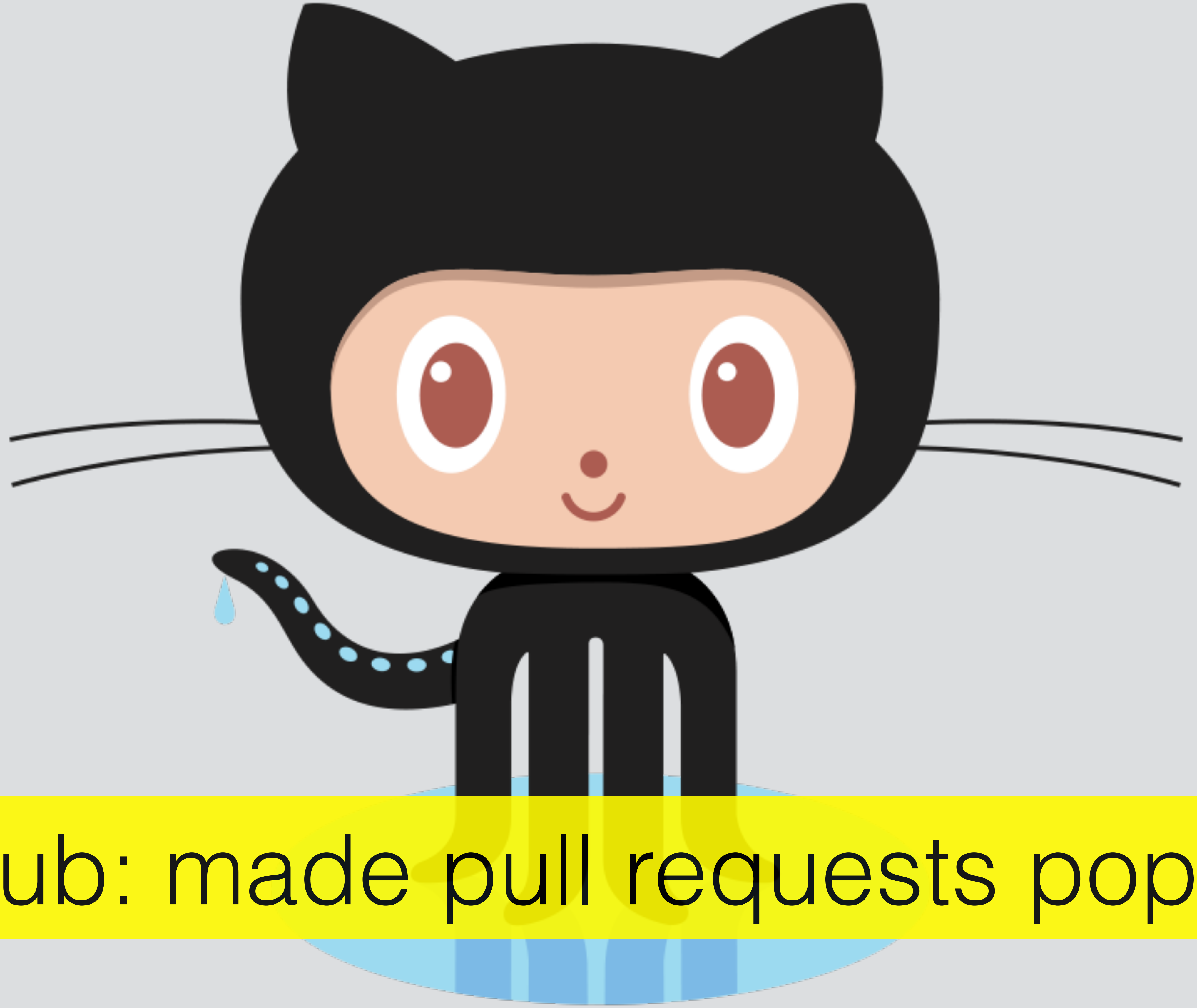
cat tax

Working Effectively with Pull Requests

Georgios Gousios // **@gousiosg**
Radboud University Nijmegen



The pull-based development model



GitHub: made pull requests popular

Projects with > 1 committers

An Exploratory Study of the Pull-based Software Development Model

Georgios Gousios
Delft University of Technology
Delft, The Netherlands
G.Gousios@tudelft.nl

Martin Pinzger
University of Klagenfurt
Klagenfurt, Austria
martin.pinzger@aau.at

Arie van Deursen
Delft University of Technology
Delft, The Netherlands
Arie.vandeursen@tudelft.nl

ABSTRACT

The advent of distributed version control systems has led to the development of a new paradigm for distributed software development; instead of pushing changes to a central repository, developers pull them from other repositories and merge them locally. Various code hosting sites, notably Github, have tapped on the opportunity to facilitate pull-based development by offering workflow support tools, such as code reviewing systems and integrated issue trackers. In this work, we explore how pull-based software development works, first on the GHTorrent corpus and then on a carefully selected sample of 291 projects. We find that the pull request model offers fast turnaround, increased opportunities for community engagement and decreased time to incorporate contributions. We show that a relatively small number of factors affect both the decision to merge a pull request and the time to process it. We also examine the reasons for pull request rejection and find that technical ones are only a small minority.

Categories and Subject Descriptors

D.2.7 [**Software Engineering**]: Distribution, Maintenance, and Enhancement—*Version control*; D.2.9 [**Software Engineering**]: Management—*Programming teams*

General Terms

Management

Keywords

pull-based development, pull request, distributed software development, empirical software engineering

1. INTRODUCTION

Pull-based development is an emerging paradigm for distributed software development. As more developers appreciate isolated development and branching [7], more projects, both closed source and, especially, open source, are being migrated to code hosting sites such as Github and Bitbucket with support for pull-based development [2]. A unique characteristic of such sites is that they allow any user to clone any public repository. The clone creates a public project that belongs to the user that cloned it, so the user can modify the repository without being part of the development team. Furthermore, such sites automate the selective contribution of commits from the clone to the source through pull requests.

Pull requests as a distributed development model in general, and as implemented by Github in particular, form a new method for collaborating on distributed software development. The novelty lays in the decoupling of the development effort from the decision to incorporate the results of the development in the code base. By separating the concerns of building artifacts and integrating changes, work is cleanly distributed between a contributor team that submits, often occasional, changes to be considered for merging and a core team that oversees the merge process, providing feedback, conducting tests, requesting changes, and finally accepting the contributions.

Previous work has identified the processes of collaboration in distributed development through patch submission and acceptance [23, 5, 32]. There are many similarities to the way pull requests work; for example, similar work team structures emerge, since typically pull requests go through an assessment process. What pull requests offer in addition is process automation and centralization of information. With pull requests, the code does not have to leave the revision control system, and therefore it can be versioned across repositories, while authorship information is effortlessly maintained. Communication about the change is context-specific, being rooted on a single pull request. Moreover, the review mechanism that Github incorporates has the additional effect of improving awareness [9]; core developers can access in an efficient way all information that relates to a pull request and solicit opinions of the community ("crowd-source") about the merging decision.

A distributed development workflow is effective if pull requests are eventually accepted, and it is efficient if the time this takes is as short as possible. Advancing our insight in the effectiveness and efficiency of pull request handling is of direct interest to contributors and developers alike. The goal of this work is to obtain a deep understanding of pull request usage and to analyze the factors that affect the efficiency of the pull-based software development model. Specifically, the questions we are trying to answer are:

RQ1 How popular is the pull based development model?

RQ2 What are the lifecycle characteristics of pull requests?

RQ3 What factors affect the decision and the time required to merge a pull request?

RQ4 Why are some pull requests not merged?

Our study is based on data from the Github collaborative development forge, as made available through our GHTorrent project [16].

45% of *collaborative* projects

Projects with > 1 committers

55% use shared repository

45% use pull requests

An Exploratory Study of the Pull-based Software Development Model

Georgios Gousios
Delft University of Technology
Delft, The Netherlands
G.Gousios@tudelft.nl

Martin Pinzger
University of Klagenfurt
Klagenfurt, Austria
martin.pinzger@aau.at

Arie van Deursen
Delft University of Technology
Delft, The Netherlands
Arie.vandeursen@tudelft.nl

ABSTRACT

The advent of distributed version control systems has led to the development of a new paradigm for distributed software development; instead of pushing changes to a central repository, developers pull them from other repositories and merge them locally. Various code hosting sites, notably Github, have tapped on the opportunity to facilitate pull-based development by offering workflow support tools, such as code reviewing systems and integrated issue trackers. In this work, we explore how pull-based software development works, first on the GHTorrent corpus and then on a carefully selected sample of 291 projects. We find that the pull request model offers fast turnaround, increased opportunities for community engagement and decreased time to incorporate contributions. We show that a relatively small number of factors affect both the decision to merge a pull request and the time to process it. We also examine the reasons for pull request rejection and find that technical ones are only a small minority.

Categories and Subject Descriptors

D.2.7 [**Software Engineering**]: Distribution, Maintenance, and Enhancement—*Version control*; D.2.9 [**Software Engineering**]: Management—*Programming teams*

General Terms

Management

Keywords

pull-based development, pull request, distributed software development, empirical software engineering

1. INTRODUCTION

Pull-based development is an emerging paradigm for distributed software development. As more developers appreciate isolated development and branching [7], more projects, both closed source and, especially, open source, are being migrated to code hosting sites such as Github and Bitbucket with support for pull-based development [2]. A unique characteristic of such sites is that they allow any user to clone any public repository. The clone creates a public project that belongs to the user that cloned it, so the user can modify the repository without being part of the development team. Furthermore, such sites automate the selective contribution of commits from the clone to the source through pull requests.

Pull requests as a distributed development model in general, and as implemented by Github in particular, form a new method for collaborating on distributed software development. The novelty lays in the decoupling of the development effort from the decision to incorporate the results of the development in the code base. By separating the concerns of building artifacts and integrating changes, work is cleanly distributed between a contributor team that submits, often occasional, changes to be considered for merging and a core team that oversees the merge process, providing feedback, conducting tests, requesting changes, and finally accepting the contributions.

Previous work has identified the processes of collaboration in distributed development through patch submission and acceptance [23, 5, 32]. There are many similarities to the way pull requests work; for example, similar work team structures emerge, since typically pull requests go through an assessment process. What pull requests offer in addition is process automation and centralization of information. With pull requests, the code does not have to leave the revision control system, and therefore it can be versioned across repositories, while authorship information is effortlessly maintained. Communication about the change is context-specific, being rooted on a single pull request. Moreover, the review mechanism that Github incorporates has the additional effect of improving awareness [9]; core developers can access in an efficient way all information that relates to a pull request and solicit opinions of the community ("crowd-source") about the merging decision.

A distributed development workflow is effective if pull requests are eventually accepted, and it is efficient if the time this takes is as short as possible. Advancing our insight in the effectiveness and efficiency of pull request handling is of direct interest to contributors and developers alike. The goal of this work is to obtain a deep understanding of pull request usage and to analyze the factors that affect the efficiency of the pull-based software development model. Specifically, the questions we are trying to answer are:

RQ1 How popular is the pull based development model?

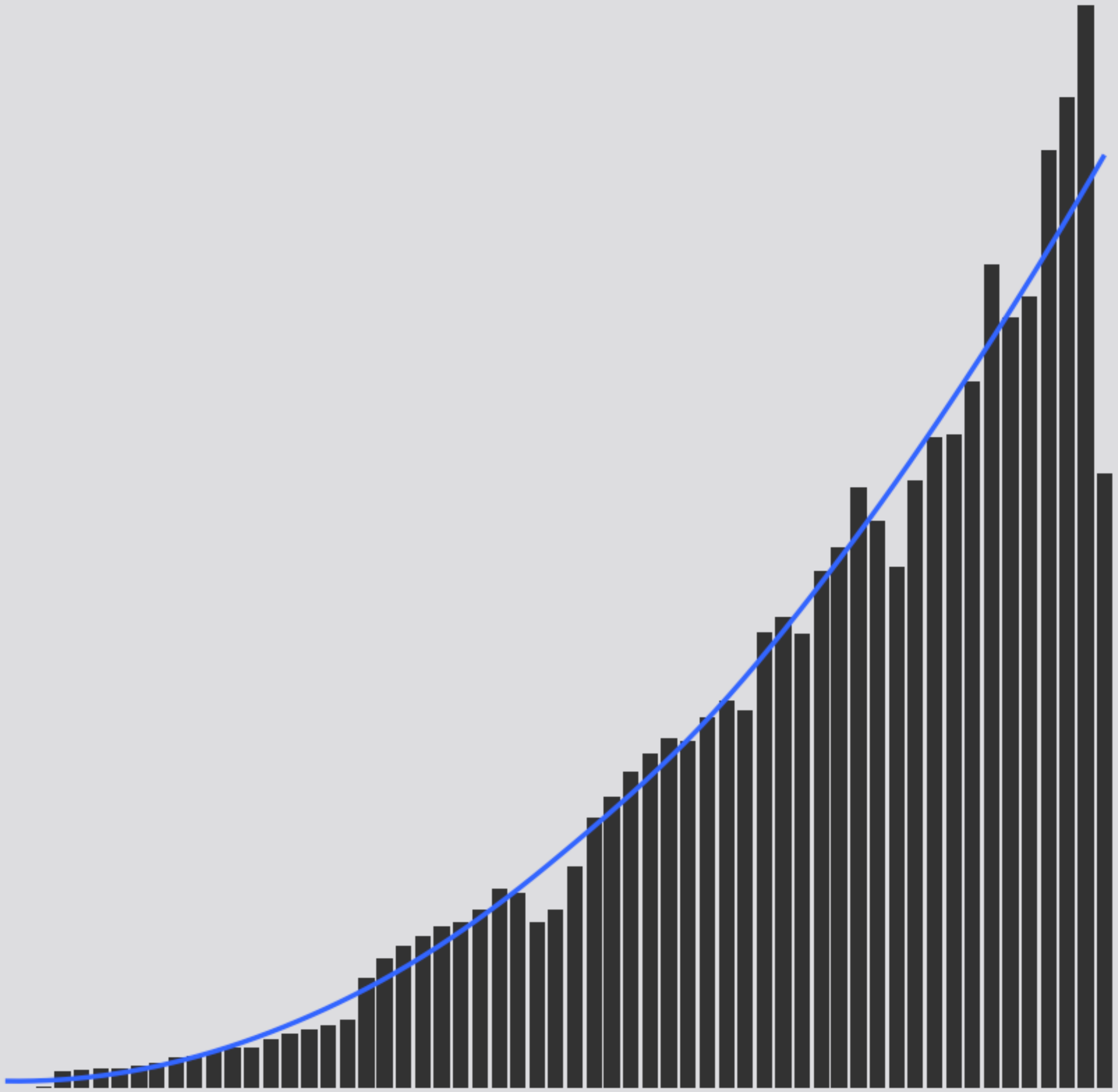
RQ2 What are the lifecycle characteristics of pull requests?

RQ3 What factors affect the decision and the time required to merge a pull request?

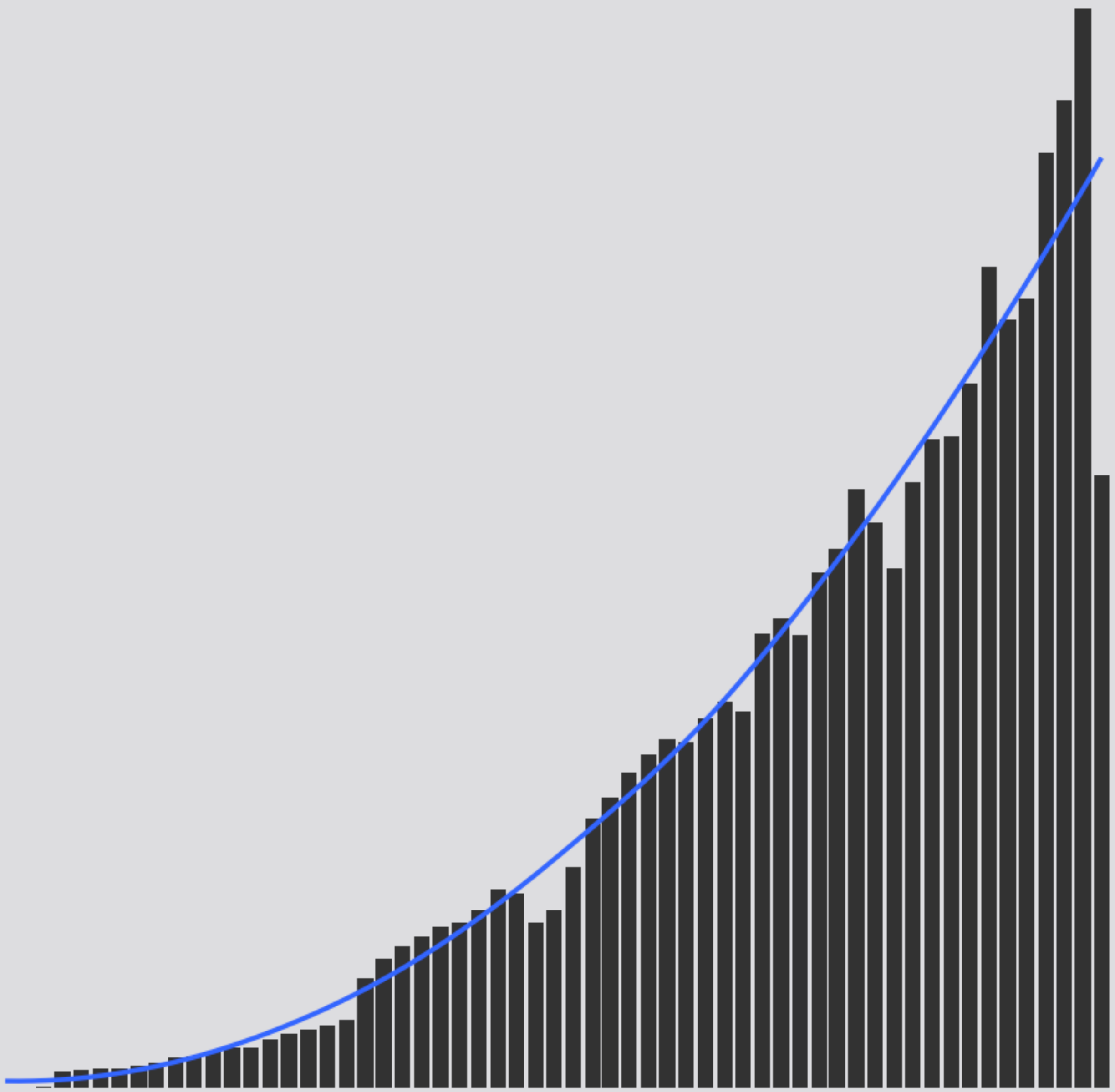
RQ4 Why are some pull requests not merged?

Our study is based on data from the Github collaborative development forge, as made available through our GHTorrent project [16].

45% of *collaborative* projects



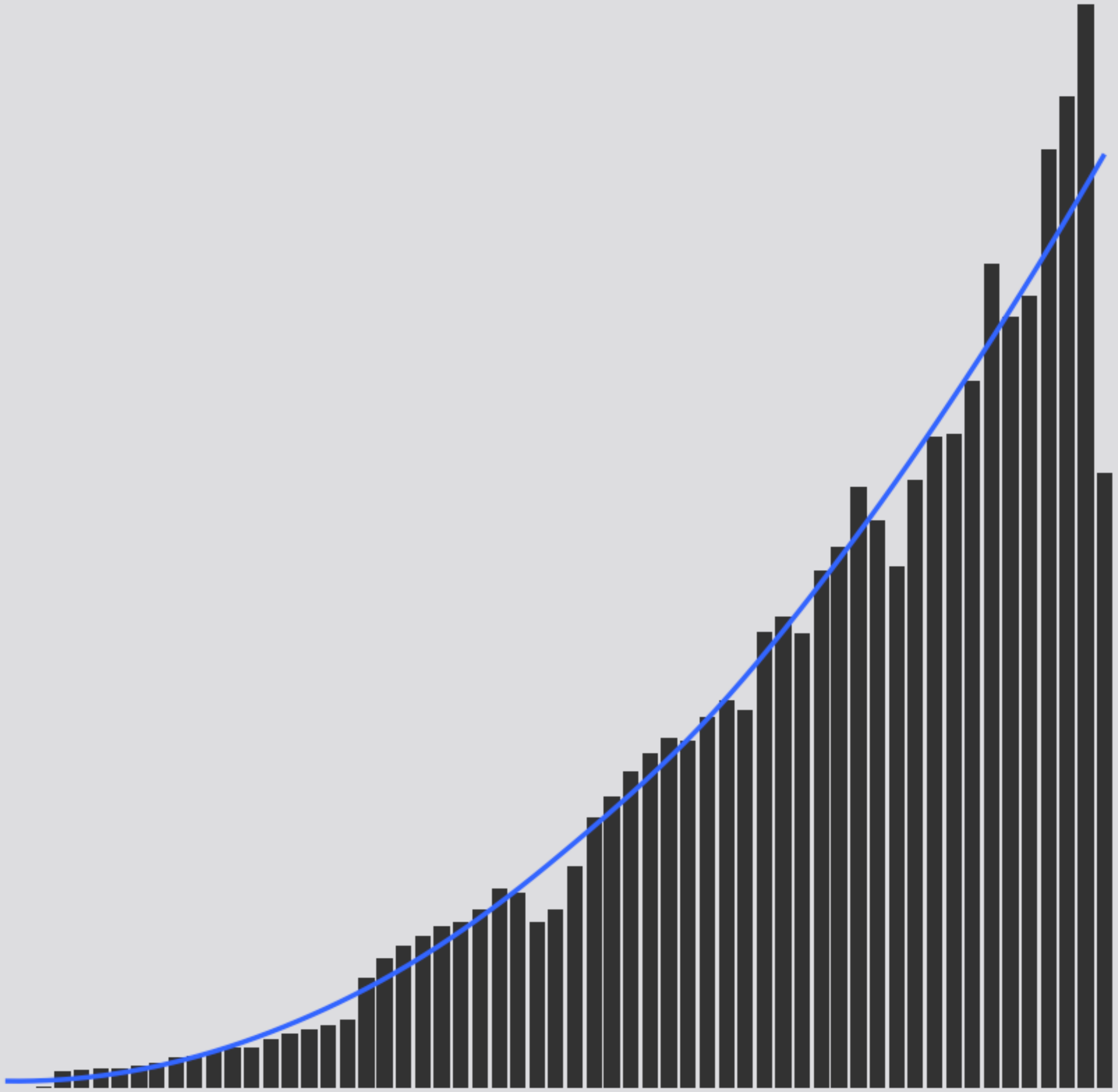
Widely popular and increasing



90k repositories

400k pull requests

Widely popular and increasing



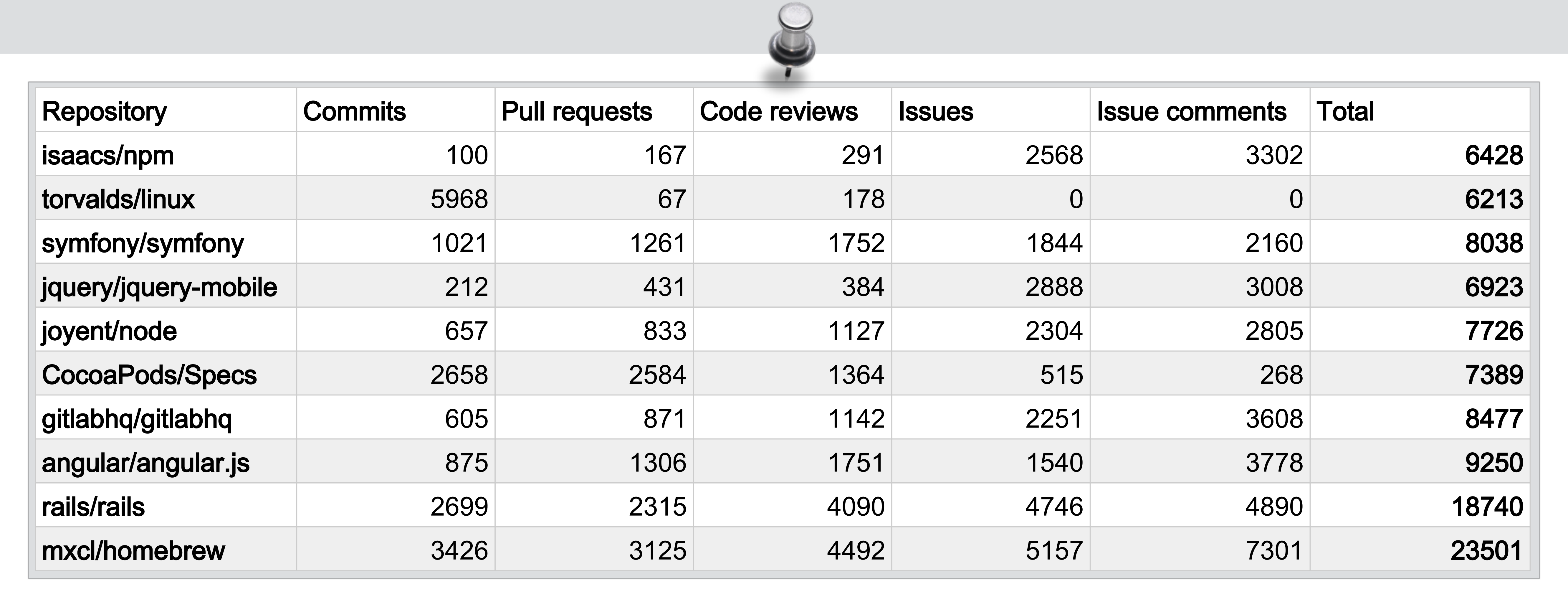
Per month

90k repositories

400k pull requests

Widely popular and increasing

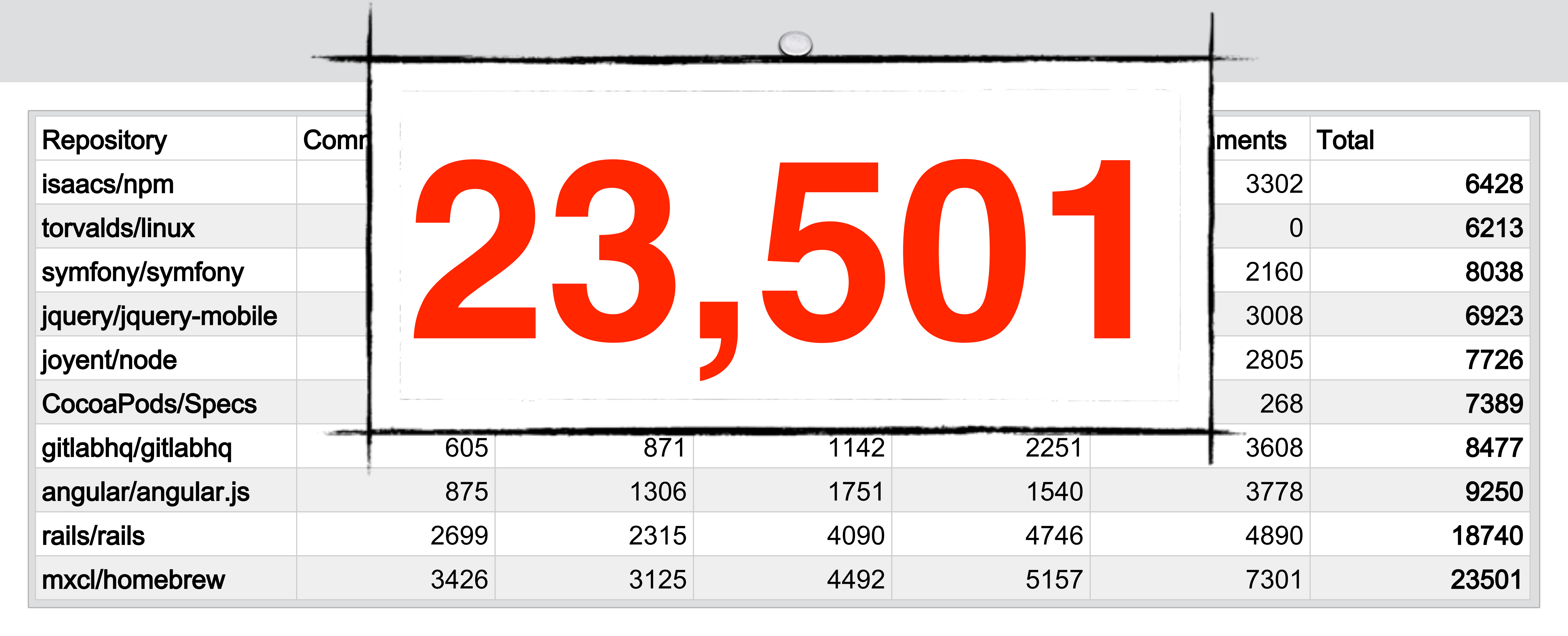
Large scale collaboration



Repository	Commits	Pull requests	Code reviews	Issues	Issue comments	Total
isaacs/npm	100	167	291	2568	3302	**6428**
torvalds/linux	5968	67	178	0	0	**6213**
symfony/symfony	1021	1261	1752	1844	2160	**8038**
jquery/jquery-mobile	212	431	384	2888	3008	**6923**
joyent/node	657	833	1127	2304	2805	**7726**
CocoaPods/Specs	2658	2584	1364	515	268	**7389**
gitlabhq/gitlabhq	605	871	1142	2251	3608	**8477**
angular/angular.js	875	1306	1751	1540	3778	**9250**
rails/rails	2699	2315	4090	4746	4890	**18740**
mxcl/homebrew	3426	3125	4492	5157	7301	**23501**

http://www.gousios.gr/blog/The-triumph-of-online-collaboration/

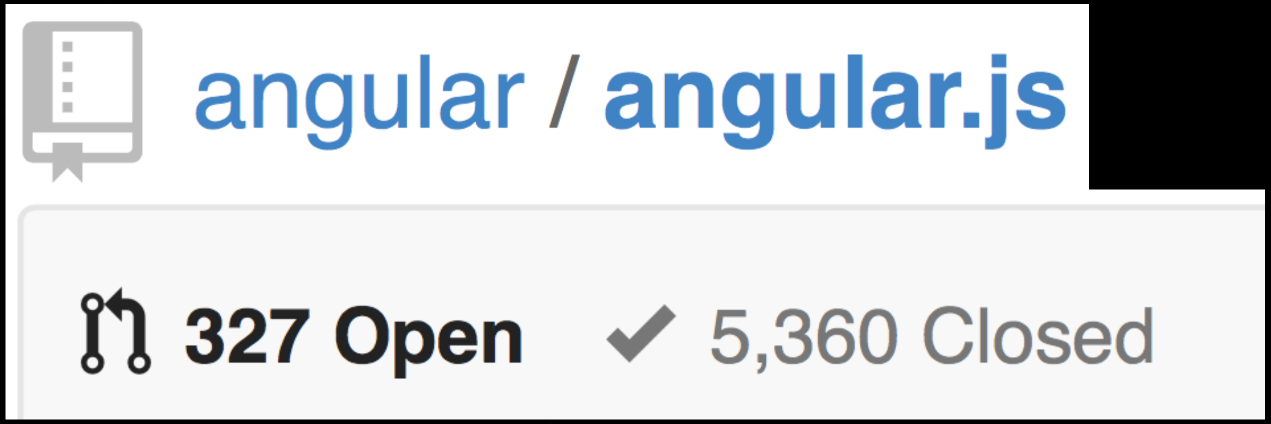
Large scale collaboration



23,501

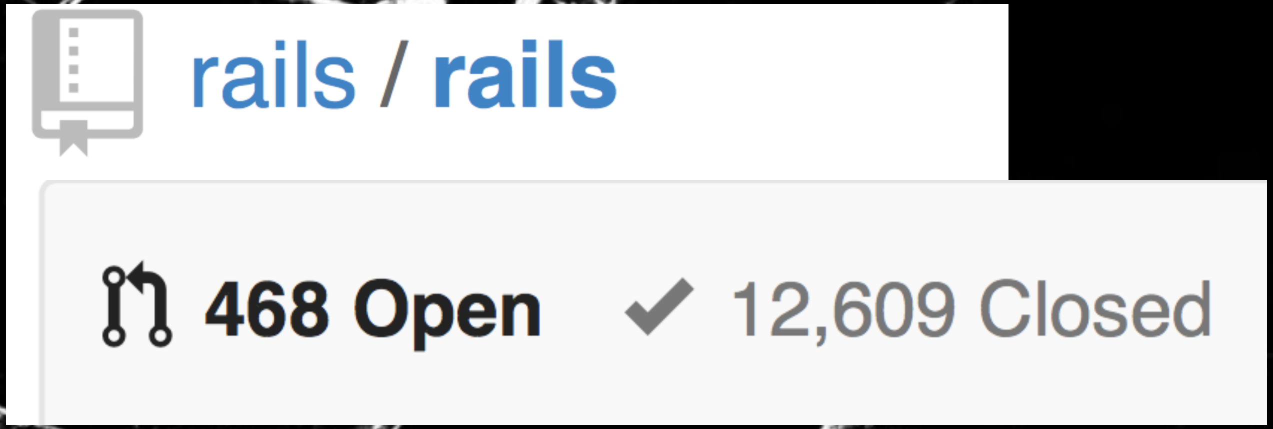
Repository	Comm				ments	Total
isaacs/npm					3302	6428
torvalds/linux					0	6213
symfony/symfony					2160	8038
jquery/jquery-mobile					3008	6923
joyent/node					2805	7726
CocoaPods/Specs					268	7389
gitlabhq/gitlabhq	605	871	1142	2251	3608	8477
angular/angular.js	875	1306	1751	1540	3778	9250
rails/rails	2699	2315	4090	4746	4890	18740
mxcl/homebrew	3426	3125	4492	5157	7301	23501

http://www.gousios.gr/blog/The-triumph-of-online-collaboration/



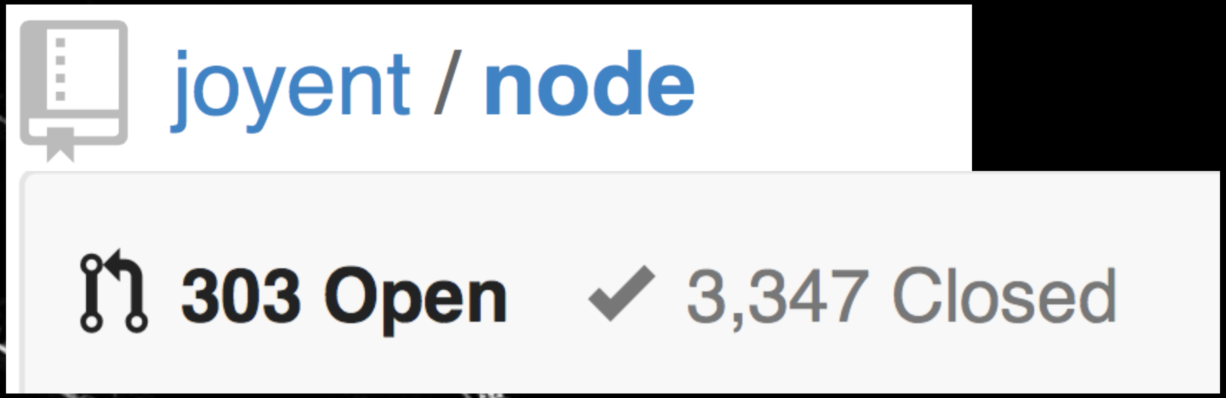
angular / angular.js

327 Open ✓ 5,360 Closed



rails / rails

468 Open ✓ 12,609 Closed

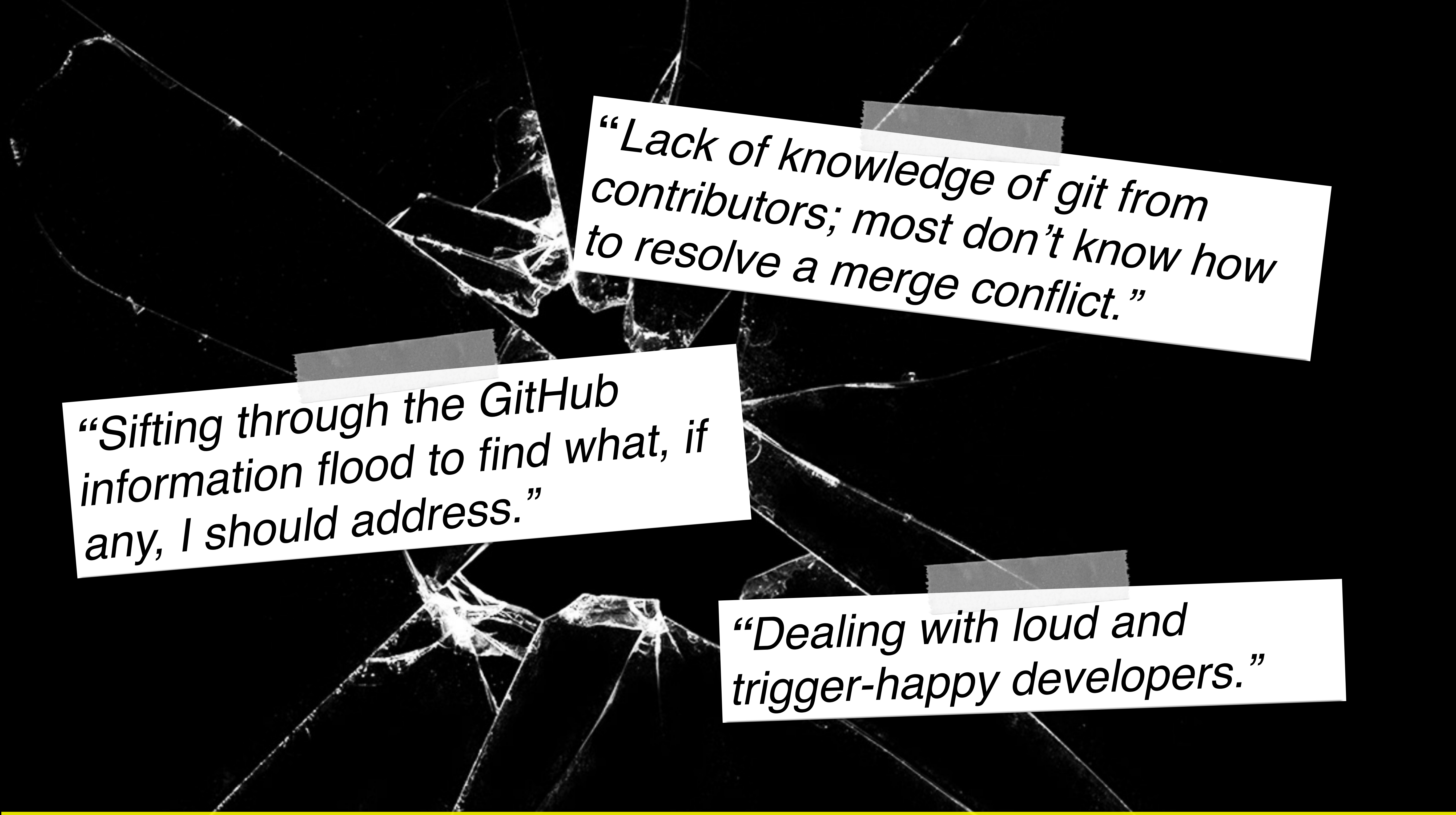


joyent / node

303 Open ✓ 3,347 Closed

Too successful?

Too successful?



“Lack of knowledge of git from contributors; most don’t know how to resolve a merge conflict.”

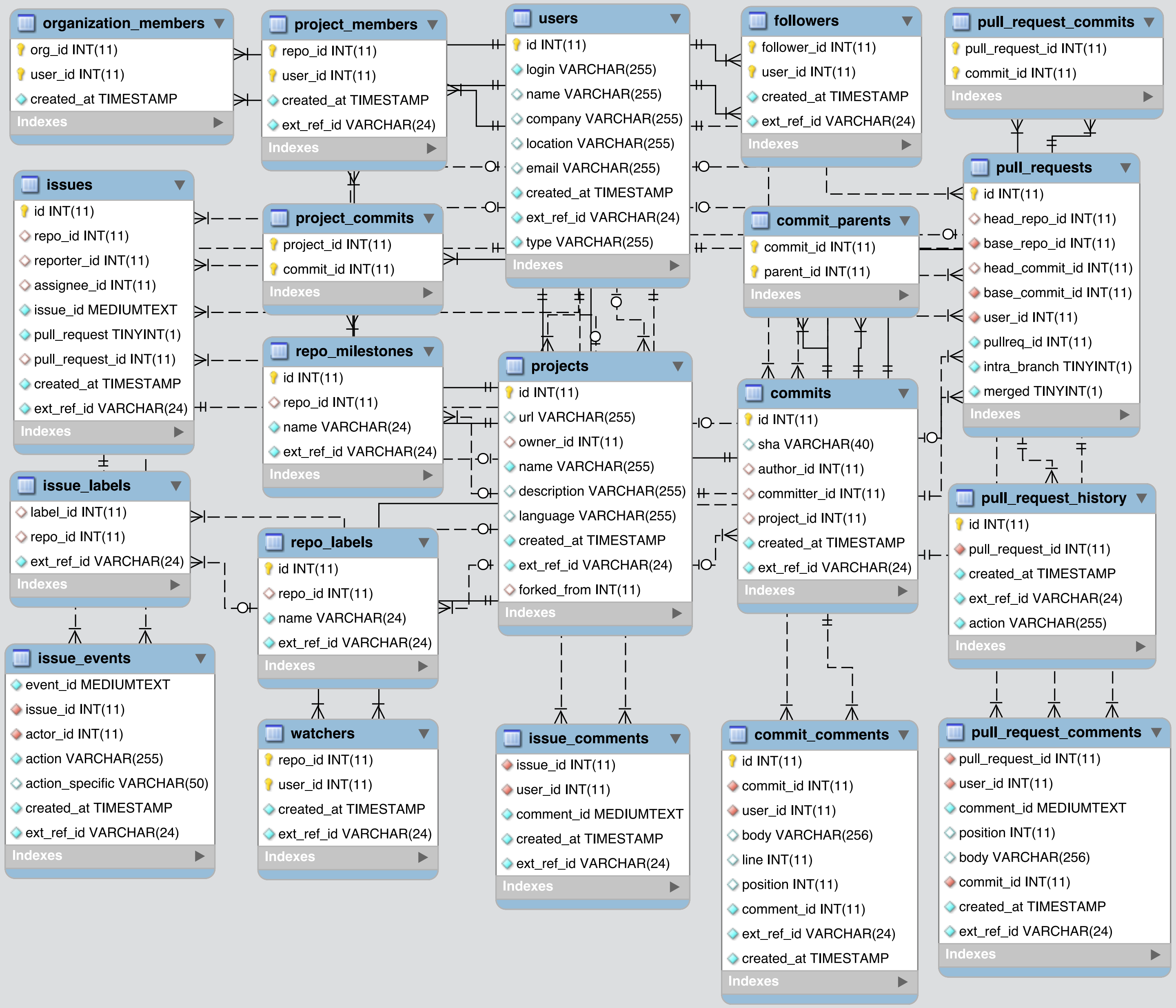
“Sifting through the GitHub information flood to find what, if any, I should address.”

“Dealing with loud and trigger-happy developers.”

GitHub Research @ RU Nijmegen

GHTorrent.org

- Query-able offline mirror of all data from the GitHub API
- Since Feb 2012
- **7TB** in MongoDB
- **1,600B** rows in MySQL
- **2GB** per hour



organization_members
- org_id INT(11)
- user_id INT(11)
- created_at TIMESTAMP
- Indexes

project_members
- repo_id INT(11)
- user_id INT(11)
- created_at TIMESTAMP
- ext_ref_id VARCHAR(24)
- Indexes

users
- id INT(11)
- login VARCHAR(255)
- name VARCHAR(255)
- company VARCHAR(255)
- location VARCHAR(255)
- email VARCHAR(255)
- created_at TIMESTAMP
- ext_ref_id VARCHAR(24)
- type VARCHAR(255)
- Indexes

followers
- follower_id INT(11)
- user_id INT(11)
- created_at TIMESTAMP
- ext_ref_id VARCHAR(24)
- Indexes

pull_request_commits
- pull_request_id INT(11)
- commit_id INT(11)
- Indexes

issues
- id INT(11)
- repo_id INT(11)
- reporter_id INT(11)
- assignee_id INT(11)
- issue_id MEDIUMTEXT
- pull_request TINYINT(1)
- pull_request_id INT(11)
- created_at TIMESTAMP
- ext_ref_id VARCHAR(24)
- Indexes

project_commits
- project_id INT(11)
- commit_id INT(11)
- Indexes

commit_parents
- commit_id INT(11)
- parent_id INT(11)
- Indexes

pull_requests
- id INT(11)
- head_repo_id INT(11)
- base_repo_id INT(11)
- head_commit_id INT(11)
- base_commit_id INT(11)
- user_id INT(11)
- pullreq_id INT(11)
- intra_branch TINYINT(1)
- merged TINYINT(1)
- Indexes

repo_milestones
- id INT(11)
- repo_id INT(11)
- name VARCHAR(24)
- ext_ref_id VARCHAR(24)
- Indexes

projects
- id INT(11)
- url VARCHAR(255)
- owner_id INT(11)
- name VARCHAR(255)
- description VARCHAR(255)
- language VARCHAR(255)
- created_at TIMESTAMP
- ext_ref_id VARCHAR(24)
- forked_from INT(11)
- Indexes

commits
- id INT(11)
- sha VARCHAR(40)
- author_id INT(11)
- committer_id INT(11)
- project_id INT(11)
- created_at TIMESTAMP
- ext_ref_id VARCHAR(24)
- Indexes

issue_labels
- label_id INT(11)
- repo_id INT(11)
- ext_ref_id VARCHAR(24)
- Indexes

repo_labels
- id INT(11)
- repo_id INT(11)
- name VARCHAR(24)
- ext_ref_id VARCHAR(24)
- Indexes

pull_request_history
- id INT(11)
- pull_request_id INT(11)
- created_at TIMESTAMP
- ext_ref_id VARCHAR(24)
- action VARCHAR(255)
- Indexes

issue_events
- event_id MEDIUMTEXT
- issue_id INT(11)
- actor_id INT(11)
- action VARCHAR(255)
- action_specific VARCHAR(50)
- created_at TIMESTAMP
- ext_ref_id VARCHAR(24)
- Indexes

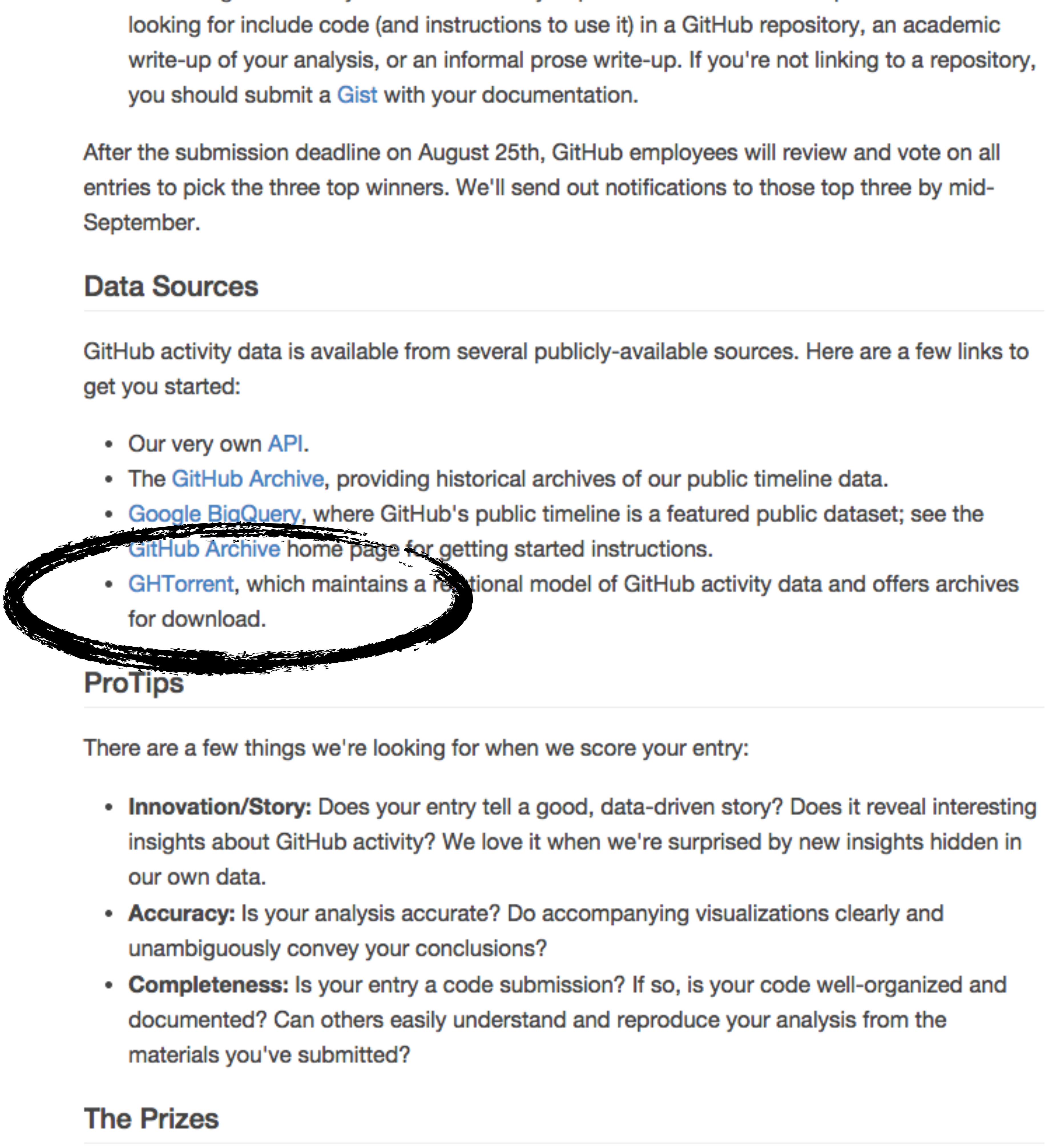
watchers
- repo_id INT(11)
- user_id INT(11)
- created_at TIMESTAMP
- ext_ref_id VARCHAR(24)
- Indexes

issue_comments
- issue_id INT(11)
- user_id INT(11)
- comment_id MEDIUMTEXT
- created_at TIMESTAMP
- ext_ref_id VARCHAR(24)
- Indexes

commit_comments
- id INT(11)
- commit_id INT(11)
- user_id INT(11)
- body VARCHAR(256)
- line INT(11)
- position INT(11)
- comment_id INT(11)
- ext_ref_id VARCHAR(24)
- created_at TIMESTAMP
- Indexes

pull_request_comments
- pull_request_id INT(11)
- user_id INT(11)
- comment_id MEDIUMTEXT
- position INT(11)
- body VARCHAR(256)
- commit_id INT(11)
- created_at TIMESTAMP
- ext_ref_id VARCHAR(24)
- Indexes

Georgios Gousios: The GHTorrent dataset and tool suite. MSR 2013: 233-236



looking for include code (and instructions to use it) in a GitHub repository, an academic write-up of your analysis, or an informal prose write-up. If you're not linking to a repository, you should submit a Gist with your documentation.

After the submission deadline on August 25th, GitHub employees will review and vote on all entries to pick the three top winners. We'll send out notifications to those top three by mid-September.

Data Sources

GitHub activity data is available from several publicly-available sources. Here are a few links to get you started:

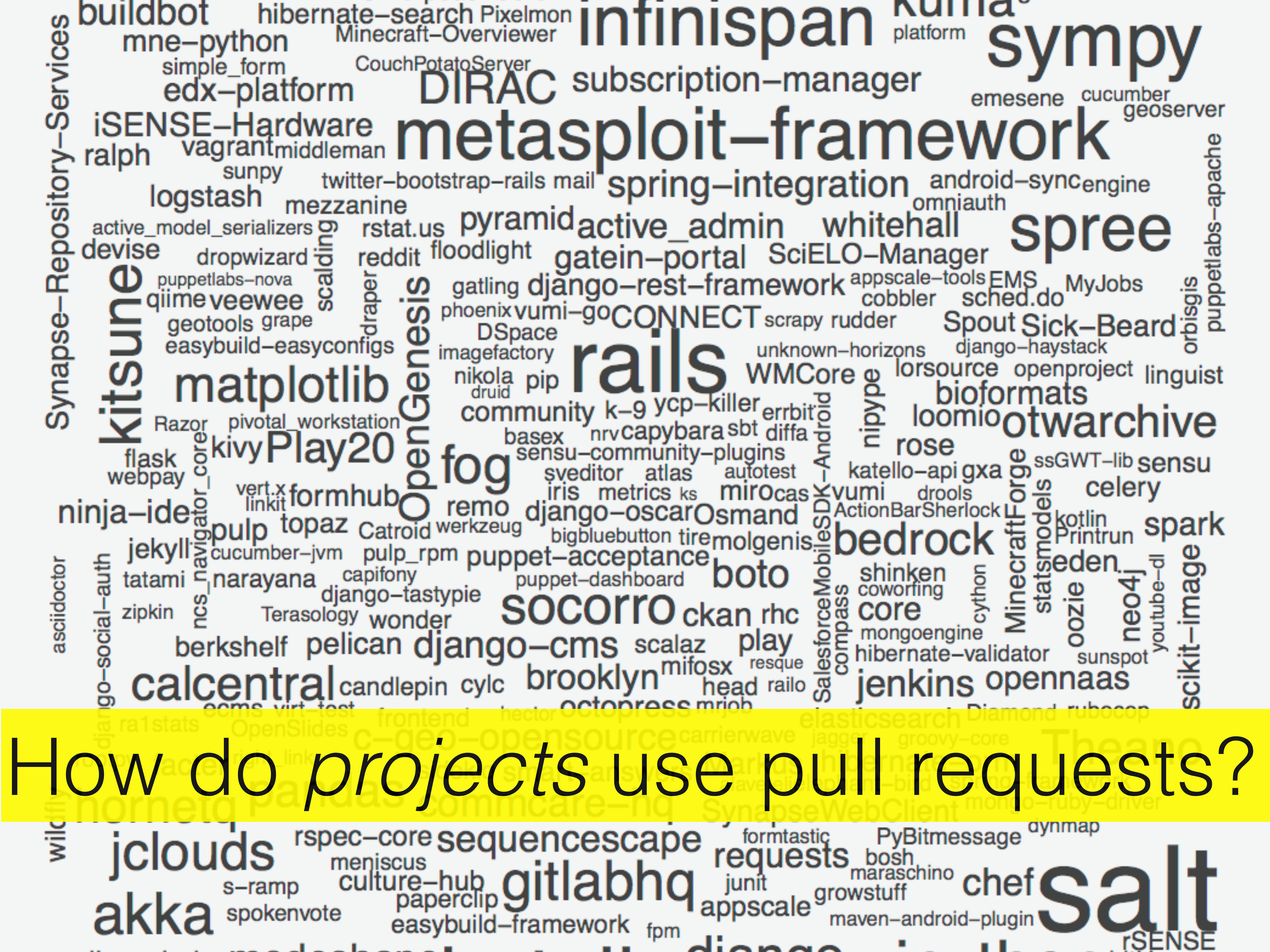
- Our very own API.
- The GitHub Archive, providing historical archives of our public timeline data.
- Google BigQuery, where GitHub's public timeline is a featured public dataset; see the GitHub Archive home page for getting started instructions.
- GHTorrent, which maintains a relational model of GitHub activity data and offers archives for download.

ProTips

There are a few things we're looking for when we score your entry:

- **Innovation/Story:** Does your entry tell a good, data-driven story? Does it reveal interesting insights about GitHub activity? We love it when we're surprised by new insights hidden in our own data.
- **Accuracy:** Is your analysis accurate? Do accompanying visualizations clearly and unambiguously convey your conclusions?
- **Completeness:** Is your entry a code submission? If so, is your code well-organized and documented? Can others easily understand and reproduce your analysis from the materials you've submitted?

The Prizes



How do *projects* use pull requests?

Pull Requests

Georgios Gousios, Martin Pinzger and Arie van Deursen:An exploratory study of the pull-based software development model. ICSE 2014: 345-355

Pull Requests

are small (< 20 lines); merged in < 1 day; are briefly discussed

Georgios Gousios, Martin Pinzger and Arie van Deursen:An exploratory study of the pull-based software development model. ICSE 2014: 345-355

Pull Requests

are small (< 20 lines); merged in < 1 day; are briefly discussed

are merged when they affect a hot project area

Georgios Gousios, Martin Pinzger and Arie van Deursen:An exploratory study of the pull-based software development model. ICSE 2014: 345-355

Pull Requests

are small (< 20 lines); merged in < 1 day; are briefly discussed

are merged when they affect a hot project area

are processed fast when project has test suite

Georgios Gousios, Martin Pinzger and Arie van Deursen:An exploratory study of the pull-based software development model. ICSE 2014: 345-355

Pull Requests

are small (< 20 lines); merged in < 1 day; are briefly discussed

are merged when they affect a hot project area

are processed fast when project has test suite

are processed fast when contributor has good track record

Georgios Gousios, Martin Pinzger and Arie van Deursen:An exploratory study of the pull-based software development model. ICSE 2014: 345-355

Pull Requests

are small (< 20 lines); merged in < 1 day; are briefly discussed

are merged when they affect a hot project area

are processed fast when project has test suite

are processed fast when contributor has good track record

are rejected mostly due to insufficient task articulation

Georgios Gousios, Martin Pinzger and Arie van Deursen:An exploratory study of the pull-based software development model. ICSE 2014: 345-355

How do *developers* use of pull requests?

Technical challenges

G. Gousios, A. Zaidman, M.-A. Storey, and A. van Deursen, “Work Practices and Challenges in Pull-Based Development: The Integrator’s Perspective,” ICSE, 2015.

maintaining quality

Technical challenges

G. Gousios, A. Zaidman, M.-A. Storey, and A. van Deursen, “Work Practices and Challenges in Pull-Based Development: The Integrator’s Perspective,” ICSE, 2015.

maintaining quality

impact assessment

Technical challenges

G. Gousios, A. Zaidman, M.-A. Storey, and A. van Deursen, “Work Practices and Challenges in Pull-Based Development: The Integrator’s Perspective,” ICSE, 2015.

maintaining quality

impact assessment

contributor experience

Technical challenges

G. Gousios, A. Zaidman, M.-A. Storey, and A. van Deursen, "Work Practices and Challenges in Pull-Based Development: The Integrator's Perspective," ICSE, 2015.

maintaining quality

impact assessment

contributor experience

volume

Technical challenges

G. Gousios, A. Zaidman, M.-A. Storey, and A. van Deursen, “Work Practices and Challenges in Pull-Based Development: The Integrator’s Perspective,” ICSE, 2015.

maintaining quality

impact assessment

contributor experience

volume

understanding the code

Technical challenges

G. Gousios, A. Zaidman, M.-A. Storey, and A. van Deursen, “Work Practices and Challenges in Pull-Based Development: The Integrator’s Perspective,” ICSE, 2015.

maintaining quality

impact assessment

contributor experience

volume

understanding the code

compliance

Technical challenges

G. Gousios, A. Zaidman, M.-A. Storey, and A. van Deursen, “Work Practices and Challenges in Pull-Based Development: The Integrator’s Perspective,” ICSE, 2015.

maintaining quality

impact assessment

contributor experience

volume

understanding the code

compliance

infrastructure setup

Technical challenges

G. Gousios, A. Zaidman, M.-A. Storey, and A. van Deursen, “Work Practices and Challenges in Pull-Based Development: The Integrator’s Perspective,” ICSE, 2015.

Social challenges

G. Gousios, A. Zaidman, M.-A. Storey, and A. van Deursen, “Work Practices and Challenges in Pull-Based Development: The Integrator’s Perspective,” ICSE, 2015.

workload

Social challenges

G. Gousios, A. Zaidman, M.-A. Storey, and A. van Deursen, “Work Practices and Challenges in Pull-Based Development: The Integrator’s Perspective,” ICSE, 2015.

workload

responsiveness

Social challenges

G. Gousios, A. Zaidman, M.-A. Storey, and A. van Deursen, “Work Practices and Challenges in Pull-Based Development: The Integrator’s Perspective,” ICSE, 2015.

workload

responsiveness

explaining rejection

Social challenges

G. Gousios, A. Zaidman, M.-A. Storey, and A. van Deursen, “Work Practices and Challenges in Pull-Based Development: The Integrator’s Perspective,” ICSE, 2015.

workload

responsiveness

explaining rejection

explaining rationale

Social challenges

G. Gousios, A. Zaidman, M.-A. Storey, and A. van Deursen, “Work Practices and Challenges in Pull-Based Development: The Integrator’s Perspective,” ICSE, 2015.

workload

responsiveness

explaining rejection

explaining rationale

motivating contributors

Social challenges

G. Gousios, A. Zaidman, M.-A. Storey, and A. van Deursen, “Work Practices and Challenges in Pull-Based Development: The Integrator’s Perspective,” ICSE, 2015.

Advice for projects owners

http://varshadaswani.com/2014/01/giving-a-17-year-old-advice/

Project owners

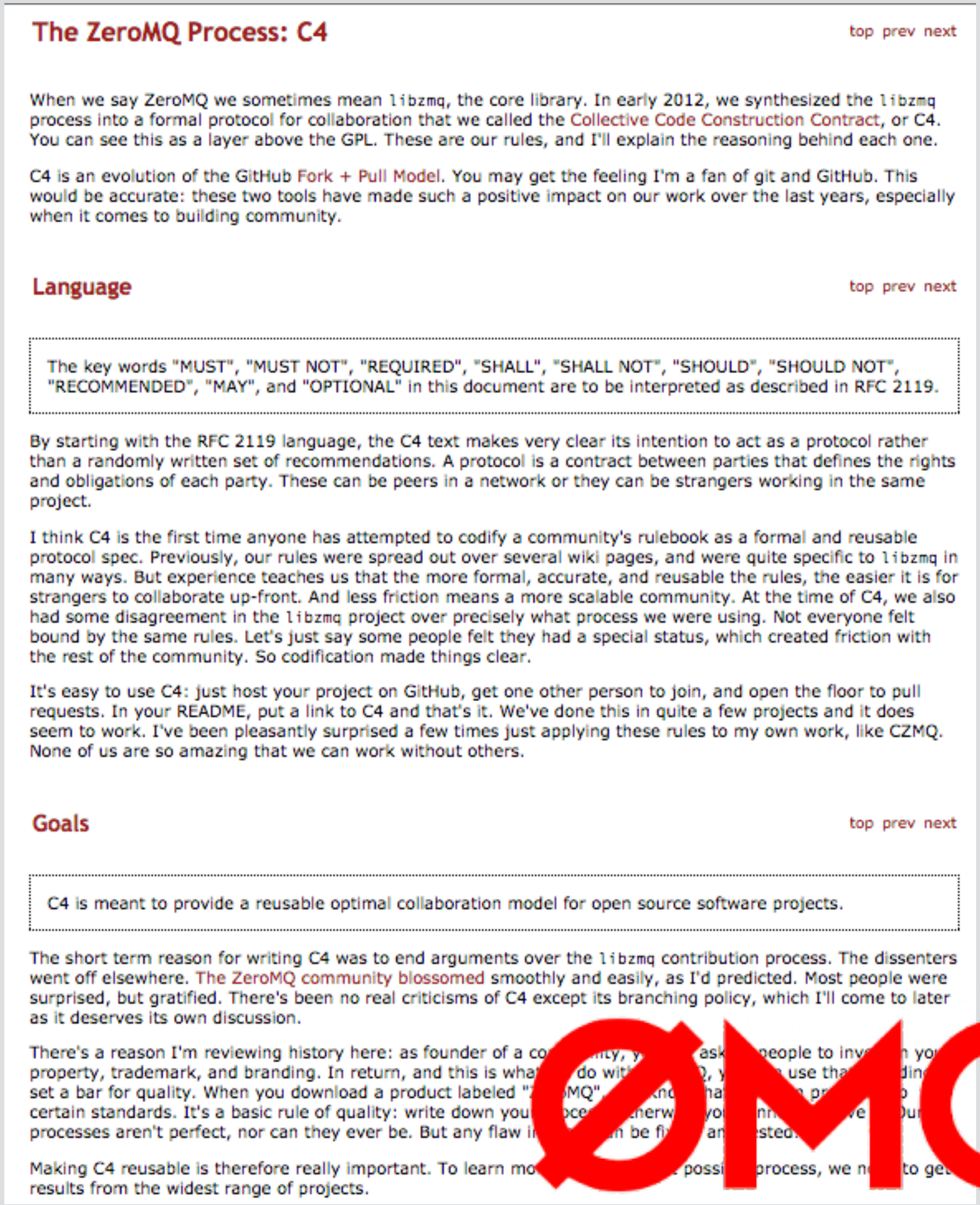
1. Provide submission guidelines
2. Invest in good tests
3. Automate
4. Be reactive and proactive
5. Monitor PR performance

Submission Guidelines

People generally expect a CONTRIBUTING.md file

- Setting up development environment
- Setting up branches
- Coding style
- Commit guidelines
- Communication options. PRs are post-hoc communication.
- Include updated list of low hanging fruit for beginners

Submission Guidelines



The ZeroMQ Process: C4

top prev next

When we say ZeroMQ we sometimes mean `libzmq`, the core library. In early 2012, we synthesized the `libzmq` process into a formal protocol for collaboration that we called the Collective Code Construction Contract, or C4. You can see this as a layer above the GPL. These are our rules, and I'll explain the reasoning behind each one.

C4 is an evolution of the GitHub Fork + Pull Model. You may get the feeling I'm a fan of git and GitHub. This would be accurate: these two tools have made such a positive impact on our work over the last years, especially when it comes to building community.

Language

top prev next

> The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

By starting with the RFC 2119 language, the C4 text makes very clear its intention to act as a protocol rather than a randomly written set of recommendations. A protocol is a contract between parties that defines the rights and obligations of each party. These can be peers in a network or they can be strangers working in the same project.

I think C4 is the first time anyone has attempted to codify a community's rulebook as a formal and reusable protocol spec. Previously, our rules were spread out over several wiki pages, and were quite specific to `libzmq` in many ways. But experience teaches us that the more formal, accurate, and reusable the rules, the easier it is for strangers to collaborate up-front. And less friction means a more scalable community. At the time of C4, we also had some disagreement in the `libzmq` project over precisely what process we were using. Not everyone felt bound by the same rules. Let's just say some people felt they had a special status, which created friction with the rest of the community. So codification made things clear.

It's easy to use C4: just host your project on GitHub, get one other person to join, and open the floor to pull requests. In your README, put a link to C4 and that's it. We've done this in quite a few projects and it does seem to work. I've been pleasantly surprised a few times just applying these rules to my own work, like CZMQ. None of us are so amazing that we can work without others.

Goals

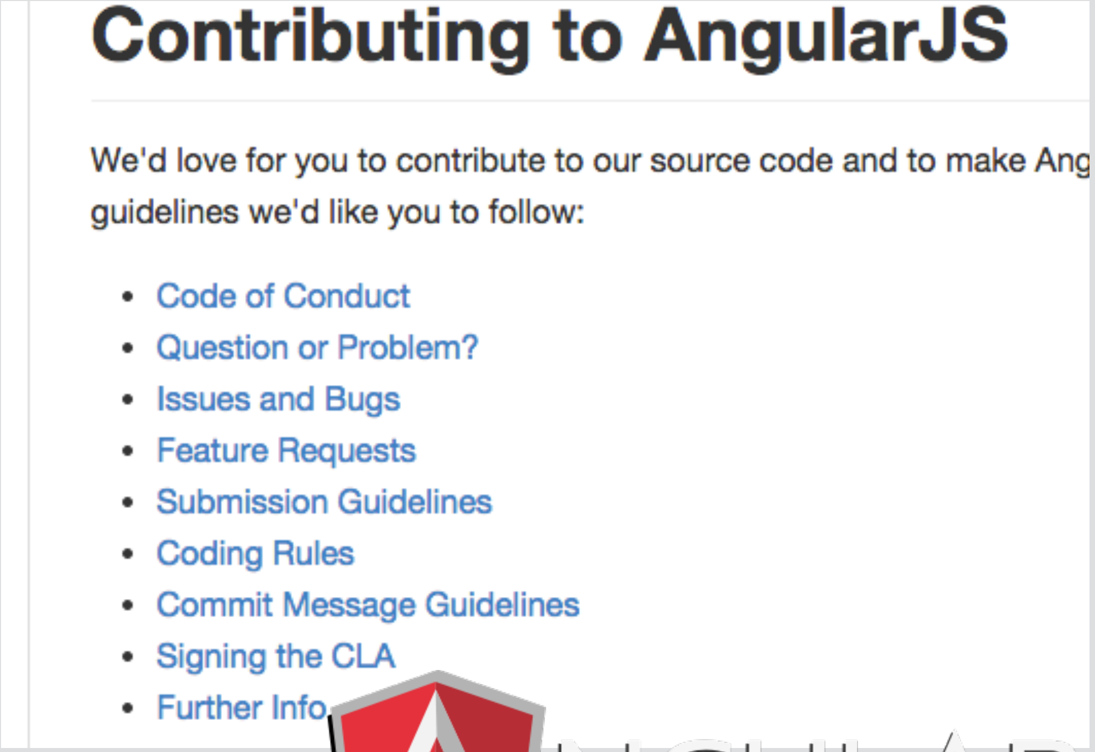
top prev next

> C4 is meant to provide a reusable optimal collaboration model for open source software projects.

The short term reason for writing C4 was to end arguments over the `libzmq` contribution process. The dissenters went off elsewhere. The ZeroMQ community blossomed smoothly and easily, as I'd predicted. Most people were surprised, but gratified. There's been no real criticisms of C4 except its branching policy, which I'll come to later as it deserves its own discussion.

There's a reason I'm reviewing history here: as founder of a co[illegible]ity, y[illegible] ask [illegible] people to inv[illegible] yo[illegible] property, trademark, and branding. In return, and this is wha[illegible] do wit[illegible]Q, y[illegible] use tha[illegible]din[illegible] set a bar for quality. When you download a product labeled "[illegible]MQ", [illegible] certain standards. It's a basic rule of quality: write down you[illegible] processes aren't perfect, nor can they ever be. But any flaw i[illegible]n be fi[illegible] an[illegible]sted.

Making C4 reusable is therefore really important. To learn mo[illegible] possi[illegible] process, we n[illegible] to get results from the widest range of projects.



Contributing to AngularJS

We'd love for you to contribute to our source code and to make Ang guidelines we'd like you to follow:

- Code of Conduct
- Question or Problem?
- Issues and Bugs
- Feature Requests
- Submission Guidelines
- Coding Rules
- Commit Message Guidelines
- Signing the CLA
- Further Info

Invest in good tests

- Run everything with a single command
- Use tools like Vagrant and Docker to simulate environments
- Measure PR coverage in CI

Automate

The Github API is brilliant. Take advantage of it.

- Building/CI

  Travis Jenkins

- Quality checks

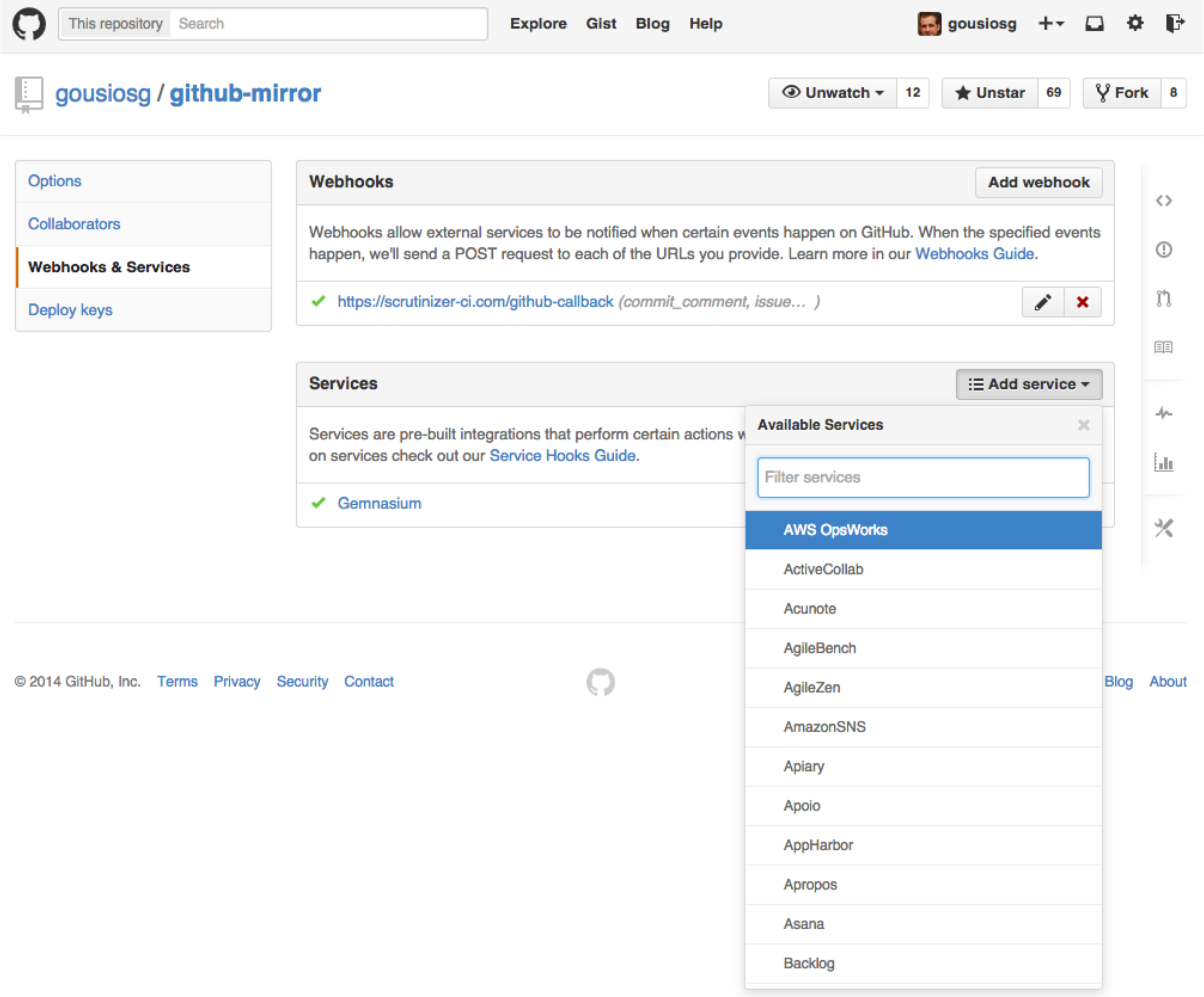
  CODE CLIMATE scrutinizer

- Testing

  COVERALLS

- Shipping

  CODESHIP



Webhooks allow external services to be notified when certain events happen on GitHub. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

✓ https://scrutinizer-ci.com/github-callback *(commit_comment, issue…)*

Services are pre-built integrations that perform certain actions w on services check out our Service Hooks Guide.

© 2014 GitHub, Inc. Terms Privacy Security Contact

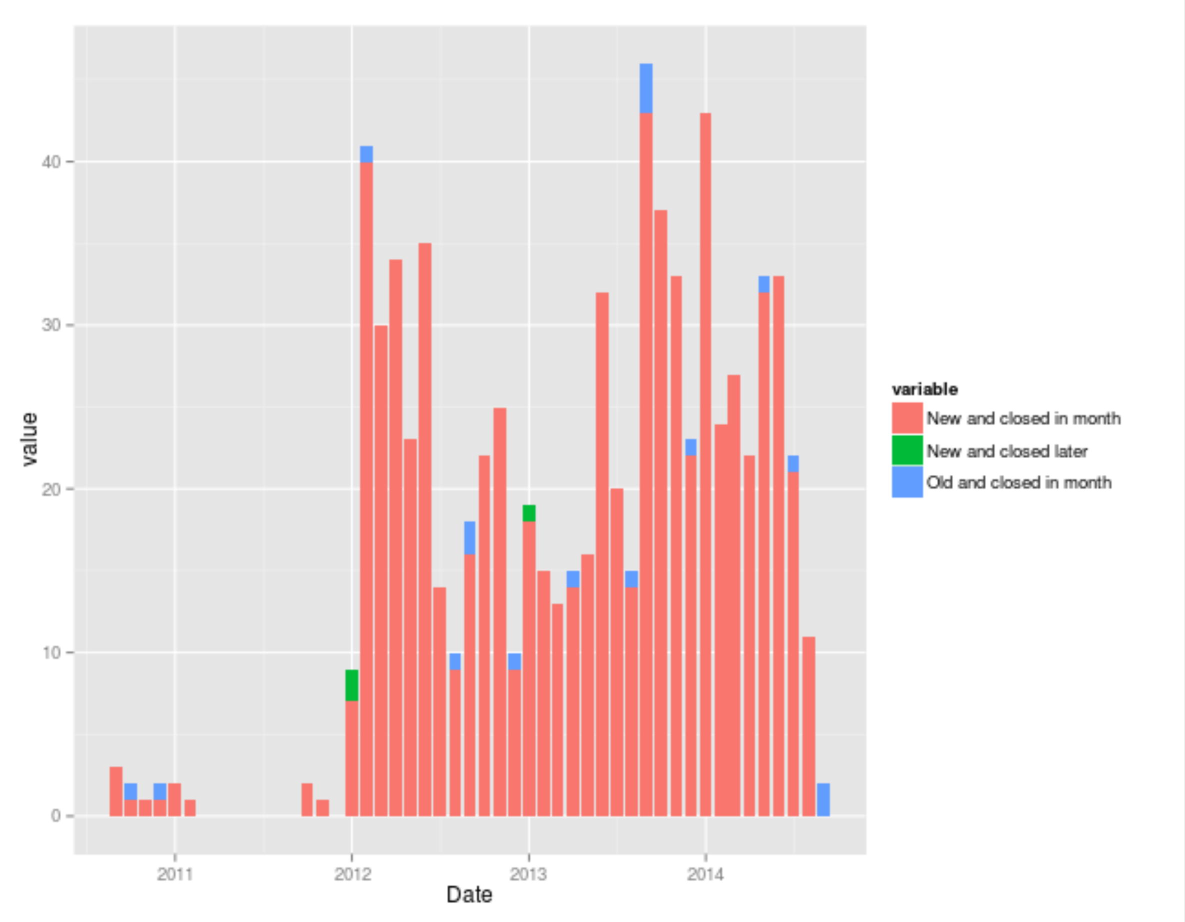
Be (pro/re)active

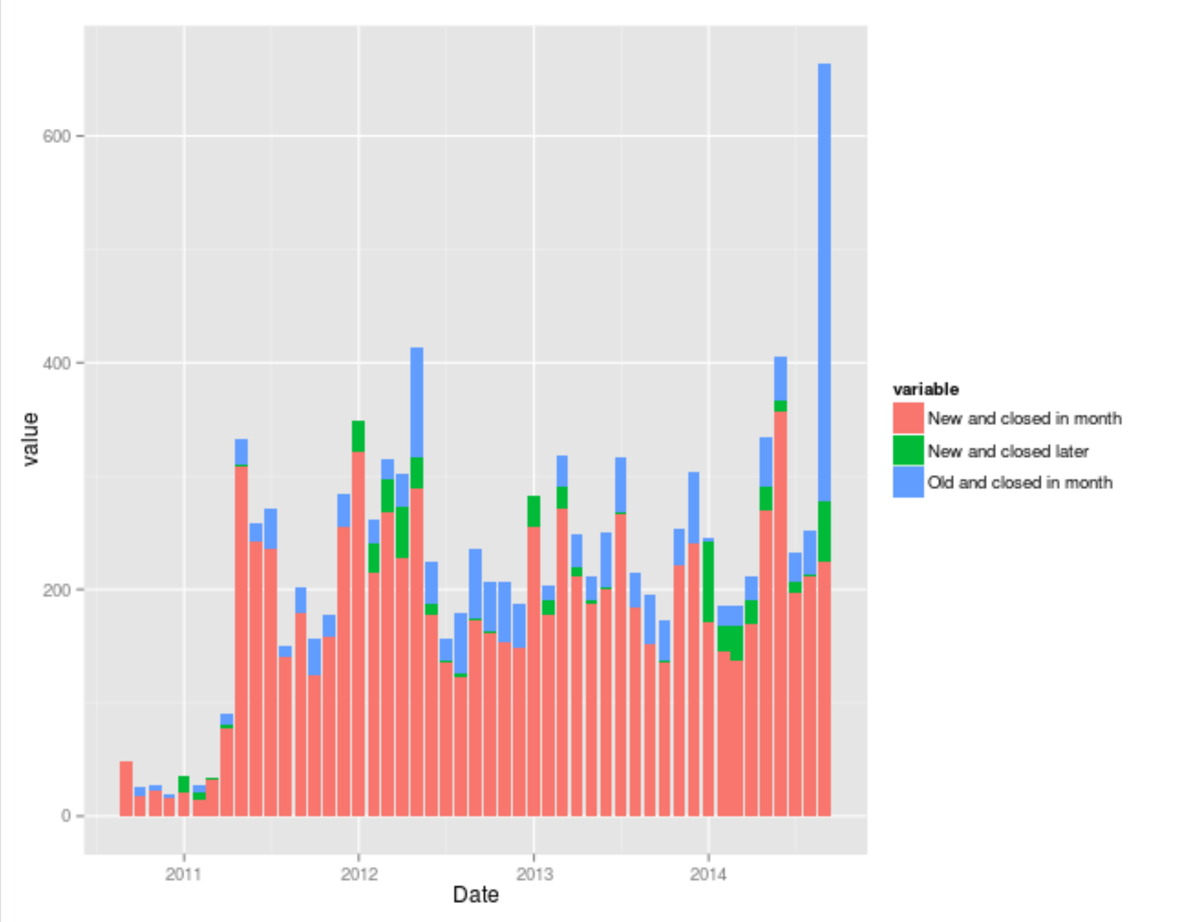
- Don’t allow PRs to linger
  - Reply to contributed requests
  - Close unwanted pull requests
- Be there when discussion diverges from technical

Monitor Performance

- Compare against the norm
  - Average pull request merged in 1 day
  - Average project merges 85% of PRs
- Monitor slow pull reqs
- Keep track of your backlog
- Monitor community engagement

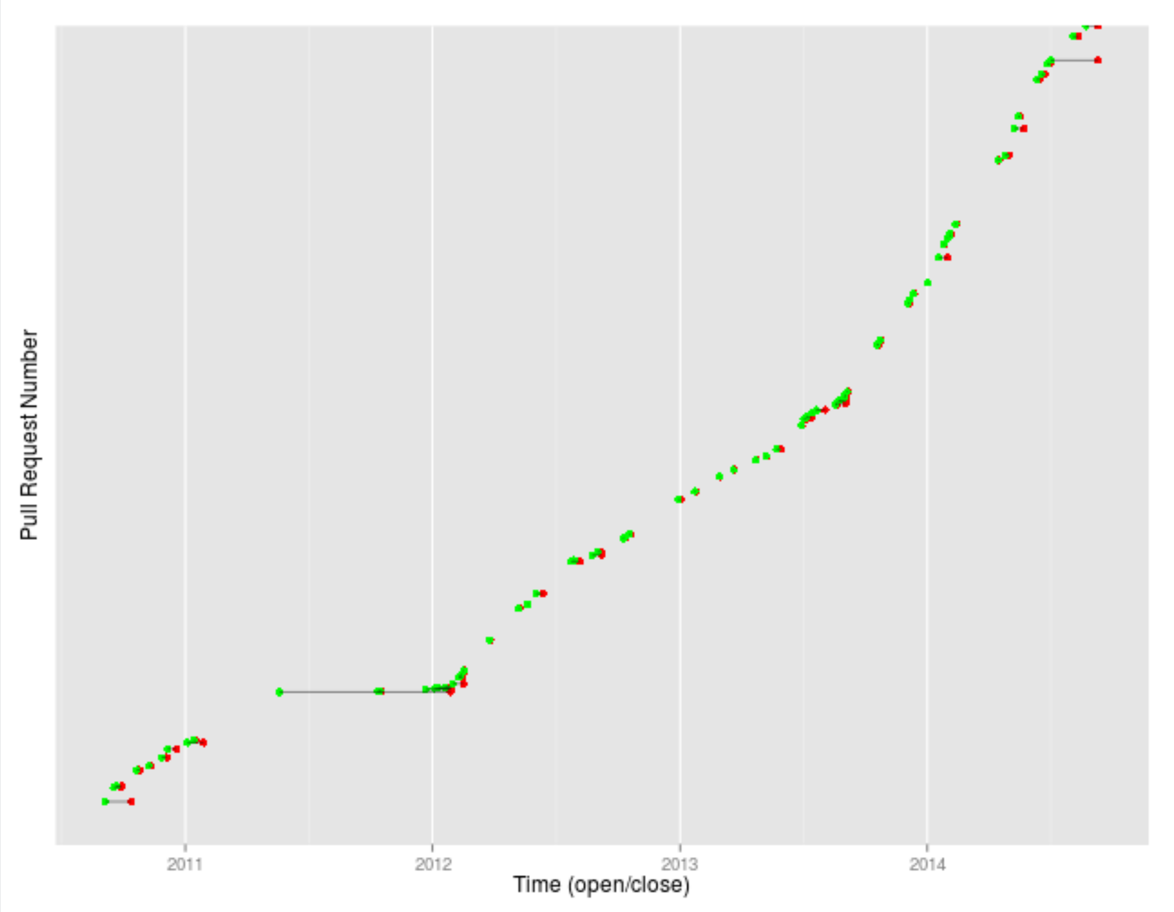
Backlog

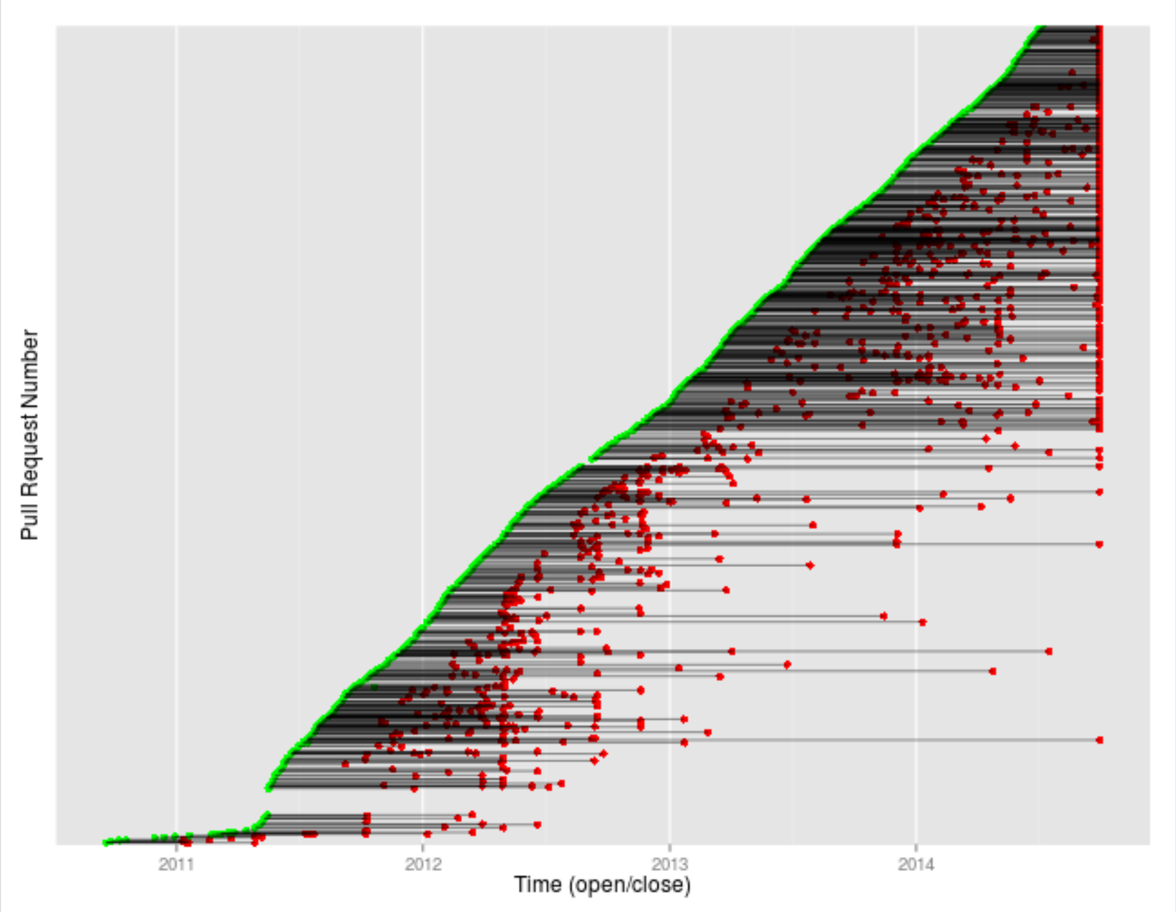




http://ghtorrent.org/pullreq-perf/

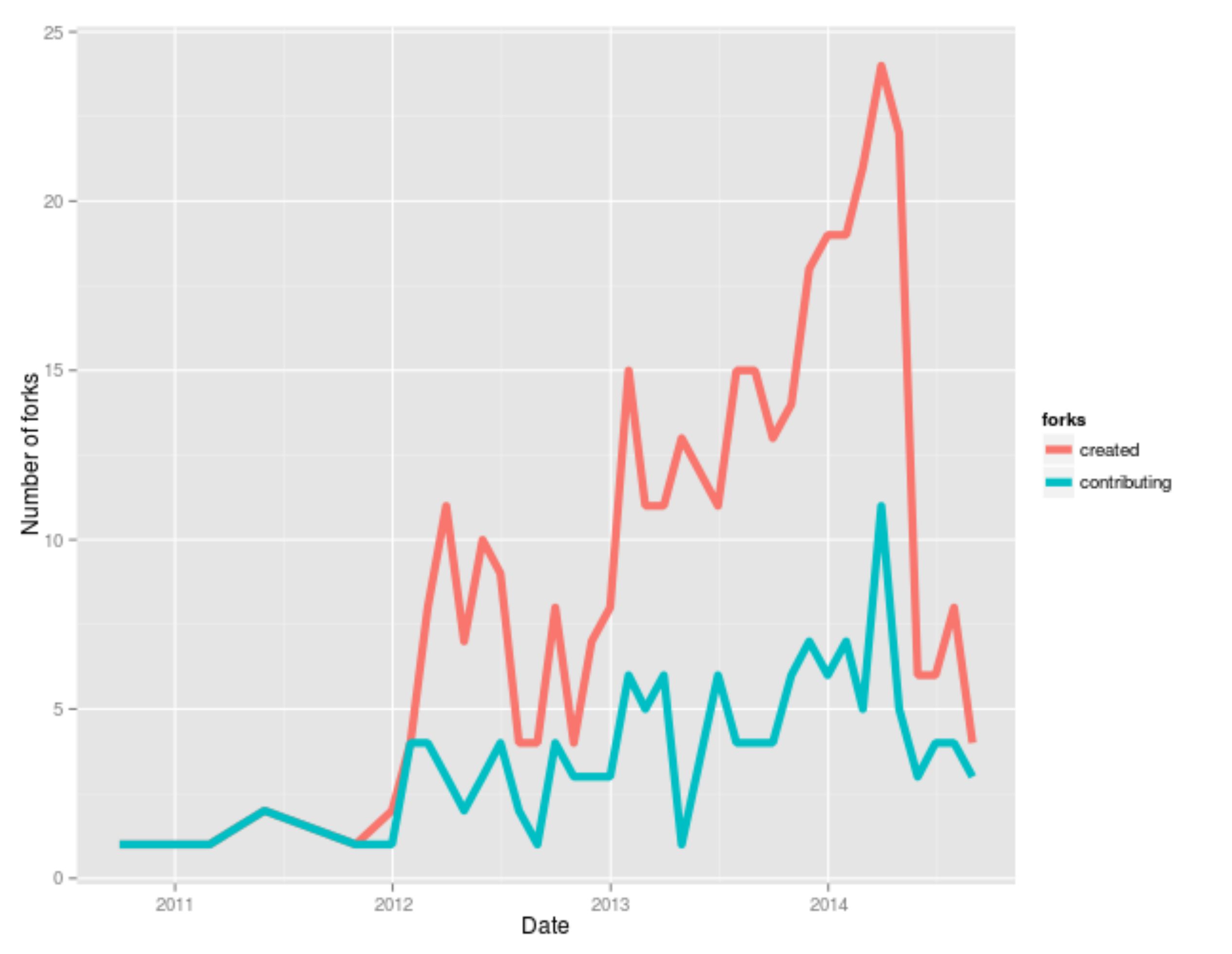
Slow pull reqs

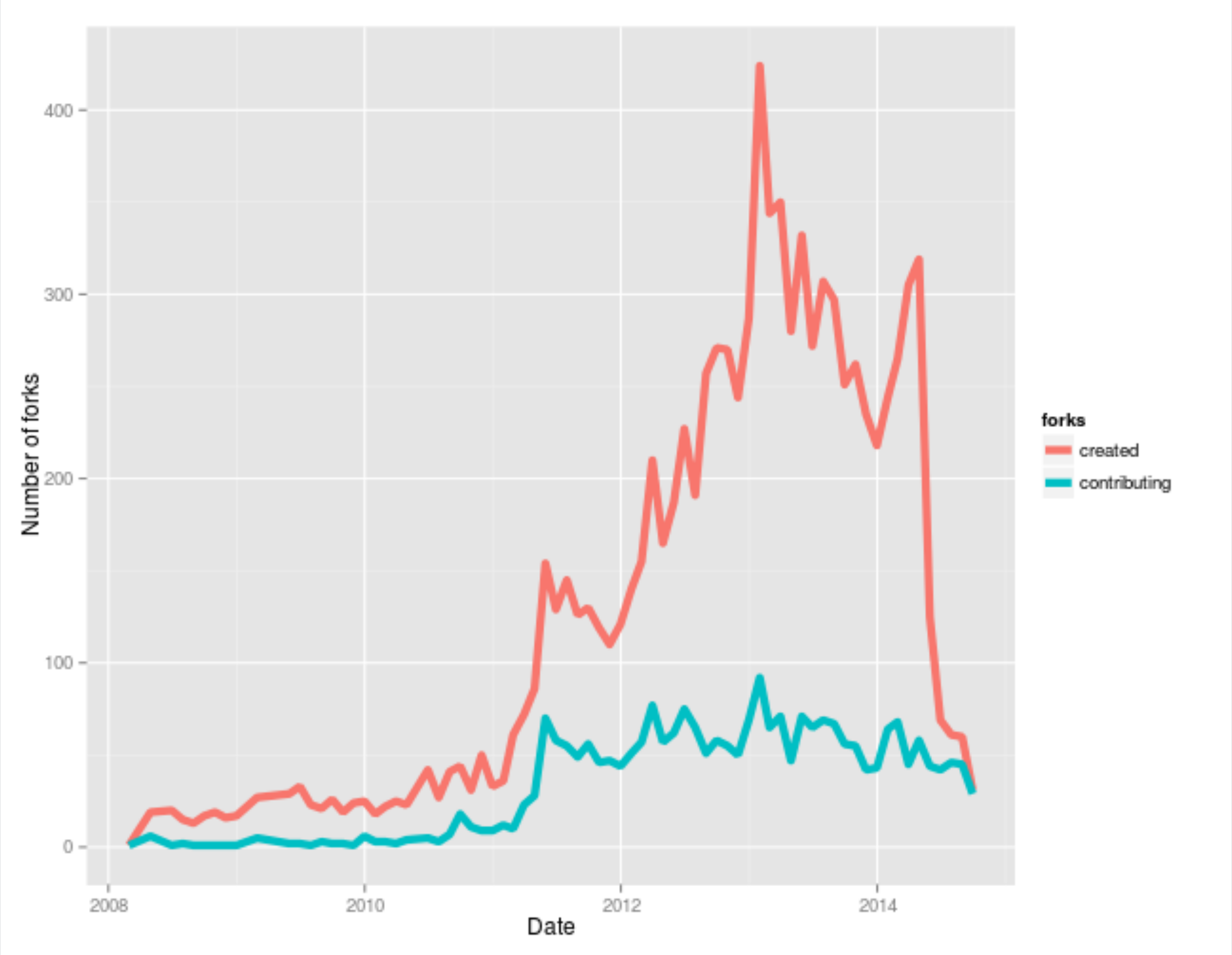




http://ghtorrent.org/pullreq-perf/

Community engagement

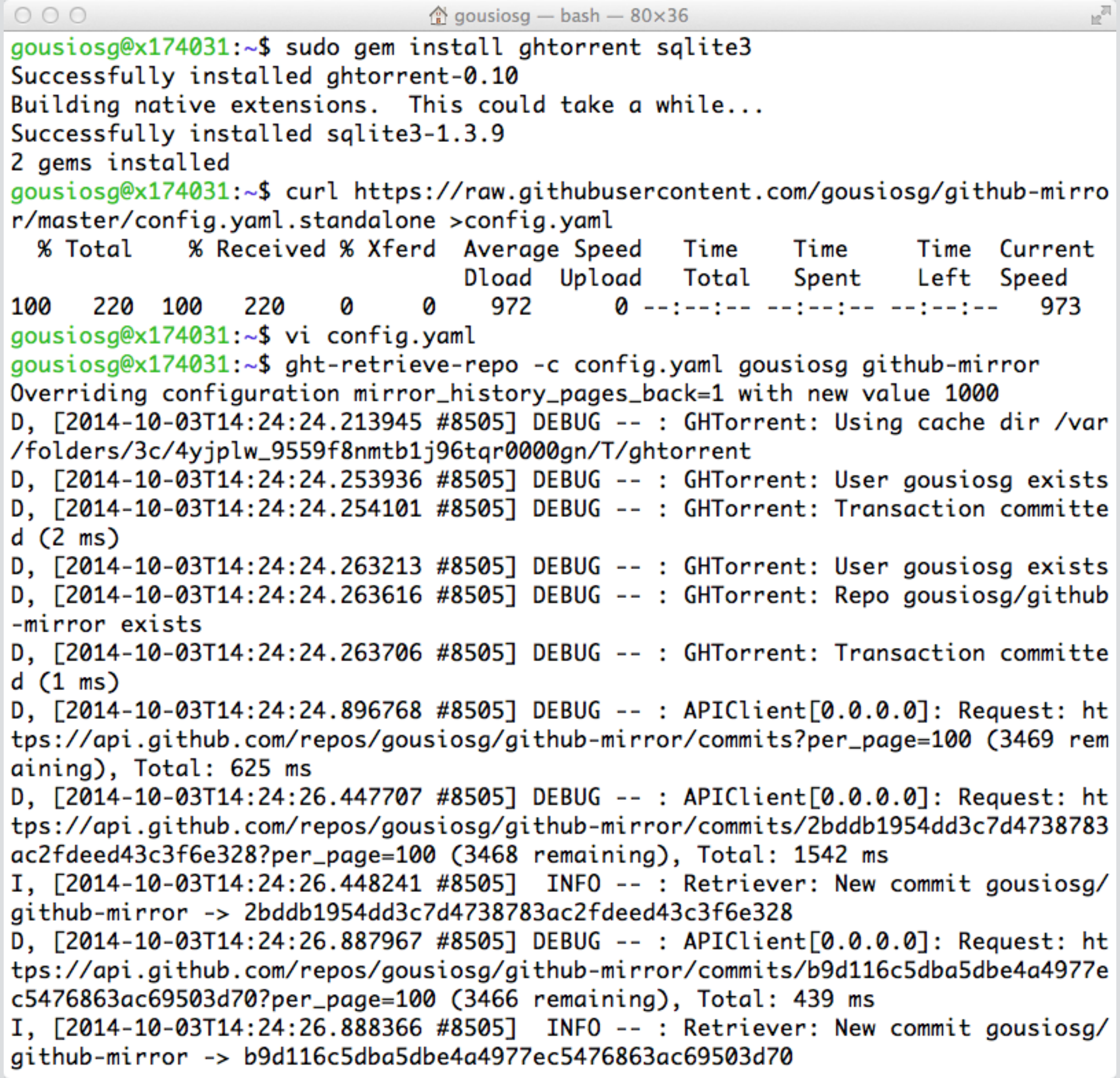




http://ghtorrent.org/pullreq-perf/

Monitoring with GHTorrent

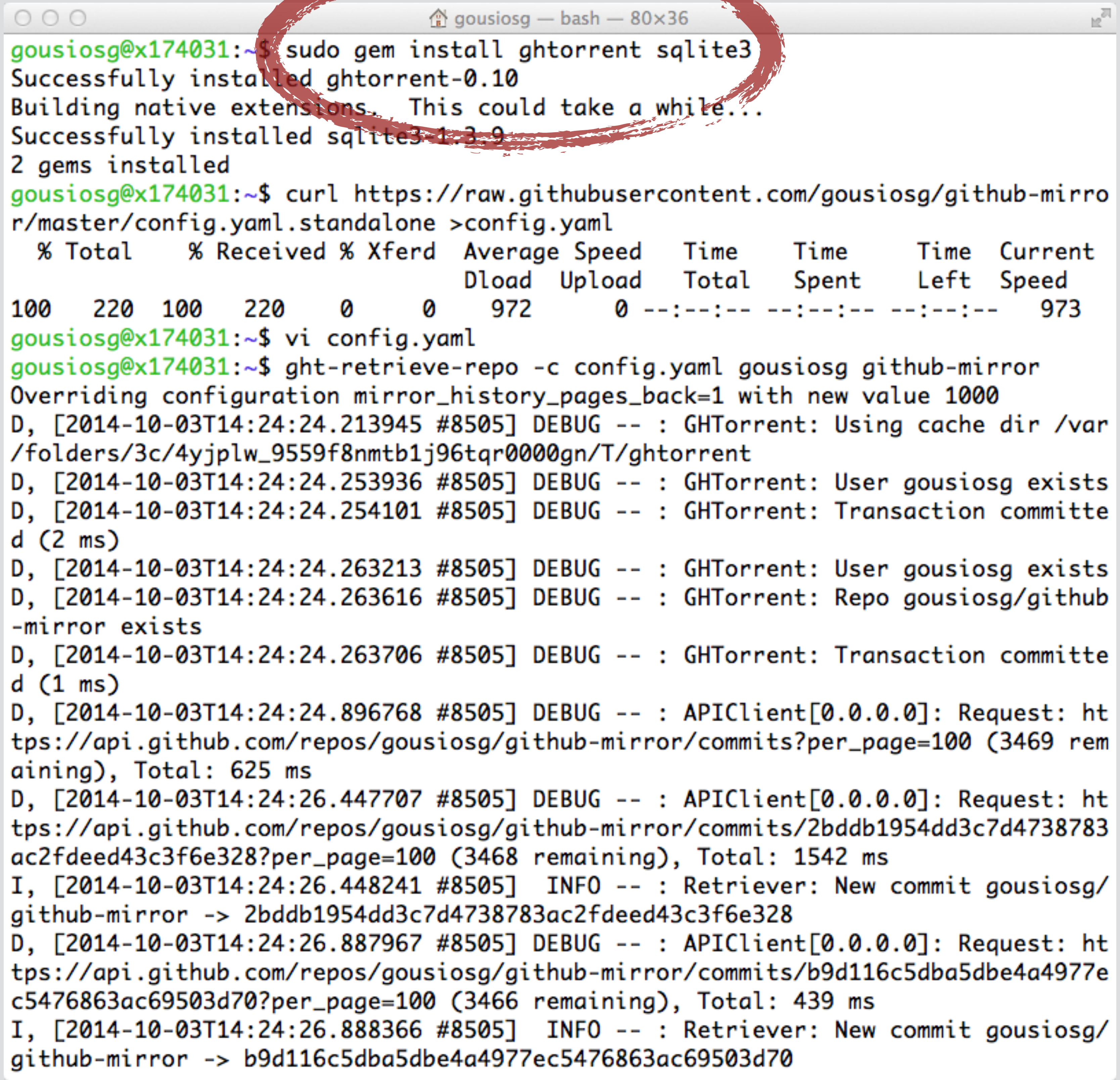
http://ghtorrent.org/



```
gousiosg@x174031:~$ sudo gem install ghtorrent sqlite3
Successfully installed ghtorrent-0.10
Building native extensions.  This could take a while...
Successfully installed sqlite3-1.3.9
2 gems installed
gousiosg@x174031:~$ curl https://raw.githubusercontent.com/gousiosg/github-mirro
r/master/config.yaml.standalone >config.yaml
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   220  100   220    0     0    972      0 --:--:-- --:--:-- --:--:--   973
gousiosg@x174031:~$ vi config.yaml
gousiosg@x174031:~$ ght-retrieve-repo -c config.yaml gousiosg github-mirror
Overriding configuration mirror_history_pages_back=1 with new value 1000
D, [2014-10-03T14:24:24.213945 #8505] DEBUG -- : GHTorrent: Using cache dir /var
/folders/3c/4yjplw_9559f8nmtb1j96tqr0000gn/T/ghtorrent
D, [2014-10-03T14:24:24.253936 #8505] DEBUG -- : GHTorrent: User gousiosg exists
D, [2014-10-03T14:24:24.254101 #8505] DEBUG -- : GHTorrent: Transaction committe
d (2 ms)
D, [2014-10-03T14:24:24.263213 #8505] DEBUG -- : GHTorrent: User gousiosg exists
D, [2014-10-03T14:24:24.263616 #8505] DEBUG -- : GHTorrent: Repo gousiosg/github
-mirror exists
D, [2014-10-03T14:24:24.263706 #8505] DEBUG -- : GHTorrent: Transaction committe
d (1 ms)
D, [2014-10-03T14:24:24.896768 #8505] DEBUG -- : APIClient[0.0.0.0]: Request: ht
tps://api.github.com/repos/gousiosg/github-mirror/commits?per_page=100 (3469 rem
aining), Total: 625 ms
D, [2014-10-03T14:24:26.447707 #8505] DEBUG -- : APIClient[0.0.0.0]: Request: ht
tps://api.github.com/repos/gousiosg/github-mirror/commits/2bddb1954dd3c7d4738783
ac2fdeed43c3f6e328?per_page=100 (3468 remaining), Total: 1542 ms
I, [2014-10-03T14:24:26.448241 #8505]  INFO -- : Retriever: New commit gousiosg/
github-mirror -> 2bddb1954dd3c7d4738783ac2fdeed43c3f6e328
D, [2014-10-03T14:24:26.887967 #8505] DEBUG -- : APIClient[0.0.0.0]: Request: ht
tps://api.github.com/repos/gousiosg/github-mirror/commits/b9d116c5dba5dbe4a4977e
c5476863ac69503d70?per_page=100 (3466 remaining), Total: 439 ms
I, [2014-10-03T14:24:26.888366 #8505]  INFO -- : Retriever: New commit gousiosg/
github-mirror -> b9d116c5dba5dbe4a4977ec5476863ac69503d70
```

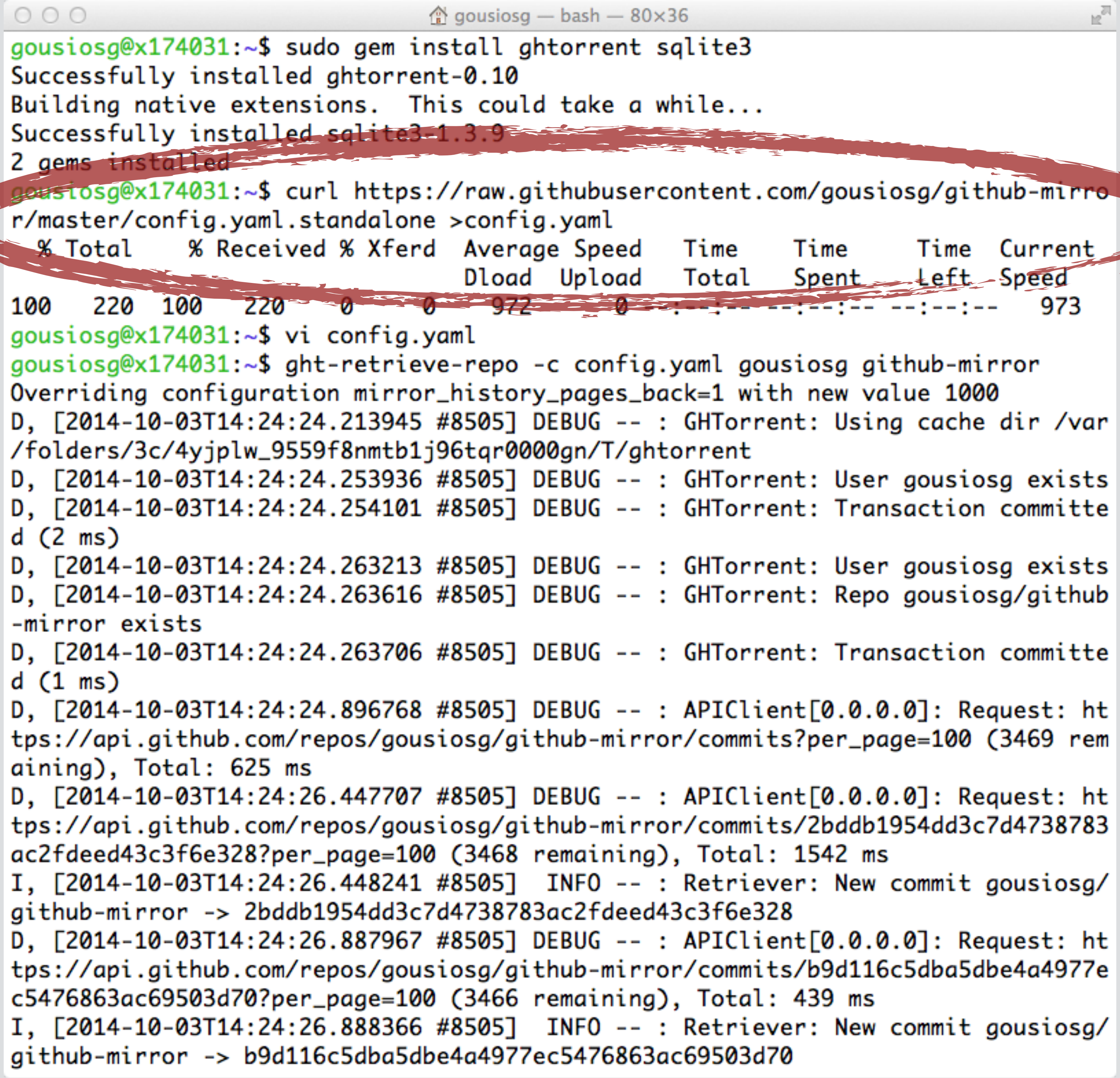
Monitoring with GHTorrent

http://ghtorrent.org/



```
gousiosg@x174031:~$ sudo gem install ghtorrent sqlite3
Successfully installed ghtorrent-0.10
Building native extensions.  This could take a while...
Successfully installed sqlite3-1.3.9
2 gems installed
gousiosg@x174031:~$ curl https://raw.githubusercontent.com/gousiosg/github-mirro
r/master/config.yaml.standalone >config.yaml
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   220  100   220    0     0    972      0 --:--:-- --:--:-- --:--:--   973
gousiosg@x174031:~$ vi config.yaml
gousiosg@x174031:~$ ght-retrieve-repo -c config.yaml gousiosg github-mirror
Overriding configuration mirror_history_pages_back=1 with new value 1000
D, [2014-10-03T14:24:24.213945 #8505] DEBUG -- : GHTorrent: Using cache dir /var
/folders/3c/4yjplw_9559f8nmtb1j96tqr0000gn/T/ghtorrent
D, [2014-10-03T14:24:24.253936 #8505] DEBUG -- : GHTorrent: User gousiosg exists
D, [2014-10-03T14:24:24.254101 #8505] DEBUG -- : GHTorrent: Transaction committe
d (2 ms)
D, [2014-10-03T14:24:24.263213 #8505] DEBUG -- : GHTorrent: User gousiosg exists
D, [2014-10-03T14:24:24.263616 #8505] DEBUG -- : GHTorrent: Repo gousiosg/github
-mirror exists
D, [2014-10-03T14:24:24.263706 #8505] DEBUG -- : GHTorrent: Transaction committe
d (1 ms)
D, [2014-10-03T14:24:24.896768 #8505] DEBUG -- : APIClient[0.0.0.0]: Request: ht
tps://api.github.com/repos/gousiosg/github-mirror/commits?per_page=100 (3469 rem
aining), Total: 625 ms
D, [2014-10-03T14:24:26.447707 #8505] DEBUG -- : APIClient[0.0.0.0]: Request: ht
tps://api.github.com/repos/gousiosg/github-mirror/commits/2bddb1954dd3c7d4738783
ac2fdeed43c3f6e328?per_page=100 (3468 remaining), Total: 1542 ms
I, [2014-10-03T14:24:26.448241 #8505]  INFO -- : Retriever: New commit gousiosg/
github-mirror -> 2bddb1954dd3c7d4738783ac2fdeed43c3f6e328
D, [2014-10-03T14:24:26.887967 #8505] DEBUG -- : APIClient[0.0.0.0]: Request: ht
tps://api.github.com/repos/gousiosg/github-mirror/commits/b9d116c5dba5dbe4a4977e
c5476863ac69503d70?per_page=100 (3466 remaining), Total: 439 ms
I, [2014-10-03T14:24:26.888366 #8505]  INFO -- : Retriever: New commit gousiosg/
github-mirror -> b9d116c5dba5dbe4a4977ec5476863ac69503d70
```

Monitoring with GHTorrent

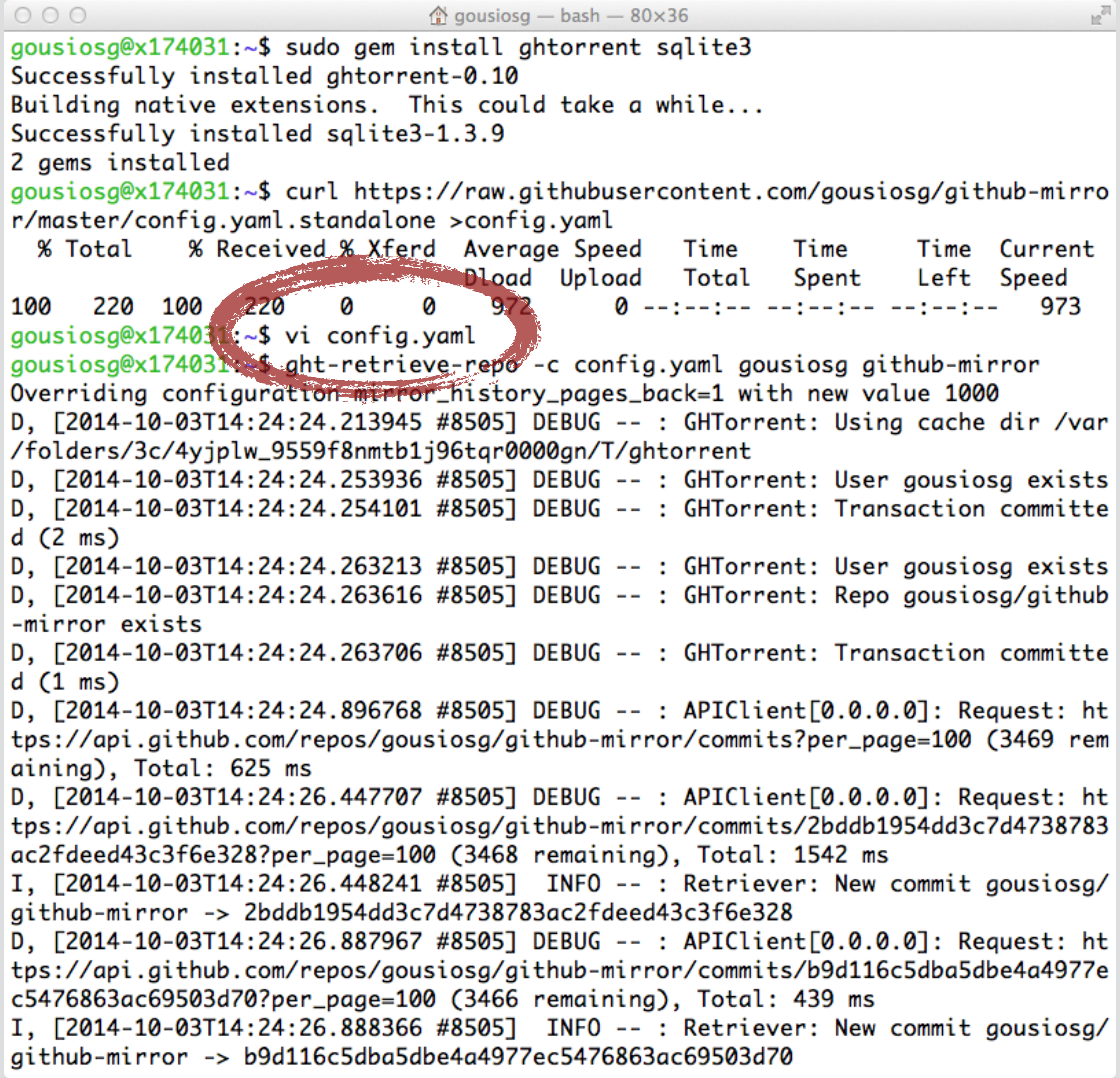


```
gousiosg@x174031:~$ sudo gem install ghtorrent sqlite3
Successfully installed ghtorrent-0.10
Building native extensions.  This could take a while...
Successfully installed sqlite3-1.3.9
2 gems installed
gousiosg@x174031:~$ curl https://raw.githubusercontent.com/gousiosg/github-mirro
r/master/config.yaml.standalone >config.yaml
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   220  100   220    0     0    972      0 --:--:-- --:--:-- --:--:--   973
gousiosg@x174031:~$ vi config.yaml
gousiosg@x174031:~$ ght-retrieve-repo -c config.yaml gousiosg github-mirror
Overriding configuration mirror_history_pages_back=1 with new value 1000
D, [2014-10-03T14:24:24.213945 #8505] DEBUG -- : GHTorrent: Using cache dir /var
/folders/3c/4yjplw_9559f8nmtb1j96tqr0000gn/T/ghtorrent
D, [2014-10-03T14:24:24.253936 #8505] DEBUG -- : GHTorrent: User gousiosg exists
D, [2014-10-03T14:24:24.254101 #8505] DEBUG -- : GHTorrent: Transaction committe
d (2 ms)
D, [2014-10-03T14:24:24.263213 #8505] DEBUG -- : GHTorrent: User gousiosg exists
D, [2014-10-03T14:24:24.263616 #8505] DEBUG -- : GHTorrent: Repo gousiosg/github
-mirror exists
D, [2014-10-03T14:24:24.263706 #8505] DEBUG -- : GHTorrent: Transaction committe
d (1 ms)
D, [2014-10-03T14:24:24.896768 #8505] DEBUG -- : APIClient[0.0.0.0]: Request: ht
tps://api.github.com/repos/gousiosg/github-mirror/commits?per_page=100 (3469 rem
aining), Total: 625 ms
D, [2014-10-03T14:24:26.447707 #8505] DEBUG -- : APIClient[0.0.0.0]: Request: ht
tps://api.github.com/repos/gousiosg/github-mirror/commits/2bddb1954dd3c7d4738783
ac2fdeed43c3f6e328?per_page=100 (3468 remaining), Total: 1542 ms
I, [2014-10-03T14:24:26.448241 #8505]  INFO -- : Retriever: New commit gousiosg/
github-mirror -> 2bddb1954dd3c7d4738783ac2fdeed43c3f6e328
D, [2014-10-03T14:24:26.887967 #8505] DEBUG -- : APIClient[0.0.0.0]: Request: ht
tps://api.github.com/repos/gousiosg/github-mirror/commits/b9d116c5dba5dbe4a4977e
c5476863ac69503d70?per_page=100 (3466 remaining), Total: 439 ms
I, [2014-10-03T14:24:26.888366 #8505]  INFO -- : Retriever: New commit gousiosg/
github-mirror -> b9d116c5dba5dbe4a4977ec5476863ac69503d70
```

http://ghtorrent.org/

Monitoring with GHTorrent

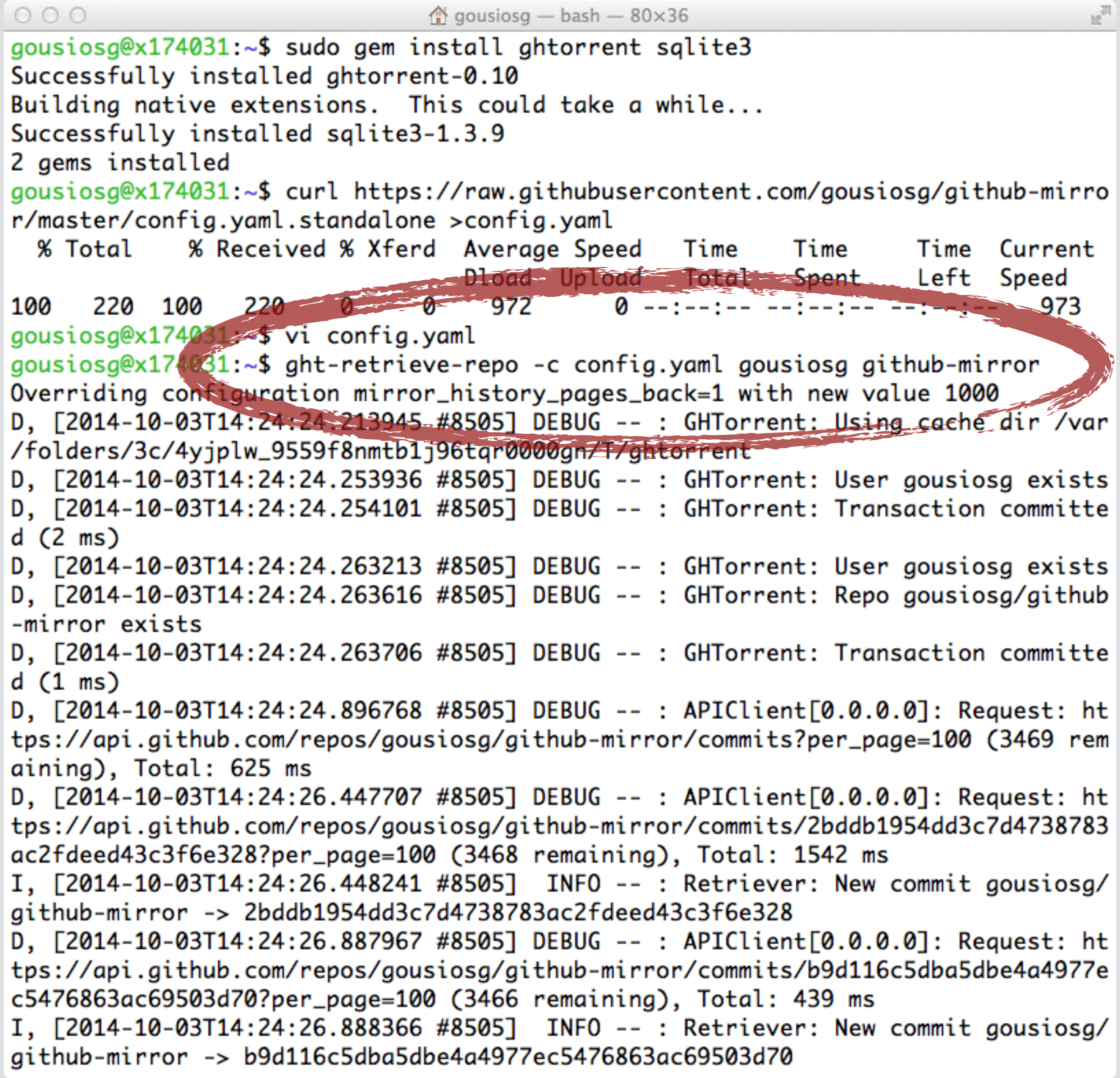
http://ghtorrent.org/



```
gousiosg@x174031:~$ sudo gem install ghtorrent sqlite3
Successfully installed ghtorrent-0.10
Building native extensions.  This could take a while...
Successfully installed sqlite3-1.3.9
2 gems installed
gousiosg@x174031:~$ curl https://raw.githubusercontent.com/gousiosg/github-mirro
r/master/config.yaml.standalone >config.yaml
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   220  100   220    0     0    972      0 --:--:-- --:--:-- --:--:--   973
gousiosg@x174031:~$ vi config.yaml
gousiosg@x174031:~$ ght-retrieve-repo -c config.yaml gousiosg github-mirror
Overriding configuration mirror_history_pages_back=1 with new value 1000
D, [2014-10-03T14:24:24.213945 #8505] DEBUG -- : GHTorrent: Using cache dir /var
/folders/3c/4yjplw_9559f8nmtb1j96tqr0000gn/T/ghtorrent
D, [2014-10-03T14:24:24.253936 #8505] DEBUG -- : GHTorrent: User gousiosg exists
D, [2014-10-03T14:24:24.254101 #8505] DEBUG -- : GHTorrent: Transaction committe
d (2 ms)
D, [2014-10-03T14:24:24.263213 #8505] DEBUG -- : GHTorrent: User gousiosg exists
D, [2014-10-03T14:24:24.263616 #8505] DEBUG -- : GHTorrent: Repo gousiosg/github
-mirror exists
D, [2014-10-03T14:24:24.263706 #8505] DEBUG -- : GHTorrent: Transaction committe
d (1 ms)
D, [2014-10-03T14:24:24.896768 #8505] DEBUG -- : APIClient[0.0.0.0]: Request: ht
tps://api.github.com/repos/gousiosg/github-mirror/commits?per_page=100 (3469 rem
aining), Total: 625 ms
D, [2014-10-03T14:24:26.447707 #8505] DEBUG -- : APIClient[0.0.0.0]: Request: ht
tps://api.github.com/repos/gousiosg/github-mirror/commits/2bddb1954dd3c7d4738783
ac2fdeed43c3f6e328?per_page=100 (3468 remaining), Total: 1542 ms
I, [2014-10-03T14:24:26.448241 #8505]  INFO -- : Retriever: New commit gousiosg/
github-mirror -> 2bddb1954dd3c7d4738783ac2fdeed43c3f6e328
D, [2014-10-03T14:24:26.887967 #8505] DEBUG -- : APIClient[0.0.0.0]: Request: ht
tps://api.github.com/repos/gousiosg/github-mirror/commits/b9d116c5dba5dbe4a4977e
c5476863ac69503d70?per_page=100 (3466 remaining), Total: 439 ms
I, [2014-10-03T14:24:26.888366 #8505]  INFO -- : Retriever: New commit gousiosg/
github-mirror -> b9d116c5dba5dbe4a4977ec5476863ac69503d70
```

Monitoring with GHTorrent

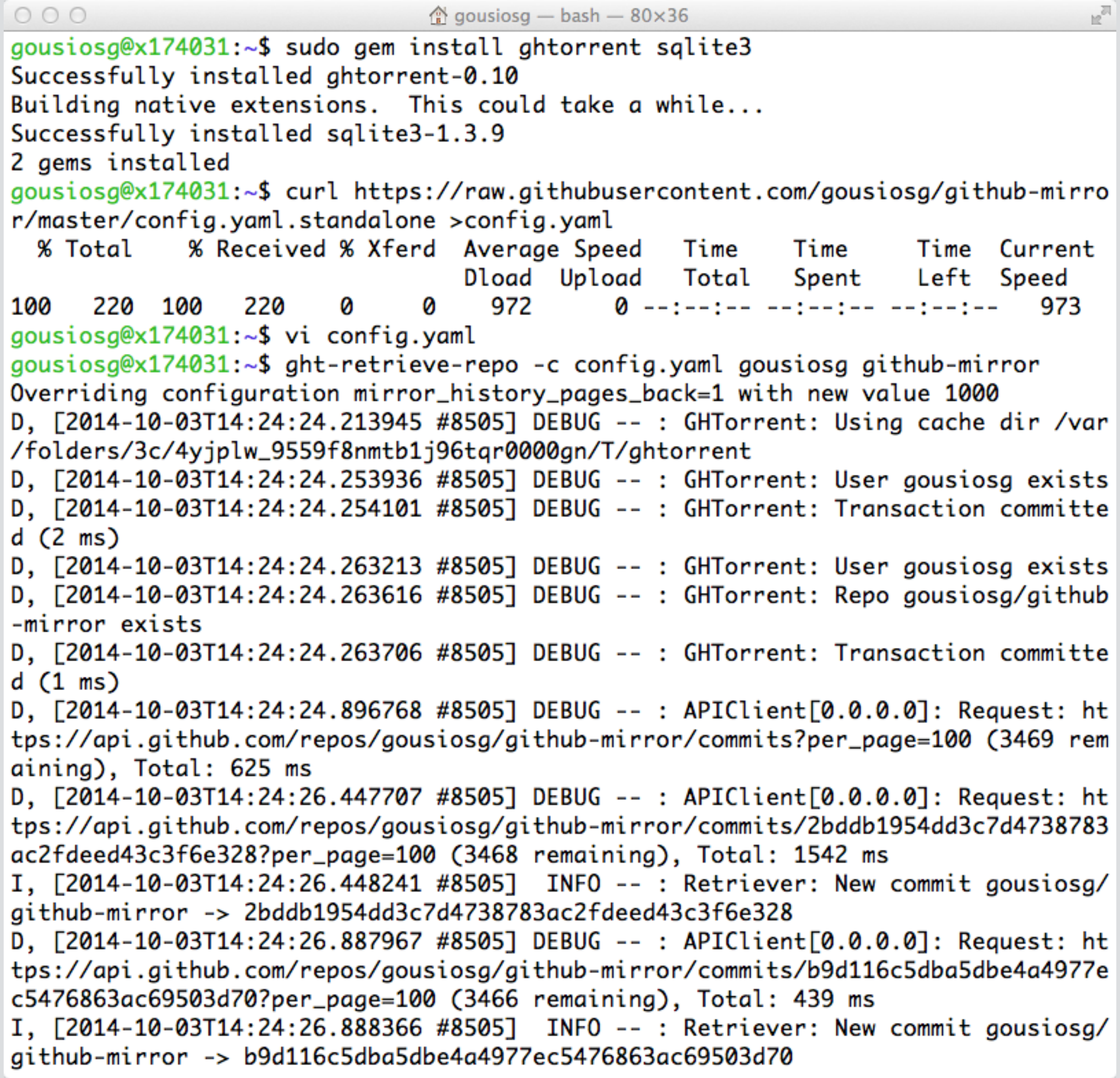
http://ghtorrent.org/



```
gousiosg@x174031:~$ sudo gem install ghtorrent sqlite3
Successfully installed ghtorrent-0.10
Building native extensions.  This could take a while...
Successfully installed sqlite3-1.3.9
2 gems installed
gousiosg@x174031:~$ curl https://raw.githubusercontent.com/gousiosg/github-mirro
r/master/config.yaml.standalone >config.yaml
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   220  100   220    0     0    972      0 --:--:-- --:--:-- --:--:--   973
gousiosg@x174031:~$ vi config.yaml
gousiosg@x174031:~$ ght-retrieve-repo -c config.yaml gousiosg github-mirror
Overriding configuration mirror_history_pages_back=1 with new value 1000
D, [2014-10-03T14:24:24.213945 #8505] DEBUG -- : GHTorrent: Using cache dir /var
/folders/3c/4yjplw_9559f8nmtb1j96tqr0000gn/T/ghtorrent
D, [2014-10-03T14:24:24.253936 #8505] DEBUG -- : GHTorrent: User gousiosg exists
D, [2014-10-03T14:24:24.254101 #8505] DEBUG -- : GHTorrent: Transaction committe
d (2 ms)
D, [2014-10-03T14:24:24.263213 #8505] DEBUG -- : GHTorrent: User gousiosg exists
D, [2014-10-03T14:24:24.263616 #8505] DEBUG -- : GHTorrent: Repo gousiosg/github
-mirror exists
D, [2014-10-03T14:24:24.263706 #8505] DEBUG -- : GHTorrent: Transaction committe
d (1 ms)
D, [2014-10-03T14:24:24.896768 #8505] DEBUG -- : APIClient[0.0.0.0]: Request: ht
tps://api.github.com/repos/gousiosg/github-mirror/commits?per_page=100 (3469 rem
aining), Total: 625 ms
D, [2014-10-03T14:24:26.447707 #8505] DEBUG -- : APIClient[0.0.0.0]: Request: ht
tps://api.github.com/repos/gousiosg/github-mirror/commits/2bddb1954dd3c7d4738783
ac2fdeed43c3f6e328?per_page=100 (3468 remaining), Total: 1542 ms
I, [2014-10-03T14:24:26.448241 #8505]  INFO -- : Retriever: New commit gousiosg/
github-mirror -> 2bddb1954dd3c7d4738783ac2fdeed43c3f6e328
D, [2014-10-03T14:24:26.887967 #8505] DEBUG -- : APIClient[0.0.0.0]: Request: ht
tps://api.github.com/repos/gousiosg/github-mirror/commits/b9d116c5dba5dbe4a4977e
c5476863ac69503d70?per_page=100 (3466 remaining), Total: 439 ms
I, [2014-10-03T14:24:26.888366 #8505]  INFO -- : Retriever: New commit gousiosg/
github-mirror -> b9d116c5dba5dbe4a4977ec5476863ac69503d70
```

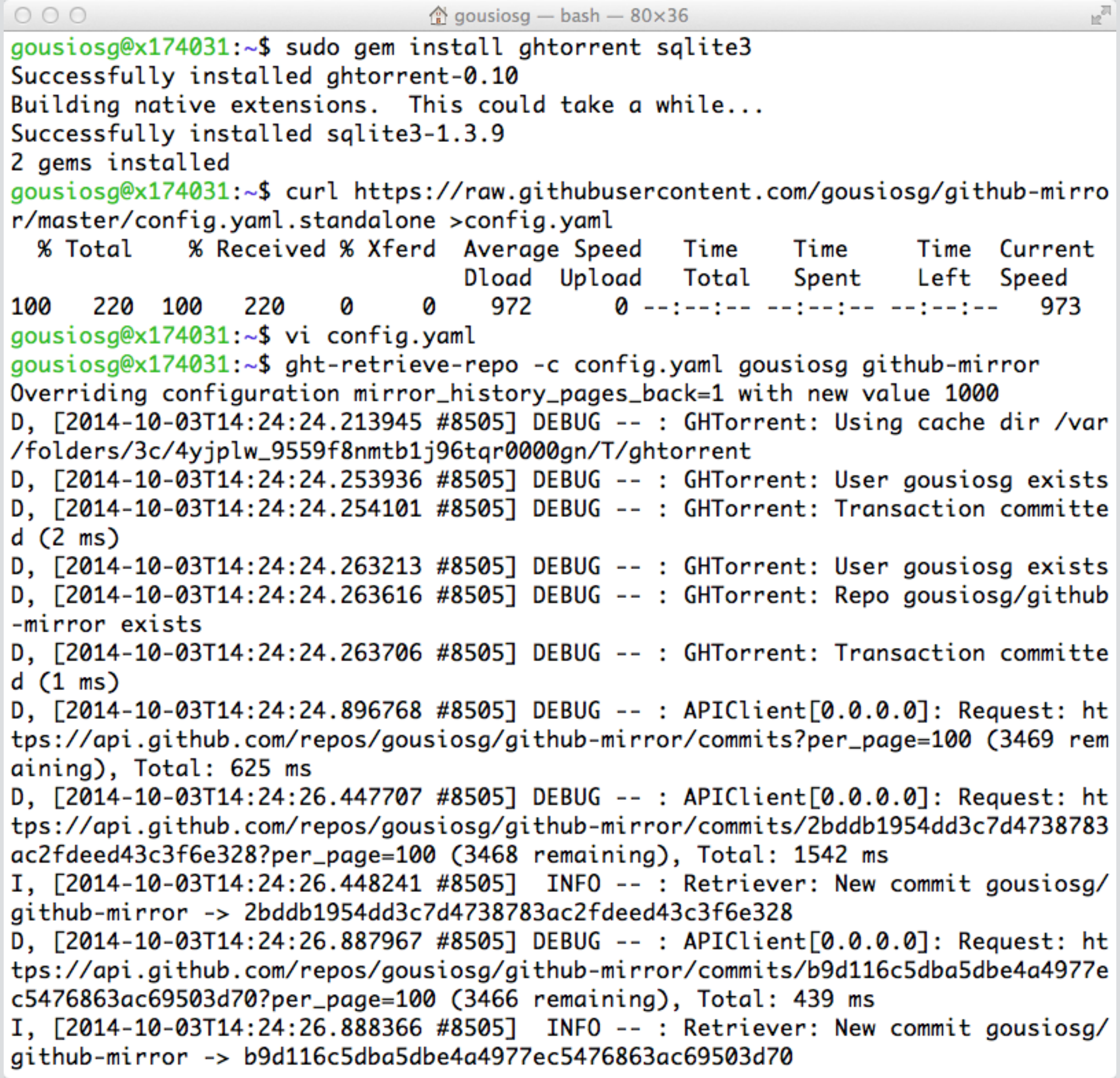
Monitoring with GHTorrent

http://ghtorrent.org/



```
gousiosg@x174031:~$ sudo gem install ghtorrent sqlite3
Successfully installed ghtorrent-0.10
Building native extensions.  This could take a while...
Successfully installed sqlite3-1.3.9
2 gems installed
gousiosg@x174031:~$ curl https://raw.githubusercontent.com/gousiosg/github-mirro
r/master/config.yaml.standalone >config.yaml
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   220  100   220    0     0    972      0 --:--:-- --:--:-- --:--:--   973
gousiosg@x174031:~$ vi config.yaml
gousiosg@x174031:~$ ght-retrieve-repo -c config.yaml gousiosg github-mirror
Overriding configuration mirror_history_pages_back=1 with new value 1000
D, [2014-10-03T14:24:24.213945 #8505] DEBUG -- : GHTorrent: Using cache dir /var
/folders/3c/4yjplw_9559f8nmtb1j96tqr0000gn/T/ghtorrent
D, [2014-10-03T14:24:24.253936 #8505] DEBUG -- : GHTorrent: User gousiosg exists
D, [2014-10-03T14:24:24.254101 #8505] DEBUG -- : GHTorrent: Transaction committe
d (2 ms)
D, [2014-10-03T14:24:24.263213 #8505] DEBUG -- : GHTorrent: User gousiosg exists
D, [2014-10-03T14:24:24.263616 #8505] DEBUG -- : GHTorrent: Repo gousiosg/github
-mirror exists
D, [2014-10-03T14:24:24.263706 #8505] DEBUG -- : GHTorrent: Transaction committe
d (1 ms)
D, [2014-10-03T14:24:24.896768 #8505] DEBUG -- : APIClient[0.0.0.0]: Request: ht
tps://api.github.com/repos/gousiosg/github-mirror/commits?per_page=100 (3469 rem
aining), Total: 625 ms
D, [2014-10-03T14:24:26.447707 #8505] DEBUG -- : APIClient[0.0.0.0]: Request: ht
tps://api.github.com/repos/gousiosg/github-mirror/commits/2bddb1954dd3c7d4738783
ac2fdeed43c3f6e328?per_page=100 (3468 remaining), Total: 1542 ms
I, [2014-10-03T14:24:26.448241 #8505]  INFO -- : Retriever: New commit gousiosg/
github-mirror -> 2bddb1954dd3c7d4738783ac2fdeed43c3f6e328
D, [2014-10-03T14:24:26.887967 #8505] DEBUG -- : APIClient[0.0.0.0]: Request: ht
tps://api.github.com/repos/gousiosg/github-mirror/commits/b9d116c5dba5dbe4a4977e
c5476863ac69503d70?per_page=100 (3466 remaining), Total: 439 ms
I, [2014-10-03T14:24:26.888366 #8505]  INFO -- : Retriever: New commit gousiosg/
github-mirror -> b9d116c5dba5dbe4a4977ec5476863ac69503d70
```

Monitoring with GHTorrent



```
gousiosg@x174031:~$ sudo gem install ghtorrent sqlite3
Successfully installed ghtorrent-0.10
Building native extensions.  This could take a while...
Successfully installed sqlite3-1.3.9
2 gems installed
gousiosg@x174031:~$ curl https://raw.githubusercontent.com/gousiosg/github-mirro
r/master/config.yaml.standalone >config.yaml
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   220  100   220    0     0    972      0 --:--:-- --:--:-- --:--:--   973
gousiosg@x174031:~$ vi config.yaml
gousiosg@x174031:~$ ght-retrieve-repo -c config.yaml gousiosg github-mirror
Overriding configuration mirror_history_pages_back=1 with new value 1000
D, [2014-10-03T14:24:24.213945 #8505] DEBUG -- : GHTorrent: Using cache dir /var
/folders/3c/4yjplw_9559f8nmtb1j96tqr0000gn/T/ghtorrent
D, [2014-10-03T14:24:24.253936 #8505] DEBUG -- : GHTorrent: User gousiosg exists
D, [2014-10-03T14:24:24.254101 #8505] DEBUG -- : GHTorrent: Transaction committe
d (2 ms)
D, [2014-10-03T14:24:24.263213 #8505] DEBUG -- : GHTorrent: User gousiosg exists
D, [2014-10-03T14:24:24.263616 #8505] DEBUG -- : GHTorrent: Repo gousiosg/github
-mirror exists
D, [2014-10-03T14:24:24.263706 #8505] DEBUG -- : GHTorrent: Transaction committe
d (1 ms)
D, [2014-10-03T14:24:24.896768 #8505] DEBUG -- : APIClient[0.0.0.0]: Request: ht
tps://api.github.com/repos/gousiosg/github-mirror/commits?per_page=100 (3469 rem
aining), Total: 625 ms
D, [2014-10-03T14:24:26.447707 #8505] DEBUG -- : APIClient[0.0.0.0]: Request: ht
tps://api.github.com/repos/gousiosg/github-mirror/commits/2bddb1954dd3c7d4738783
ac2fdeed43c3f6e328?per_page=100 (3468 remaining), Total: 1542 ms
I, [2014-10-03T14:24:26.448241 #8505]  INFO -- : Retriever: New commit gousiosg/
github-mirror -> 2bddb1954dd3c7d4738783ac2fdeed43c3f6e328
D, [2014-10-03T14:24:26.887967 #8505] DEBUG -- : APIClient[0.0.0.0]: Request: ht
tps://api.github.com/repos/gousiosg/github-mirror/commits/b9d116c5dba5dbe4a4977e
c5476863ac69503d70?per_page=100 (3466 remaining), Total: 439 ms
I, [2014-10-03T14:24:26.888366 #8505]  INFO -- : Retriever: New commit gousiosg/
github-mirror -> b9d116c5dba5dbe4a4977ec5476863ac69503d70
```

http://ghtorrent.org/

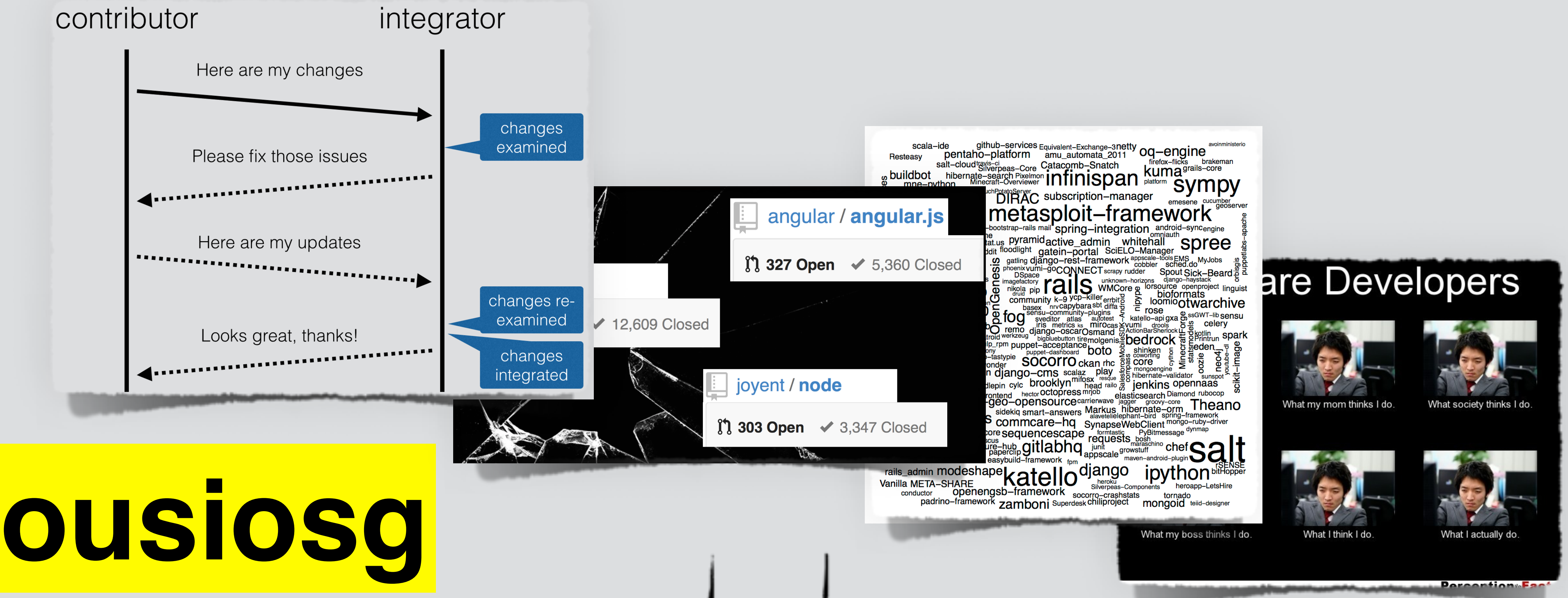
SQLite

Contributors

Minimize friction

Contributors

1. Obey the guidelines (when available)
2. Invest time to learn the tools
3. Keep it short, hot and isolated



@gousiosg

Project owners

1. Provide submission guidelines
2. Invest in good tests
3. Automate
4. Be *reactive*
5. Monitor PR performance

Contributors

1. Obey the guidelines (when available)
2. Invest time to learn the tools
3. Keep it short, hot and isolated

There is more to Scrum than Scrum

Bianca Griffioen

✉ bgriffioen@vxcompany.com

@BiaGrif

in https://nl.linkedin.com/in/biancagriffioen

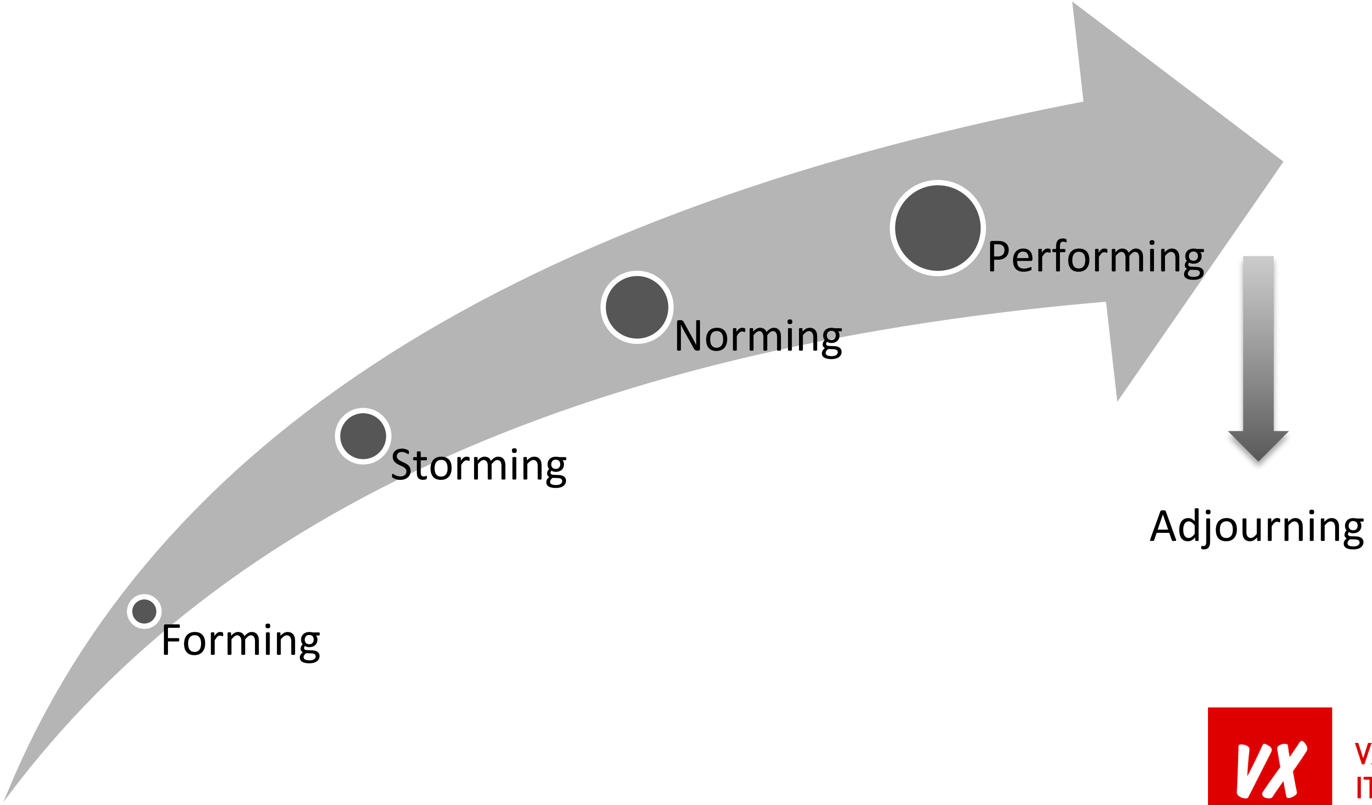
☎ 06-21506814

VX Company
IT Services

EK KWALIFICATIEDUEL
NEDERLAND
KAZACHSTAN
10-10-2014
AMSTERDAM ARENA
NO TO RACISM
RESPECT
KE.COM/

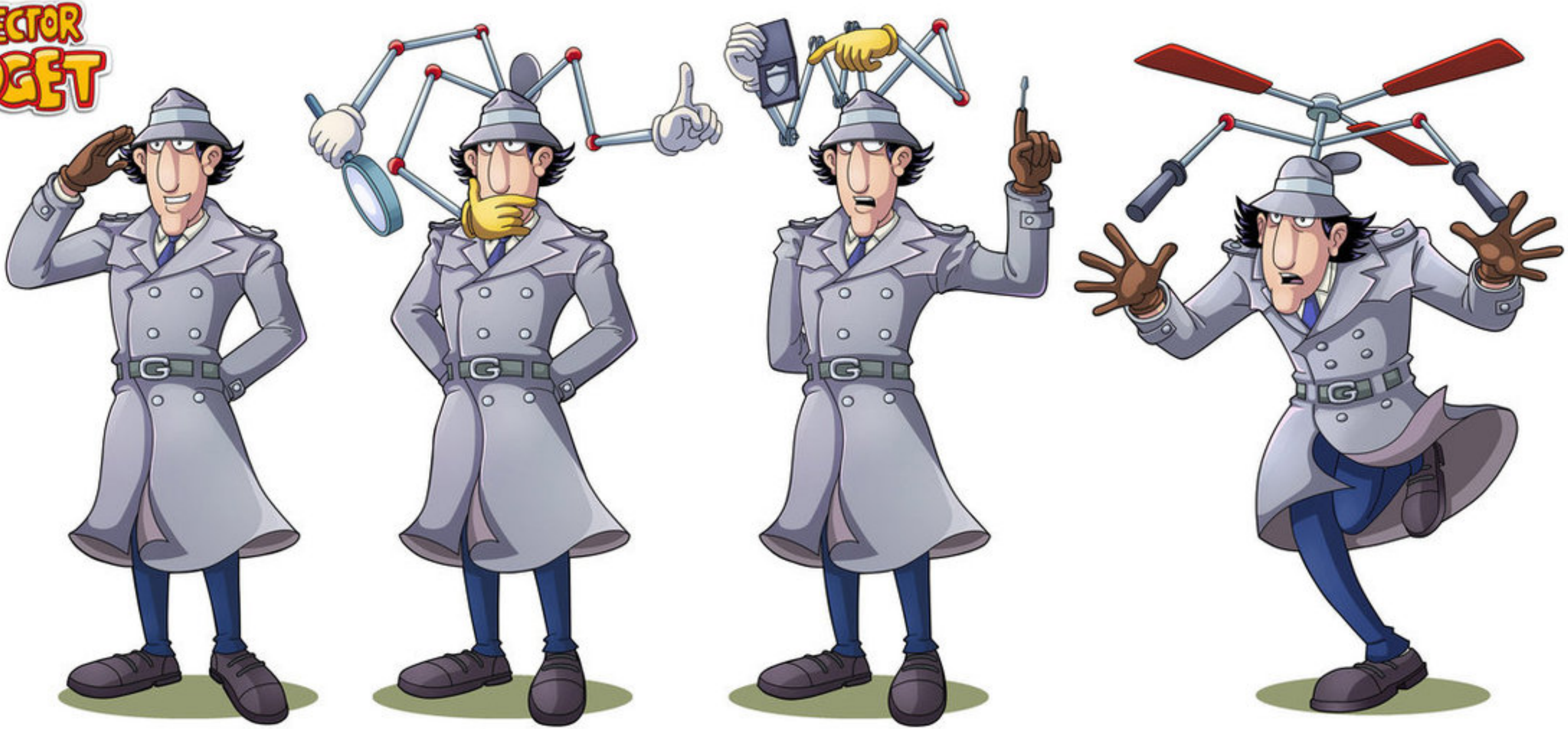
Being with several people in one room is not the same as being a team

Bruce Tuckman – Stages of group development



VX Company
IT Services

INSPECTOR GADGET



INSPECTOR GADGET
CHARACTER LINEUP
ART: José Cobá

A Scrum Master does not need to have any special skills (1)

Two types of Scrum Master

- the Scrum Master who guides the Scrum process
- the Scrum Master who guides the Scrum process *and* works with the team on items of the Sprint Backlog

Scrum Master has to

- have good communication skills
- have organizational awareness and sensitivity
- be able to be a servant leader
- … understand Scrum!

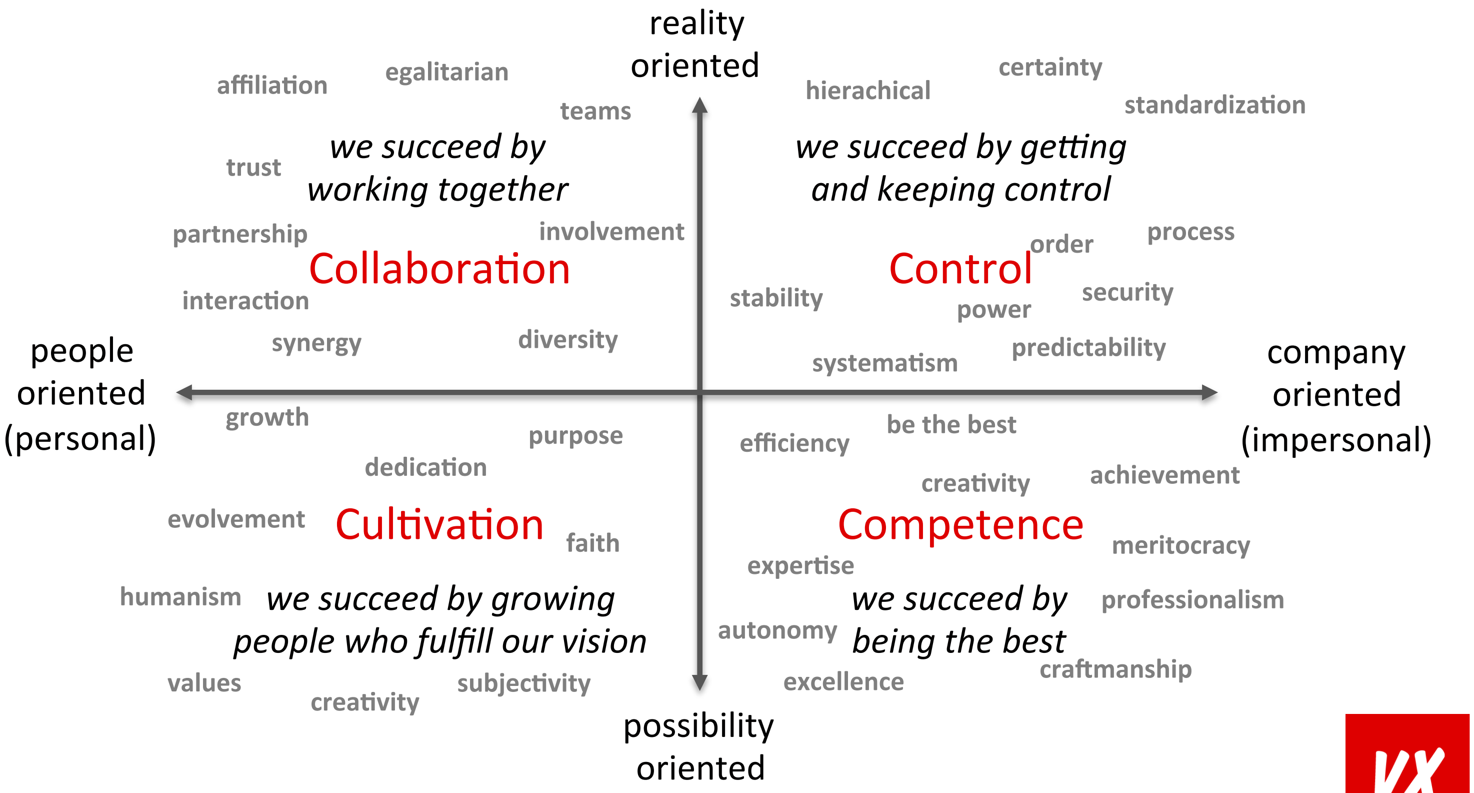
A Scrum Master does not need to have any special skills (2)

Scrum Master has many different stances

- Coach
- Teacher
- Mentor
- Advisor
- Servant Leader
- Facilitator

Not every organisational culture is “fit” for Scrum

Schneider Culture Model



reality oriented

possibility oriented

people oriented (personal)

company oriented (impersonal)

Collaboration

we succeed by working together

affiliation, egalitarian, teams, trust, partnership, involvement, interaction, synergy, diversity

Control

we succeed by getting and keeping control

hierachical, certainty, standardization, order, process, stability, power, security, systematism, predictability

Cultivation

we succeed by growing people who fulfill our vision

growth, purpose, dedication, evolvement, faith, humanism, values, creativity, subjectivity

Competence

we succeed by being the best

efficiency, be the best, creativity, achievement, meritocracy, expertise, professionalism, autonomy, craftmanship, excellence

VX Company
IT Services

Pictures

Dutch soccer team:
http://www.goal.com/nl/news/7131/ek-2016/2014/10/10/5173984/in-beeld-nederland-kazachstan/netherlands-kazakhstan-euro-2016-qualifier-10102014

Inspector Gadget: tohruichi.deviantart.com

Culture:
http://www.yourarticlelibrary.com/culture/culture-the-meaning-characteristics-and-functions/9577/