

# Mining Software Repositories

John Businge

# References



## **The Road Ahead for Mining Software Repositories**

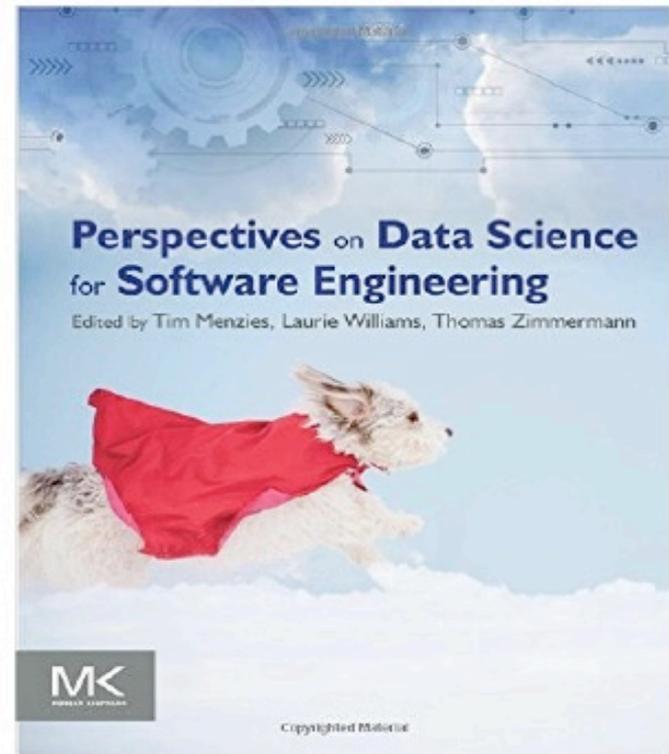
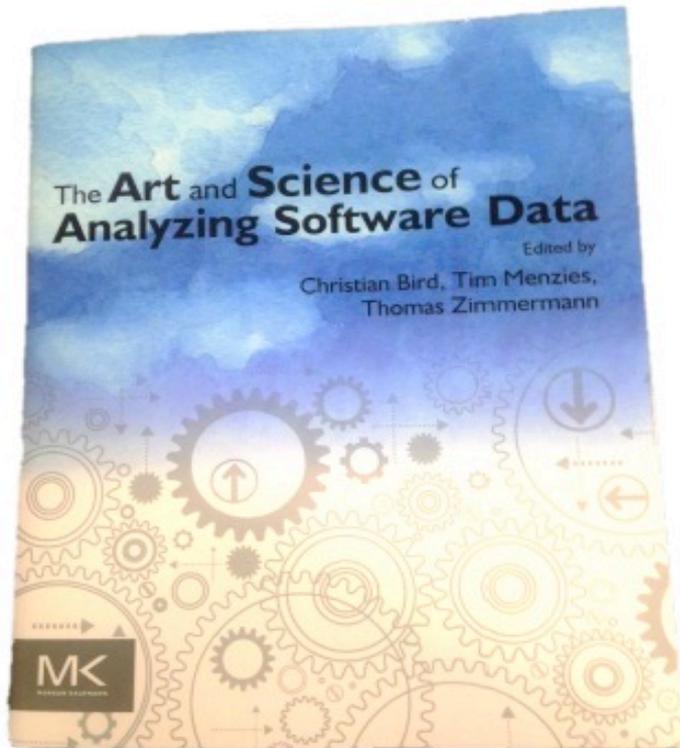
Ahmed E. Hassan  
Software Analysis and Intelligence Lab (SAIL)  
School of Computing, Queen's University, Canada  
ahmed@cs.queensu.ca

## **Software Intelligence: The Future of Mining Software Engineering Data**

Ahmed E. Hassan  
School of Computing  
Queen's University  
Kingston, ON, Canada  
ahmed@cs.queensu.ca

Tao Xie  
Department of Computer Science  
North Carolina State University  
Raleigh, NC, USA  
xie@csc.ncsu.edu

# More References



# Acknowledgement



**Ahmed E. Hassan**

Queen's University

[www.cs.queensu.ca/~ahmed](http://www.cs.queensu.ca/~ahmed)

[ahmed@cs.queensu.ca](mailto:ahmed@cs.queensu.ca)

**Tao Xie**

North Carolina State University

[www.csc.ncsu.edu/faculty/xie](http://www.csc.ncsu.edu/faculty/xie)

[xie@csc.ncsu.edu](mailto:xie@csc.ncsu.edu)

**Bram Adams**

Polytechnique Montréal, Canada

<http://mcis.polymtl.ca/index.html>

[lab.mcis@gmail.com](mailto:lab.mcis@gmail.com)

With updates by **John Businge** from the University of Nevada Las Vegas

# Lecture Goals



- **Learn about:**

- Classic and notable research and researchers in mining SE data
- Data mining and data processing techniques and how to apply them to SE data
- Risks in using SE data due to e.g., noise

- **After the lecture, you should be able to:**

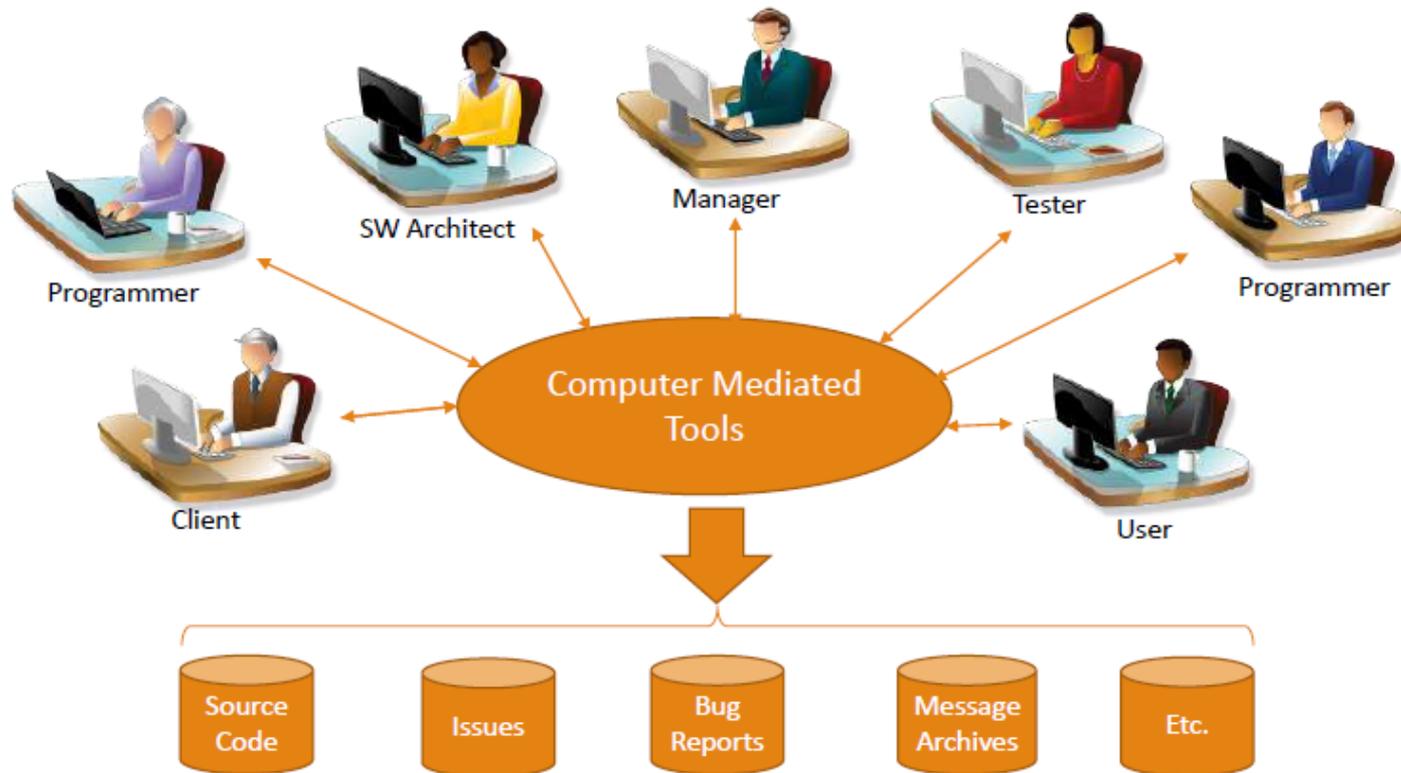
- Retrieve SE data
- Prepare SE data for mining
- Mine interesting information from SE data

# Why mine SE data?

- **SE data can be used to:**
  - Gain empirically-based understanding of software development
  - Predict, plan, and understand various aspects of a project
  - Support future development and project management activities



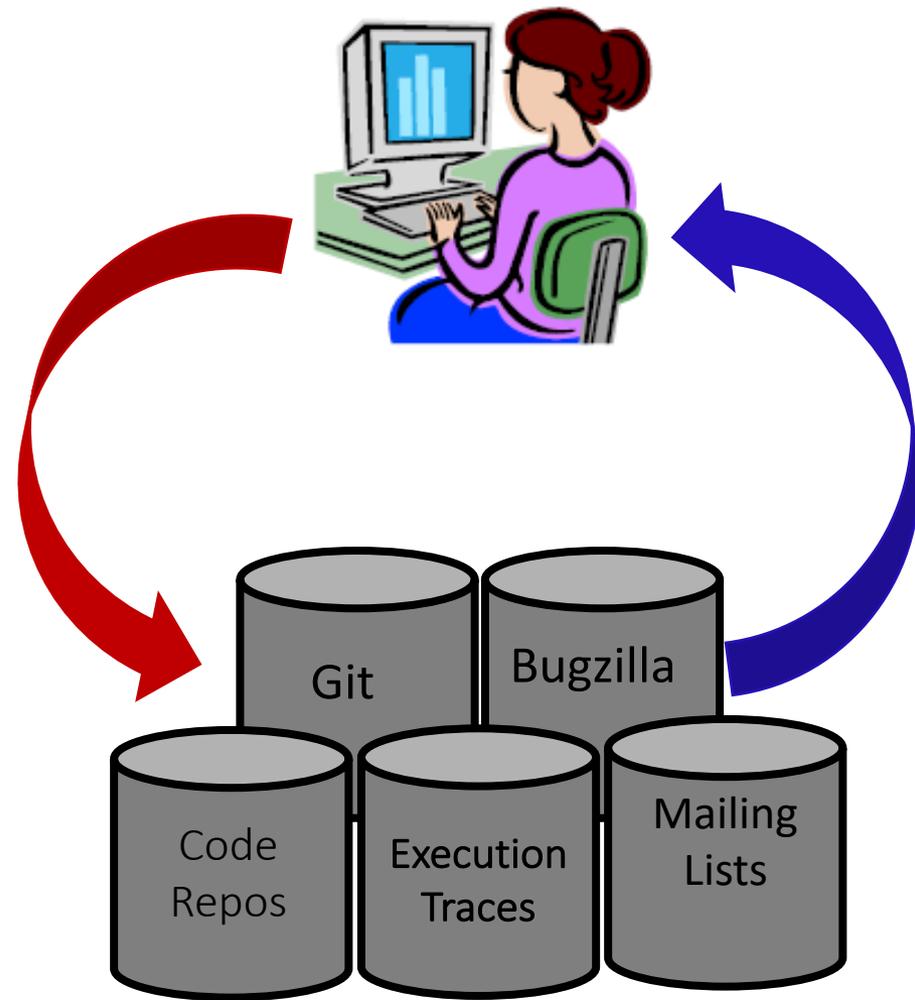
# How is SE Data generated?



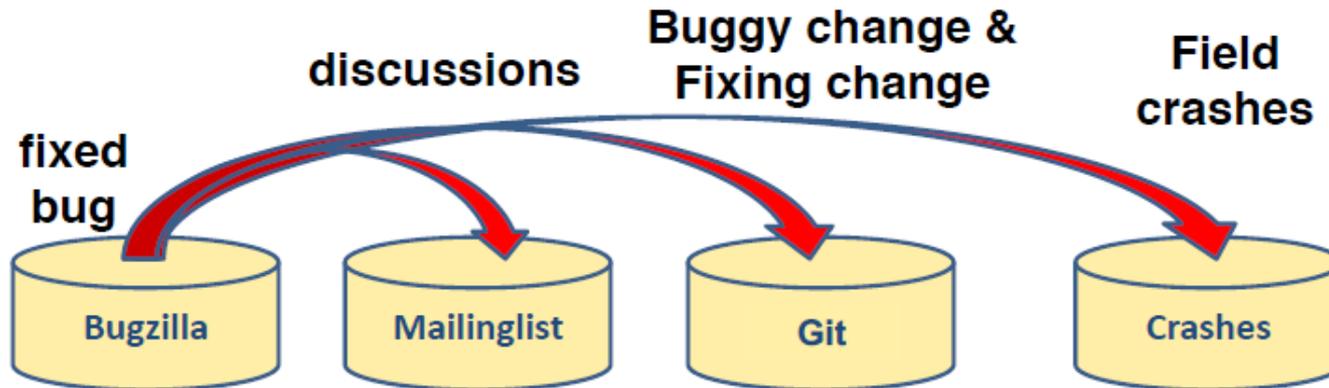
Current and historical artifacts and interactions are registered in software repositories

# What is MSR?

- Transforming static record keeping SE into active data
- Making SE data actionable by uncovering patterns and trends



# MSR researchers analyze and cross-link repositories



New bug report  
Estimate fix effort  
Suggest experts and fix!

# Study Outline



- **Part I:** What can we learn from SE data?
  - A sample of notable findings for different SE data types
- **Part II:** How can we mine SE data?
  - Understand the structure of SE data

# MSR studies – Bugs – Part I

## Using imports to predict Bugs

71% of files that import compiler packages, had to be fixed later on.

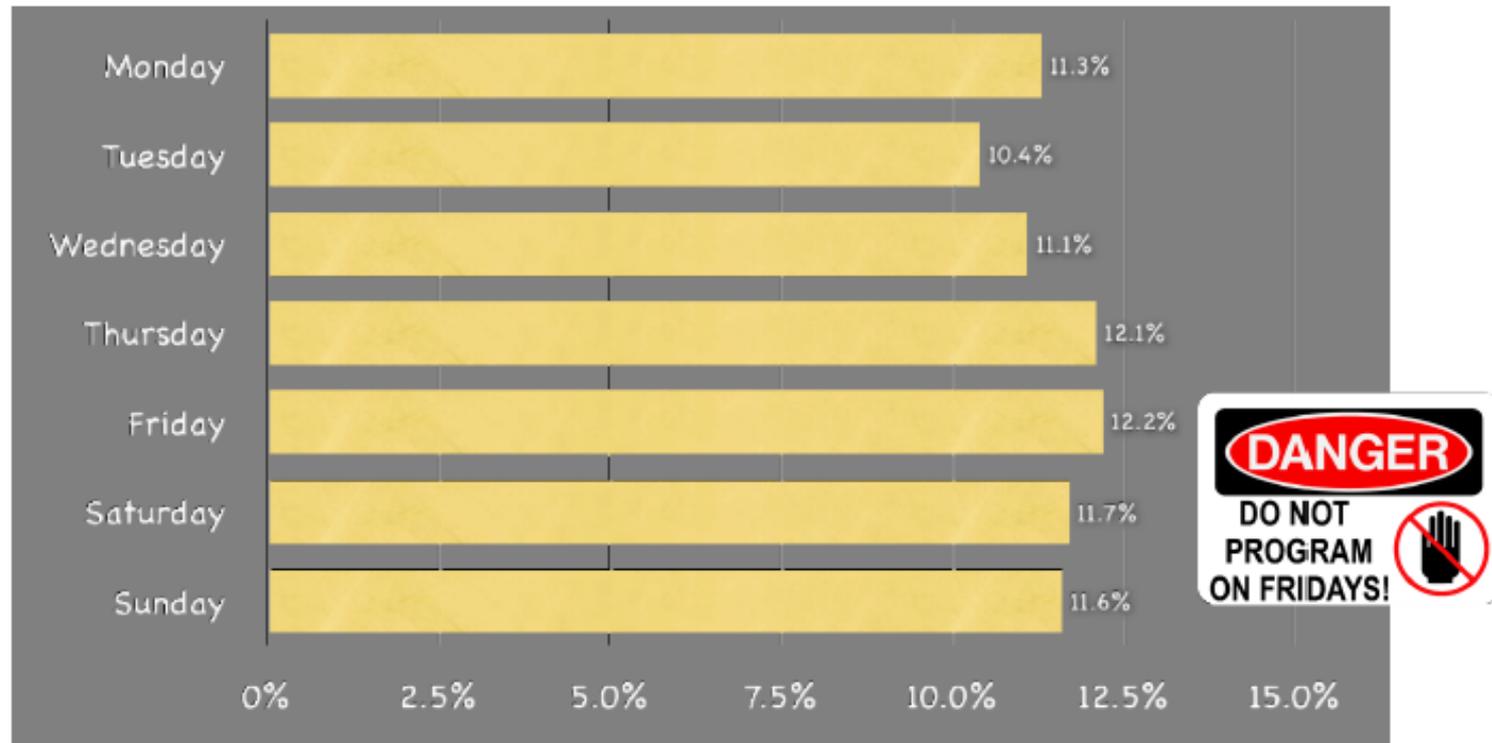
```
import org.eclipse.jdt.internal.compiler.lookup.*;
import org.eclipse.jdt.internal.compiler.*;
import org.eclipse.jdt.internal.compiler.ast.*;
import org.eclipse.jdt.internal.compiler.util.*;
...
import org.eclipse.pde.core.*;
import org.eclipse.jface.wizard.*;
import org.eclipse.ui.*;
```

[Schröter et al. 06]

14% of all files that import ui packages, had to be fixed later on.

# MSR studies - Bugs

Do not program on Friday ;-)



Percentage of bug-introducing changes for eclipse

[Zimmermann et al. 05]



# MSR studies – Sentiment Analysis



10/23/12 3:56 AM

Disable 'showmatch' option Matching parens are highlighted even without this option; what it does is jump the cursor to the matching paren which is [redacted] insane.



10/23/12 3:28 AM

styles everywhere, tutorial for first user login, fixing some css [redacted]



10/23/12 2:57 AM

[redacted] again



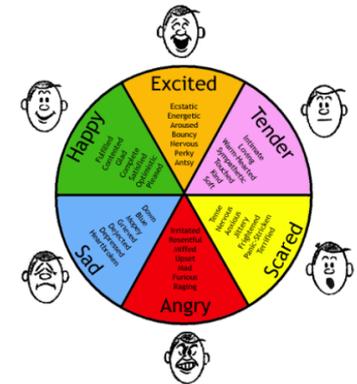
10/23/12 2:30 AM

more [redacted]



10/23/12 2:29 AM

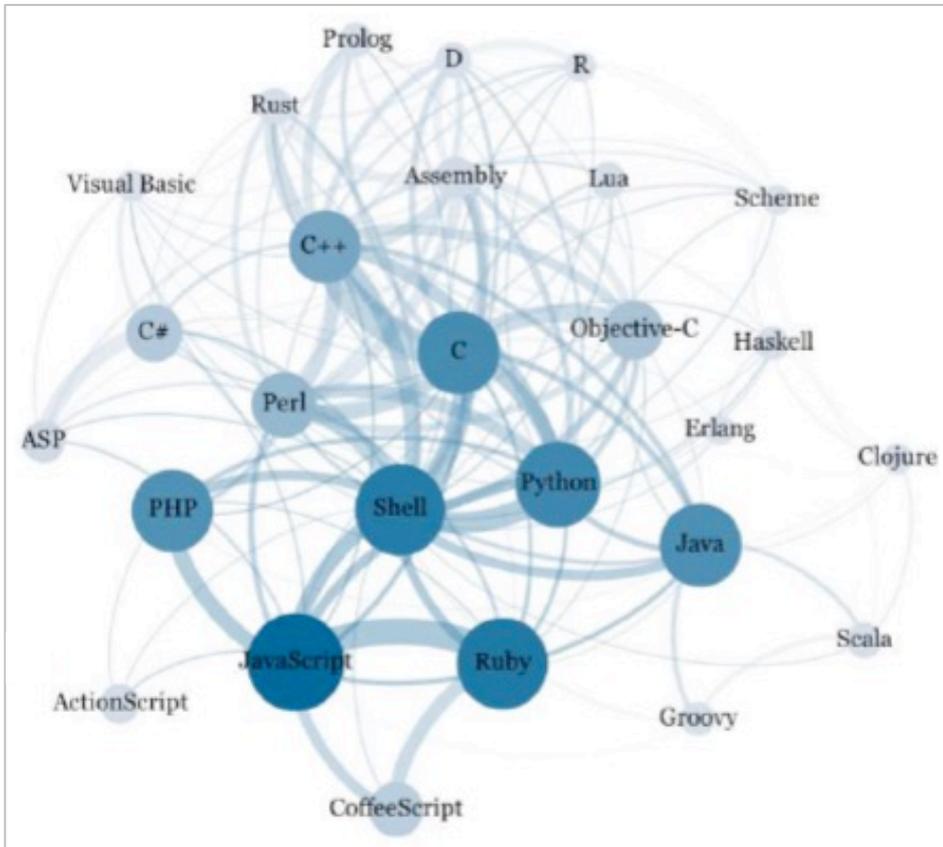
Security [redacted] worked out



<http://www.commitlogsfromlastnight.com/>

# MSR studies – Programming languages

## Programming language relations



A **Ruby** programmers is **very likely to know Javascript**, while a **Perl** programmer is not

**Java** is a popular programming language but stands primarily alone

<https://github.com/mjwillson/ProgLangVisualise>

# MSR studies – Changes by programmers

## Programming language relations

[Zimmermann et al., 2005]

Mining Version Histories to Guide Software Changes

The screenshot shows the Eclipse IDE with the following components:

- Package Explorer:** Shows a project structure with files like `CompareMessages.java`, `CompareNavigator.java`, and `ComparePreferencePage.java`.
- Code Editor:** Displays the source code for `OverlayPreferenceStore`. A red box highlights the `fKeys` field in the constructor: `public final OverlayPreferenceStore OverlayKey[] fKeys; new OverlayPreferenceStore OverlayKey[]`. A blue box highlights the `initDefaults` method: `public static void initDefaults(IPreferenceStore store) {`.
- Related Changes Table:** A table at the bottom of the editor window showing related changes. The first row is highlighted in blue.

Symbol	File	Support	Confidence
<code>initDefaults(IPreferenceStore store)</code>	<code>ComparePreferencePage.java</code>	7	1.0
<code>org.eclipse.compare.plugin.properties</code>	<code>plugin.properties</code>	6	0.875
<code>org.eclipse.compare.htmlnotes.compare.html</code>	<code>htmlnotes_compare.html</code>	6	0.75
<code>TextMergeViewer(Composite parent, int style, CompareConfiguration configuration)</code>	<code>TextMergeViewer.java</code>	6	0.75
<code>propertyChanged(PropertyChangeEvent event)</code>	<code>TextMergeViewer.java</code>	6	0.75
<code>createGeneralPage(Composite parent)</code>	<code>ComparePreferencePage.java</code>	5	0.625
<code>createTextComparePage(Composite parent)</code>	<code>ComparePreferencePage.java</code>	5	0.625
<code>handleDispose(DisposeEvent event)</code>	<code>TextMergeViewer.java</code>	4	0.5

A) The user inserts a new preference into the field fKeys

B) ROSE suggests locations for further changes, e.g. the function initDefaults()

After the programmer has made some changes to the source (above), ROSE suggests locations (below) where, in similar transactions in the past, further changes were made

- Suggests and predicts likely changes
- Prevents errors due to incomplete changes

# How can we mine SE Data – Part II

## Repositories of Repositories



January 2020:  
100 Million repositories  
40 Million Users



January 2020:  
430K repositories  
3.7 Million Users



April 2019  
28 Million repositories  
10 Million Users



April 2019  
28K projects



# How can we mine SE Data form GitHub

---



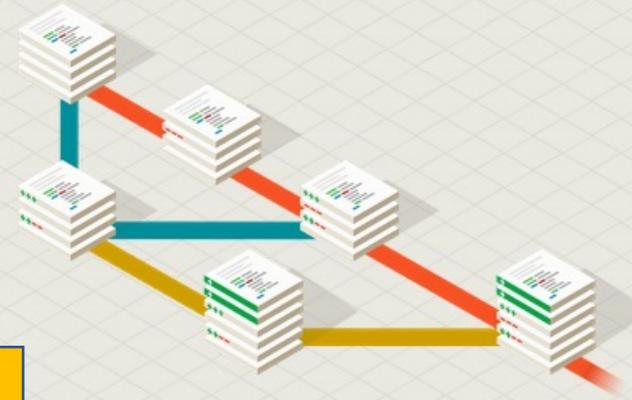
# How can we mine SE Data



Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.



Easiest to obtain local copy (distributed version control!), and has distinction between authors and committers, but ... branches can be pain to analyze



# How can we mine SE Data

The screenshot shows the GitHub homepage for user 'bramadams'. At the top, there is a search bar and navigation links for 'Explore', 'Gist', 'Blog', and 'Help'. Below the navigation, there are tabs for 'News Feed', 'Pull Requests', and 'Issues'. The main content area features the 'GitHub Bootcamp' guide, which consists of four steps:

- 1 Set up Git**: A quick guide to help you get started with Git.
- 2 Create repositories**: Repositories are where you'll work and collaborate on projects.
- 3 Fork repositories**: Forking creates a new, unique project from an existing one.
- 4 Work together**: Send pull requests, follow friends. Star and watch projects.

Below the bootcamp guide, there is a yellow box with the text: "Access to thousands of Git-based projects via GitHub". To the right of this box, there is a notification for "Better Word Highlighting in Diffs" and a list of repositories the user contributes to:

Repositories you contribute to	
inuکشuk/jekyll-scholar	152 ★
smcintosh/moosetracks	1 ★

# How can we mine GitHub Data

GitHub Developer

Docs ▾ Blog Forum Versions ▾

## REST API v3

Reference Guides Libraries

### Overview

This describes the resources that make up the official GitHub REST API v3. If you have any problems or requests, please contact [GitHub Support](#) or [GitHub Premium Support](#).

- i. [Current version](#)
- ii. [Schema](#)
- iii. [Authentication](#)
- iv. [Parameters](#)
- v. [Root endpoint](#)
- vi. [GraphQL global node IDs](#)
- vii. [Client errors](#)
- viii. [HTTP redirects](#)
- ix. [HTTP verbs](#)
- x. [Hypertext](#)

▼ Overview

- [Media Types](#)
- [OAuth Authorizations API](#)
- [Other Authentication Methods](#)
- [Troubleshooting](#)
- [API Previews](#)
- [Versions](#)
- ▶ Activity
- ▶ Checks
- ▶ Gists
- ▶ Git Data
- ▶ GitHub Actions
- ▶ GitHub Apps
- GitHub Marketplace
- ▶ Interactions
- ▶ Issues

How can we obtain GitHub data?

# How can we mine GitHub Data

GitHub Developer Docs Blog Forum Versions Search...

## REST API v3

Reference Guides Libraries

### Overview

This describes the resources that make up the official GitHub REST API v3. If you have any problems or requests, please contact [GitHub Support](#) or [GitHub Premium Support](#).

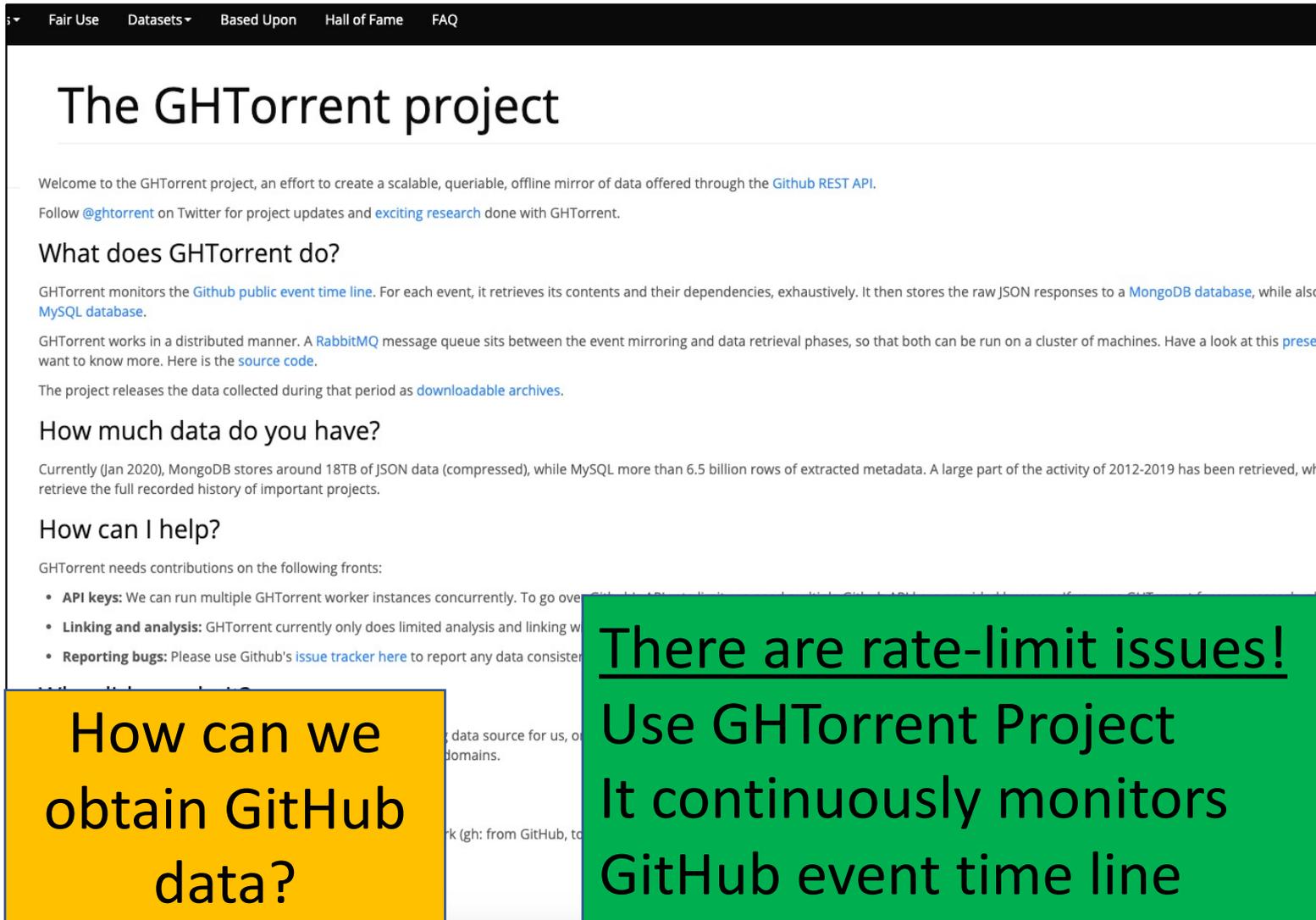
- [Current version](#)
- [Schema](#)
- [Authentication](#)
- [Parameters](#)
- [Root endpoint](#)
- [GraphQL global node IDs](#)
- [Client errors](#)
- [HTTP redirects](#)
- [HTTP verbs](#)
- [Hypertext](#)

- Overview
  - Media Types
  - OAuth Authorizations API
  - Other Authentication Methods
  - Troubleshooting
  - API Previews
  - Versions
- Activity

**How can we obtain GitHub data?**

**There are rate-limit issues!  
You are only allowed only a limited number of GitHub requests per hour**

# How can we mine GitHub Data



The screenshot shows the GHTorrent project website. The navigation bar includes links for Fair Use, Datasets, Based Upon, Hall of Fame, and FAQ. The main heading is "The GHTorrent project". Below it, there is a welcome message and a list of links for Twitter, research, and source code. The page is divided into sections: "What does GHTorrent do?", "How much data do you have?", and "How can I help?". The "How can I help?" section lists three points: API keys, linking and analysis, and reporting bugs. Two callout boxes are overlaid on the bottom of the screenshot: a yellow one on the left and a green one on the right.

Fair Use Datasets Based Upon Hall of Fame FAQ

## The GHTorrent project

Welcome to the GHTorrent project, an effort to create a scalable, queryable, offline mirror of data offered through the [Github REST API](#).

Follow [@ghtorrent](#) on Twitter for project updates and [exciting research](#) done with GHTorrent.

### What does GHTorrent do?

GHTorrent monitors the [Github public event time line](#). For each event, it retrieves its contents and their dependencies, exhaustively. It then stores the raw JSON responses to a [MongoDB database](#), while also [MySQL database](#).

GHTorrent works in a distributed manner. A [RabbitMQ](#) message queue sits between the event mirroring and data retrieval phases, so that both can be run on a cluster of machines. Have a look at this [presentation](#) if you want to know more. Here is the [source code](#).

The project releases the data collected during that period as [downloadable archives](#).

### How much data do you have?

Currently (Jan 2020), MongoDB stores around 18TB of JSON data (compressed), while MySQL more than 6.5 billion rows of extracted metadata. A large part of the activity of 2012-2019 has been retrieved, which allows us to retrieve the full recorded history of important projects.

### How can I help?

GHTorrent needs contributions on the following fronts:

- **API keys:** We can run multiple GHTorrent worker instances concurrently. To go over [Github API rate limits](#), we need more keys. If you have a key, please [contact us](#).
- **Linking and analysis:** GHTorrent currently only does limited analysis and linking with [external data sources](#). We would like to see more [external data sources](#) for us, or [domains](#).
- **Reporting bugs:** Please use Github's [issue tracker here](#) to report any data consistency issues.

data source for us, or domains.

rk (gh: from GitHub, to

**How can we obtain GitHub data?**

**There are rate-limit issues!**  
**Use GHTorrent Project**  
**It continuously monitors GitHub event time line**

# How can we mine GitHub Data

GitHub Developer

Docs ▾ Blog Forum Versions ▾ Search...

## REST API v3

Reference Guides Libraries

### Overview

This describes the resources that make up the official GitHub REST API v3. If you have any problems or requests, please contact [GitHub Support](#) or [GitHub Premium Support](#).

- i. [Current version](#)
- ii. [Schema](#)
- iii. [Authentication](#)
- iv. [Parameters](#)
- v. [Root endpoint](#)
- vi. [GraphQL global node IDs](#)
- vii. [Client errors](#)
- viii. [HTTP redirects](#)
- ix. [HTTP verbs](#)
- x. [Hypertext](#)

▼ Overview

- [Media Types](#)
- [OAuth Authorizations API](#)
- [Other Authentication Methods](#)
- [Troubleshooting](#)
- [API Previews](#)
- [Versions](#)
- ▶ Activity

How can we obtain GitHub data?

There are rate-limit issues!  
GitHub allows one to use authentication where u get a token (5000 requests/hr)