

# Mining Software Repositories

John Businge

# References



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

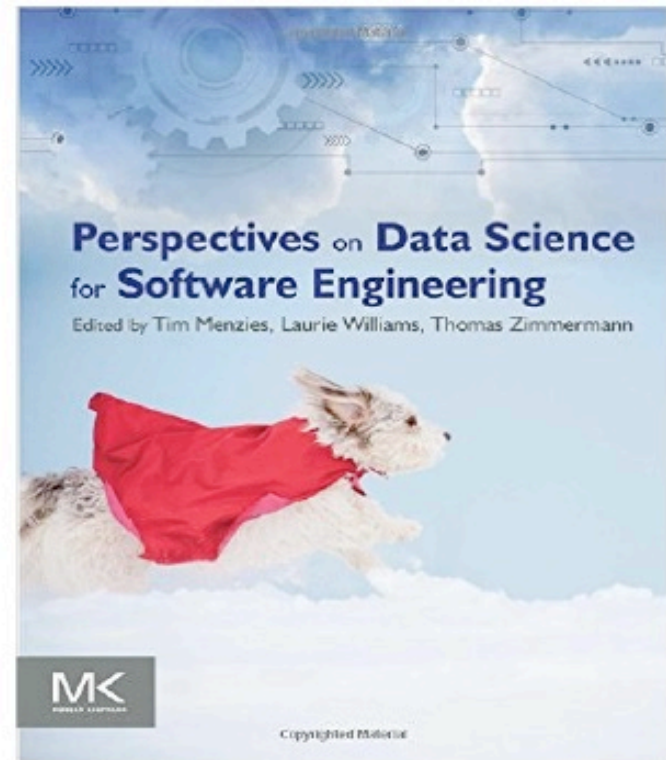
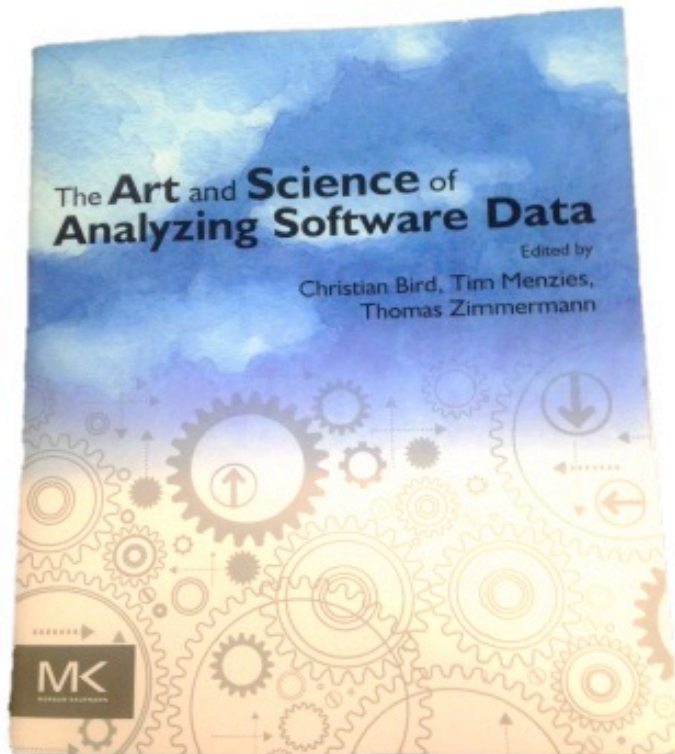
Ahmed E. Hassan  
Software Analysis and Intelligence Lab (SAIL)  
School of Computing, Queen's University, Canada  
ahmed@cs.queensu.ca

## **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 University of Antwerp, Belgium

# 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

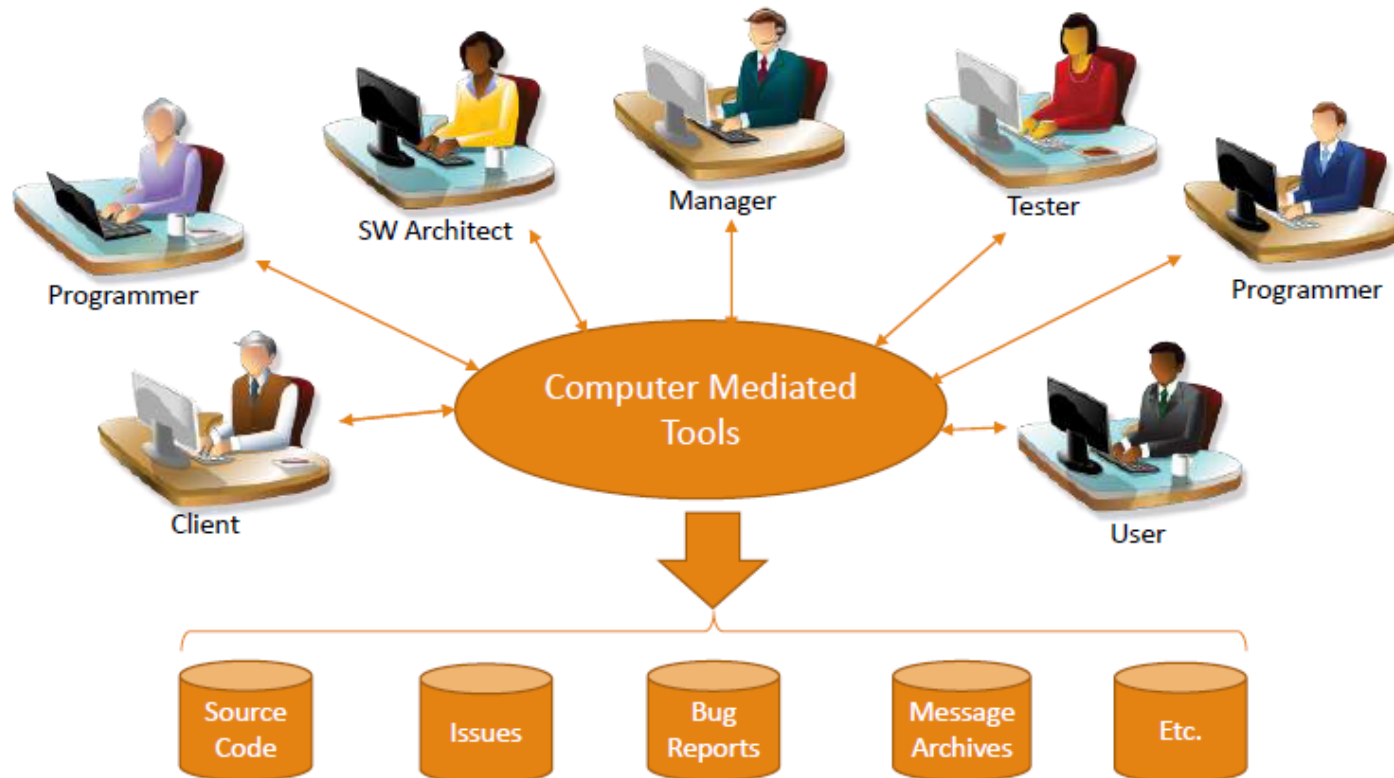
# Why mine SE data?

- **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



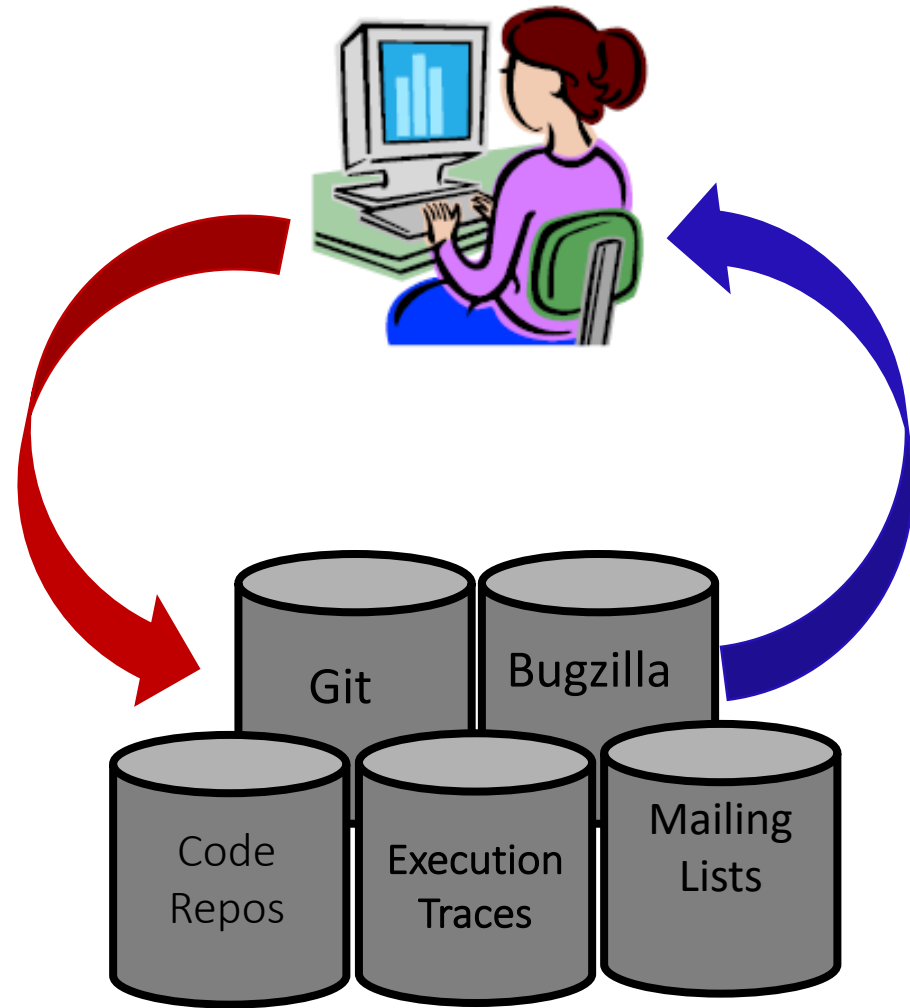
# How is SE Data generated?



Current and historical artifacts and interactions are registered in software repositories

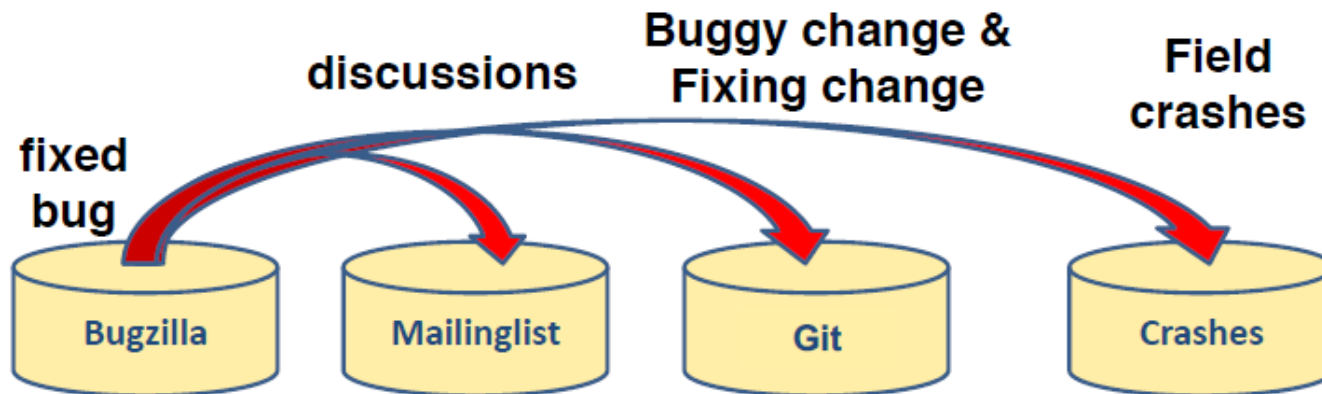
# What is MSR?

- 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.

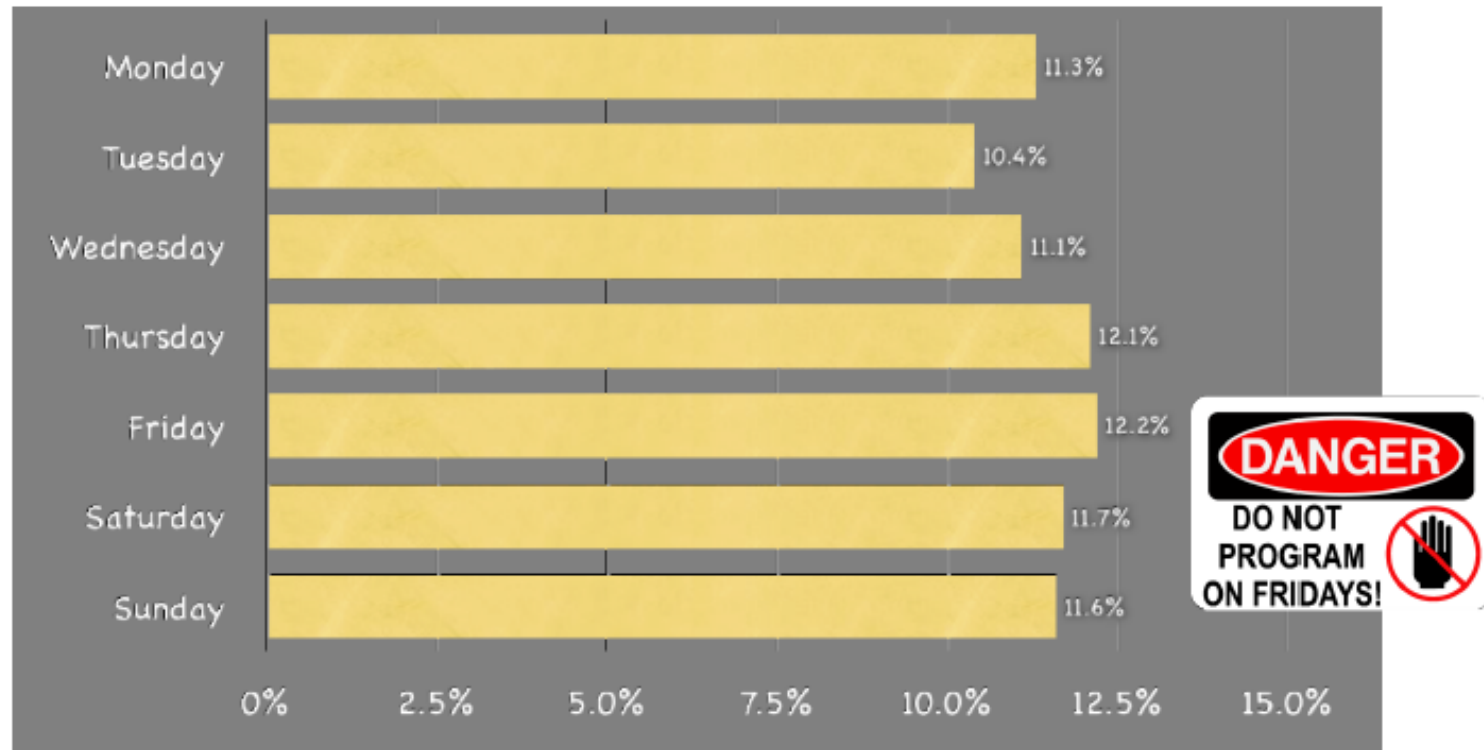
```
import org.eclipse.jdt.internal.compiler.lookup.*;
import org.eclipse.jdt.internal.compiler.*;
import org.eclipse.jdt.internal.compiler.ast.*;
import org.eclipse.jdt.internal.compiler.util.*;
...
import org.eclipse.pde.core.*;
import org.eclipse.jface.wizard.*;
import org.eclipse.ui.*;
```

[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



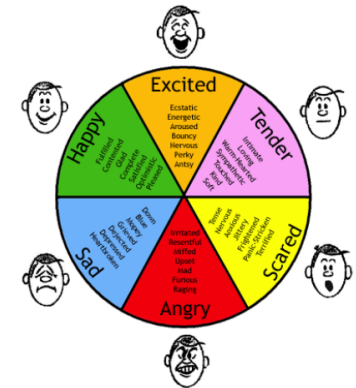
10/23/12 2:30 AM

more [REDACTED]



10/23/12 2:29 AM

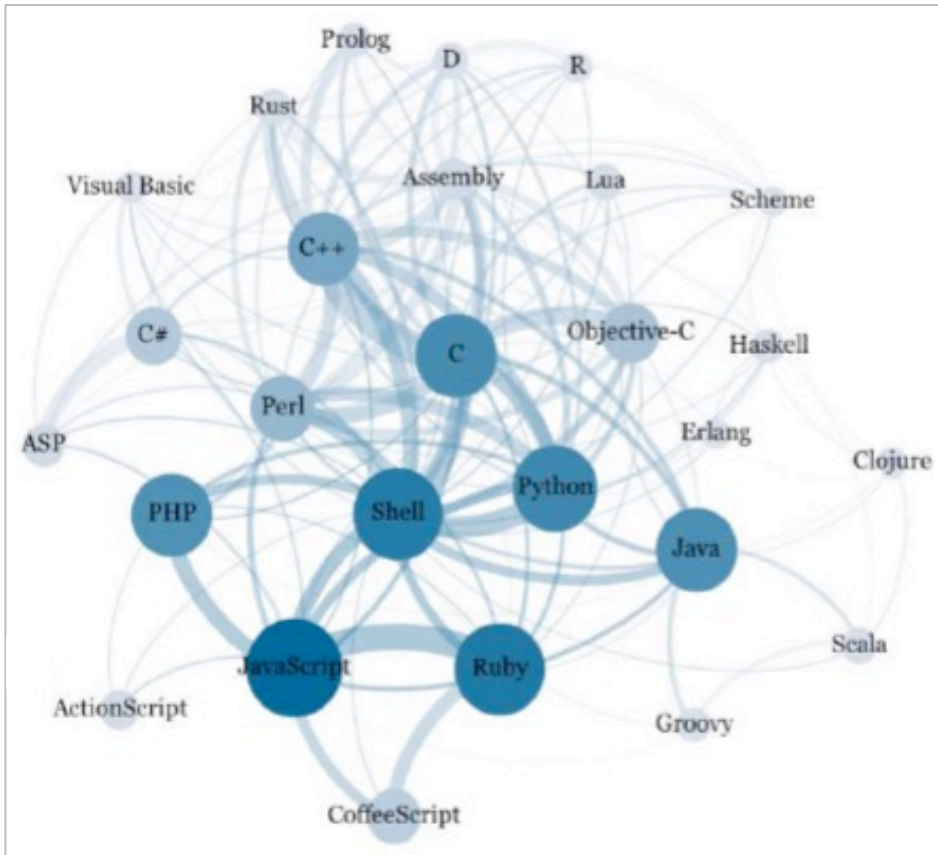
Security [REDACTED] worked out



<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 file 'ComparePreferencePage.java' open. The code defines an 'OverlayPreferenceStore' and an 'initDefaults' method. Annotations include 'A) The user inserts a new preference into the field fKeys[]' pointing to a line in the 'OverlayPreferenceStore' and 'B) ROSE suggests locations for further changes, e.g. the function initDefaults()' pointing to the 'initDefaults' method. A 'Related Changes' table is visible at the bottom.

Symbol	File	Support	Confidence
initDefaults(PreferenceStore store)	ComparePreferencePage.java	8	1.0
org.eclipse.compare.plugin.properties	plugin.properties	7	0.875
org.eclipse.compare.hudnotes.compare.html	hudnotes_compare.html	6	0.75
TextMergeViewer.java	TextMergeViewer.java	6	0.75
propertyChange(PropertyChangeEvent event)	TextMergeViewer.java	6	0.75
createGeneralPage(Composite parent)	ComparePreferencePage.java	5	0.625
createTextComparePage(Composite parent)	ComparePreferencePage.java	5	0.625
handleDispose(DisposeEvent event)	TextMergeViewer.java	4	0.5

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



Explore Gist Blog Help


bramadams + -

bramadams

News Feed Pull Requests Issues


### GitHub Bootcamp

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.

Access to thousands of  
Git-based projects via  
GitHub





**Better Word Highlighting in Diffs**

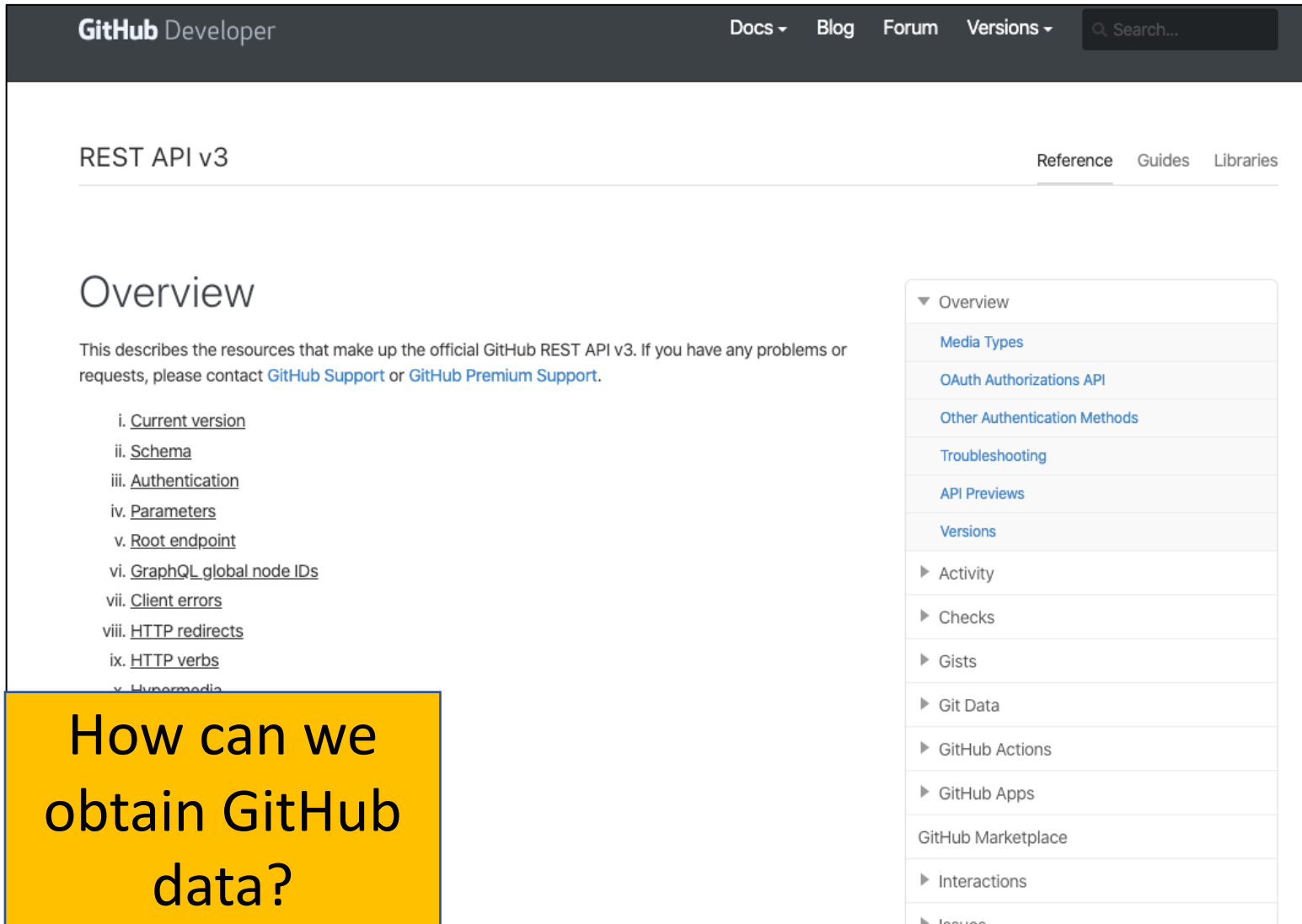
Commits, compare views, and pull requests now highlight individual changed words.

[View 70 new broadcasts](#)

Repositories you contribute to

 <a href="#">inukshuk/jekyll-scholar</a>	152 ★
 <a href="#">smcintosh/moosetracks</a>	1 ★

# How can we mine GitHub Data



The screenshot shows the GitHub Developer REST API v3 Overview page. The page has a dark header with the GitHub Developer logo and navigation links for Docs, Blog, Forum, and Versions. A search bar is located on the right. The main content area is titled 'REST API v3' and includes tabs for Reference, Guides, and Libraries. The 'Overview' section is active, displaying a list of links for various API features. A yellow callout box is overlaid on the bottom left of the page, containing the text 'How can we obtain 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 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](#).

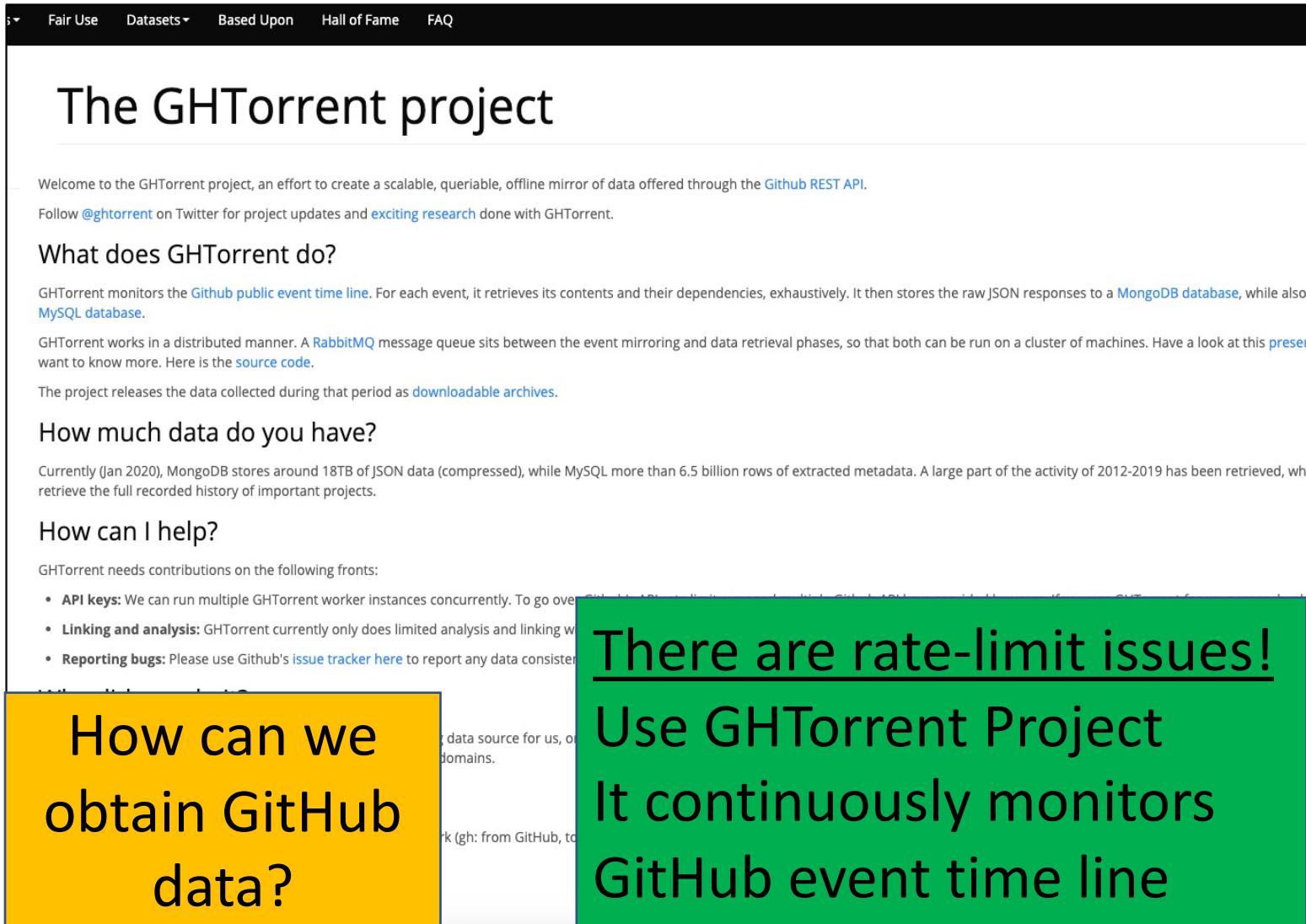
- i. [Current version](#)
- ii. [Schema](#)
- iii. [Authentication](#)
- iv. [Parameters](#)
- v. [Root endpoint](#)
- vi. [GraphQL global node IDs](#)
- vii. [Client errors](#)
- viii. [HTTP redirects](#)
- ix. [HTTP verbs](#)
- x. [Hypertext](#)

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

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

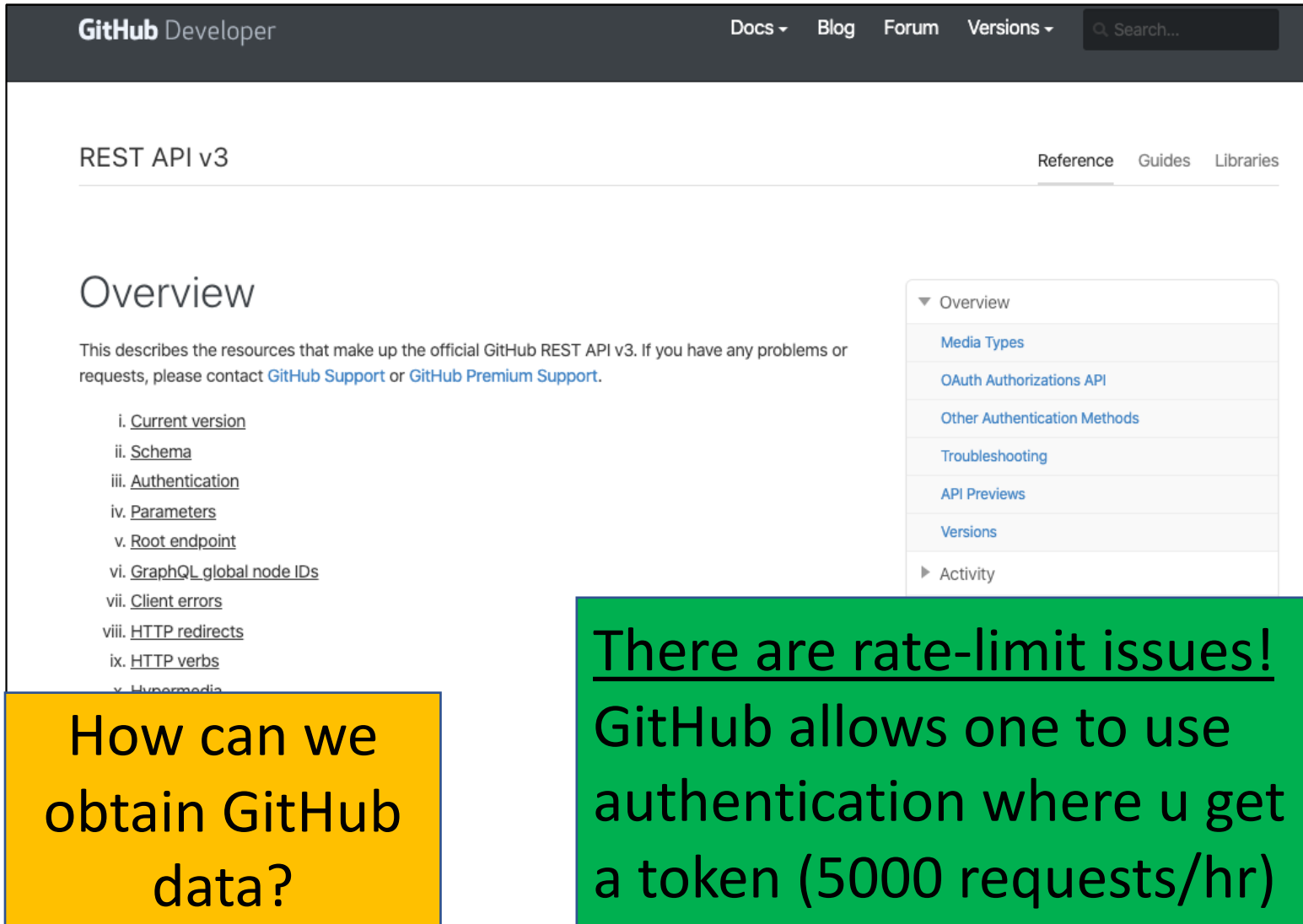


The screenshot shows the GHTorrent project website. The navigation bar includes links for Fair Use, Datasets, Based Upon, Hall of Fame, and FAQ. The main heading is "The GHTorrent project". The text describes the project as a scalable, queriable, offline mirror of data offered through the [Github REST API](#). It mentions following [@ghorrent](#) on Twitter and [exciting research](#) done with GHTorrent. A section titled "What does GHTorrent do?" explains that it monitors the [Github public event time line](#), retrieves contents and dependencies, and stores raw JSON responses to a [MongoDB database](#) and a [MySQL database](#). It also mentions a [RabbitMQ](#) message queue and a [source code](#) link. Another section titled "How much data do you have?" states that as of Jan 2020, MongoDB stores around 18TB of JSON data (compressed), while MySQL stores more than 6.5 billion rows of extracted metadata. A section titled "How can I help?" lists contributions: API keys, linking and analysis, and reporting bugs. Two callout boxes are overlaid on the bottom of the screenshot: an orange one on the left and a green one on the right.

**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



The screenshot shows the GitHub Developer REST API v3 Overview page. The page has a dark header with 'GitHub Developer' on the left and navigation links 'Docs', 'Blog', 'Forum', and 'Versions' on the right, along with a search bar. Below the header, the page title 'REST API v3' is followed by tabs for 'Reference', 'Guides', and 'Libraries'. The main content area is titled 'Overview' and contains a paragraph about the resources and a list of links: '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](#)', and 'x. [Hypertext](#)'. On the right side, there is a sidebar with a '▼ Overview' section containing links to 'Media Types', 'OAuth Authorizations API', 'Other Authentication Methods', 'Troubleshooting', 'API Previews', and 'Versions', and an '► Activity' section. Two callout boxes are overlaid on the page: a yellow one at the bottom left and a green one at the bottom right.

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

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)