

Mining Software Repositories

John Businge

References



The Road Ahead for Mining Software Repositories

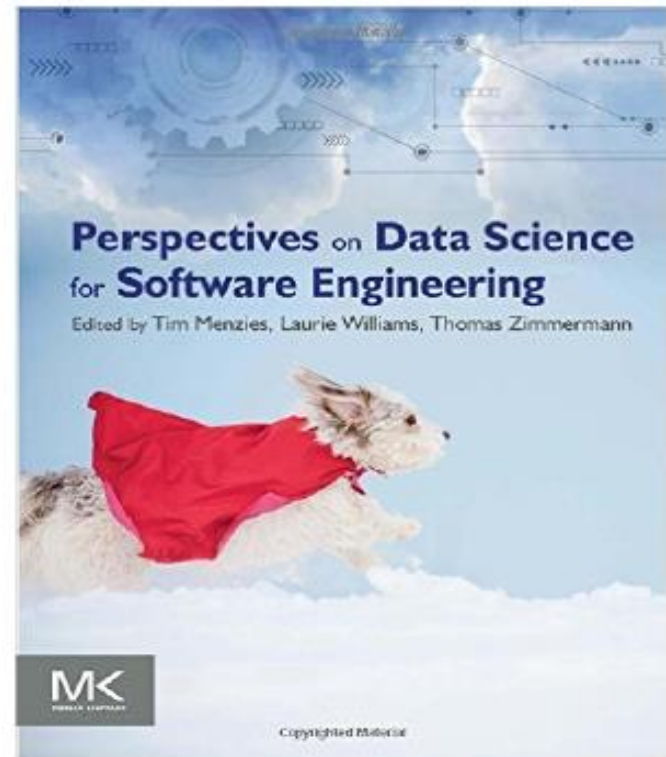
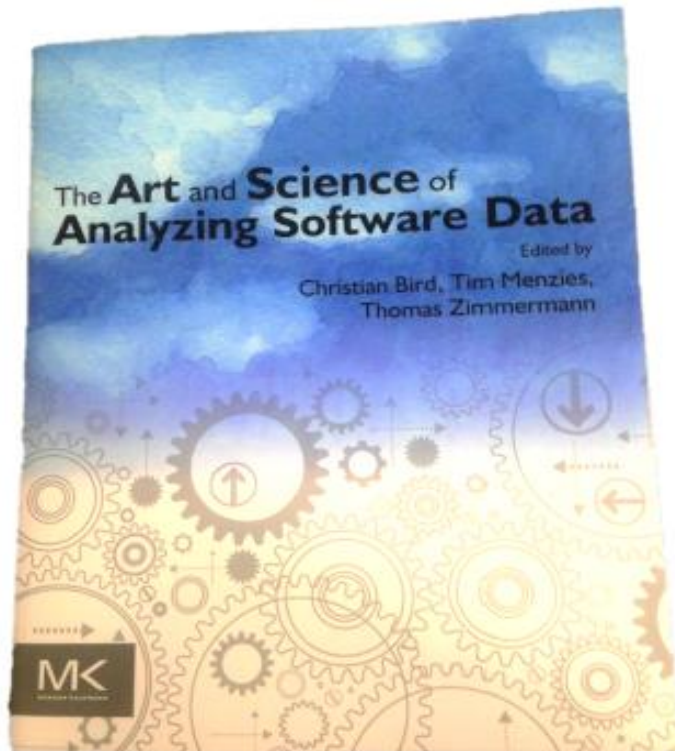
Ahmed E. Hassan
Software Analysis and Intelligence Lab (SAIL)
School of Computing, Queen's University, Canada
ahmed@cs.queensu.ca

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

ahmed@cs.queensu.ca

Tao Xie

North Carolina State University

www.csc.ncsu.edu/faculty/xie

xie@csc.ncsu.edu

Bram Adams

Polytechnique Montréal, Canada

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

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:**

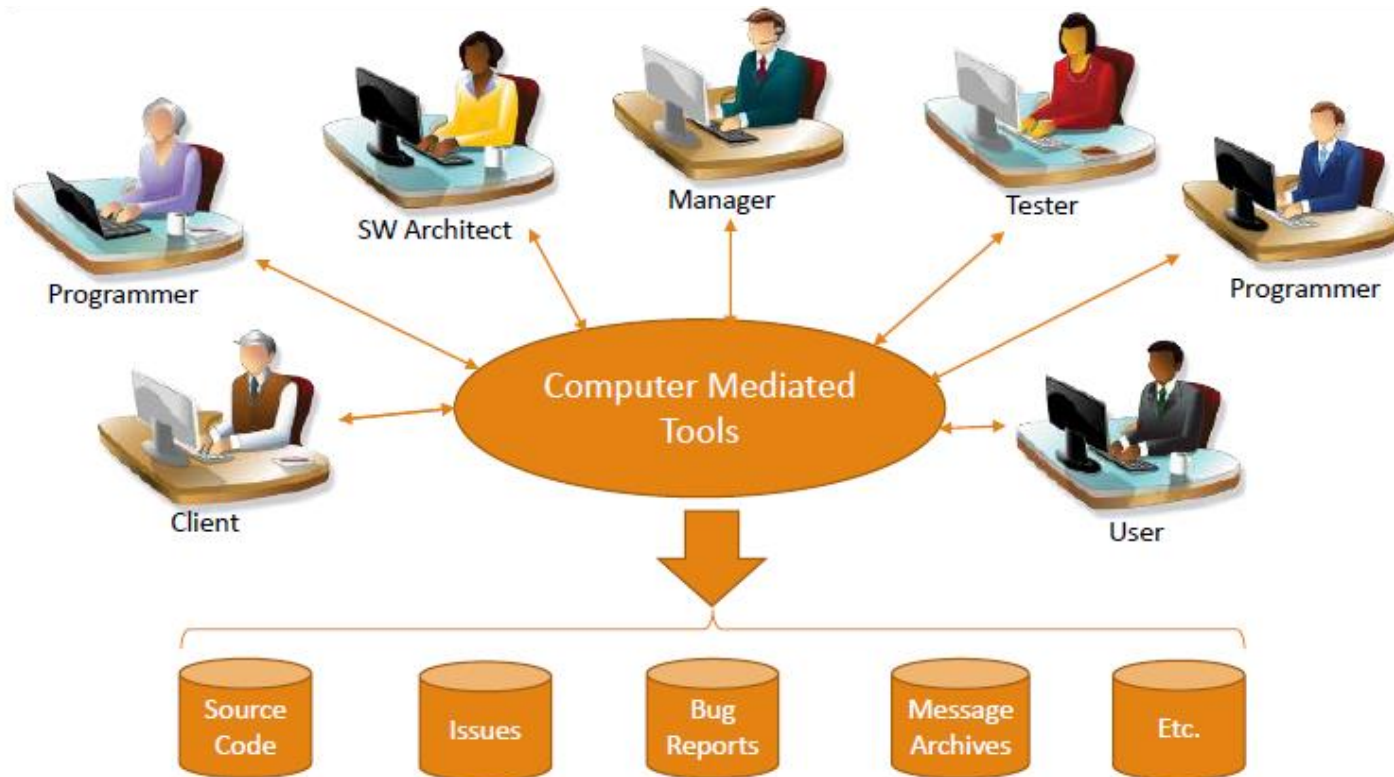
- Retrieve SE data
- Prepare SE data for mining
- Mine interesting information from SE data

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



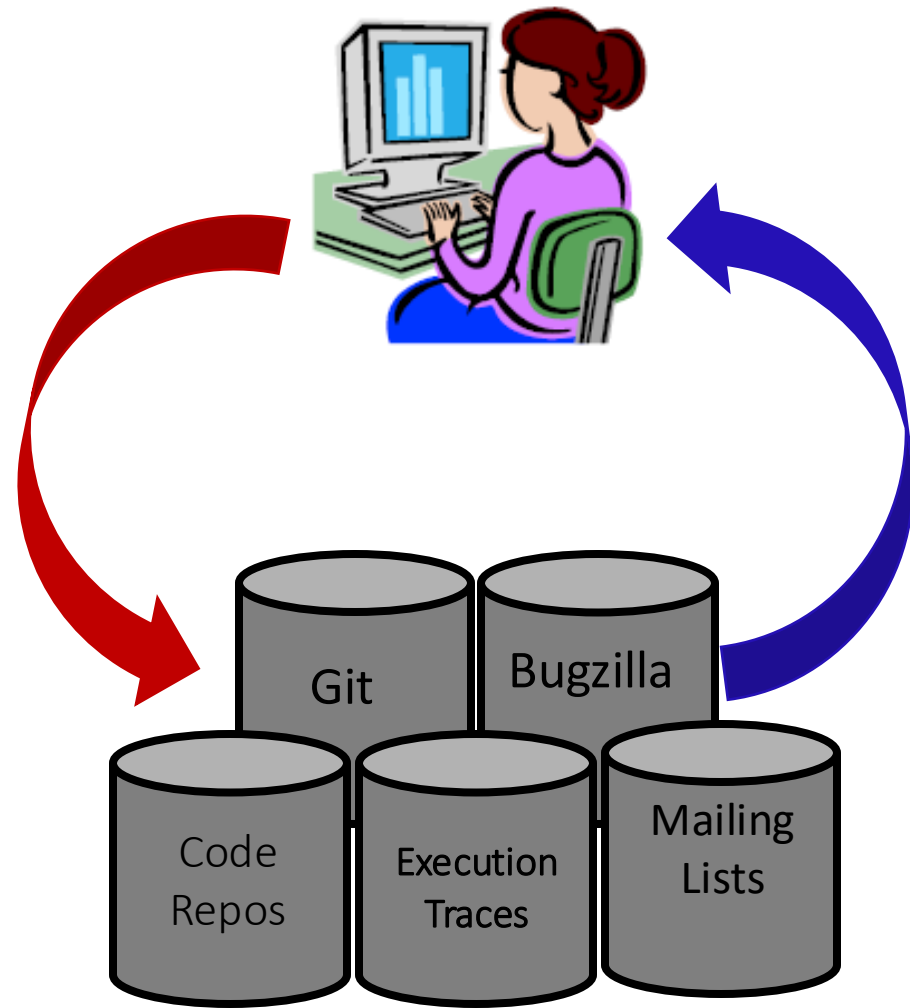
How is SE Data generated?



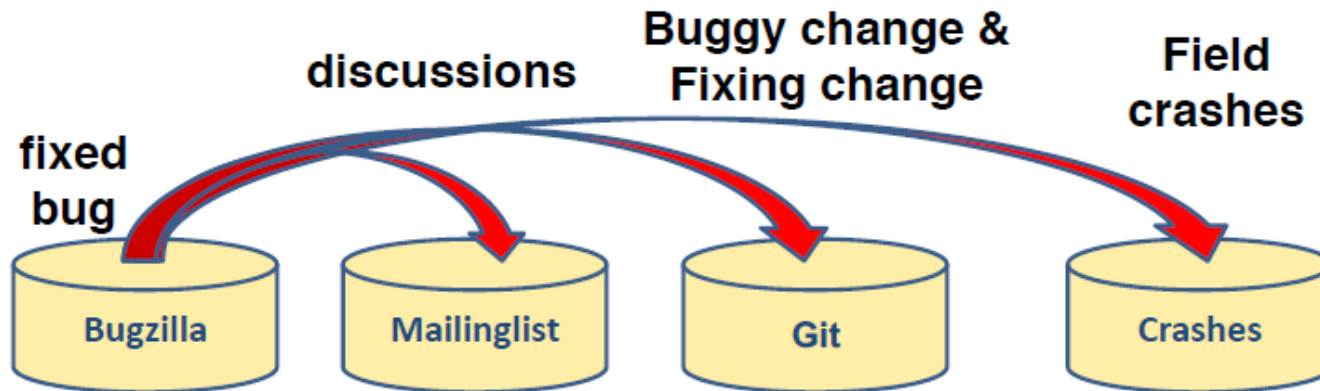
Current and historical artifacts and interactions are registered in software repositories

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.

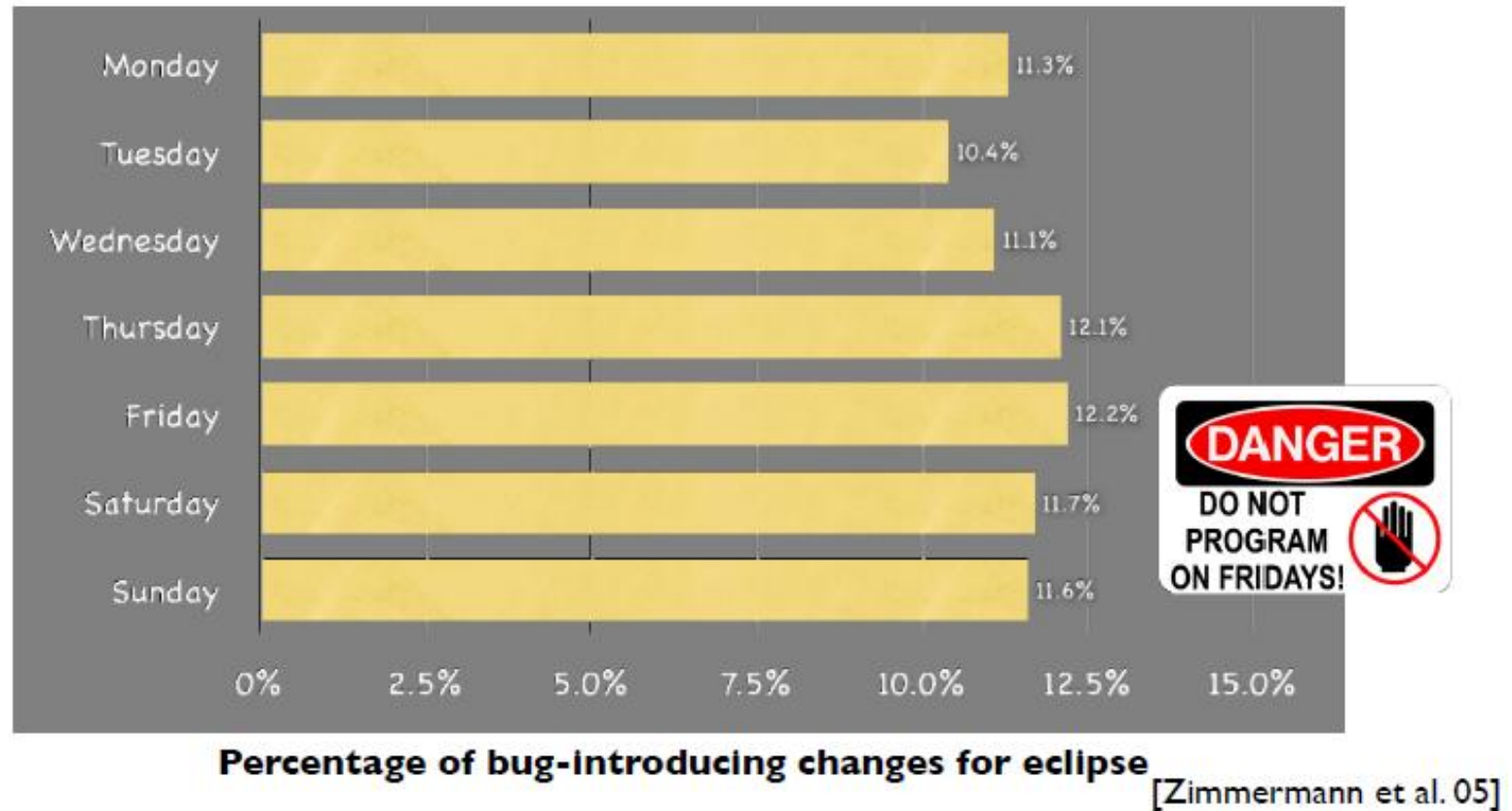
```
import org.eclipse.jdt.internal.compiler.lookup.*;  
import org.eclipse.jdt.internal.compiler.*;  
import org.eclipse.jdt.internal.compiler.ast.*;  
import org.eclipse.jdt.internal.compiler.util.*;  
...  
import org.eclipse.pde.core.*;  
import org.eclipse.jface.wizard.*;  
import org.eclipse.ui.*;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

[Schröter et al. 06]

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

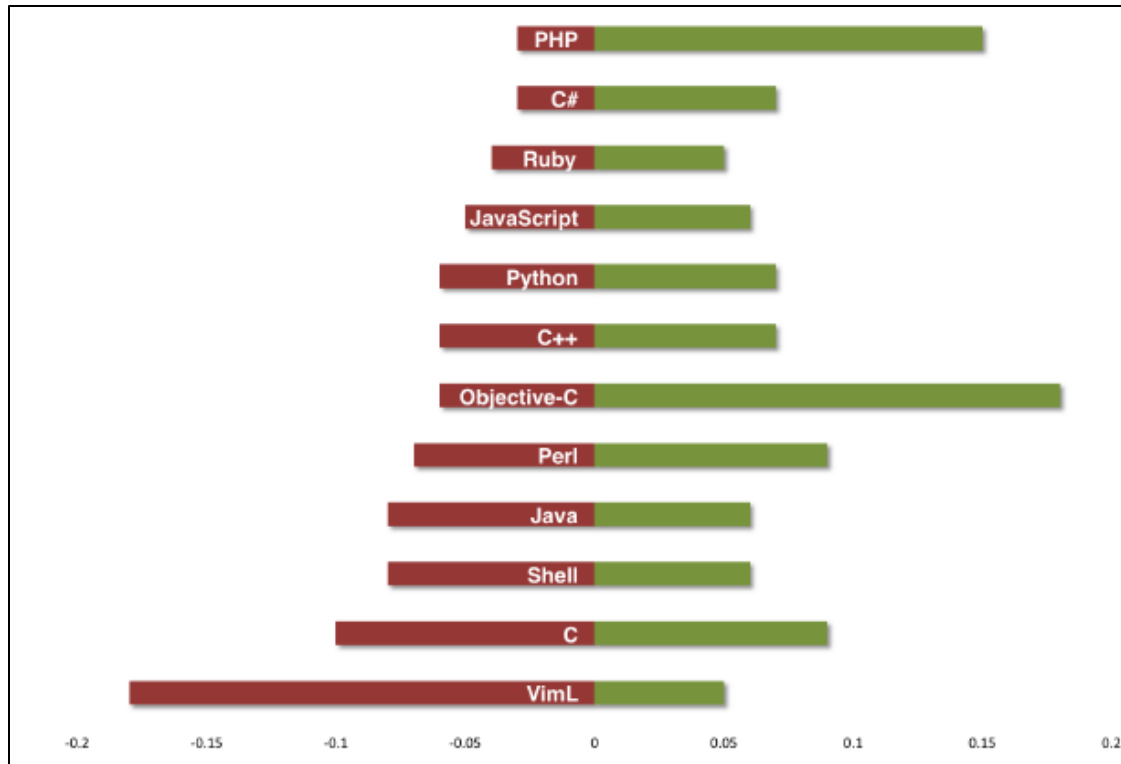
MSR studies - Bugs

Do not program on Friday ;-)



MSR studies – Sentiment Analysis

Anger vs. Joy



How they stack up?

- **PHP**, **Object-C**, and **C#** are net positive
- **Java**, **Shell**, and **C** are fairly even while **VimL** is just bad news.

[Doll and Grigorik, 2012]

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]



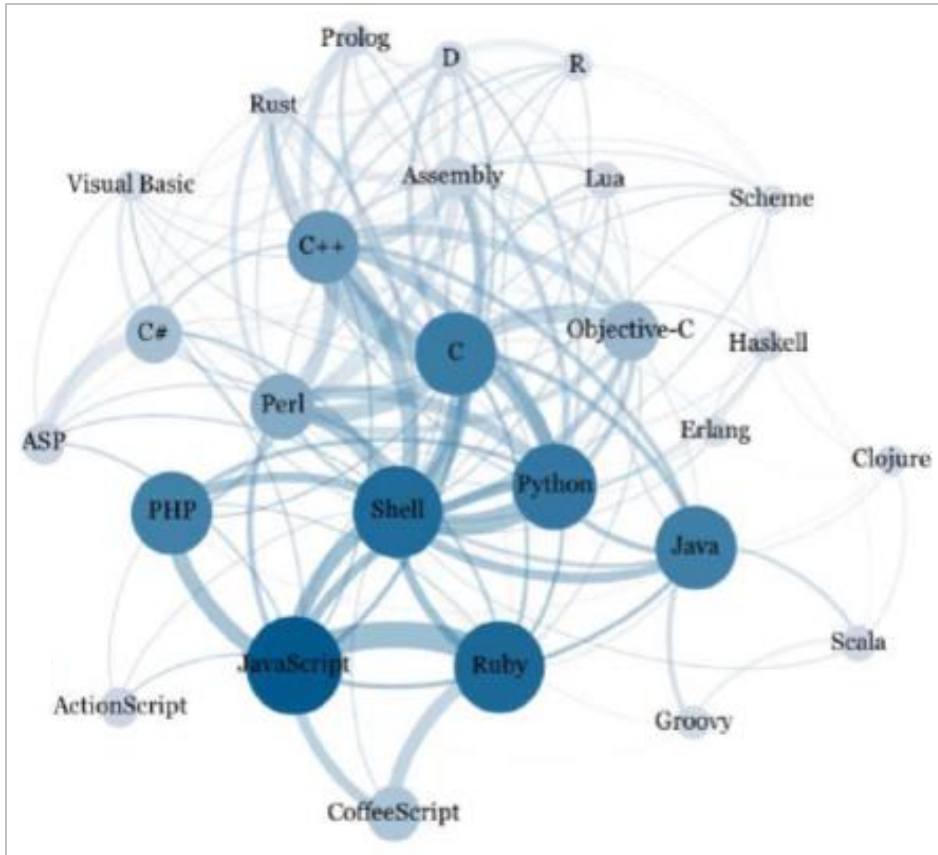
10/23/12 2:29 AM

Security [REDACTED] worked out



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

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

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

Symbol	File	Support	Confidence
initDefaults(IPreferenceStore store)	ComparePreferencePage.java	7	1.0
org.eclipse.compare.plugin.properties	plugin.properties	0	0.875
org.eclipse.compare.subnotes_compare.html	subnotes_compare.html	6	0.75
TextMergeViewer(Composite parent, int style, CompareConfiguration configuration)	TextMergeViewer.java	6	0.75
propertyChanged(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



July 2025:
150 million developers
518 repositories



January 2020:
430K repositories
3.7 Million Users



April 2019
28 Millinon 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

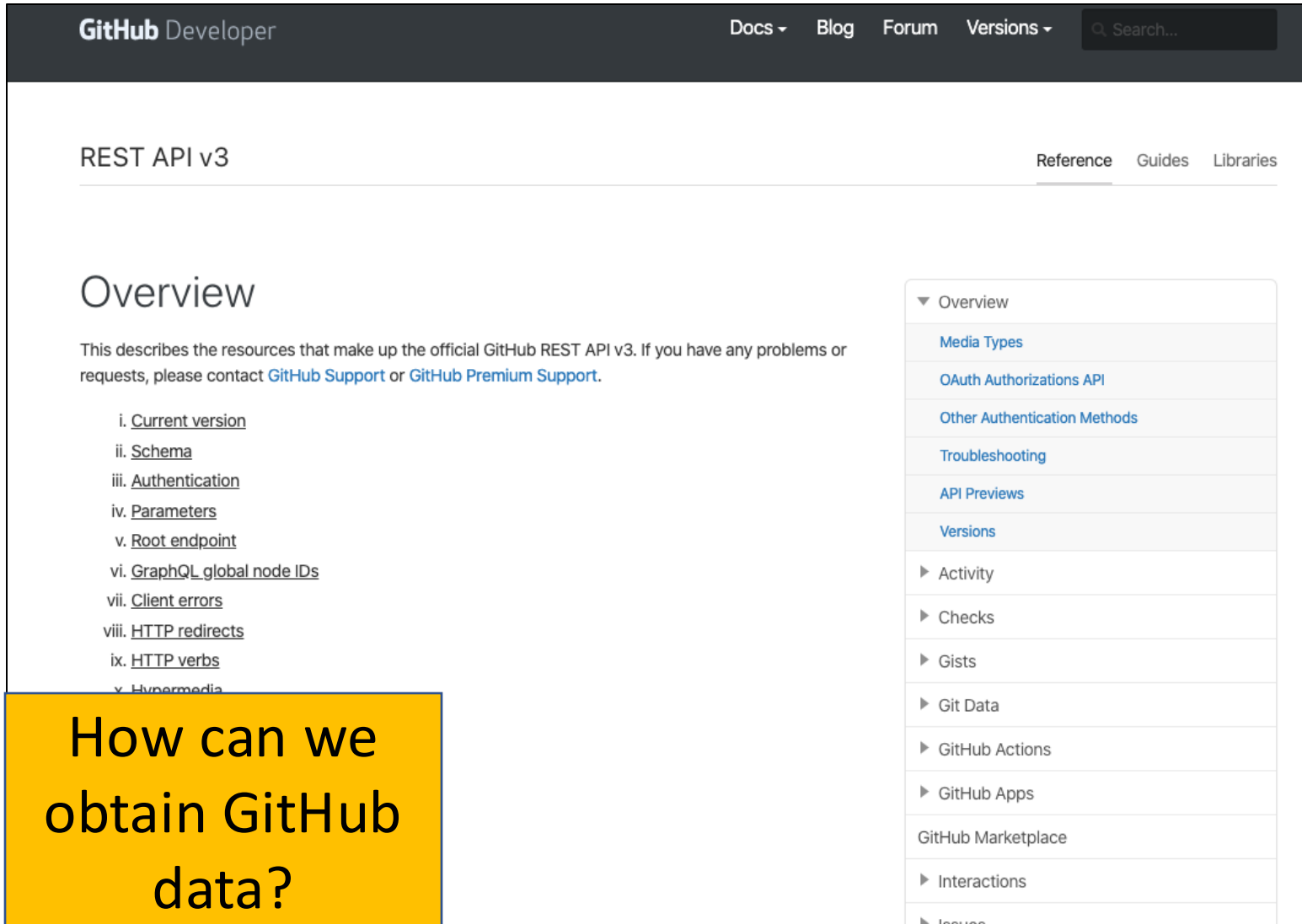
The screenshot shows the GitHub homepage for user 'bramadams'. At the top is a search bar and navigation links: Explore, Gist, Blog, Help. Below the navigation bar are links for News Feed, Pull Requests, and Issues. A 'GitHub Bootcamp' section is prominently displayed, featuring four steps:

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

Repositories you contribute to	
inuksuk/jekyll-scholar	152 ★
smcintosh/moosetracks	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 logo and navigation links: Docs, Blog, Forum, Versions, and a search bar. Below the header, the page title is 'REST API v3' with tabs for Reference, Guides, and Libraries. The main content area is titled 'Overview' and contains a paragraph describing the resources of the GitHub REST API v3. Below this paragraph is 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. [Hypermedia](#). On the right side of the page is a sidebar with a table of contents. The first section is 'Overview' with links to Media Types, OAuth Authorizations API, Other Authentication Methods, Troubleshooting, API Previews, and Versions. Below this are sections for Activity, Checks, Gists, Git Data, GitHub Actions, GitHub Apps, GitHub Marketplace, Interactions, and Issues. 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](#).

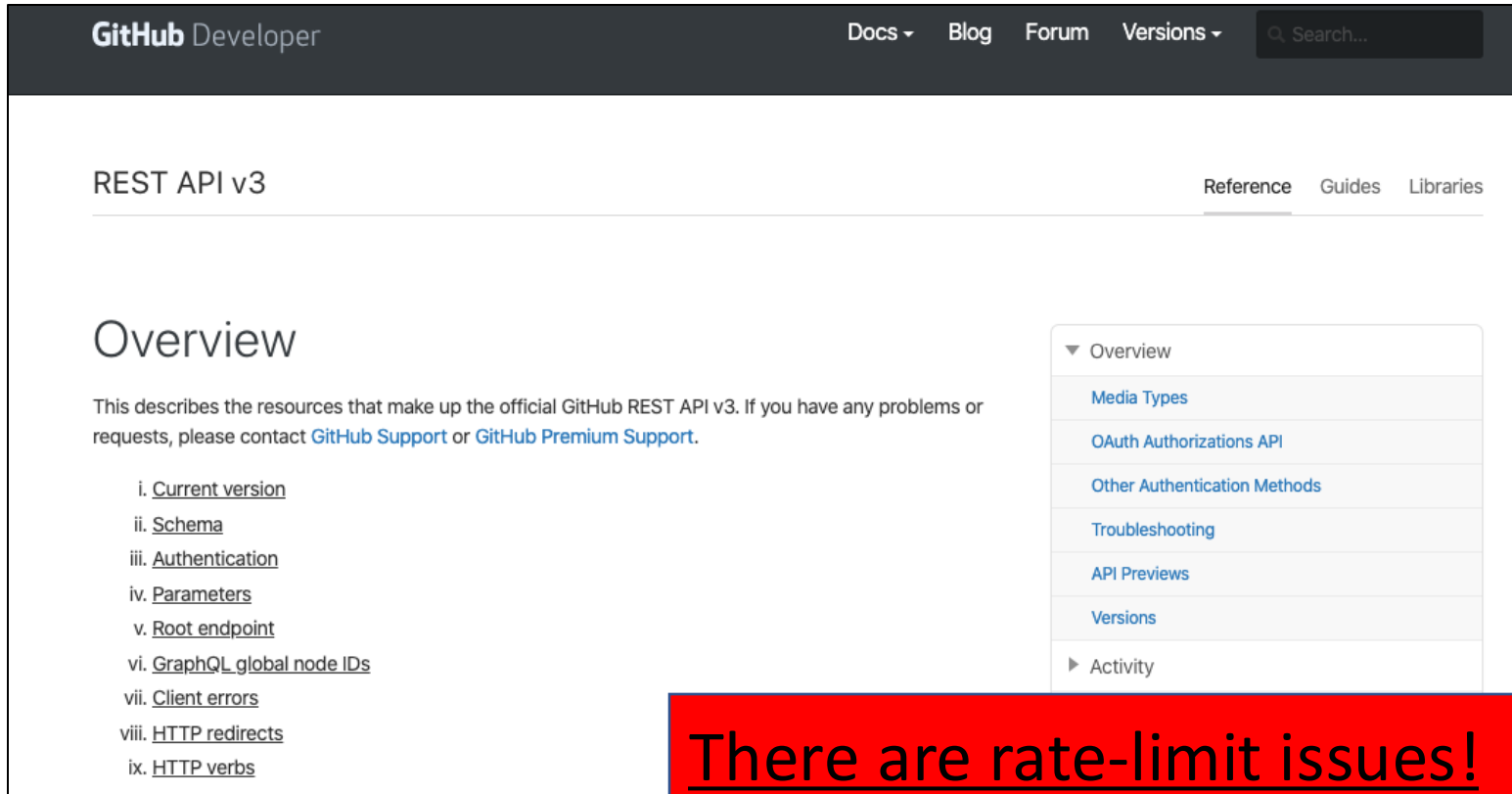
- i. [Current version](#)
- ii. [Schema](#)
- iii. [Authentication](#)
- iv. [Parameters](#)
- v. [Root endpoint](#)
- vi. [GraphQL global node IDs](#)
- vii. [Client errors](#)
- viii. [HTTP redirects](#)
- ix. [HTTP verbs](#)
- x. [Hypermedia](#)

▼ 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](#).

- 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. [Hypermedia](#)

▼ 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

GitHub GraphQL API documentation

To create integrations, retrieve data, and automate your workflows, use the GitHub GraphQL API. The GitHub GraphQL API offers more precise and flexible queries than the GitHub REST API.

Overview

Start here [View all →](#)

Forming calls with GraphQL

Learn how to authenticate to the GraphQL API, then learn how to create and run queries and mutations.

Introduction to GraphQL

Learn useful terminology and concepts for using the GitHub GraphQL API.

Using GraphQL Clients

You can run queries on real GitHub data using various GraphQL clients and libraries.

Popular

Explorer

Public schema

Download the public schema for the GitHub GraphQL API.

Using pagination in the GraphQL API

Learn how to traverse data sets using cursor based pagination with the GraphQL API.

How can we
obtain GitHub
data?

There are rate-limit issues!
Use GraphQL API
Rate limit is less of an issue