# Software Developers' Work Habits and Expertise

## Sketching, Code Plagiarism, and Expertise Development

**Sebastian Baltes**

@s_baltes

Universität Trier

empirical-software.engineering

# Goals of Research

Observe
Describe
Explain
Predict

} Phenomena $\Rightarrow$ Expand knowledge

# Goals of my PhD Research

Observe
Describe
Explain
(Predict)

Software Developers' Work Habits

⇒

Expand knowledge:

- Identify requirements for better tool support
- Point to possible process improvements
- Communicate results back to practitioners

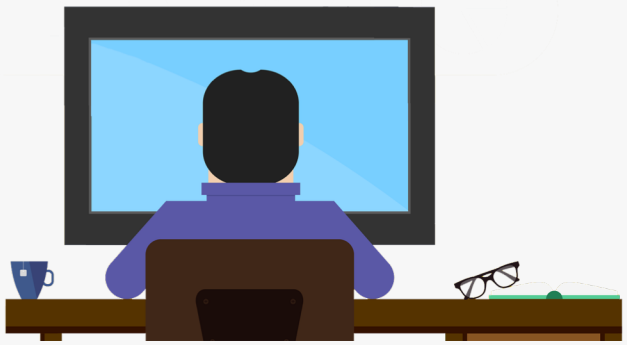
# Research Statement

*"For me, thoroughly analyzing and understanding the **state-of-practice** is an essential first step towards **improving** how software is being developed, because too often, decisions are still rather opinion-based than **data-informed**."*

# Habit?


„A **settled tendency** or **usual manner of behavior**“

https://www.merriam-webster.com/dictionary/habit



https://news.harvard.edu/gazette/story/2018/04/5-healthy-habits-may-increase-life-expectancy-by-decade-or-more/

# Studied Habits

2013

Sketches and Diagrams in Practice

Sketching

SketchLink

How Developers Locate Performance Bugs

LivelySketches

Towards a Theory of Software Development Expertise

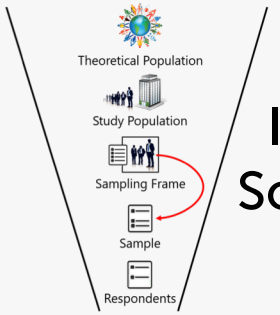
Expertise Development

Code Plagiarism


stackoverflow Code Snippets in GitHub Projects

2018

# “Parallel Thread“




Issues in Sampling Software Developers

2013

Methodology

Open Data

SOTorrent

Interdisciplinary Research

airbnb

Constructing Urban Tourism Space Digitally

2018

# Studied Habits

2013

Sketches and Diagrams in Practice

Sketching

SketchLink

How Developers Locate Performance Bugs

LivelySketches

Towards a Theory of Software Development Expertise

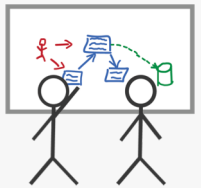
Expertise Development

Code Plagiarism


stackoverflow Code Snippets in GitHub Projects

2018

## Sketching



# Sketches and Diagrams in Practice

FSE 2014

Sebastian Baltes
Computer Science
University of Trier
Trier, Germany
s.baltes@uni-trier.de

Stephan Diehl
Computer Science
University of Trier
Trier, Germany
diehl@uni-trier.de

## ABSTRACT

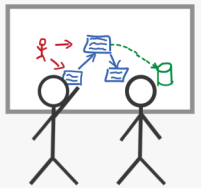
Sketches and diagrams play an important role in the daily work of software developers. In this paper, we investigate the use of sketches and diagrams in software engineering practice. To this end, we used both quantitative and qualitative methods. We present the results of an exploratory study in three companies and an online survey with 394 participants. Our participants included software developers, software architects, project managers, consultants, as well as researchers. They worked in different countries and on projects from a wide range of application areas. Most questions in the survey were related to the last sketch or diagram that the participants had created. Contrary to our expectations and previous work, the majority of sketches and

## 1. INTRODUCTION

Over the past years, studies have shown the importance of sketches and diagrams in software development [6,11,43]. Most of these visual artifacts do not follow formal conventions like the *Unified Modeling Language* (UML), but have an informal, ad-hoc nature [6,11,23,25]. Sketches and diagrams are important because they depict parts of the mental model developers build to understand a software project [21]. They may contain different views, levels of abstraction, formal and informal notations, pictures, or generated parts [6, 11,41,42]. Developers create sketches and diagrams mainly to understand, to design, and to communicate [6]. Media for sketch creation include whiteboards, engineering notebooks, scrap papers, but also software tools like Photoshop

https://empirical-software.engineering/projects/sketches/

Sketching



# Navigate, Understand, Communicate: How Developers Locate Performance Bugs


Sebastian Baltes*, Oliver Moseler*, Fabian Beck†, and Stephan Diehl*
* University of Trier, Germany
† VISUS, University of Stuttgart, Germany

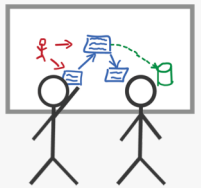
***Abstract*—*Background:*** **Performance bugs can lead to severe issues regarding computation efficiency, power consumption, and user experience. Locating these bugs is a difficult task because developers have to judge for every costly operation whether runtime is consumed necessarily or unnecessarily. *Objective:* We wanted to investigate how developers, when locating performance bugs, navigate through the code, understand the program, and communicate the detected issues. *Method:* We performed a qualitative user study observing twelve developers trying to fix documented performance bugs in two open source projects. The developers worked with a profiling and analysis tool that visually depicts runtime information in a list representation and embedded into the source code view. *Results:* We identified typical navigation strategies developers used for pinpointing the bug, for instance, following method calls based on runtime consumption. The integration of visualization and code helped developers to**

directly because the steps and tools required to optimize a non-functional requirement like performance are substantially different from those applied for fixing a functional bug. These differences include: (i) developers cannot analyze whether a program is correct regarding performance because there only exist better or worse solutions; (ii) developers need to investigate not only program state but also runtime consumption; and (iii) collecting runtime information requires to set up realistic benchmarks that differ from usual regression tests. Also, Jin et al. [1] already pointed at the lack of studies on how performance bugs are fixed by developers.

The user study presented in this paper aims at filling this gap by investigating how developers *navigate* through code, *understand* performance problems, and *communicate*

https://empirical-software.engineering/projects/debugging/

## Sketching



# Linking Sketches and Diagrams to Source Code Artifacts

Sebastian Baltes, Peter Schmitz, and Stephan Diehl
Computer Science
University of Trier
Trier, Germany
{s.baltes,diehl}@uni-trier.de

FSE 2014

**ABSTRACT**

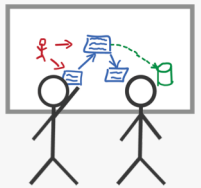
Recent studies have shown that sketches and diagrams play an important role in the daily work of software developers. If these visual artifacts are archived, they are often detached from the source code they document, because there is no adequate tool support to assist developers in capturing, archiving, and retrieving sketches related to certain source code artifacts. This paper presents *SketchLink*, a tool that aims at increasing the value of sketches and diagrams created during software development by supporting developers in these tasks. Our prototype implementation provides a web application that employs the camera of smartphones and tablets to capture analog sketches, but can also be used on desktop

or generated parts [5, 8, 20, 21]. Developers create sketches and diagrams mainly to understand, to design, and to communicate [1, 5]. Media used for sketch creation include not only whiteboards and scrap paper, but also software tools like Photoshop and PowerPoint [5, 10, 17, 22].

Sketches and diagrams are important because they depict parts of the mental model developers build to understand a software project [13]. Understanding source code is one of the most important problems developers face on a daily basis [5, 12, 13, 19]. However, this task is often complicated by documentation that is frequently poorly written and out of date [9, 15]. Sketches and diagrams, whether formal or informal, can fill in this gap and serve as a supplement to conventional documentation like source code comments. To this

https://empirical-software.engineering/projects/sketchlink/

## Sketching



# Round-Trip Sketches: Supporting the Lifecycle of Software Development Sketches from Analog to Digital and Back

Sebastian Baltes, Fabrice Hollerich, and Stephan Diehl
Department of Computer Science
University of Trier
Trier, Germany
Email: research@sbaltes.com, diehl@uni-trier.de

**VISSOFT 2017**

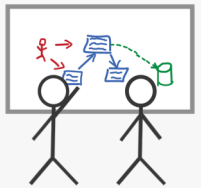
***Abstract*—Sketching is an important activity for understanding, designing, and communicating different aspects of software systems such as their requirements or architecture. Often, sketches start on paper or whiteboards, are revised, and may evolve into a digital version. Users may then print a revised sketch, change it on paper, and digitize it again. Existing tools focus on a paperless workflow, i.e., archiving analog documents, or rely on special hardware—they do not focus on integrating digital versions into the analog-focused workflow that many users**

media [13], because digital sketches can more easily be edited, copied, organized, and shared [18]. Even if a digital version exists, analog sketches may be kept as a memory aid [19]. Context information is often needed to understand informal sketches [20] and information may get lost due to the transient nature of sketches [12], [14].

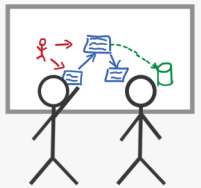
Despite the widespread usage of sketches in many domains, to the best of our knowledge there is currently no tool that

https://empirical-software.engineering/projects/livelysketches/

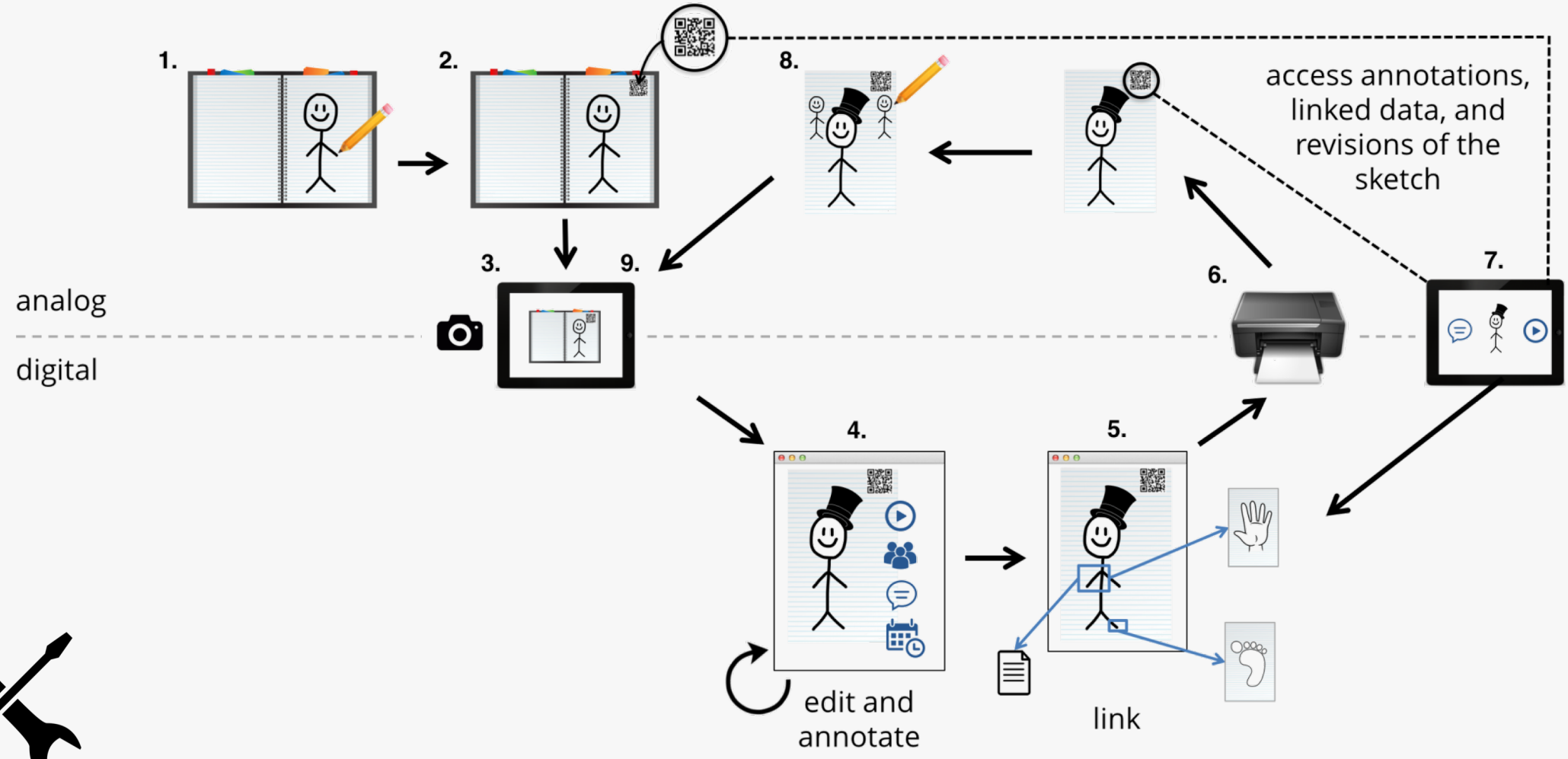
## Sketching




https://www.youtube.com/watch?v=mG6xCiQpS80



# LivelySketches




# Studied Habits

2013

Sketches and Diagrams in Practice

SketchLink

Sketching

How Developers Locate Performance Bugs

LivelySketches

Towards a Theory of Software Development Expertise

Expertise Development

Code Plagiarism


stackoverflow Code Snippets in GitHub Projects

2018

# Expertise Development

## Towards a Theory of Software Development Expertise

Sebastian Baltes
University of Trier
Trier, Germany
research@sbaltes.com

ESEC/FSE 2018

Stephan Diehl
University of Trier
Trier, Germany
diehl@uni-trier.de

**ABSTRACT**

Software development includes diverse tasks such as implementing new features, analyzing requirements, and fixing bugs. Being an expert in those tasks requires a certain set of skills, knowledge, and experience. Several studies investigated individual aspects of software development expertise, but what is missing is a comprehensive theory. We present a first conceptual theory of software development expertise that is grounded in data from a mixed-method survey with 335 software developers and in literature on expertise and expert performance. Our theory currently focuses on programming, but already provides valuable insights for researchers, developers, and employers. The theory describes important properties of software development expertise and which factors foster or hinder its formation, including how developers' performance may decline over time. Moreover, our quantitative results show that developers' expertise self-assessments are context-dependent and that experience is not necessarily related to expertise.

Tomorrow

expert performance [78]. Bergersen et al. proposed an instrument to measure programming skill [9], but their approach may suffer from learning effects because it is based on a fixed set of programming tasks. Furthermore, aside from programming, software development involves many other tasks such as requirements engineering, debugging [62, 96, 100], in which a software developer is expected to be good at.

Researchers investigated certain aspects of software development expertise (SDExp) such as the influence of programming experience [95], desired attributes of software engineers [63], or the time it takes for developers to become "fluent" in software projects [117]. However, there is currently no theory combining those individual aspects. Such a theory could help structuring existing knowledge about SDExp in a concise and precise way and hence facilitate its communication [44]. Despite many arguments in favor of developing and using theories [46, 56, 85, 109], theory-driven research is not very common in software engineering [97].

https://empirical-software.engineering/projects/expertise/

# Studied Habits

2013

Sketches and Diagrams in Practice

Sketching

SketchLink

How Developers Locate Performance Bugs

LivelySketches

Towards a Theory of Software Development Expertise

Expertise Development

2018

Code Plagiarism


stackoverflow Code Snippets in GitHub Projects

# Code Plagiarism



Empirical Software Engineering
https://doi.org/10.1007/s10664-018-9650-5

CrossMark

## Usage and attribution of Stack Overflow code snippets in GitHub projects

Sebastian Baltes[1] · Stephan Diehl[1]

Published online: 01 October 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

**Abstract**

Stack Overflow (SO) is the most popular question-and-answer website for software developers, providing a large amount of copyable code snippets. Using those snippets raises maintenance and legal issues. SO's license (CC BY-SA 3.0) requires attribution, i.e., referencing the original question or answer, and requires derived work to adopt a compatible license. While there is a heated debate on SO's license model for code snippets and the

https://empirical-software.engineering/projects/snippets/

# Usage and Attribution of


stackoverflow Code Snippets

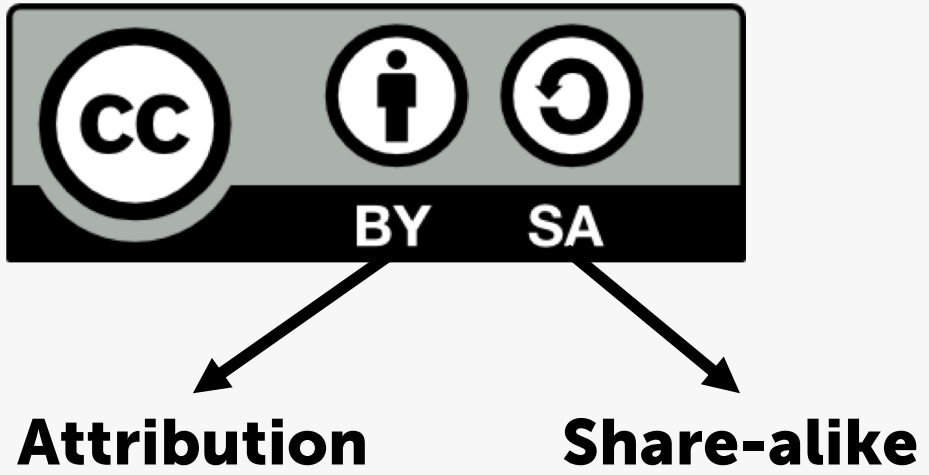
in GitHub Projects

# Question 1

Who admits regularly copying non-trivial code snippets from Stack Overflow?

# Question 2

Who knew that all content on Stack Overflow is licensed under CC BY-SA?



# Background


*"Well, but these snippets are rather trivial and not protected by copyright."*

- **Not all** code snippets on Stack Overflow are copyrightable
- *"If two programmers would provide substantially the same piece of code, the code is not **creative** under copyright law"* [Engelfriet 2016]
- *"A snippet that is **more than one or two lines** of standard function calls would typically be creative enough for copyright"* [Engelfriet 2016]
- No *"international **standard** for originality"* [Creative Commons 2017b]

http://theconversation.com/why-universities-cant-be-expected-to-police-copyright-infringement-82677

889

Here's what I do:

1. First of all I check what providers are enabled. Some may be disabled on the device, some may be disabled in application manifest.
2. If any provider is available I start location listeners and timeout timer. It's 20 seconds in my example, may not be enough for GPS so you can enlarge it.
3. If I get update from location listener I use the provided value. I stop listeners and timer.
4. If I don't get any updates and timer elapses I have to use last known values.
5. I grab last known values from available providers and choose the most recent of them.

Here's how I use my class:

```
LocationResult locationResult = new LocationResult(){
    @Override
    public void gotLocation(Location location){
        //Got the location!
    }
};
MyLocation myLocation = new MyLocation();
myLocation.getLocation(this, locationResult);
```

And here's MyLocation class:

```
import java.util.Timer;
import java.util.TimerTask;
import android.content.Context;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;

public class MyLocation {
    Timer timer1;
    LocationManager lm;
    LocationResult locationResult;
    boolean gps_enabled=false;
    boolean network_enabled=false;

    public boolean getLocation(Context context, LocationResult result)
    {
        //I use LocationResult callback class to pass location value from MyLocat
        locationResult=result;
        if(lm==null)
            lm = (LocationManager) context.getSystemService(Context.LOCATION_SERV

        //exceptions will be thrown if provider is not permitted.
        try{gps_enabled=lm.isProviderEnabled(LocationManager.GPS_PROVIDER);}catch
        try{network_enabled=lm.isProviderEnabled(LocationManager.NETWORK_PROVIDER

        //don't start listeners if no provider is enabled
        if(!gps_enabled && !network_enabled)
            return false;

        if(gps_enabled)
            lm.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, locatio
        if(network_enabled)
            lm.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, loc
```

Somebody may also want to modify my logic. For example if you get update from Network provider don't stop listeners but continue waiting. GPS gives more accurate data so it's worth waiting for it. If timer elapses and you've got update from Network but not from GPS then you can use value provided from Network.

One more approach is to use LocationClient http://developer.android.com/training/location/retrieve-current.html. But it requires Google Play Services apk to be installed on user device.

share improve this answer

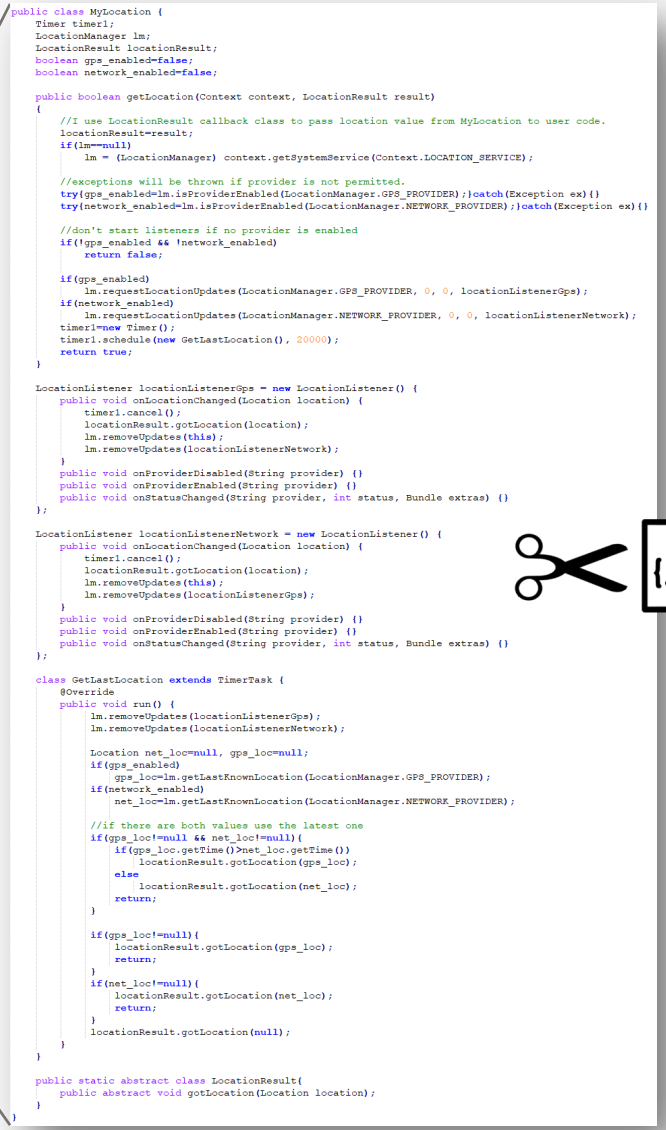
edited Jun 25 '13 at 9:33

answered Jun 30 '10 at 0:07

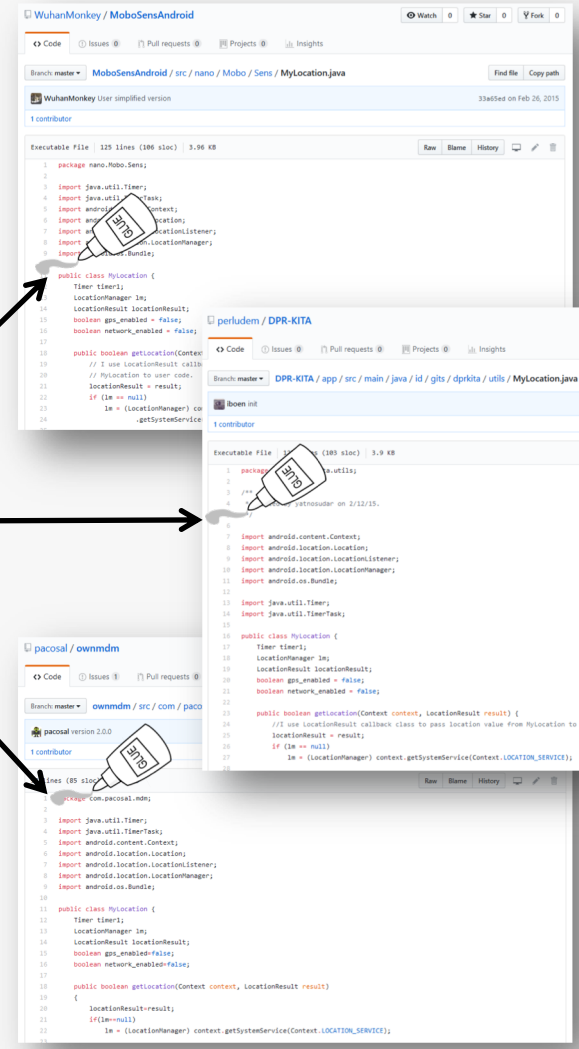
Fedor
40k ● 9 ● 71 ● 86

stackoverflow

GitHub







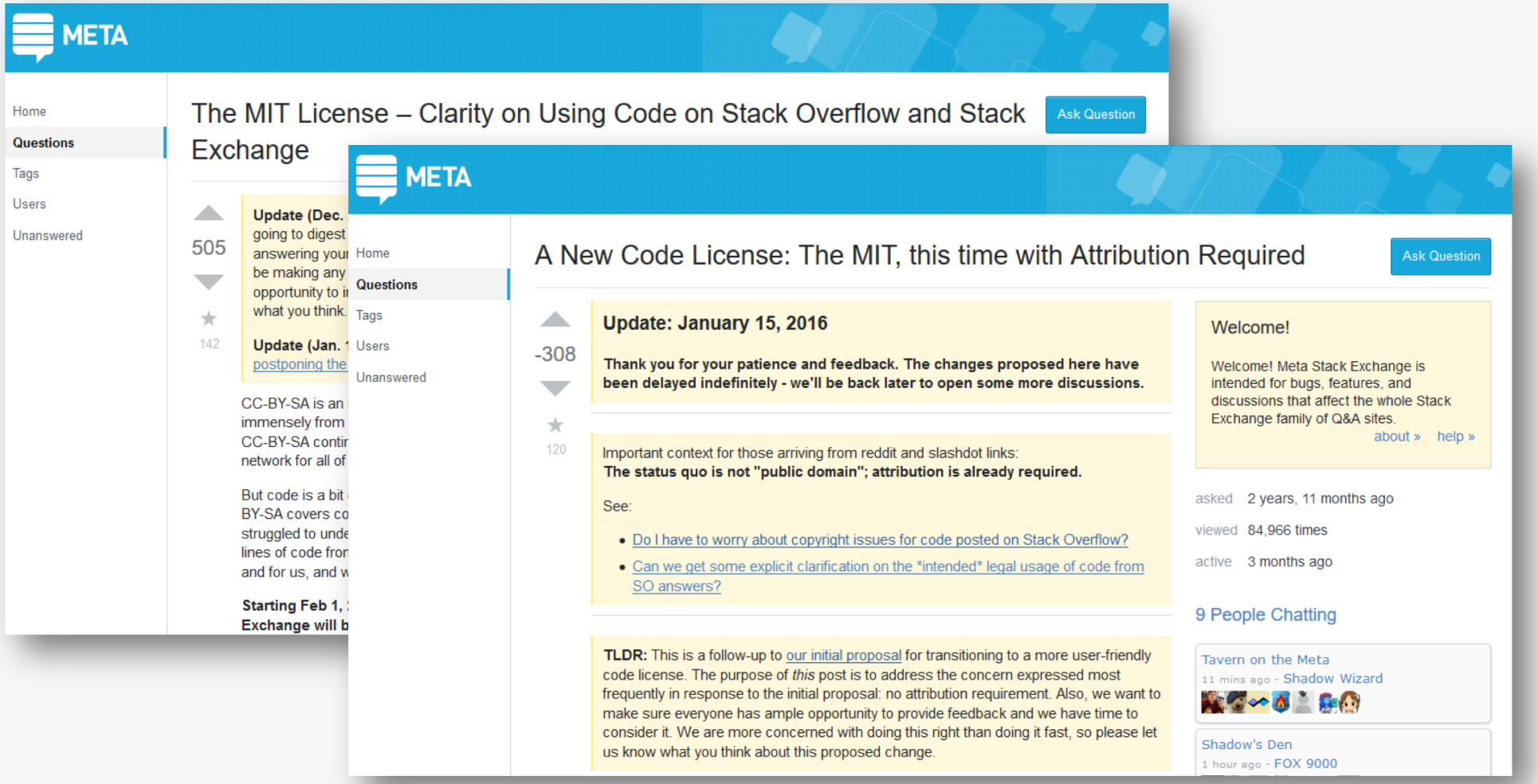
https://stackoverflow.com/a/3145655



https://github.com/pacosal/ownmdm/blob/master/src/com/pacosal/mdm/MyLocation.java

# *"Do Stack Overflow authors care about attribution?"*

https://meta.stackexchange.com/q/273168

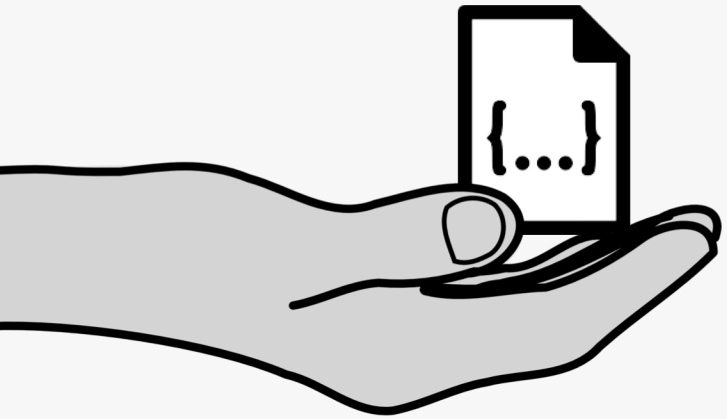



https://meta.stackexchange.com/q/272956

# Implications of Stack Overflow's License

**Permissive Licenses**

- Permit using the licensed source code in proprietary software **without publishing changes** or the derived work
- *Examples:* MIT, Apache, and BSD license families



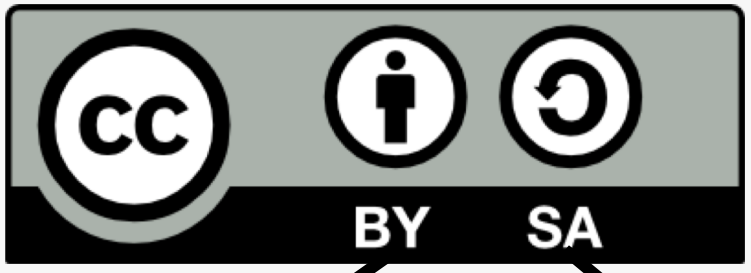

**Copyleft Licenses**

- Requires either modifications to the licensed content or the complete derived work to be **published under the same or a compatible license** (share-alike)
- *Examples (weak copyleft):* Mozilla/Eclipse Public Licenses
- *Examples (viral copyleft):* GNU General Public Licenses, Creative Commons Share-Alike Licenses (e.g., **CC BY-SA**)

# Implications of Stack Overflow's License

*"You must give **appropriate credit**, provide a link to the license, and indicate if changes were made."*

*If you remix, transform, or build upon the material, you must **distribute your contributions** under the same license as the original.*

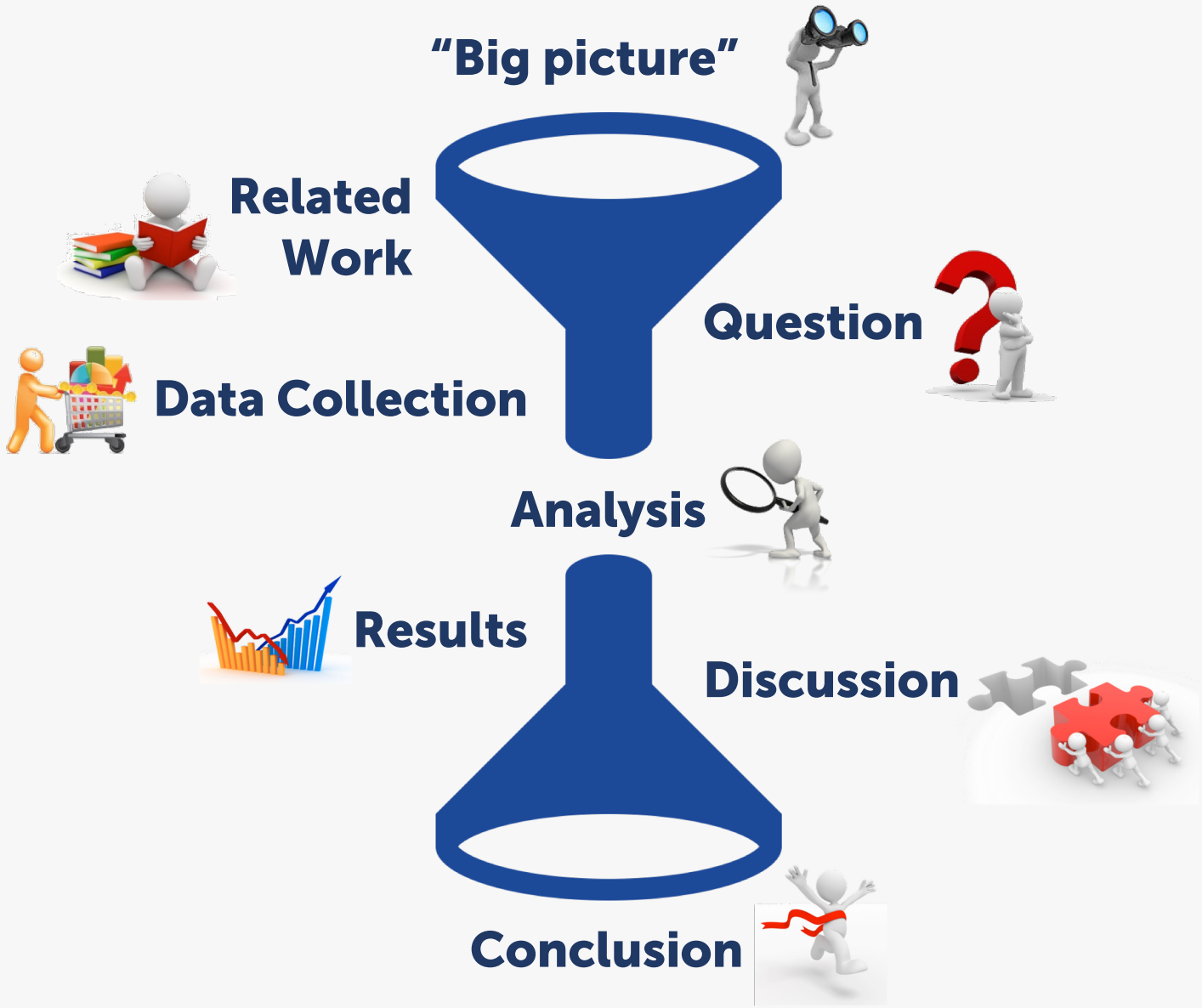



**Attribution** **Share-alike**

# Implications of Stack Overflow's License

- Courts in the US and Europe ruled that open source licenses are **enforceable contracts**
- Developers are able to **sue** when terms like the share-alike requirement are violated:
  - **Interdict distribution** of derived work
  - **Claim monetary damages**
- USA: DMCA takedown notices for allegedly infringed copyright
  - See, e.g., https://github.com/github/dmca
- Risk in mergers and acquisitions of companies
  - See, e.g., FSF vs. Cisco lawsuit

# General Scientific Workflow



# Our Research Questions

**RQ1:** How often is code from Stack Overflow posts used in public GitHub projects without the required attribution?

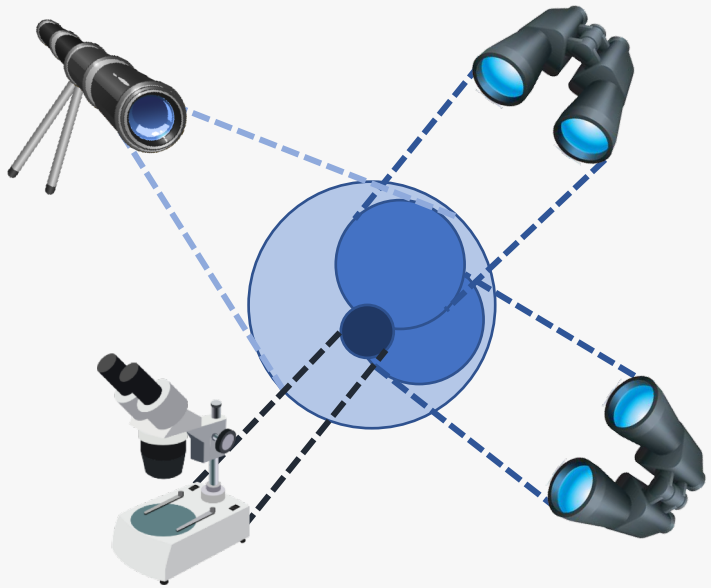
**RQ2:** How often does the license of repositories containing code copied from Stack Overflow conflict with Stack Overflow's license?

**RQ3:** Do developers adhere to the attribution requirements defined in the Stack Overflow terms of service?

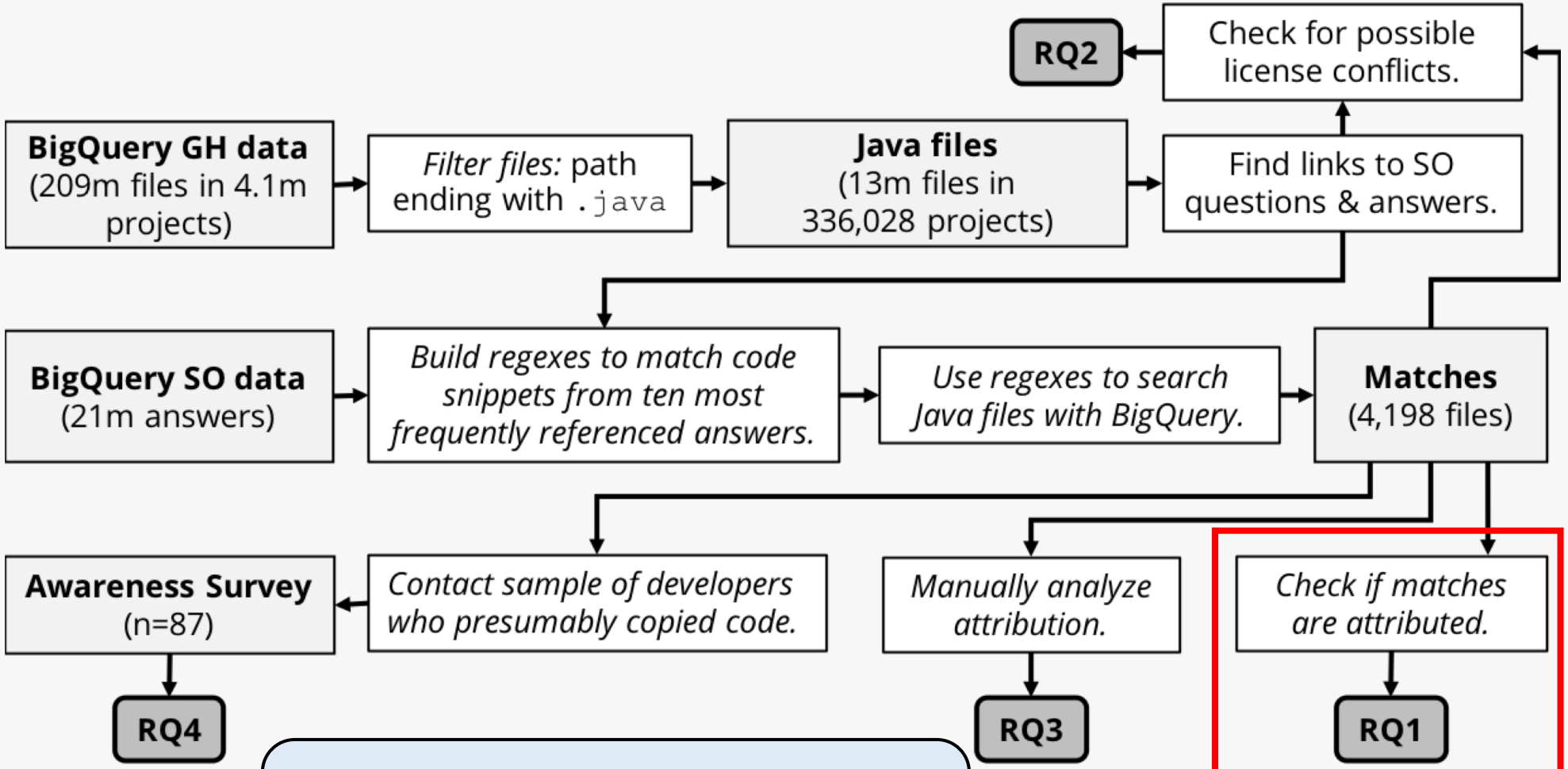
**RQ4:** Are software developers aware of the licensing of Stack Overflow code snippets and its implications?

# RQ1: Triangulation

- Term “triangulation” is an analogy to land surveying
- Increase validity of research by studying a phenomenon from several points of view
- Cross-validation from two or more sources:
  - Different data sources
  - Different aspects of the same phenomenon
  - Different research instruments
  - Different researchers
- **Here:** Use three different approaches to estimate the attribution ratio of snippets copied from Stack Overflow into GitHub projects.



# Phase 1: Research Design




*10 Java snippets, all Java files on GitHub**

* All Java files in the Google BigQuery GitHub dataset

# Phase 1: Exemplary Regex

```java
public static String humanReadableByteCount(long bytes, boolean si) {
    int unit = si ? 1000 : 1024;
    if (bytes < unit) return bytes + " B";
    int exp = (int) (Math.log(bytes) / Math.log(unit));
    String pre = (si ? "kMGTPE" : "KMGTPE").charAt(exp-1) + (si ? "" : "i");
    return String.format("%.1f %sB", bytes / Math.pow(unit, exp), pre);
}
```

```
((?i:String[\s]+\w+\([^\{]*long[^\{]+\)[\s]*\{[\s\S]+if[\s]*\([^<]+<[^\)]+\)
[\s\S]*return[^;]+\+[^;]*\"\ B\"[\s\S]+int[\s][^\=]+\=[\s]*\([\s]*int[\s]*\)
[\s]*\([\s]*Math[\s]*\.[\s]*log[\s]*\([^\)]+\)[\s]*\/[\s]*Math[\s]*\.[\s]*lo
g[\s]*\([^\)]+\)[\s]*\)[\s\S]+return[^\}]+String[\s]*\.[\s]*format[\s]*\([^\
}]+\}))
```

https://stackoverflow.com/a/3758880

# Phase 1: Recall

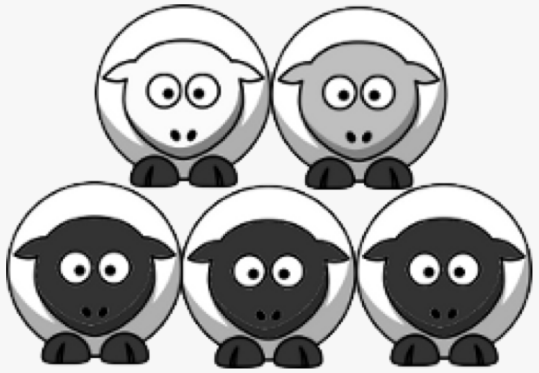
**Table 3** RQ 1 – Phase 1: Ten most frequently referenced code snippets from SO Java answers; estimated ratio of unattributed usages detected using regular expressions; number of matched files (ALL), distinct matches (DISTINCT), distinct matches with reference to SO (REF), distinct matches without reference to SO (NO-REF)

| Rank | Matches | | | | Recall | Attribution | |
|---|---|---|---|---|---|---|---|
| | ALL | DISTINCT | REF | NO-REF | REF/$F_{AQ}$ | REF/DISTINCT | $F_{AQ}$/DIST. |
| 1 | 997 | 448 | 97 | 351 | 79.5% | 21.7% | 27.2% |
| 2 | 1,843 | 913 | 60 | 853 | 60.0% | 6.6% | 11.0% |
| 3 | 2,662 | 902 | 87 | 815 | 80.6% | 9.6% | 12.0% |
| 4 | 420 | 170 | 18 | 152 | 94.7% | 10.6% | 11.2% |
| 5 | 1,492 | 402 | 25 | 377 | 73.5% | 6.2% | 8.5% |
| 6 | 2,642 | 807 | 65 | 742 | 87.8% | 8.1% | 9.2% |
| 7 | 160 | 124 | 12 | 112 | 29.3% | 9.7% | 33.1% |
| 8 | 355 | 174 | 22 | 152 | 61.1% | 12.6% | 20.7% |
| 9 | 295 | 225 | 5 | 220 | 10.6% | 2.2% | 20.9% |
| 10 | 65 | 33 | 11 | 22 | 42.3% | 33.3% | 78.8% |
| All | 10,931 | 4,198 | 402 | 3,796 | *M* 61.9% | *M* 12.1% | *M* 23.2% |

# Phase 2: Research Design

- **Goal:** Search for clones of a sample of Stack Overflow snippets in a sample of GitHub projects using a more scalable approach
- Why samples?
  - Code clone detection is computationally expensive
- Which snippets and projects to select?
  - Random samples: Many “toy” projects on GitHub and many irrelevant snippets on Stack Overflow
  - Sampling based on distribution of certain properties

*Sample of Java snippets, sample of Java projects*



# Background: Sampling

Theoretical Population

*Who do you want to generalize to?*

Study Population

*What populations can you get access to?*

Sampling Frame

*How can you get access to them?*

Random Selection

Sample

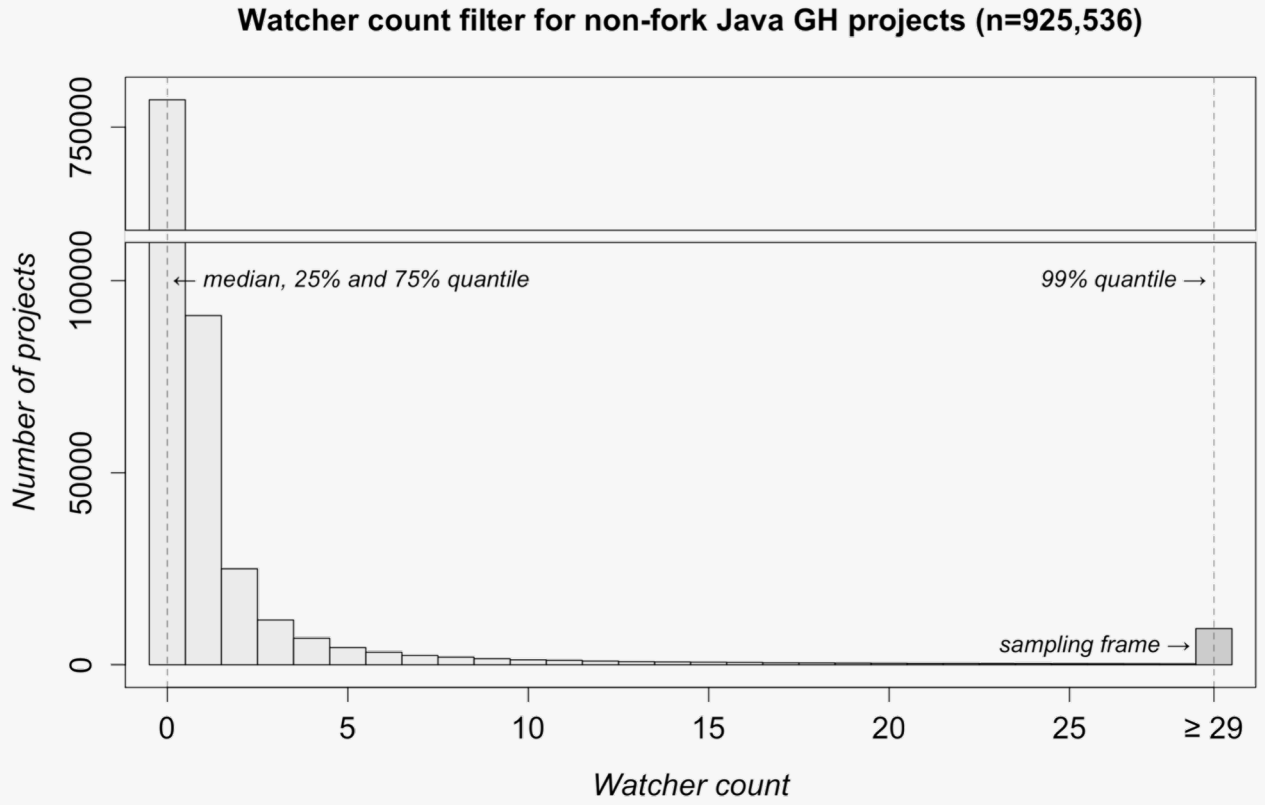
*Who is in your study?*

Respondents

*Who participated?*

# Phase 2: GitHub Project Sample

- Focus on popular GitHub projects
- High precision in selecting “engineered” software projects [Munaiah et al. 2017]
- Greater (potential) impact of licensing issues




Sample size:
3,000 / 2,313

GitHub

# Phase 2: Stack Overflow Snippet Samples

- Snippets from 100 most frequently referenced Stack Overflow answers (phase 1) $\Rightarrow S_{\text{top100}}$
- Snippets from answers referenced in GitHub projects $\Rightarrow S_{\text{gh}}$

**Definition 1** Let $C$ (copies) be a relation over a set of code snippets $S$ and a set of source code files $F$:

$$C \subseteq S \times F$$

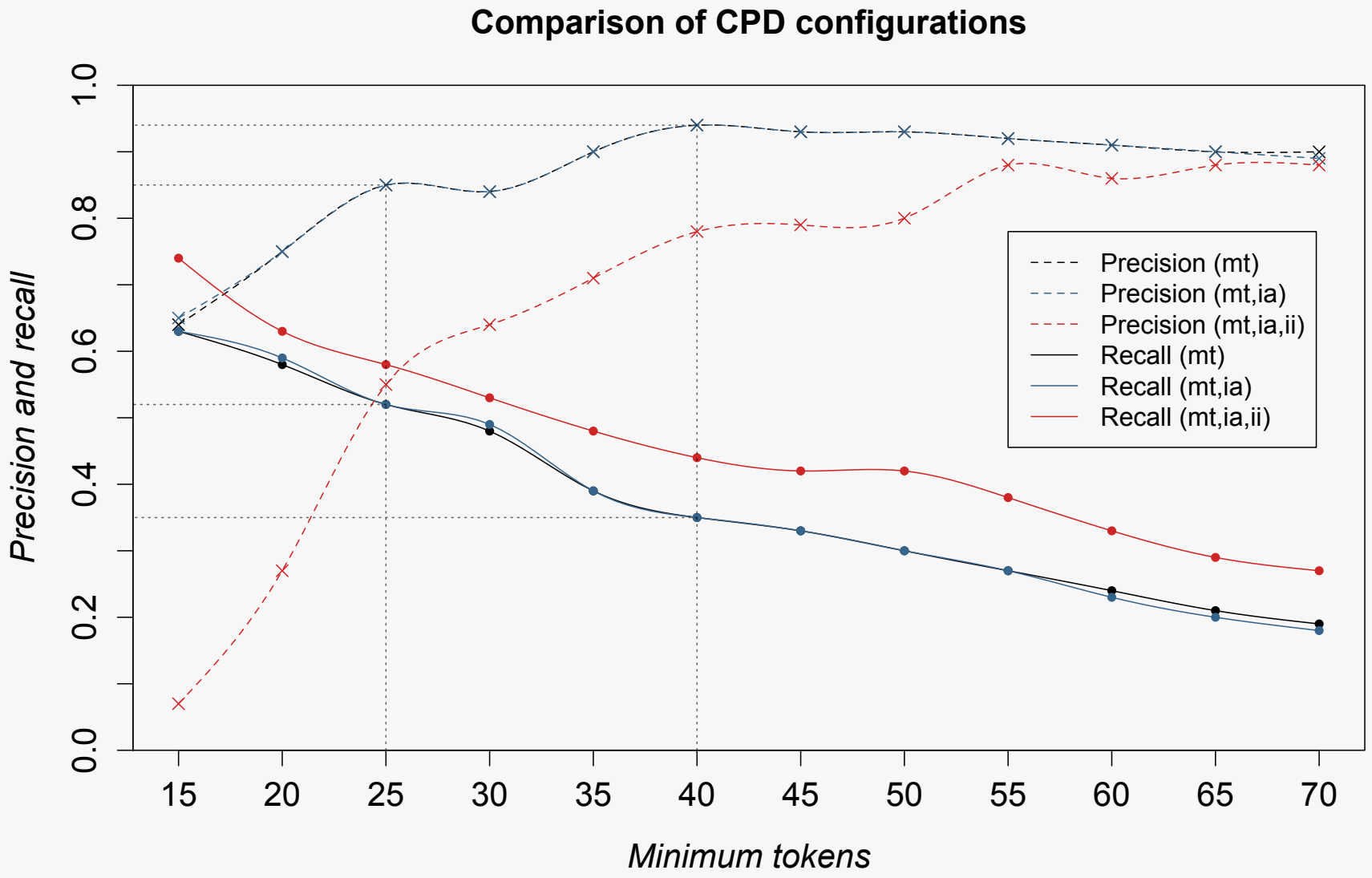
Let $C_{\text{so}} \subseteq C$ be the set of copies identified by an SO answer URL in the source code file and $C_{\text{cpd}} \subseteq C$ be the set of copies identified by CPD. Then we define precision and recall as follows:

$$\text{precision} = \frac{|C_{\text{so}} \cap C_{\text{cpd}}|}{|C_{\text{cpd}}|} \qquad \text{recall} = \frac{|C_{\text{so}} \cap C_{\text{cpd}}|}{|C_{\text{so}}|}$$

Sample size:
111 / 137

# Phase 2: Code Clone Detector Calibration


https://pmd.github.io/

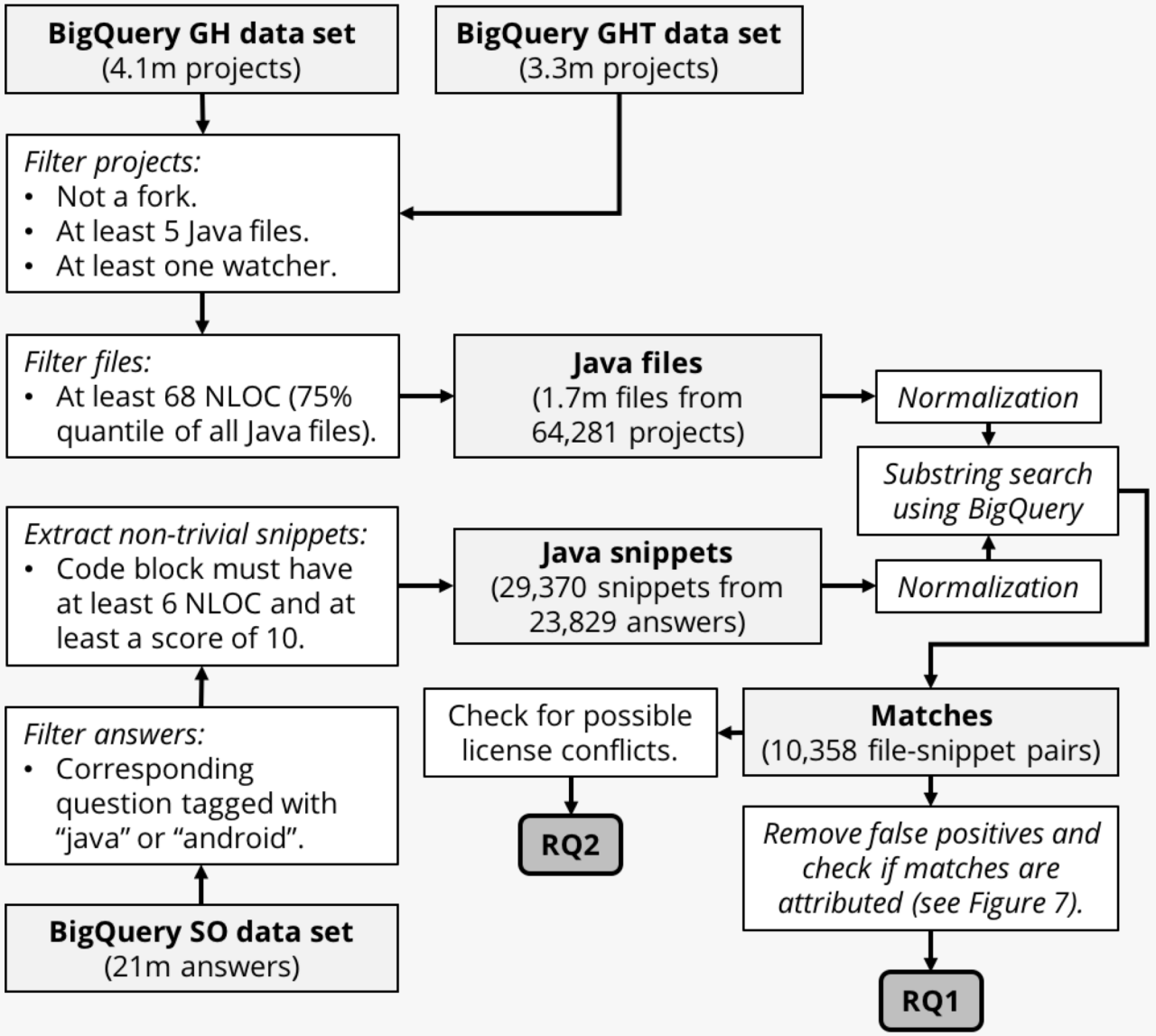


# Phase 3: Research Design

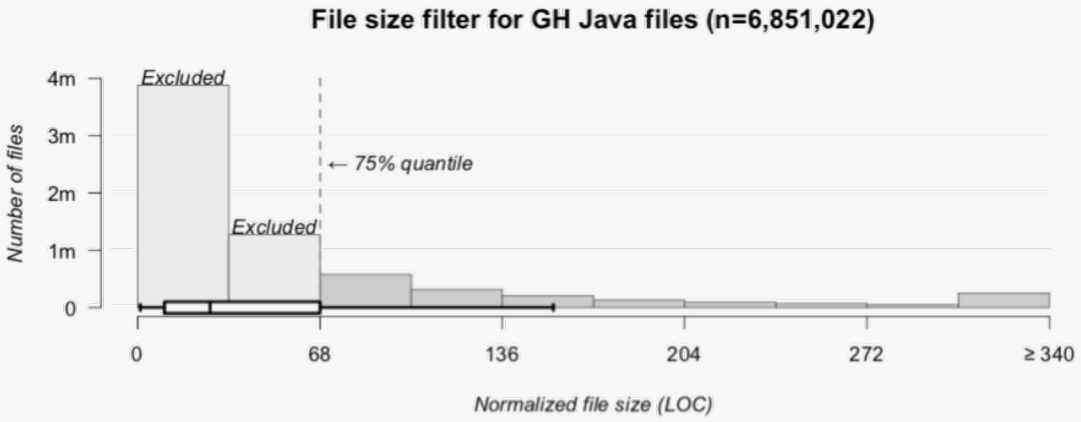
- **Goal:** Address shortcomings of phases 1 and 2
  - External sources
  - Small sample sizes
  - Some rather short snippets
- Select as many projects and snippets as possible and search only for (almost) exact matches

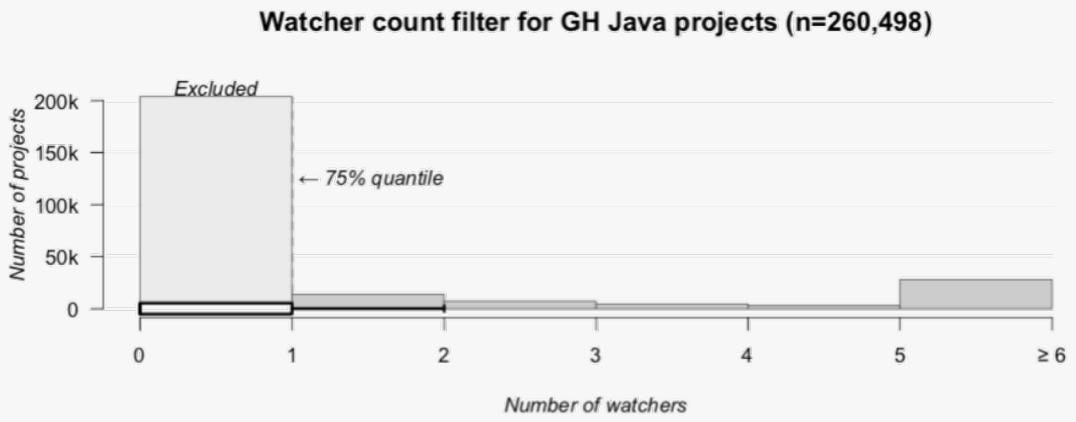
*Many Java snippets, many Java projects*

# Phase 3: Research Design

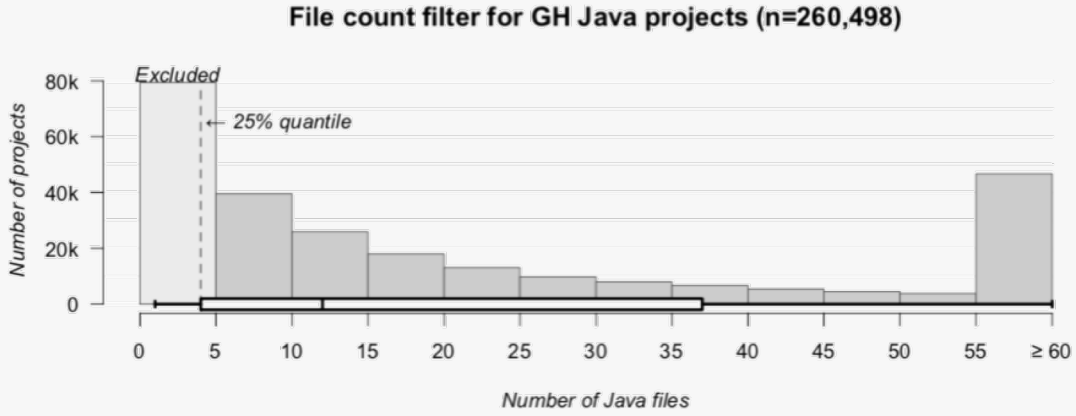


# Phase 3: Filtering GitHub Projects

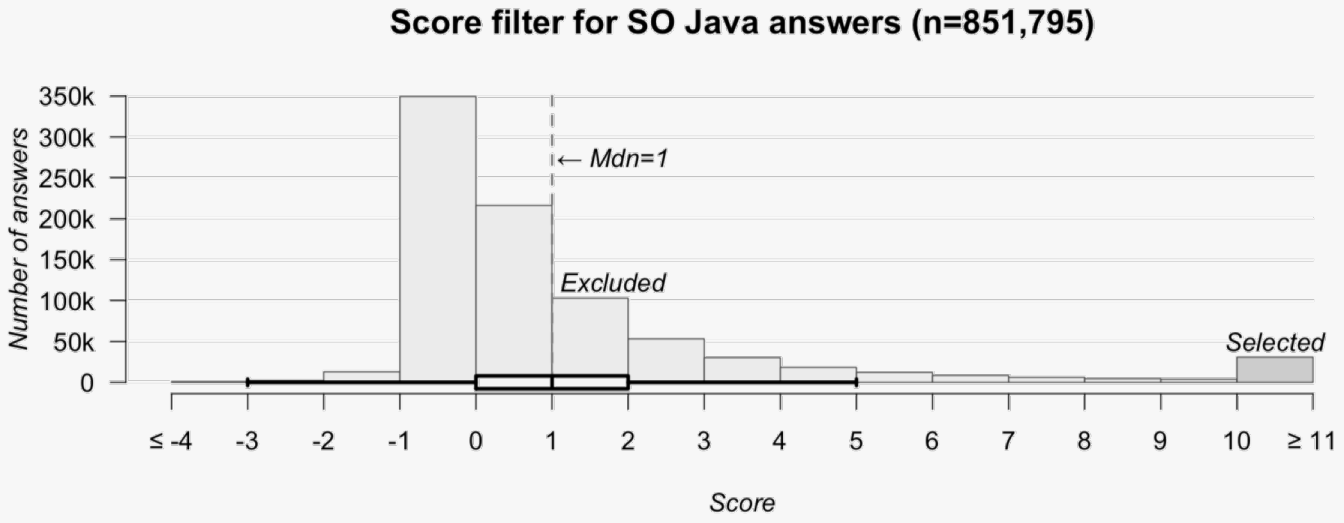


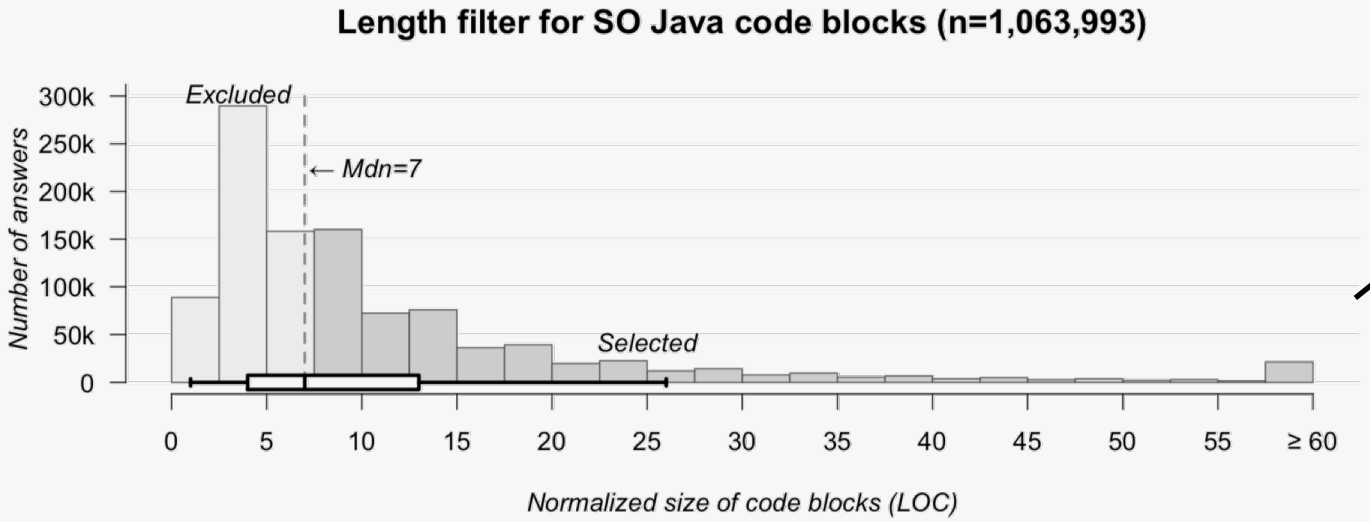







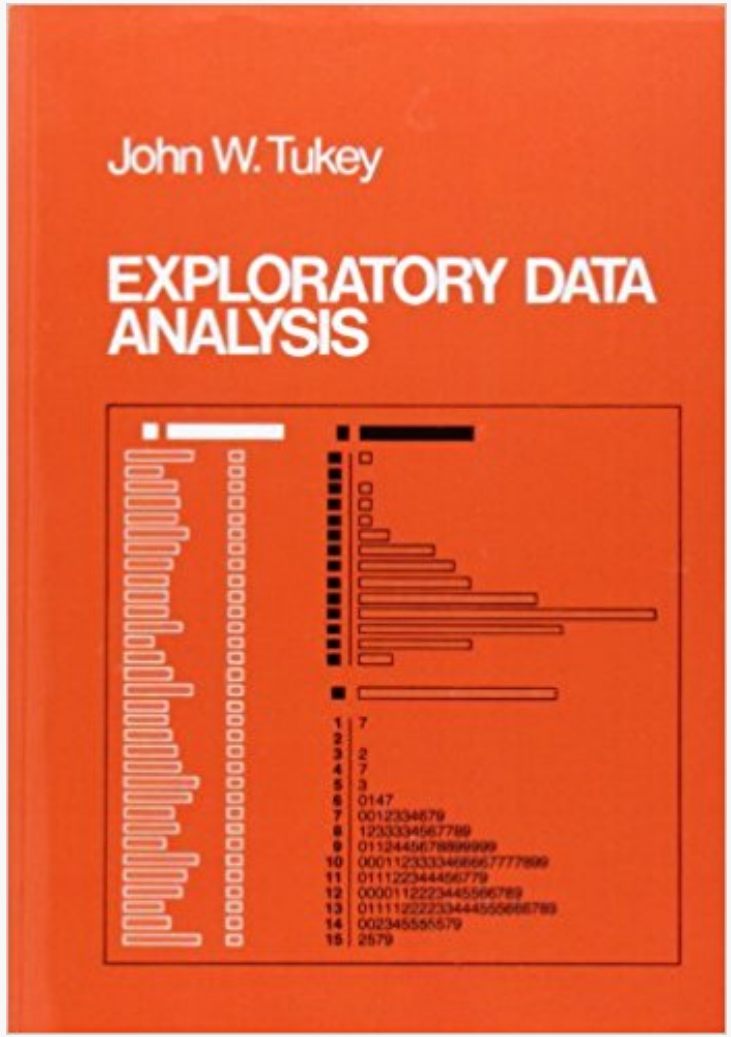
# Phase 3: Filtering Stack Overflow Snippets



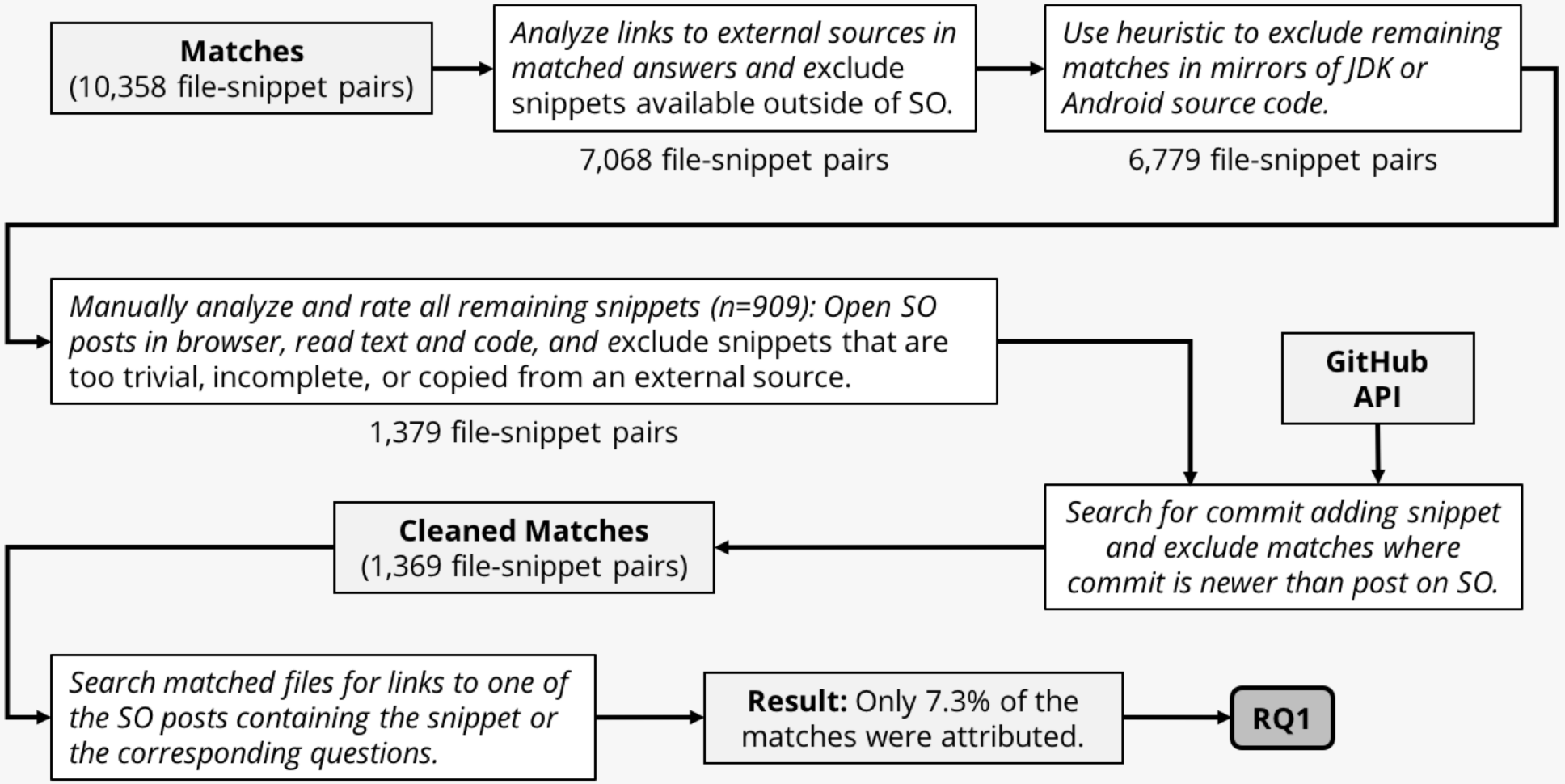




**Proxies for originality**

# Background: Exploratory Data Analysis



# Phase 3: Snippets with External Source



# RQ1: Results

**Table 8** Summary of results from phases 1 to 3: Distinct references to answers (A) or questions (Q) on Stack Overflow (SO) in the Java files from GitHub analyzed in each phase; number of analyzed files and repositories, files/repos containing a reference to SO, files/repos containing a copy of a SO snippet, attributed copies of SO snippets

| Ph. | References | | Files | | | | Repositories | | |
|---|---|---|---|---|---|---|---|---|---|
| | A | Q | Count | Ref | Copy | Attr | Count | Ref | Copy |
| 1 | 5,014 | 16,298 | 13.3m | 18,605 | 4,198 | 402 | 336k | 11,086 | 3,291 |
| | 23.5% | 76.5% | | 0.09% | 0.03% | **9.6%** | | 3.3% | 1.0% |
| 2 | 209 | 463 | 445k | 634 | 297 | 70 | 2,313 | 274 | 199 |
| | 31.1% | 68.9% | | 0.14% | 0.07% | **23.6%** | | 11.9% | 8.6% |
| 3 | 1,551 | 4,843 | 1.7m | 5,354 | 1,369 | 104 | 64,281 | 3,536 | 1,332 |
| | 24.3% | 75.7% | | 0.31% | 0.08% | **7.6%** | | 5.5% | 2.1% |

# Our Research Questions

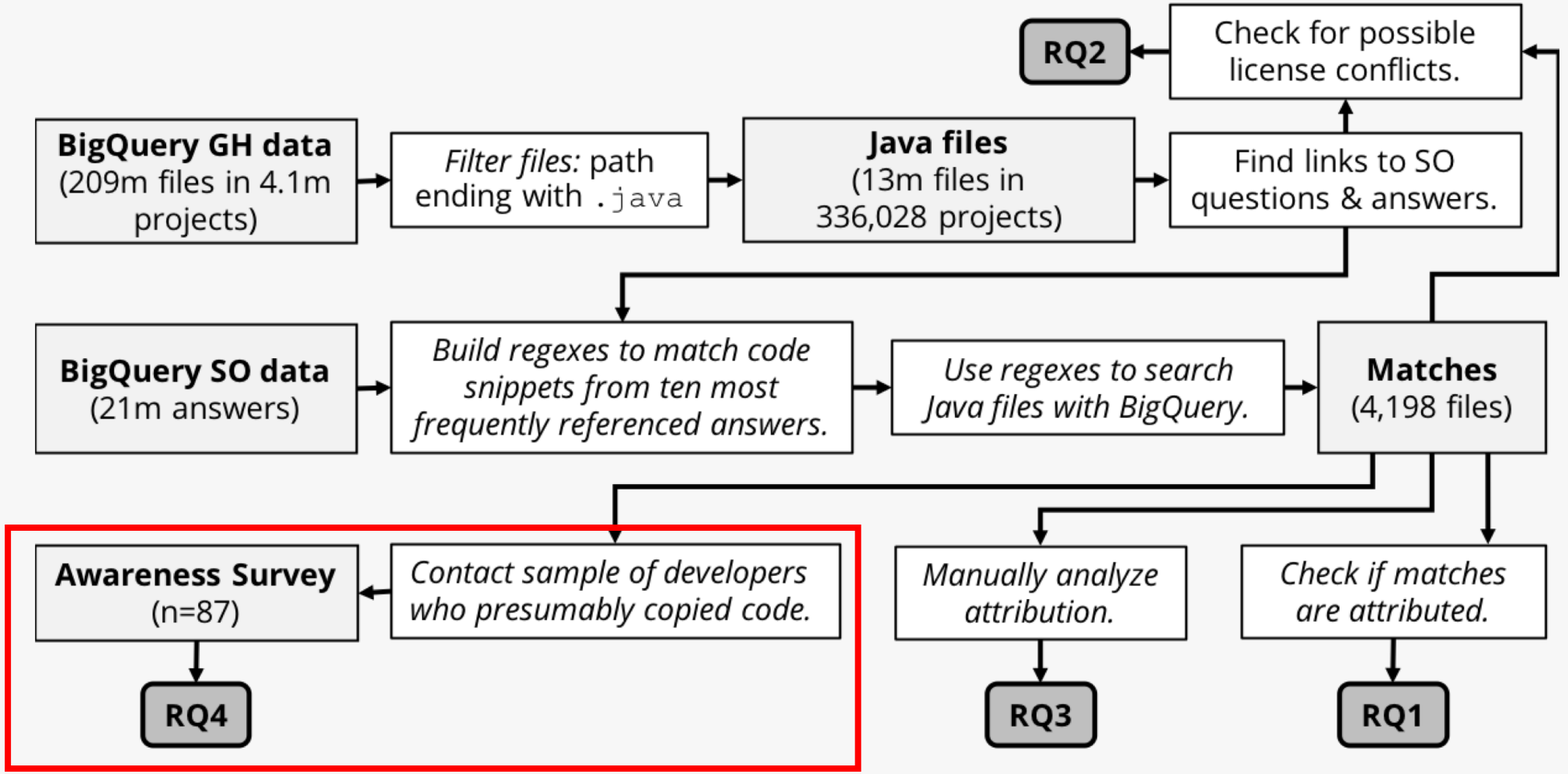
**RQ1:** How often is code from Stack Overflow posts used in public GitHub projects without the required attribution?

**RQ2:** How often does the license of repositories containing code copied from Stack Overflow conflict with Stack Overflow's license?

**RQ3:** Do developers adhere to the attribution requirements defined in the Stack Overflow terms of service?

**RQ4:** Are software developers aware of the licensing of Stack Overflow code snippets and its implications?

# Research Design: Phase 1



# Survey Results

- Contacted owners of GitHub projects containing copies of Stack Overflow snippets
- Received 87 responses (11.8% response rate)
- **75%** did **not know** that Stack Overflow content is licensed under CC BY-SA
- **41%** admitted **regularly copying** code from Stack Overflow
- Many thankful comments

# Survey Results: Stack Overflow Snippet in JDK

JDK / JDK-8170860

Get rid of the humanReadableByteCount() method in openjdk/hotspot

**Details**

| | | | |
|---|---|---|---|
| Type: | Bug | Status: | RESOLVED |
| Priority: | P2 | Resolution: | Fixed |
| Affects Version/s: | 9 | Fix Version/s: | 9 |
| Component/s: | hotspot | | |
| Labels: | noreg-self testbug | | |
| Subcomponent: | gc | | |
| Resolved In Build: | b156 | | |

implement the method humanReadableByteCount which body was copied from the Stack Overflow site: https://stackoverflow.com/a/3758880

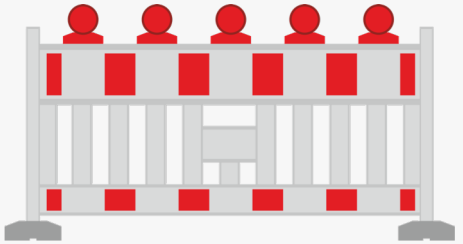
It's just a few lines of code, but it could cause legal issues. The method should be either re-implemented or removed.

Besides the potential legal issues, duplicating a code is not a good practice.

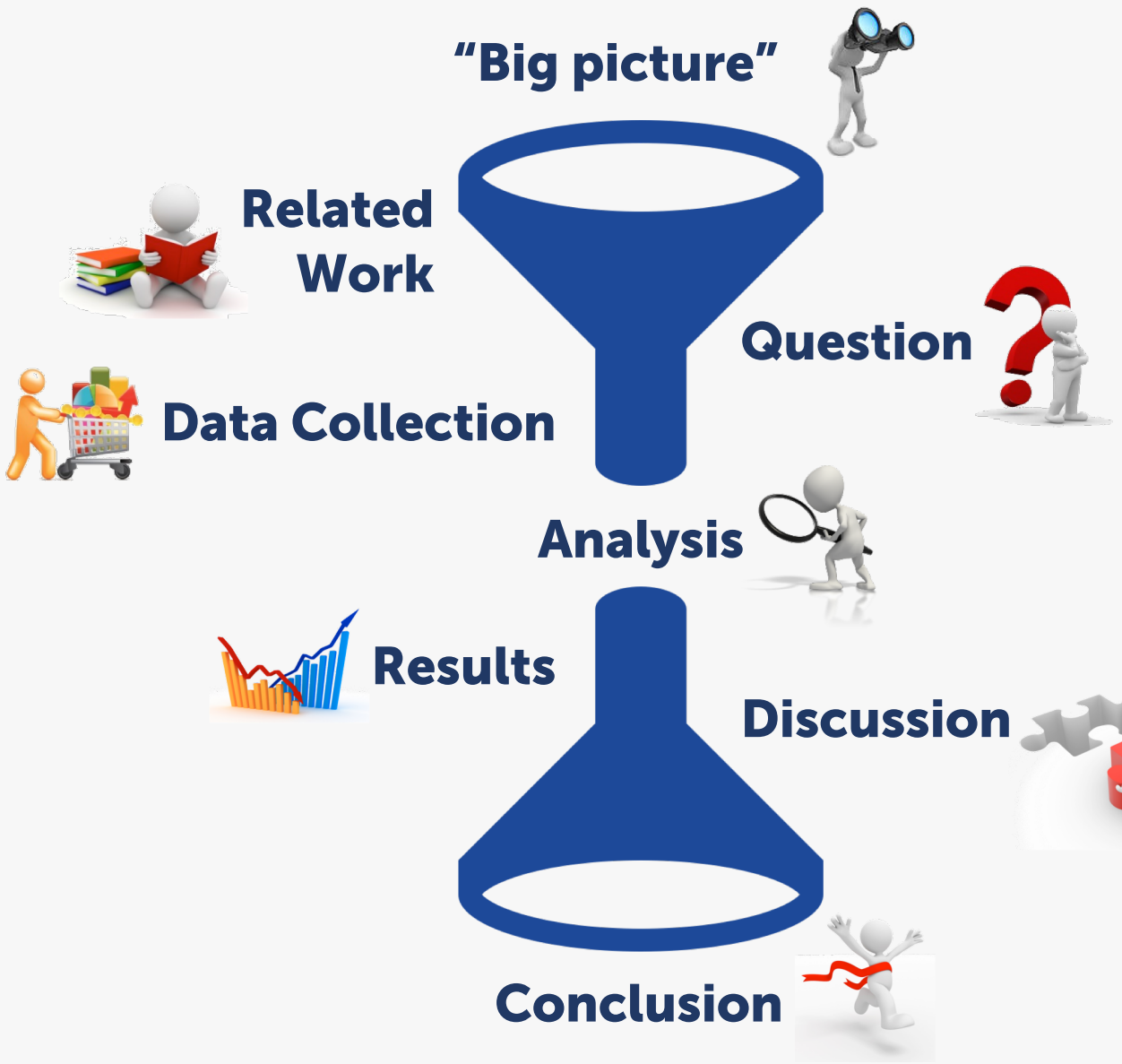
https://bugs.openjdk.java.net/browse/JDK-8170860

# Limitations

- Focus on Java, **generalizability** to other programming languages is limited
- In phases 1 and 2, we we only considered relatively **small samples** of snippets
  - Still found a considerable number of files with copies
  - Number of attributions was even smaller in phase 3, where we included more snippets and only searched for exact matches
- **External sources**
  - Analysis in paper
  - Excluded in phase 3
- Not all matches may be protected by **copyright**
  - Used proxies for originality



# Background: Verifiability




**Provenance tracking**

**Publish results, raw data, and provenance information**

# Usage and Attribution of

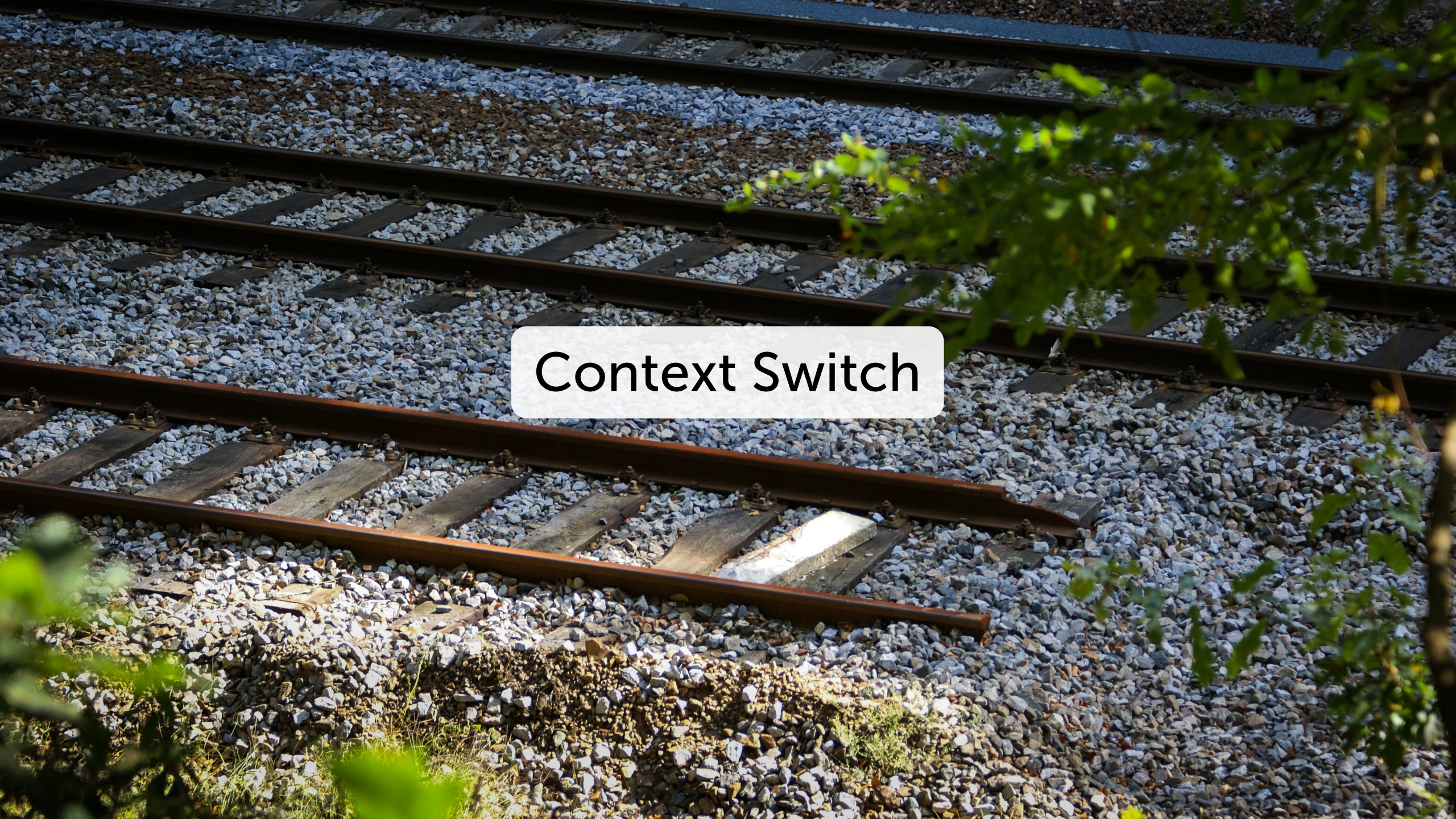

stackoverflow Code Snippets

in GitHub Projects

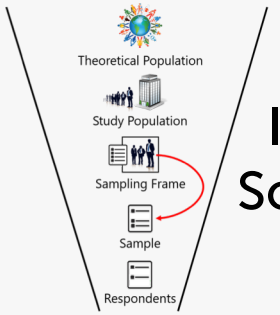
Sebastian Baltes
@s_baltes

**snippets.sbaltes.com**

*Supplementary material available on Zenodo.*


Context Switch

# “Parallel Thread”




Issues in Sampling Software Developers

2013

Methodology

Open Data

SOTorrent

Interdisciplinary Research

airbnb

Constructing Urban Tourism Space Digitally

2018

# SOTorrent

## Studying the Origin, Evolution, and Usage of Stack Overflow Code Snippets

**Sebastian Baltes**
@s_baltes

**sotorrent.org**

*Dataset available on Zenodo and BigQuery*

# Corresponding Research Papers

**SOTorrent: Reconstructing and Analyzing the Evolution of Stack Overflow Posts**

Sebastian Baltes
Lorik Dumani
research@sbaltes.com
dumani@uni-trier.de
University of Trier, Germany

Christoph Treude
christoph.treude@adelaide.edu.au
University of Adelaide, Australia

Stephan Diehl
diehl@uni-trier.de
University of Trier, Germany

ABSTRACT

Stack Overflow (SO) is the most popular questio
site for software developers, providing a larg
snippets and free-form text on a wide variety o
software artifacts, questions and answers on S
for example when bugs in code snippets are fix
to work with a more recent library version, or
code snippet is edited for clarity. To be able to a
on SO evolves, we built *SOTorrent*, an open da
official SO data dump. *SOTorrent* provides acces
tory of SO content at the level of whole posts ar
code blocks. It connects SO posts to other platfo
URLs from text blocks and by collecting refer

**SOTorrent: Studying the Origin, Evolution, and Usage of Stack Overflow Code Snippets**

Sebastian Baltes
*University of Trier, Germany*
research@sbaltes.com

Christoph Treude
*University of Adelaide, Australia*
christoph.treude@adelaide.edu.au

Stephan Diehl
*University of Trier, Germany*
diehl@uni-trier.de

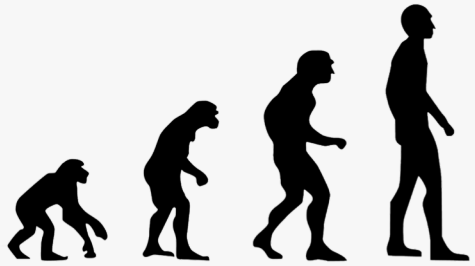
***Abstract*—Stack Overflow (SO) is the most popular question-and-answer website for software developers, providing a large amount of copyable code snippets. Like other software artifacts, code on SO evolves over time, for example when bugs are fixed or APIs are updated to the most recent version. To be able to analyze how code and the surrounding text on SO evolves, we built *SOTorrent*, an open dataset based on the official SO data dump. *SOTorrent* provides access to the version history of SO content at the level of whole posts and individual text and code blocks. It connects code snippets from SO posts to other platforms by aggregating URLs from surrounding text blocks and comments, and by collecting references from GitHub files to SO posts. Our vision is that researchers will use *SOTorrent* to investigate and understand the evolution and maintenance of code on SO and its relation to other platforms such as GitHub.**

dataset [16] that enables researchers to analyze the versi
history of SO posts at the level of individual text and co
blocks (see Figure 1 for exemplary posts). The official S
data dump [1] keeps track of different versions of e
posts, but does not contain information about diffe
between versions at a more fine-grained level. In parti
extracting different versions of the same code snippet from
the history of a post is challenging and required us to develop
a complex strategy, involving the eva uation of 134 different
string similarity metrics [15]. Beside providing access to the
version history, our dataset links SO p
in two ways: (1) by extracting linked
of SO posts and from post comment

**MSR 2018/2019**

# Why Reconstruct and Analyze SO Post Evolution?

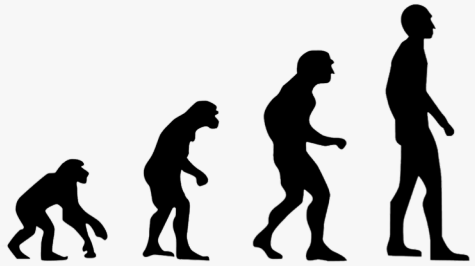
- The content of **14.3 million posts** has been **edited** after creation
  (SO data dump 2018-03-01)
- Like other **software artifacts**, SO posts **evolve over time**:
  - Bugs in code snippets are fixed
  - Clarifications are added in text documenting the code
  - Snippets are updated to new language/library versions
- **Copying code** from Stack Overflow (SO) is common, despite licensing, security, and maintainability implications



# Why Reconstruct and Analyze SO Post Evolution?

**Evolution of code on SO** differs from regular software projects:

- **Short** code snippets (12 LOC on average)
- **No bug tracking** system (just comments and new answers)
- **No versioning** for individual snippets (just whole posts)



# Example

## Read/convert an InputStream to a String

3101

929

If you have `java.io.InputStream` object, how should you process that object and produce a `String` ?

Suppose I have an `InputStream` that contains text data, and I want to convert this to a `String` . For example, so I can write the contents of the stream to a log file.

What is the easiest way to take the `InputStream` and convert it to a `String` ?

```
public String convertStreamToString(InputStream is) {
    // ???
}
```

java string io stream inputstream

share improve this question

edited May 19 '17 at 8:58 Pehlaj 4,824 ● 6 ● 25 ● 43

asked Nov 21 '08 at 16:47 Johnny Maelstrom 15.9k ● 5 ● 16 ● 17

## Question

https://stackoverflow.com/q/309424

2034

Here's a way using only standard Java library (note that the stream is not closed, YMMV).

```
static String convertStreamToString(java.io.InputStream is) {
    java.util.Scanner s = new java.util.Scanner(is).useDelimiter("\\A");
    return s.hasNext() ? s.next() : "";
}
```

I learned this trick from "Stupid Scanner tricks" article. The reason it works is because Scanner iterates over tokens in the stream, and in this case we separate tokens using "beginning of the input boundary" (\A) thus giving us only one token for the entire contents of the stream.

**Note, if you need to be specific about the input stream's encoding, you can provide the second argument to `Scanner` constructor that indicates what charset to use (e.g. "UTF-8").**

Hat tip goes also to Jacob, who once pointed me to the said article.

**EDITED:** Thanks to a suggestion from Patrick, made the function more robust when handling an empty input stream. **One more edit:** nixed try/catch, Patrick's way is more laconic.

share improve this answer

edited Sep 2 '17 at 1:27

answered Mar 26 '11 at 20:40 Pavel Repin 25.3k ● 1 ● 27 ● 36

## Answer

https://stackoverflow.com/a/5445161

2034

Here's a way using only standard Java library (note that the stream is not closed, YMMV).

```
static String convertStreamToString(java.io.InputStream is) {
    java.util.Scanner s = new java.util.Scanner(is).useDelimiter("\\A");
    return s.hasNext() ? s.next() : "";
}
```

I learned this trick from "Stupid Scanner tricks" article. The reason it works is because Scanner iterates over tokens in the stream, and in this case we separate tokens using "beginning of the input boundary" (\A) thus giving us only one token for the entire contents of the stream.

**Note, if you need to be specific about the input stream's encoding, you can provide the second argument to `Scanner` constructor that indicates what charset to use (e.g. "UTF-8").**

Hat tip goes also to Jacob, who once pointed me to the said article.

**EDITED:** Thanks to a suggestion from Patrick, made the function more robust when handling an empty input stream. **One more edit:** nixed try/catch, Patrick's way is more laconic.

share improve this answer

edited Sep 2 '17 at 1:27

answered Mar 26 '11 at 20:40



Pavel Repin

25.3k ● 1 ● 27 ● 36

https://stackoverflow.com/a/5445161

2034

Here's a way using only standard Java library (note that the stream is not closed, YMMV).

```
static String convertStreamToString(java.io.InputStream is) {
    java.util.Scanner s = new java.util.Scanner(is).useDelimiter("\\A");
    return s.hasNext() ? s.next() : "";
}
```

**Code snippet**

I learned this trick from "Stupid Scanner tricks" article. The reason it works is because Scanner iterates over tokens in ase we separate tokens using boundary" (\A) thus giv the entire contents of the st

**Source of snippet**

**Reference to JDK**

**Note, if you need to be specific about the input stream's encoding, you can provide the second argument to `Scanner` constructor that indicates what charset to use (e.g. "UTF-8").**

Hat tip goes also to Jacob, who once pointed me to the said article.

**EDITED:** Thanks to a suggestion from Patrick, made the function more robust when handling an empty input stream. **One more edit:** nixed try/catch, Patrick's way is more laconic.

share in er

**Post edits**

edited Sep 2

**Reasons for edits**

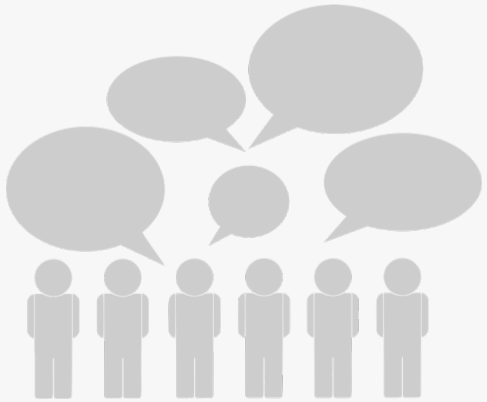
ered Mar 26 '11 at 20:40



Pavel Repin

25.3k ● 1 ● 27 ● 36

# Comments



**EDITED:** Thanks to a suggestion from Patrick, made the function more robust when handling an empty input stream. **One more edit:** nixed try/catch, Patrick's way is more laconic.

share improve this answer — edited Sep 2 '17 at 1:27 — answered Mar 26 '11 at 20:40 Pavel Repin 25.3k ●1 ●27 ●36

7 Thanks, for my version of this I added a finally block that closes the input stream, so the user doesn't have to since you've finished reading the input. Simplifies the caller code considerably. – user486646 Apr 21 '12 at 17:07

4 @PavelRepin @Patrick in my case, an empty inputStream caused a NPE during Scanner construction. I had to add `if (is == null) return "";` right at the beginning of the method; I believe this answer needs to be updated to better handle null inputStreams. – CFL_Jeff Aug 9 '12 at 13:36

The problem with this approach I find is it does not handle CR/LF translations too well. So you have to make sure your line endings are consistent. – Archimedes Trajano Feb 28 '13 at 12:13

@ArchimedesTrajano does `IOUtils.copy(inputStream, writer, encoding)` deal with CR/LF translations better? I think CR/LF consistency is entirely unrelated issue. Not saying it isn't an issue. – Pavel Repin Mar 1 '13 at 9:18

95 For Java 7 you can close in a try-with: `try(java.util.Scanner s = new java.util.Scanner(is)) { return s.useDelimiter("\\A").hasNext() ? s.next() : ""; }` – earcam Jun 13 '13 at 5:24

3 Unfortunately this solution seems to go and lose the exceptions thrown in my underlying stream implementation. – Taig Jul 16 '13 at 7:59

excellent trick! any ideas about performance of Scanner vs reading the stream in a more verbose way? – isapir Aug 28 '13 at 19:54

@Igal I didn't measure it. If you do, gist it and I'll append your results to the answer. – Pavel Repin Aug 28 '13 at 23:13

11 FYI, hasNext blocks on console input streams (see here). (Just ran into this issue right now.) This solution works fine otherwise... just a heads up. – Ryan Feb 24 '14 at 5:36

1 @earcam thanks for the tip! For those wondering how this works, it's thanks to try-with-resources – Mark Mar 14 '15 at 21:33

1 looks like a neat trick, but it seems there are some limitations. For me it hangs when reading InputStream from Socket. When testing with something like ByteArrayInputStream it works nicely. Reading from socket results in a hang. – Normunds Kalnberzins Dec 16 '15 at 14:16

If the `Scanner` is going to be *"giving us only one token for the entire contents of the stream"* anyways, why not use a normal stream reader? `Scanner` is meant to pre-parse tokens out of the stream, not for being the stream reader (without any parsing being done). – XenoRo Dec 28 '15 at 14:06

@AlmightyR `Scanner` has built-in stream reading logic and we're telling it that the stream has just one token. A special case of Scanner usage. Fair game. Good point though. This stuff is clearly a hack. – Pavel Repin Jan 15 '16 at 1:23

1 be careful ,using this method with socket stream is slow ! Scanner#next() hangs for a little while. – WestFarmer Apr 20 '16 at 10:22

1 nice answer, the article link is on oracle website community.oracle.com/blogs/pat/2004/10/23/stupid-scanner-tricks – Eng. Samer T Jul 23 '17 at 16:04

**Bug report**

**Alternative solution**

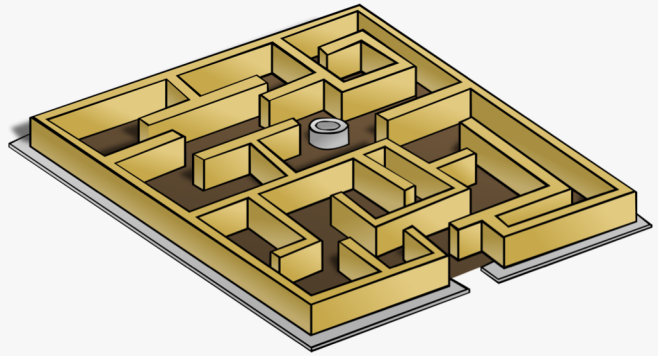
**Bug report**

**Bug report**

**Comment by author**

This stuff is clearly a hack.

Even for such a simple code snippet, the **context** is quite **complex**:



- The snippet is based on an **external source**
- Hidden in the **comments**, the author acknowledges that his solution is *“clearly a hack”*
- There are several **bug reports** pointing to issues
- Has the snippet been **edited** to fix those issues?
- Is the snippet **safe** to use?


SOTorrent

# Retrieve all versions of a code snippet:

```
SELECT PostHistoryId, Content, Length, LineCount, PredSimilarity
FROM PostBlockVersion
WHERE PostId=5445161 AND LocalId=2 AND PredEqual=0
ORDER BY PostHistoryId DESC;
```

**Most recent version**

| PostHistoryId | Content | Length | LineCount | PredSimilarity |
|---|---|---|---|---|
| 155295527 | static String convertStreamToString(java.io.In… | 192 | 4 | 0.7532467532467533 |
| 154620092 | static String convertStreamToString(java.io.In… | 352 | 13 | 0.7532467532467533 |
| 44935719 | static String convertStreamToString(java.io.In… | 192 | 4 | 0.9846153846153847 |
| 31249705 | public static String convertStreamToString(jav… | 199 | 4 | 0.9523809523809523 |
| 30827994 | String convertStreamToString(java.io.InputStr… | 185 | 4 | 0.6875 |
| 25270546 | String convertStreamToString(java.io.InputStr… | 239 | 7 | 0.9714285714285714 |
| 21289331 | public String convertStreamToString(java.io.I… | 246 | 7 | 0.8157894736842105 |
| 21230790 | import java.util.Scanner; import java.util.No… | 298 | 10 | 0.8405797101449275 |

# Retrieve line-based difference for latest version:

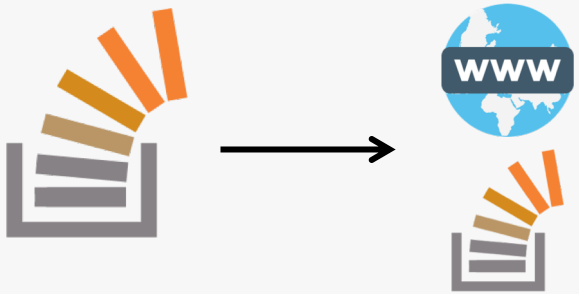
```
SELECT PostHistoryId, LocalId, PredLocalId, PostBlockDiffOperationId, Text
FROM PostBlockDiff
WHERE PostHistoryId=155295527 AND LocalId=2;
```

**Changed lines**

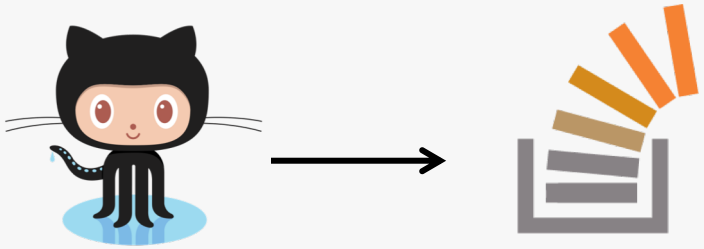
| PostHistoryId | LocalId | PredLocalId | PostBlockDiffOperationId | | Text |
|---|---|---|---|---|---|
| 155295527 | 2 | 2 | 0 | Equal | static String convertStreamToString(java.io.InputStream is) { |
| 155295527 | 2 | 2 | -1 | **Delete** | java.util.Scanner s = new java.util.Scanner(is).useDelimiter("\\A");… |
| 155295527 | 2 | 2 | 1 | **Insert** | if (is == null) { return ""; } java.util.Scanner s… |
| 155295527 | 2 | 2 | 0 | Equal | } |

# Extracting Links From Stack Overflow Posts

- Extracted **31.4m links** from 11.6m posts, pointing to 567k different domains using a regular expression (SOTorrent 2018-05-04)




- Extracted **6.0m links** from 438k GitHub repos, pointing to 147k posts using Google BigQuery (SOTorrent 2018-05-04)



# Retrieve links from a post version:

```
SELECT PostId, PostHistoryId, Domain, Url
FROM PostVersionUrl
WHERE PostHistoryId=155295527;
```

| PostId | PostHistoryId | Domain | Url |
|---|---|---|---|
| 5445161 | 155295527 | community.oracle.com | https://community.oracle.com/blogs/pat/2004/10/23/stupid-scanner-tricks |
| 5445161 | 155295527 | download.oracle.com | http://download.oracle.com/javase/8/docs/api/java/util/Scanner.html |
| 5445161 | 155295527 | stackoverflow.com | https://stackoverflow.com/users/68127/jacob-gabrielson |
| 5445161 | 155295527 | stackoverflow.com | https://stackoverflow.com/users/101272/patrick |

# Retrieve links from GitHub repos to post:

```
SELECT PostId, RepoName, Branch, Path, FileExt, Size, Copies
FROM PostReferenceGH
WHERE PostId=5445161;
```

**Referenced in 103 distinct repos**

| PostId | RepoName | Branch | Path | FileExt | Size |
|---|---|---|---|---|---|
| 5445161 | resource4j/resource4j | master | core/src/main/java/com/github/resource4j/object… | .java | 2077 |
| 5445161 | yugecin/opsu-dance | master | src/itdelatrisu/opsu/Utils.java | .java | 16107 |
| 5445161 | Roojin/persian-calendar-view | master | persiancalendar/src/main/java/ir/mirrajabi/persia… | .java | 16833 |
| 5445161 | FITeagle/sfa | master | src/main/java/org/fiteagle/north/sfa/dm/SFA_XM… | .java | 5426 |
| 5445161 | Steguer/ProjetAndroid | master | ProjetAndroid/libs/android-maps-utils/demo/src/… | .java | 1140 |
| 5445161 | ScottSWu/opsu | master | src/itdelatrisu/opsu/Utils.java | .java | 17943 |
| 5445161 | massimiliano76/freedomotic | master | plugins/devices/restapi-v3/src/main/java/com/fre… | .java | 3315 |

…

**MSR Mining Challenge 2019**

Abstracts due Feb 1, 2019

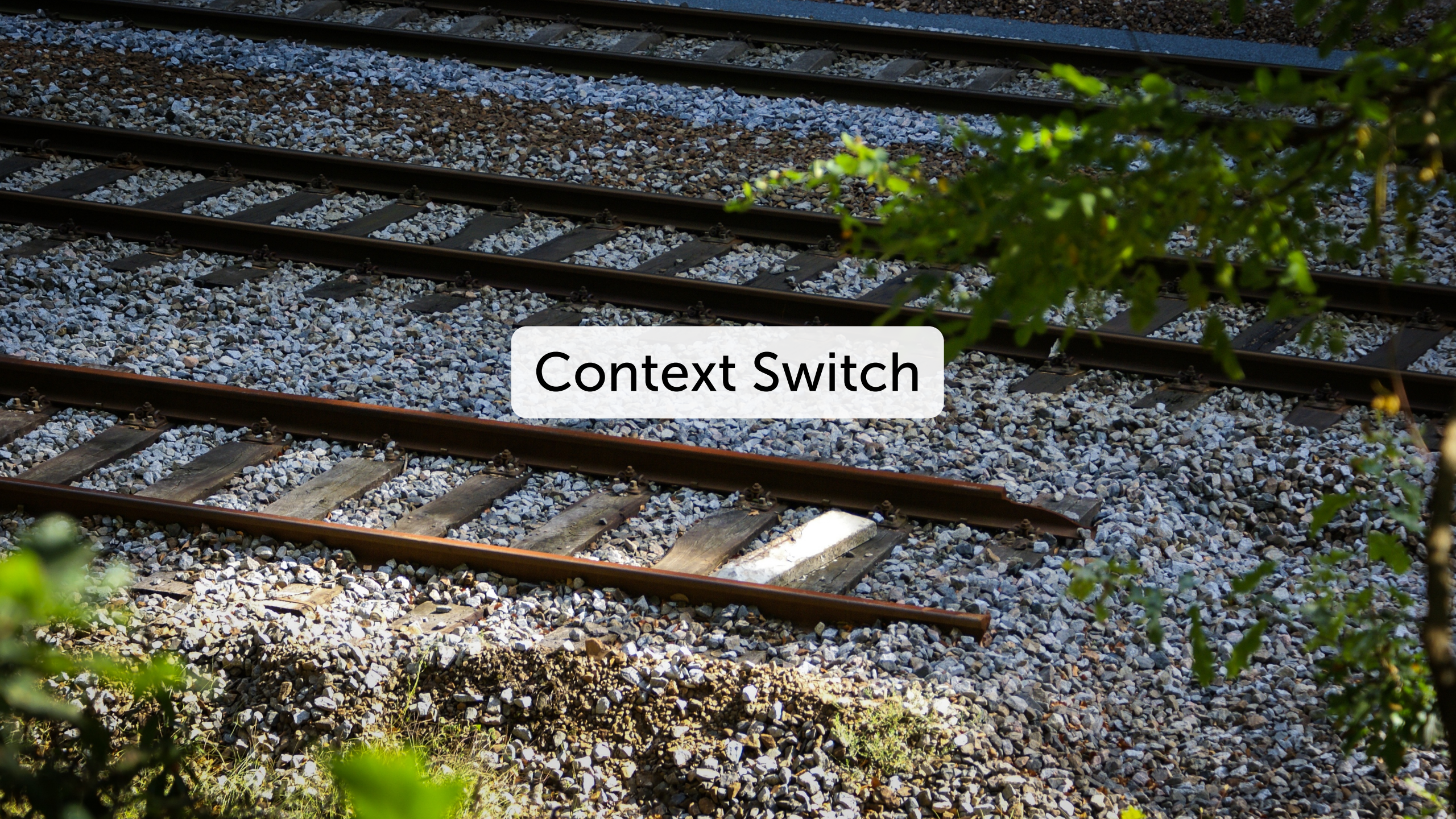
Papers due Feb 6, 2019

Sebastian Baltes

@s_baltes

**sotorrent.org**

*Dataset available on Zenodo and BigQuery*


Context Switch

# Studied Habits

2013

Sketches and Diagrams in Practice

Sketching

SketchLink

How Developers Locate Performance Bugs

LivelySketches

Towards a Theory of Software Development Expertise

Expertise Development

2018

**Code Plagiarism**


stackoverflow Code Snippets in GitHub Projects

# Question 3

How could we better support developers struggling with licenses of online code snippets?

- What could Stack Overflow as a platform do?
- What could project owners/companies do?