

An Exploratory Study on Confusion in Code Reviews



Felipe Ebert
@felipe_fifo



Fernando Castor
@fernandocastor



Nicole Novielli
@NicoleNovielli



Alexander Serebrenik
@aserebrenik

Code Reviews & Confusion

```
Integer euclidAlgorithm (int A, int B) {
    A=Math.abs(A);
    B=Math.abs(B);
    while (B!=0){
        if (A>B) A=A-B;
        else B=B-A;
    }
    return A;
}

Integer euclidAlgorithm (int A, int B) {
    A=Math.abs(A);
    B=Math.abs(B);
    while (B!=0){
        if (A>B) A=A-B;
        else B=B-A;
    }
    return A;
}

Integer euclidAlgorithm (int A, int B) {
    A=Math.abs(A);
    B=Math.abs(B);
    while (B!=0){
        if (A>B) A=A-B;
        else B=B-A;
    }
    return A;
}
```

Previously...

Confusion in Code Reviews: Reasons, Impacts, and Coping Strategies

Felipe Ebert*, Fernando Castor*, Nicole Novielli[†], Alexander Serebrenik[‡]

*Federal University of Pernambuco, Brazil, {fe,castor}@cin.ufpe.br

[†]University of Bari, Italy, nicole.novielli@uniba.it

[‡]Eindhoven University of Technology, The Netherlands, a.serebrenik@tue.nl

Abstract—Code review is a software quality assurance practice widely employed in both open source and commercial software projects to detect defects, transfer knowledge and encourage adherence to coding standards. Notwithstanding, code reviews can also delay the incorporation of a code change into a code base, thus slowing down the overall development process. Part of this delay is often a consequence of reviewers not understanding, becoming confused by, or being uncertain about the intention, behavior, or effect of a code change.

We investigate the reasons and impacts of confusion in code reviews, as well as the strategies developers adopt to cope with confusion. We employ a concurrent triangulation strategy to combine the analyses of survey responses and of the code review comments, and build a comprehensive confusion framework structured along the dimensions of the review process, the artifact being reviewed, the developers themselves and the relation between the developer and the artifact. The most frequent reasons for confusion are the *missing rationale*, *discussion of non-functional requirements of the solution*, and *lack of familiarity with existing code*. Developers report that confusion *delays the merge decision*, *decreases review quality*, and results in *additional discussions*. To cope with confusion developers *request information*, *improve familiarity with existing code*, and *discuss off-line*.

Based on the results, we provide a series of implications for tool builders, as well as insights and suggestions for researchers. The results of our work offer empirical justification for the need to improve code review tools to support developers facing confusion.

Index Terms—code review; confusion; survey; cards sorting.

I. INTRODUCTION

Code review is an important practice for software quality assurance, which has been widely adopted in both open source and commercial software projects [1], [2], [3], [4], [5]. The benefits of code reviews are well-known. Active participation of developers in code reviews decreases the number of post-release defects and improves the software quality [6], [7]; knowledge transfer and adherence to the project coding standards are additional benefits of code reviews [8], [9], [10].

However, code reviews also incur cost on software development projects as they can delay the merge of a code change in the repository and, hence, slowdown the overall development process [11], [12]. Indeed, the time spent by a developer reviewing code is non-negligible [1] and may take up to 10-15% of the overall time spent on software development activities [13], [6]. The merge of a code change in the repository can be even further delayed if the reviewers experience difficulties in understanding the change, *i.e.*, they are not certain about its correctness, run-time behavior and

impact on the system [6], [9], [14], [15], [16]. As such, we believe that a proper understanding of the main reasons for confusion in code reviews represents a necessary starting point towards reducing the cost and enhancing the effectiveness of this practice, thus improving the overall development process.

We focus on confusion experienced by developers during code review, its reasons and impacts, as well as on the strategies adopted by developers to cope with it. By *confusion* we mean “any situation where the person is uncertain about anything or unable to understand something” [17]. We do not distinguish between lack of knowledge, confusion, and uncertainty. Indeed, confusion (which also encompasses doubt and uncertainty) and lack of knowledge are strictly connected (*e.g.*, confusion could be determined by lack of knowledge) [18].

Our goals are threefold. First, we aim at obtaining empirically-driven actionable insights for both researchers and tool builders, on what are the main causes of confusion in code reviews. Thus, we formulate our first research question:

RQ1. *What are the reasons for confusion in code reviews?*

We have observed that the three most frequent reasons for confusion are *missing rationale*, *discussion of the solution: non-functional*, and *lack of familiarity with existing code*.

Second, while confusion can be expected to negatively affect code reviews, we would like to identify specific impacts of confusion. By monitoring these impacts developers and managers can curb the undesirable consequences. As such, we formulate our second research question:

RQ2. *What are the impacts of confusion in code reviews?*

Our results suggest that the merge decision is *delayed* when developers experience confusion, there is an increase in the number of messages exchanged during the discussion, and the *review quality decreases*. However, we also observed unexpected consequences of confusion, such as helping to find a *better solution*. This suggests that communicating uncertainty and doubts might be beneficial for collaborative code development, *i.e.*, by inducing critical reflection [19] or triggering knowledge transfer [9].

Finally, we believe that understanding the strategies adopted by the developers to deal with confusion can further inform the design of tools to support code reviewers in fulfilling their information needs associated to the experience of uncertainty and doubt. As such, we formulate our third research question:

RQ3. *How do developers cope with confusion during code reviews?*



What are the gaps in the literature?



What are the gaps in the literature?

•Solutions?

•Impacts?

Preliminary study

Topics related to the code change

How often do you feel confused when reviewing code changes due to:

	Not at all	Less than once a month	Once a month	Once a week	Once a day	More than once a day
Missing code change rationale	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Lack of understanding of the system behavior	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Lack of documentation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Long or complex code change	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Lack of context	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The impact of code change	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Lack of tests	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Preliminary study

Group	Reason for Confusion	Mean	Median	Dimension
1	Long or complex code change	2.40	2	Artifact
	Organization of work	2.33	2	Review Process
	Dependency between different code changes	2.24	2	Review Process
	Lack of documentation	2.20	2	Artifact
	Missing code change rationale	2.19	2	Artifact
2	Lack of tests	2.14	2	Artifact
	Lack of familiarity with the existing code	2.11	2	Link
	Lack of understanding of the system behavior	2.08	2	Artifact
	Not having enough time	2.03	2	Review Process
	Disagreement with the strategy proposed in the code change	2.01	2	Artifact
	The impact of code change	2.00	2	Artifact
	Lack of understanding of the correctness of the code change	1.96	2	Artifact
3	Lack of context	1.93	2	Artifact
	Discussion of the solution related to non-functional aspects	1.83	2	Artifact
	Lack of understanding of the code change	1.82	2	Link
	Fatigue	1.80	2	Developer
	Lack of understanding of the intention of peers' comments	1.77	2	Developer
	Lack of understanding of the problem	1.77	2	Link

Systematic Mapping Study

#1	<p><i>Long or complex code change</i> ("code review" OR "code inspection" OR (("peer code review" OR "peer review") AND software)) AND ((long OR large OR huge OR big OR complex OR decompose OR composite OR cumbersome OR tricky OR intricate OR complicate OR tangled) AND ("code change" OR changeset OR commit OR "patch set" OR patch OR "pull request"))</p>
----	---

Systematic Mapping Study

#1	<p><i>Long or complex code change</i> ("code review" OR "code inspection" OR (("peer code review" OR "peer review") AND software)) AND ((long OR large OR huge OR big OR complex OR decompose OR composite OR cumbersome OR tricky OR intricate OR complicate OR tangled) AND ("code change" OR changeset OR commit OR "patch set" OR patch OR "pull request"))</p>
----	---

IEEE Xplore[®]
Digital Library

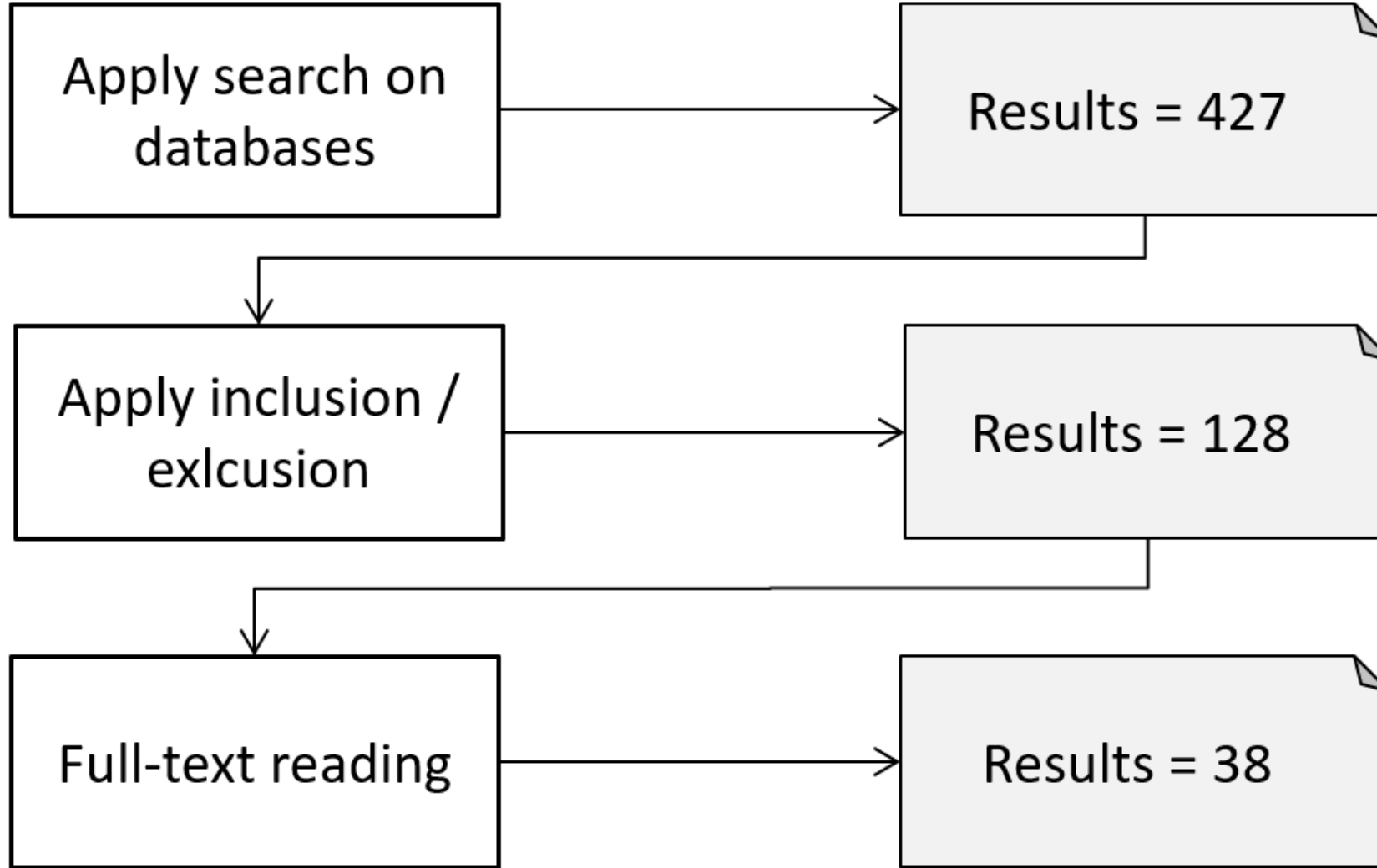
ACM  DIGITAL
LIBRARY



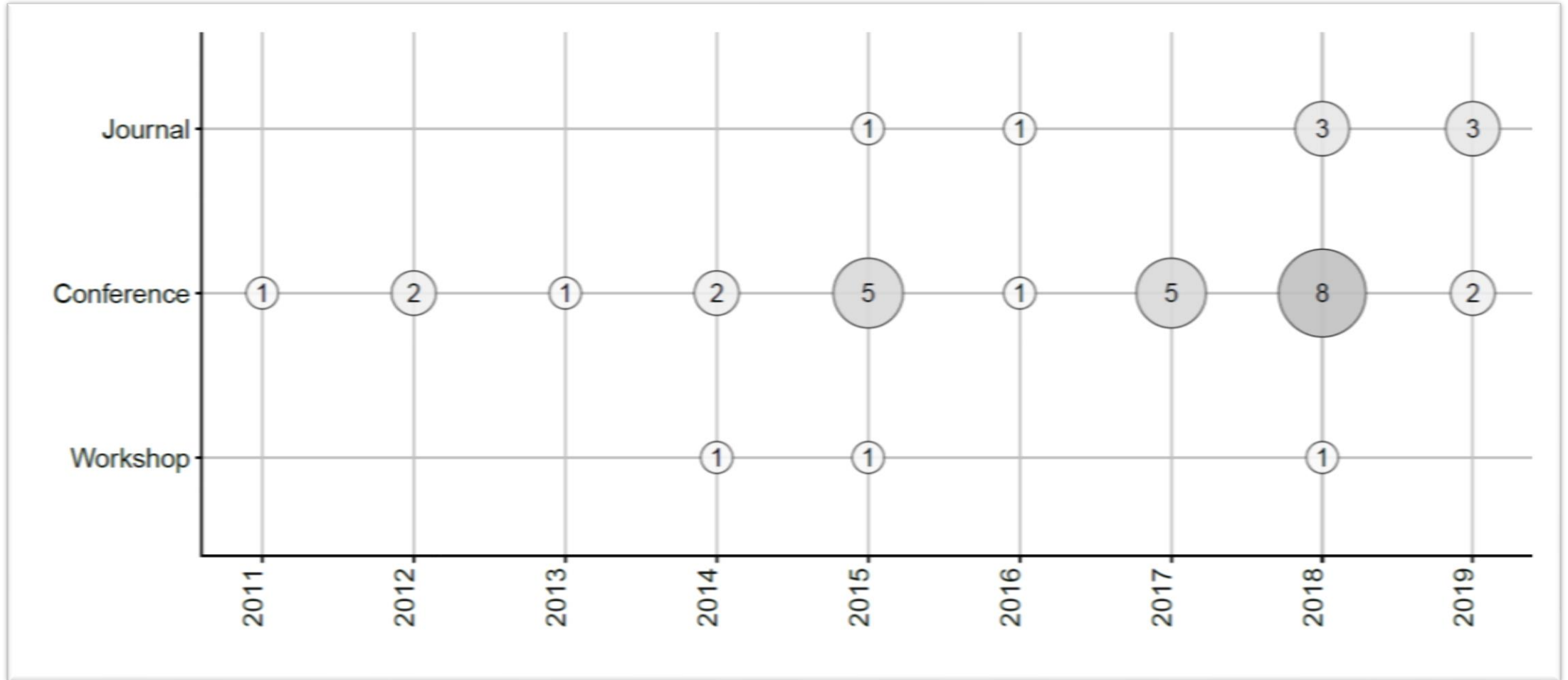
ELSEVIER
Scopus



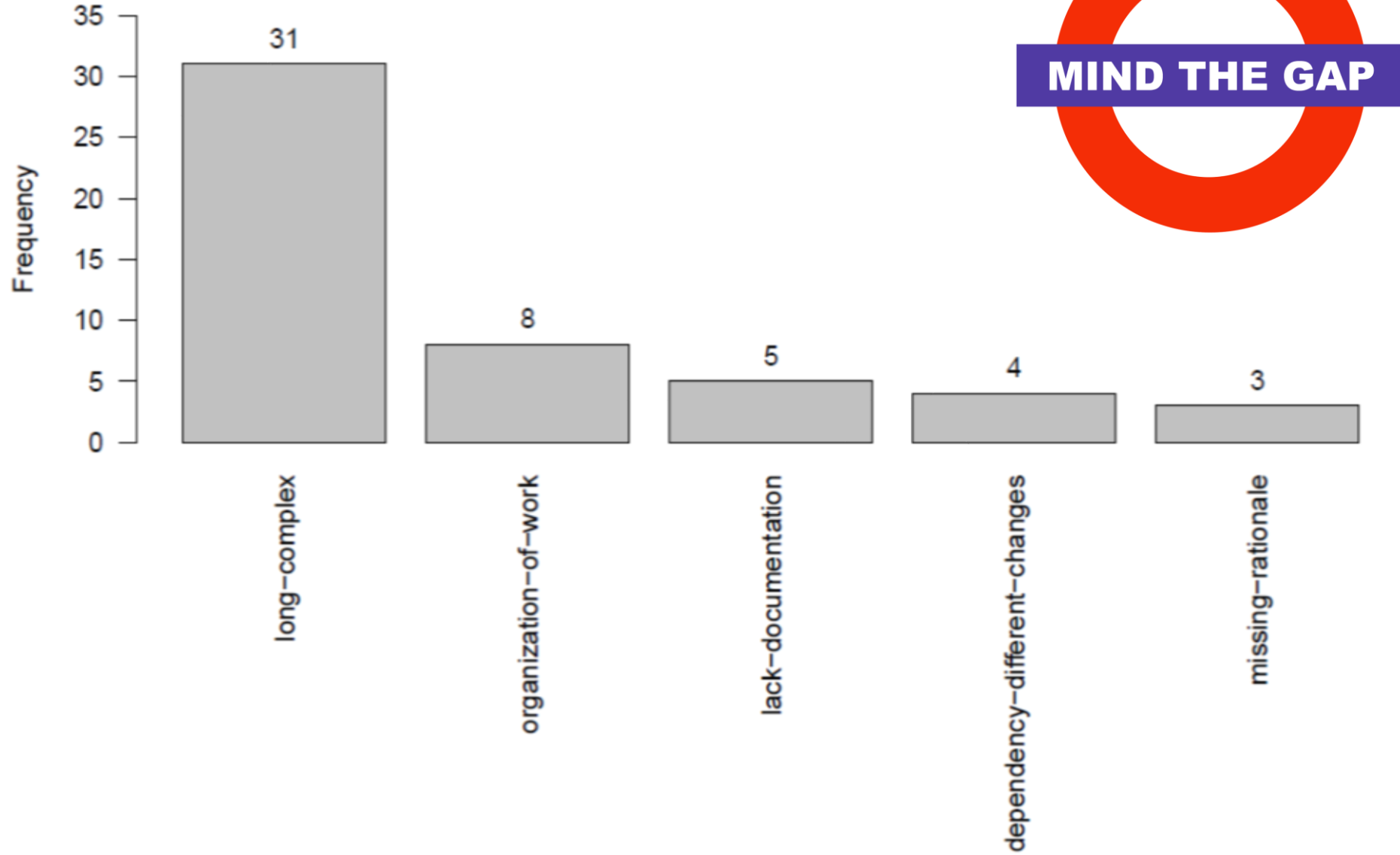
SpringerLink

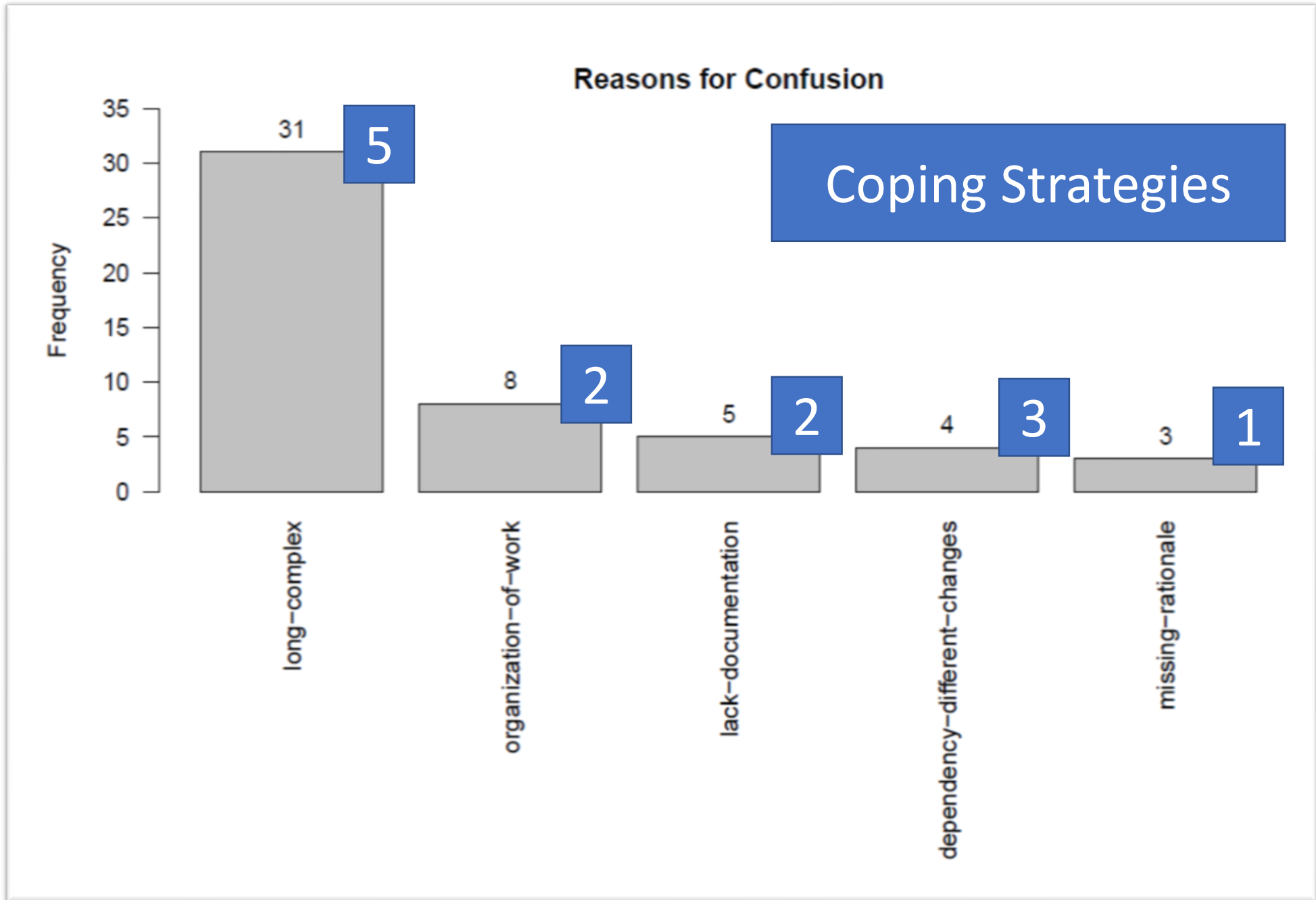


Evolution on the Literature

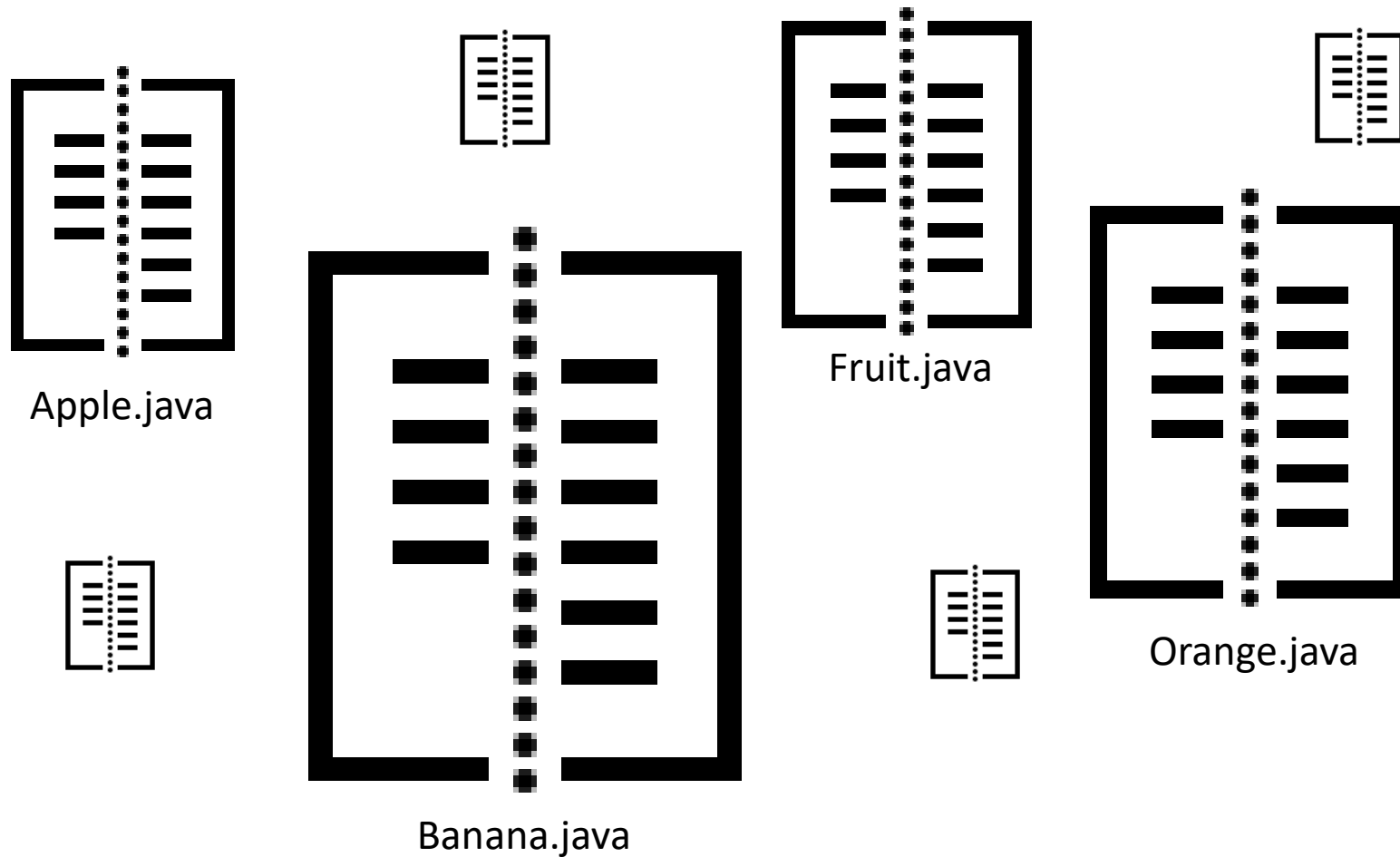


Reasons for Confusion

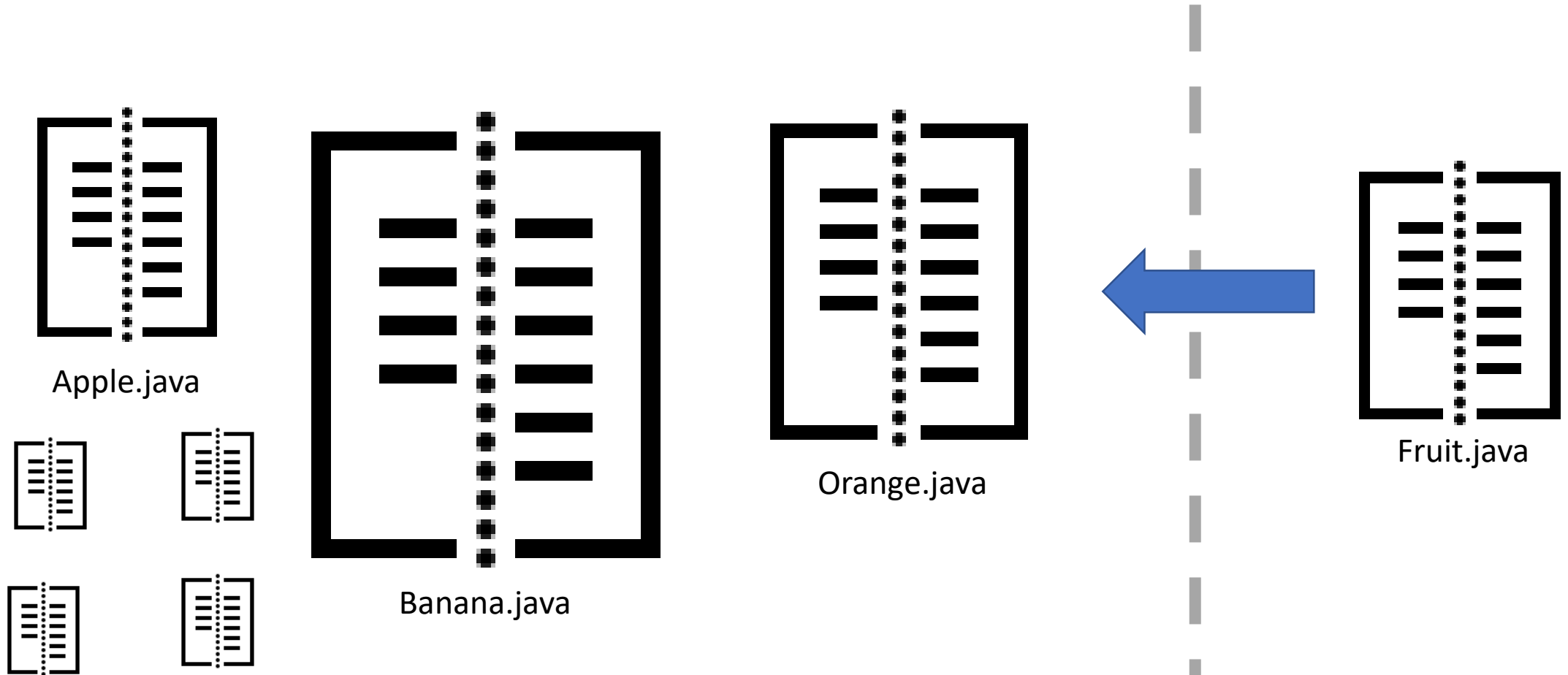




Problem: long or complex code change



Solution: Make use of salient files



Relationships: Reasons and Impacts

Reasons for confusion vs Impacts	<i>Delaying</i>	<i>Decreased review quality</i>	<i>Increased development effort</i>	<i>Review rejection</i>	<i>Frustration</i>
Long or complex change	x	x	x	x	
Organization of work	x	x	x	x	
Dependency between changes	x				
Lack of documentation				x	
Missing rationale					x

Implications

- Improved framework of confusion
 - 13 ==> 21 coping strategies

Implications

- Improved framework of confusion
 - 13 ==> 21 coping strategies
- Guideline on how to cope with confusion

Implications

- Improved framework of confusion
 - 13 ==> 21 coping strategies
- Guideline on how to cope with confusion
- Tools builders
 - Cluster code changes performing the same task

Implications

- Improved framework of confusion
 - 13 ==> 21 coping strategies
- Guideline on how to cope with confusion
- Tools builders
 - Cluster code changes performing the same task
- Research agenda
 - Automatic rationale extraction

Implications

- Improved framework of confusion
 - 13 ==> 21 coping strategies
- Guideline on how to cope with confusion
- Tools builders
 - Cluster code changes performing the same task
- Research agenda
 - Automatic rationale extraction

Learning Developer Tools from Code
+ other artifacts

Summary



What are the gaps in the literature?

- What are the **solutions** proposed by researchers for the most frequent reasons for confusion?
- What **relationships** has previous research established between the most frequent reasons for confusion and their impacts?

Preliminary study

Topics related to the code change

How often do you feel confused when reviewing code changes due to:

	Not at all	Less than once a month	Once a month	Once a week	Once a day	More than once a day
Missing code change rationale	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Lack of understanding of the system behavior	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Lack of documentation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



Relationships: Reasons and Impacts

Reasons for confusion vs Impacts	Delaying	Decreased review quality	Increased development effort	Review rejection	Frustration
Long or complex change	x	x	x	x	
Organization of work	x	x	x	x	
Dependency between changes	x				
Lack of documentation				x	
Missing rationale					x

Thank you!

f.ebert@tue.nl

<https://felipeebert.github.io>

@felipe_fifo