FROM COLLABORATION TO RESOLUTION OF MERGE CONFLICTS

TO EVALUATING AI'S ROLE ON SOFTWARE DEVELOPMENT

Gustavo do Vale

GUSTAVO ANDRADE DO VALE

Formação

Bel. Sistemas de Informação
Universidade Federal de Lavras - UFLA

Mestre em Ciência da Computação
Universidade Federal de Minas Gerais - UFMG

PhD em Ciência da Computação
Universität des Saarlandes

Experiência

Professor (2023 - Atual)
Unilavras, Lavras, Brasil

CEO (2023 - Atual)
Grupo Vale, Lavras, Brasil

Professor (2022 - 2025)
Fagammon, Lavras, Brasil

Experiência (Cont.)

PhD Intern (2022)
Meta (ex-Facebook), Londres, Reino Unido

Pesquisador (2020 - 2024)
Universität des Saarlandes, Saarbrücken, Alemanha

Senior IT Consultant (2018 - 2022)
msg systems, Passau, Alemanha

Assistente de Pesquisa (2016 - 2020)
Passau Universität, Passau, Alemanha

Assistente de Pesquisa (2014-2016)
Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, Brasil

Gerente de Projetos (2010-2013)
Comp Júnior, Lavras, Brasil

Estagiário (2009-2012)
Diretoria de Gestão da Tecnologia da Informação (DGTI -UFLA), Lavras, Brasil

INVESTIGATING THE MERGE CONFLICT LIFE-CYCLE TAKING THE SOCIAL DIMENSION INTO ACCOUNT

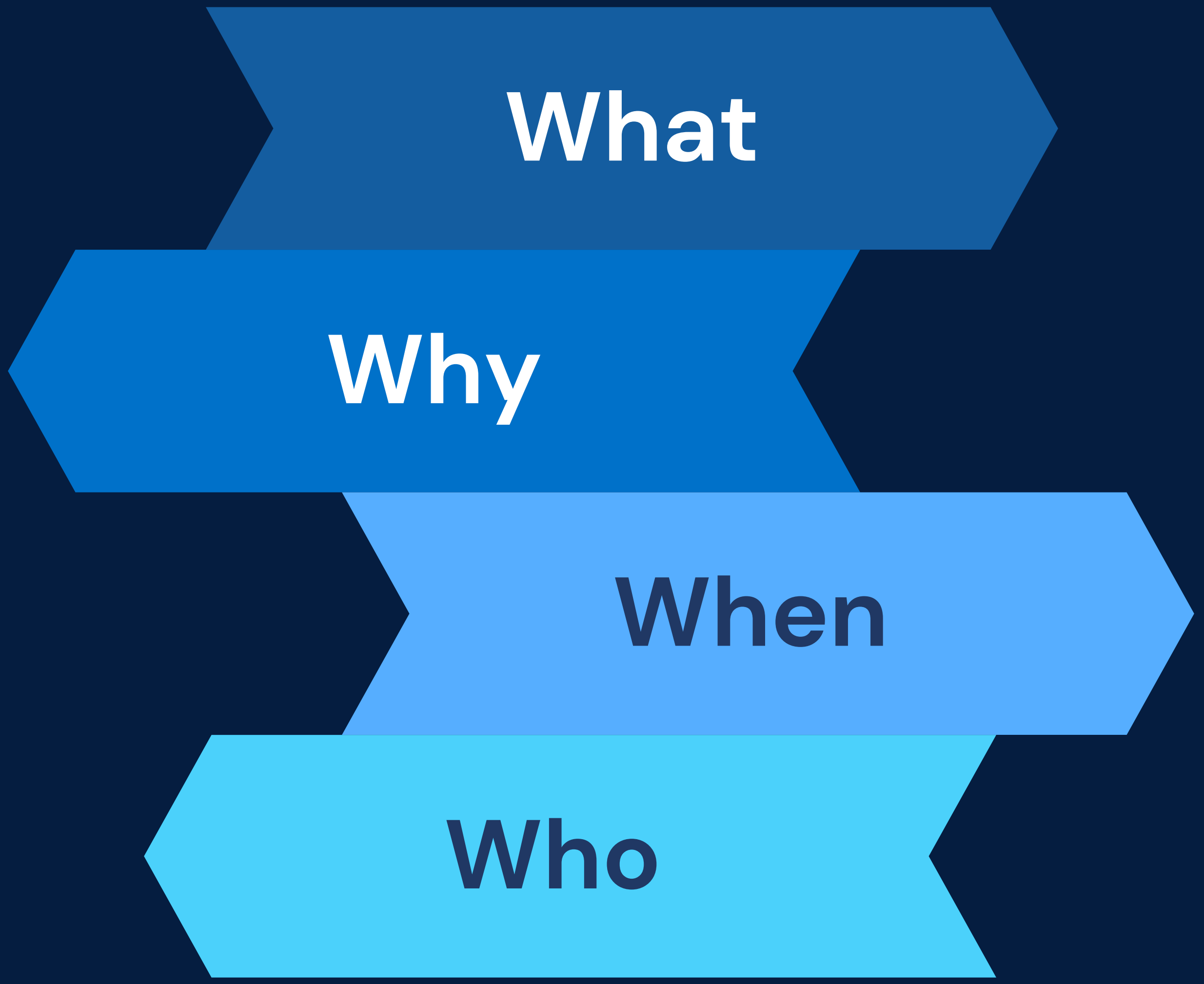
PhD Thesis: *Gustavo do Vale*

COLLABORATIVE SOFTWARE DEVELOPMENT

Success depends on the ability to coordinate social and technical assets

VERSION CONTROL SYSTEMS

Tools used to facilitate collaborative software development



Packages, files, chunks changed

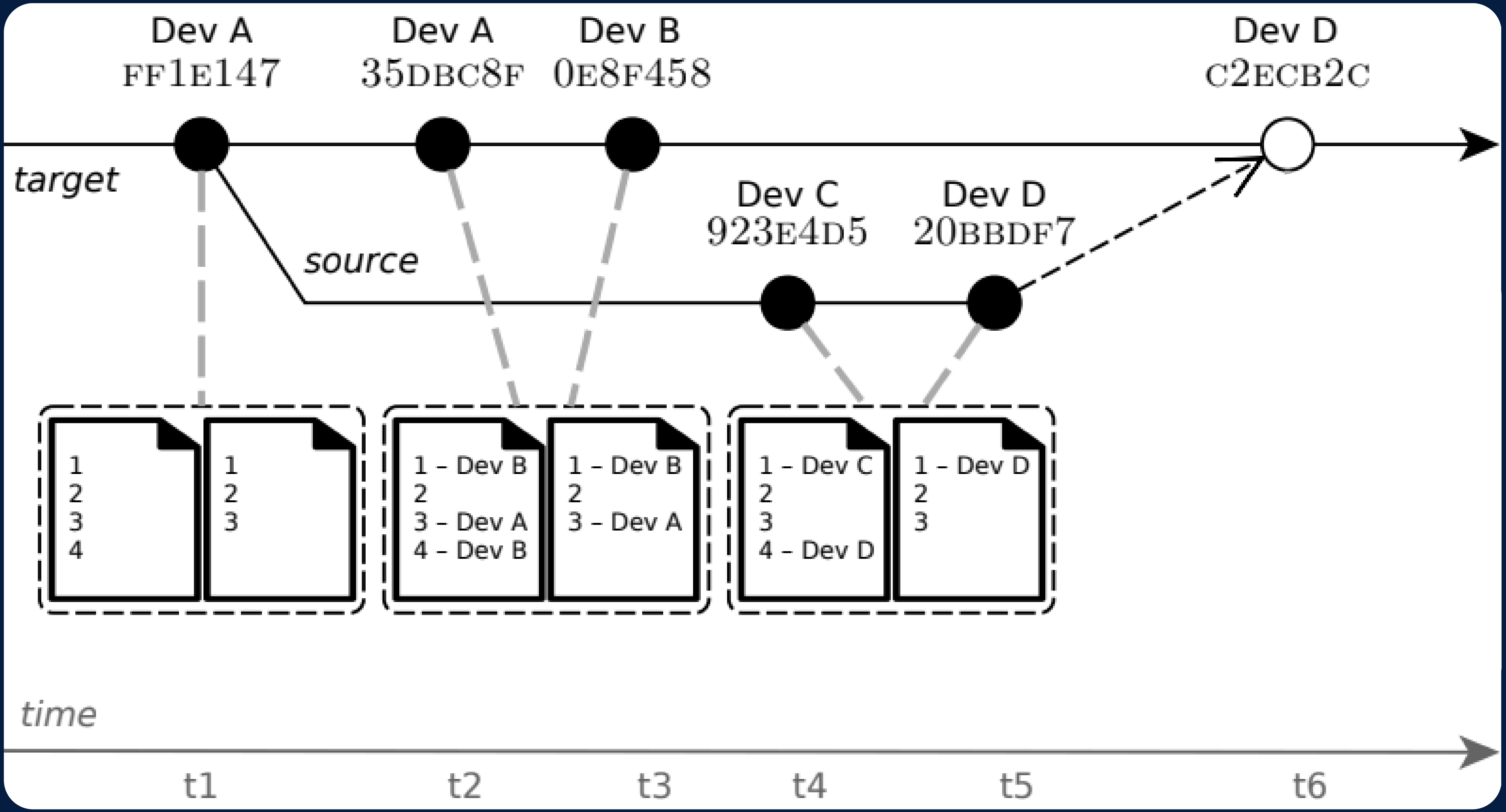
The reasons behind

The date changes happened

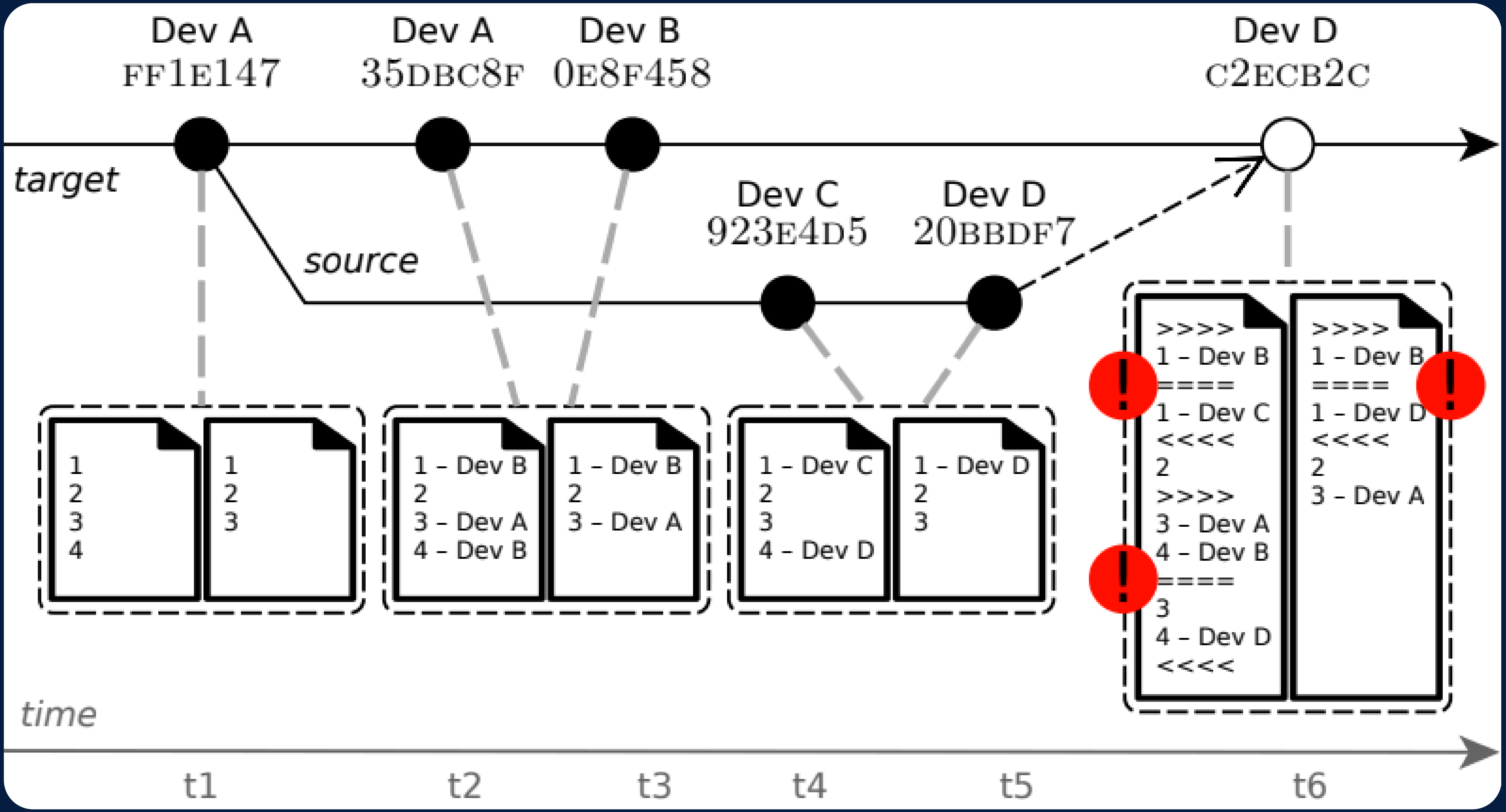
Developer(s) involved

Developers can browse and revert changes

MERGE SCENARIOS



MERGE CONFLICTS



ISSUES DUE TO MERGE CONFLICTS

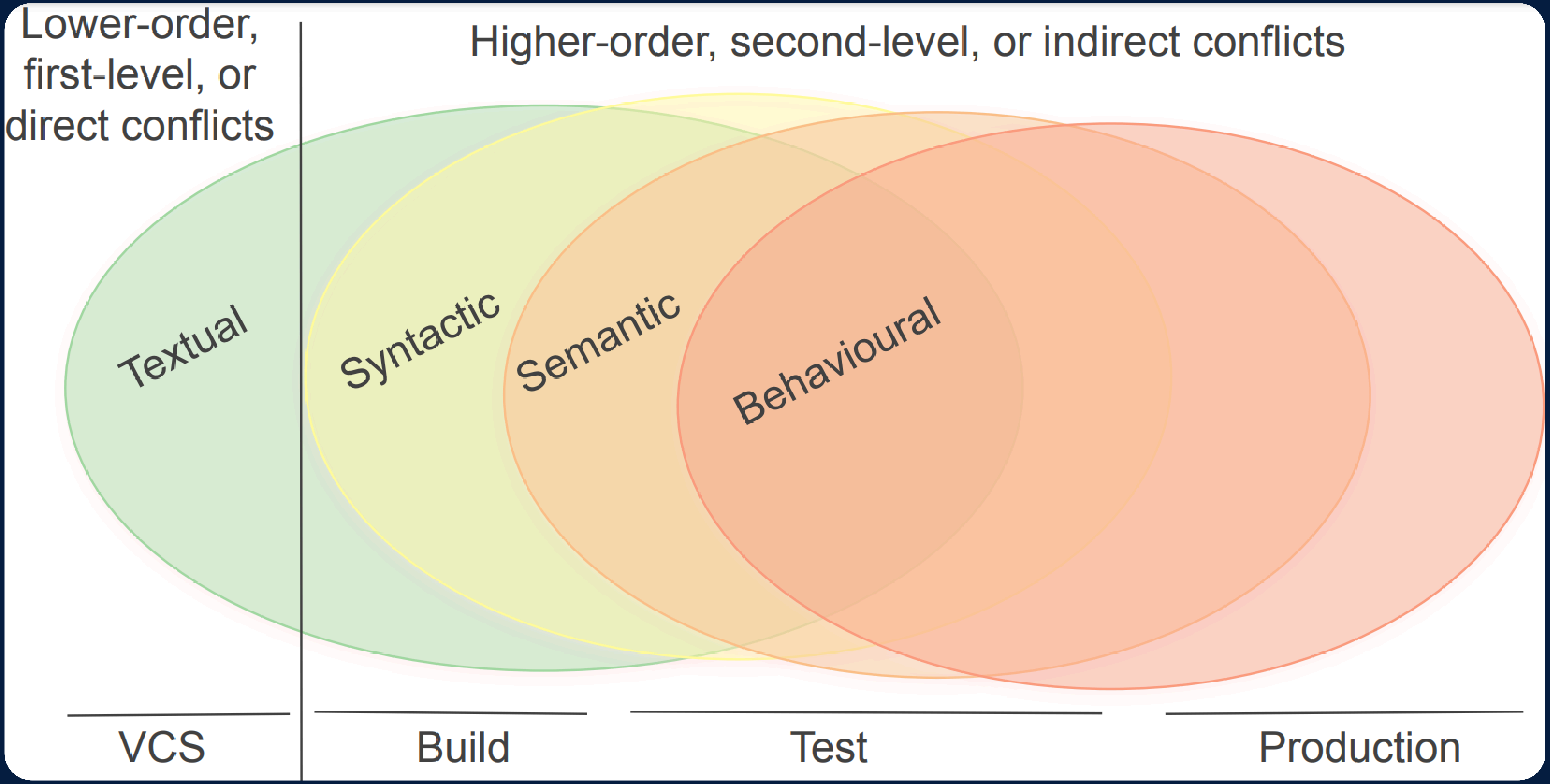
Developers distraction

Negative impact on team productivity, motivation, and keeping the schedule

Resolving them is a difficult, time-consuming, and error-prone

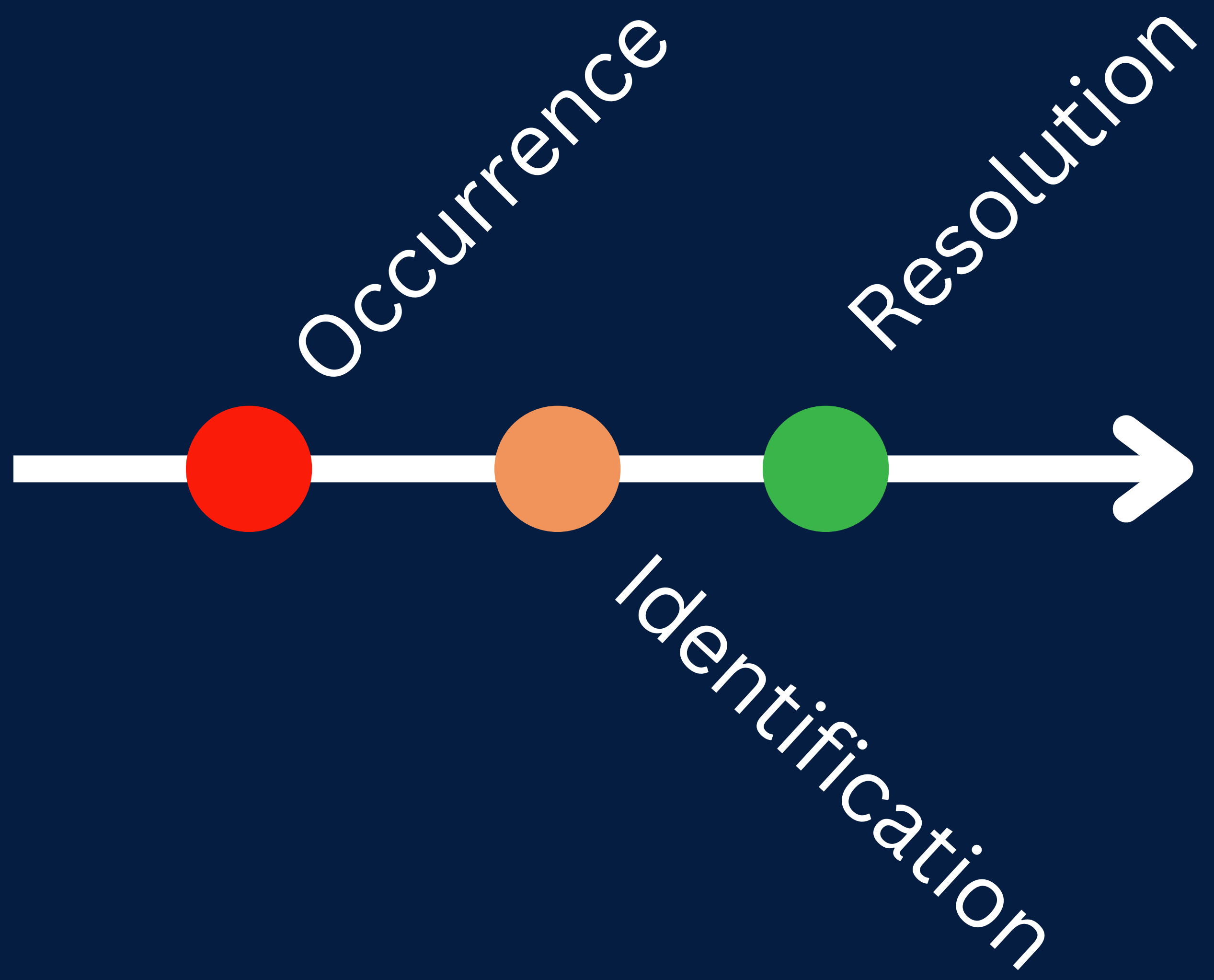
MERGE CONFLICT RATE AND TYPES

Conflict rate varies from 0% to 87.84% of merge scenarios


Lower-order, first-level, or direct conflicts
Higher-order, second-level, or indirect conflicts
Textual
Syntactic
Semantic
Behavioural
VCS
Build
Test
Production

STUDIES RELATED TO MERGE CONFLICTS

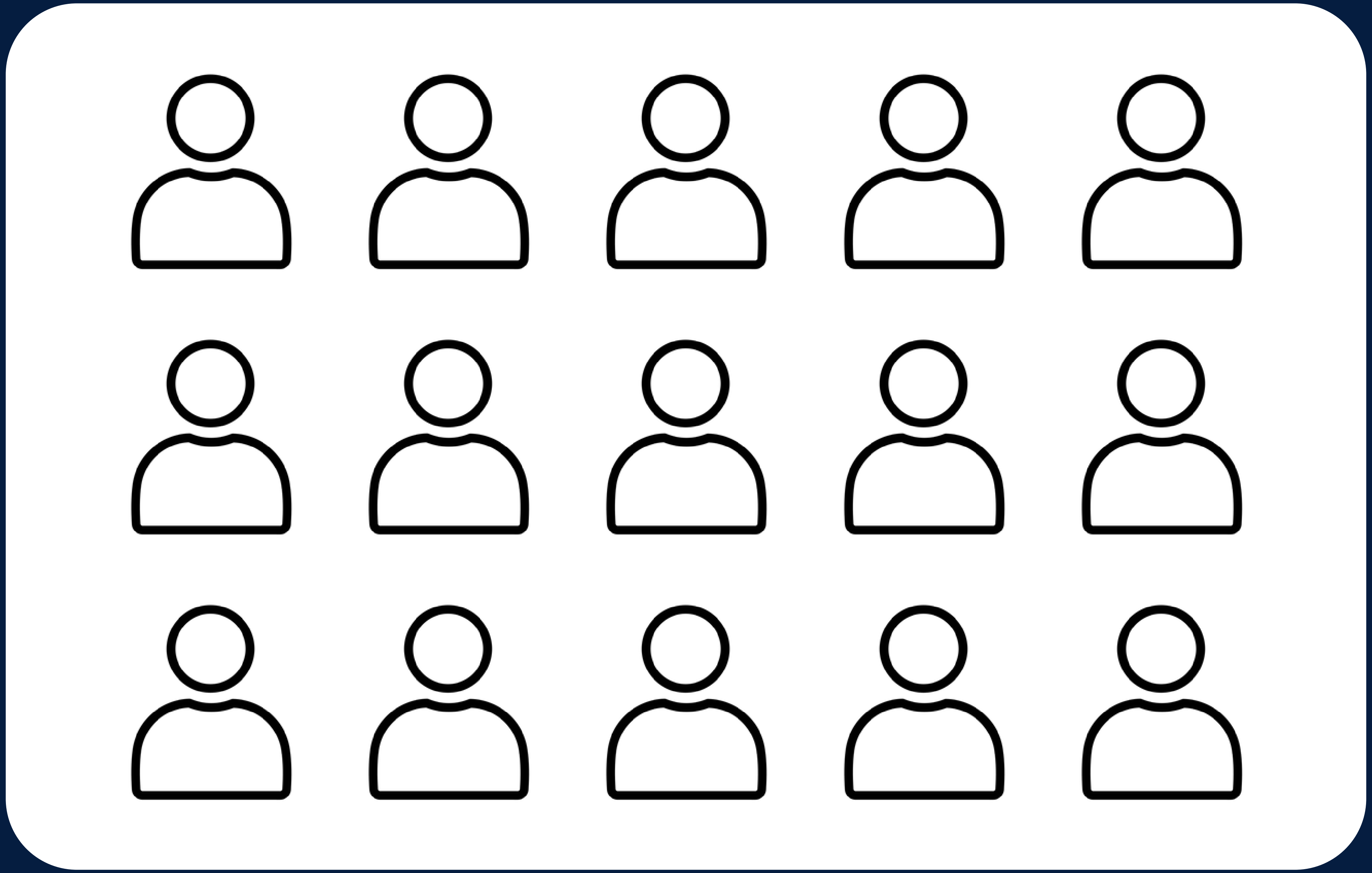
- Avoiding Merge Conflicts
- Merge Strategies
- Characterising Merge Conflicts
- Conflict Resolution



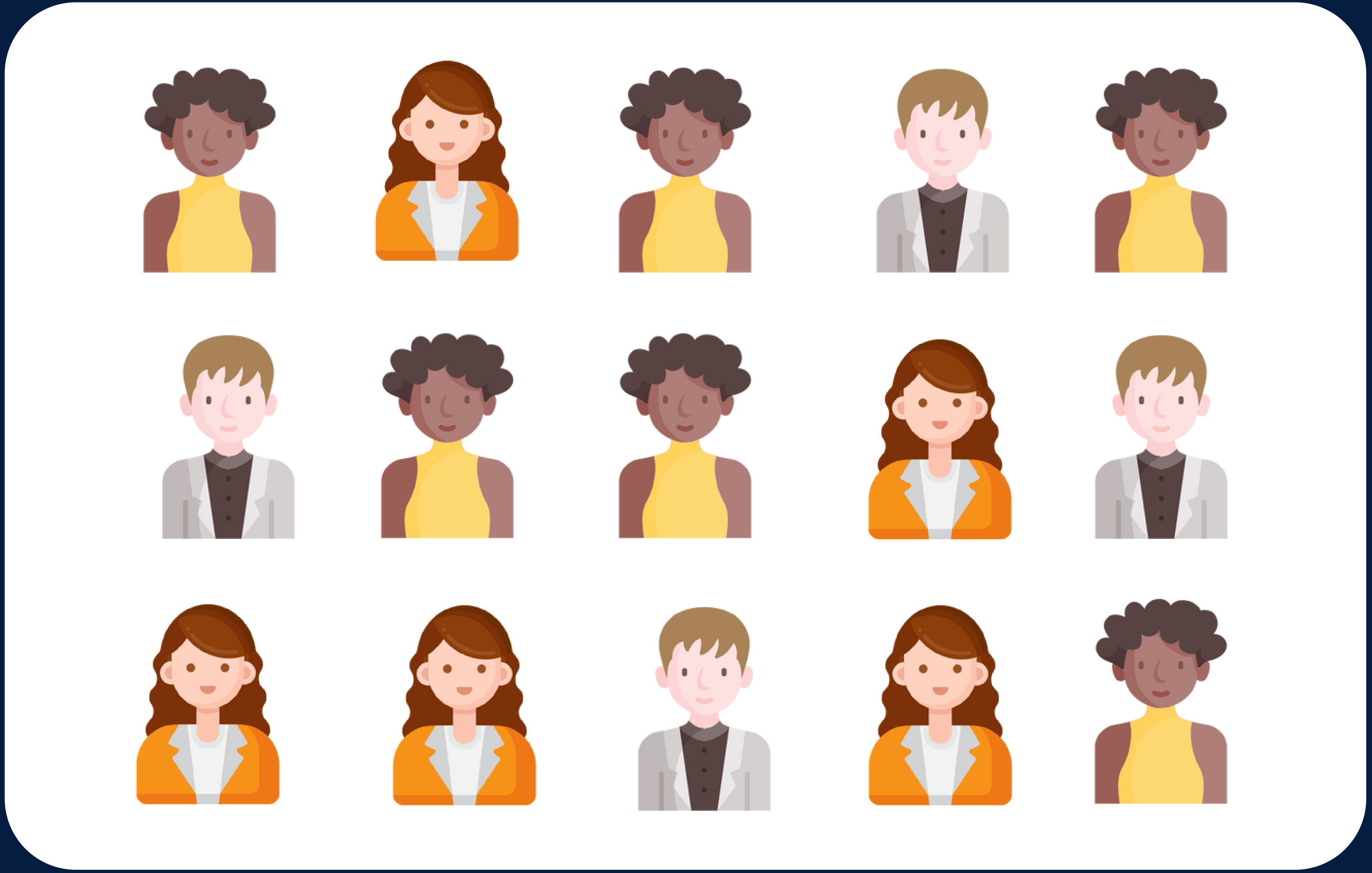
OPPORTUNITY

The social dimension is often ignored!!!

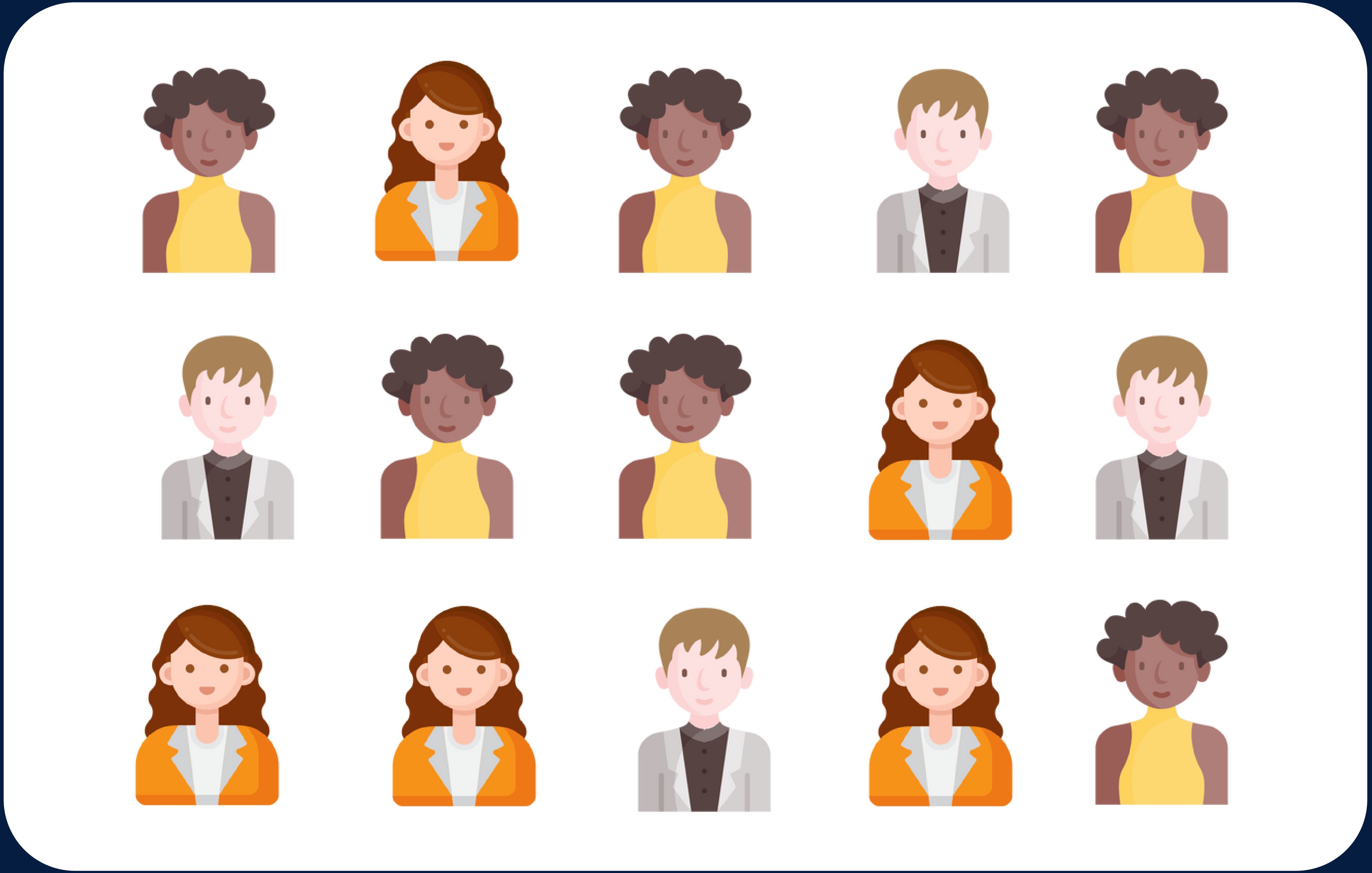
VISON 1 - DEVELOPER ROLES



VISON 1 - DEVELOPER ROLES



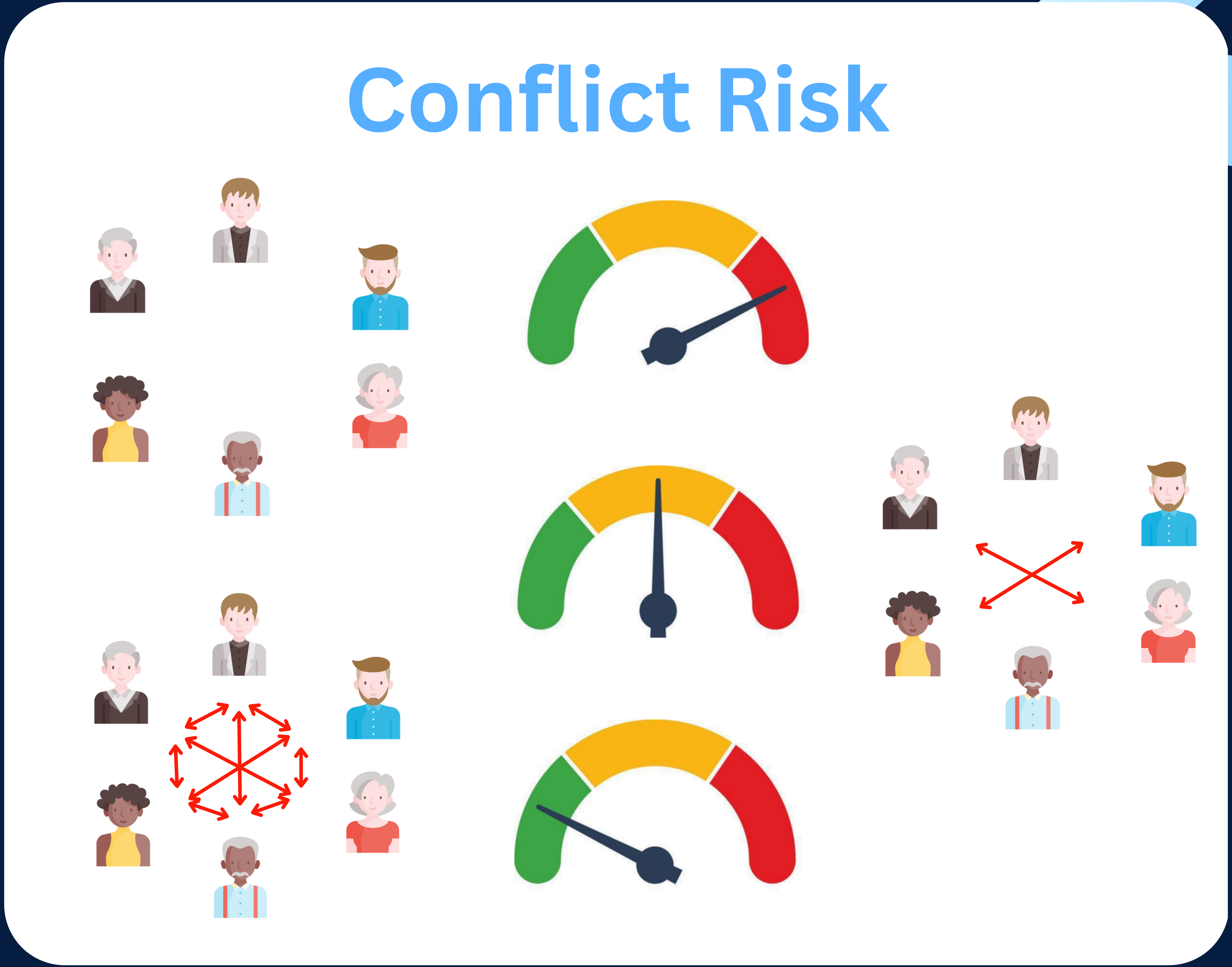
VISON 1 - DEVELOPER ROLES



Conflict risk

VISON 2 - COMMUNICATION ACTIVITY

VISON 2 - COMMUNICATION ACTIVITY



THESIS GOAL

Understand the role the social dimension plays in the merge conflict life-cycle

EMPIRICAL STUDIES

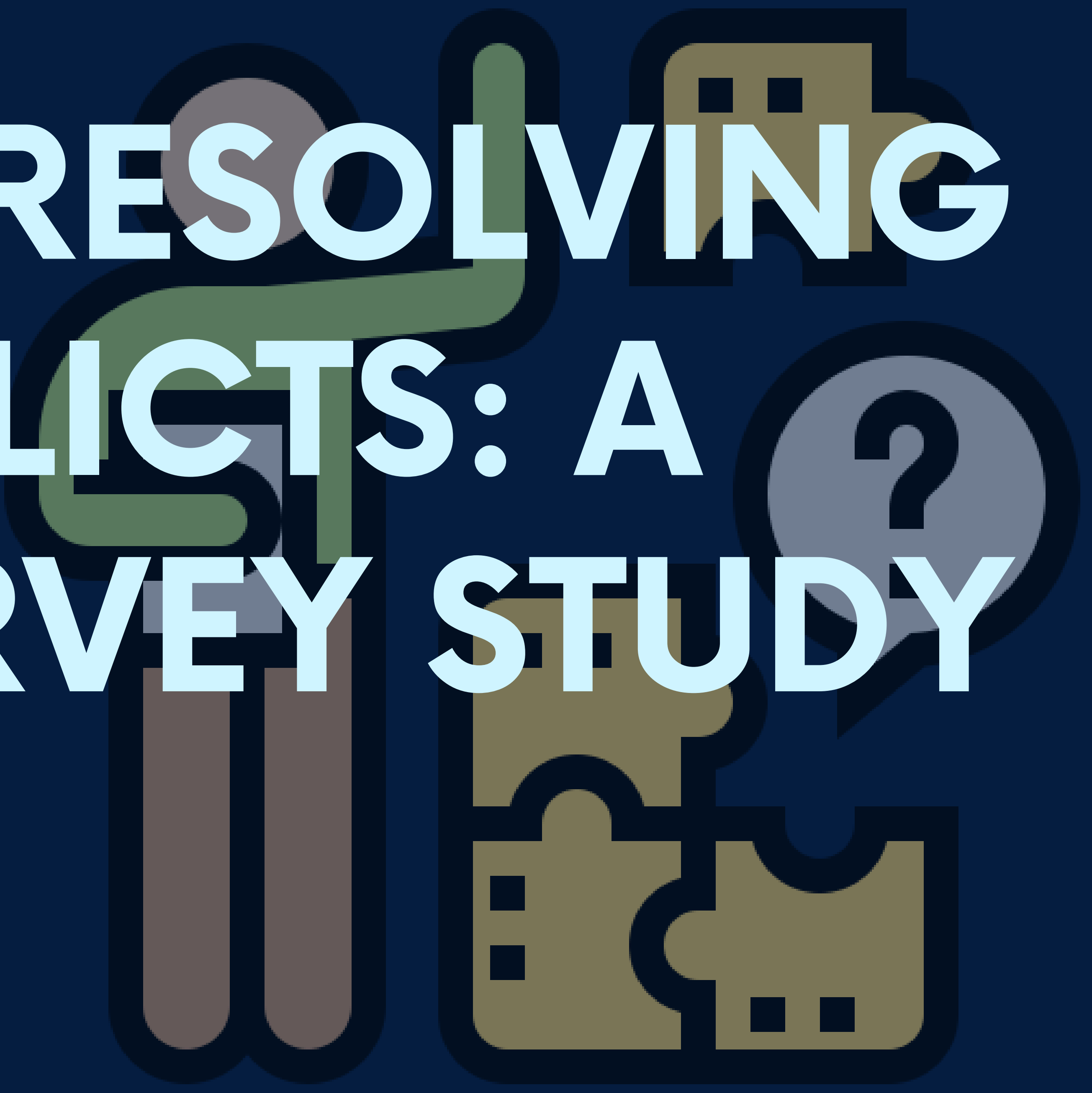
1. On the Relation between GitHub Communication Activity and Merge Conflicts

2. Predicting Merge Conflicts Considering Social and Technical Assets

3. Behind Developer Contributions on Conflicting Merge Scenarios

4. Challenges of Resolving Merge Conflicts: A Mining and Survey Study

CHALLENGES OF RESOLVING MERGE CONFLICTS: A MINING AND SURVEY STUDY



4

4

Problem

The **understanding of challenges and strategies on the resolution of merge conflicts is limited** in practice

4

Motivation

An empirical study mining historical data may not only confirm and add nuances to previous findings but also pin down the **most impacting and recurring factors**.

These factors may either serve as **best practices for developers saving time on merge conflict resolution** or as guidelines for tool builders to better support practitioners.

4

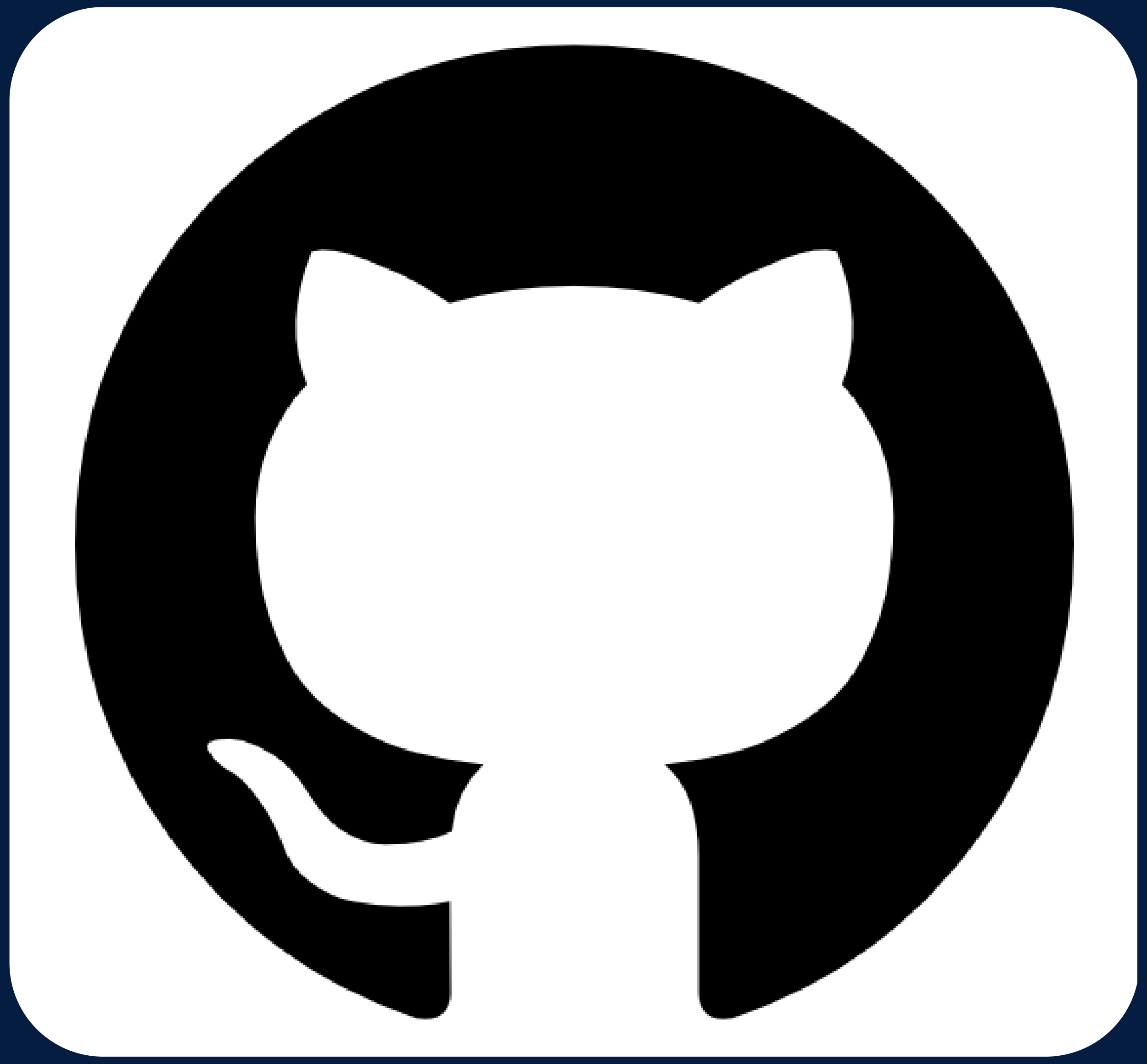
Study Steps

Which factors do make merge conflicts longer to resolve in practice?

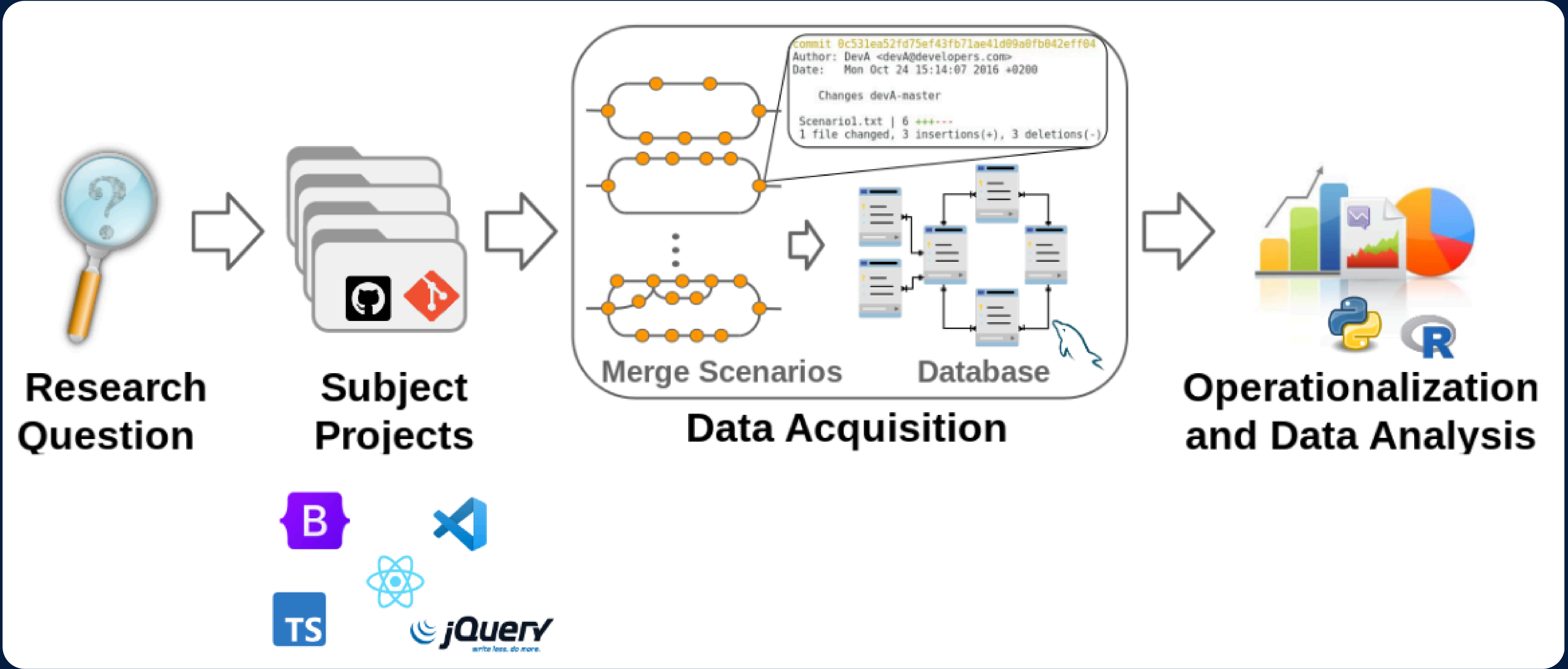
Mining study

Survey study

MINING STUDY



STUDY SETTINGS



Research Question

Subject Projects

```
commit 0c531ea52fd75ef43fb71ae41d09a0fb042eff04
Author: DevA <devA@developers.com>
Date:   Mon Oct 24 15:14:07 2016 +0200

    Changes devA-master

 Scenario1.txt | 6 +++---
 1 file changed, 3 insertions(+), 3 deletions(-)
```

Merge Scenarios

Database

Data Acquisition

Operationalization and Data Analysis

4CsNet

4

Experiment Variables

Dependent Variable

#SecondsToMerge

Independent Variables

CodeComplexity
#ConfChunks
#ConfFiles
%FormatingChanges
%IntegratorKnowledge

Directly related to merge conflicts

#Chunks
#Devs
#Files
#LoC

Indirectly related to merge conflicts

4

Statistical Analyses

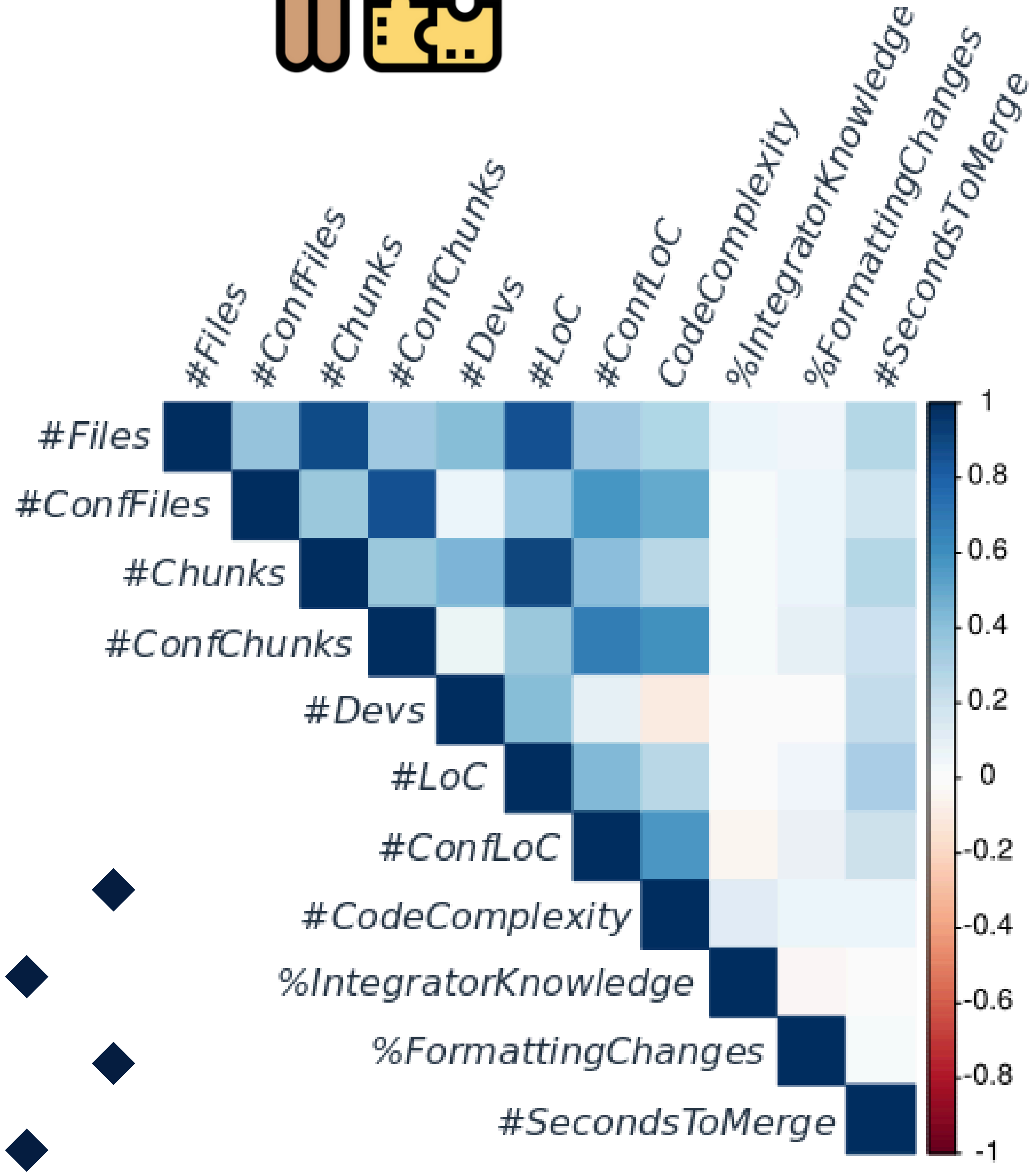
Correlation Analysis

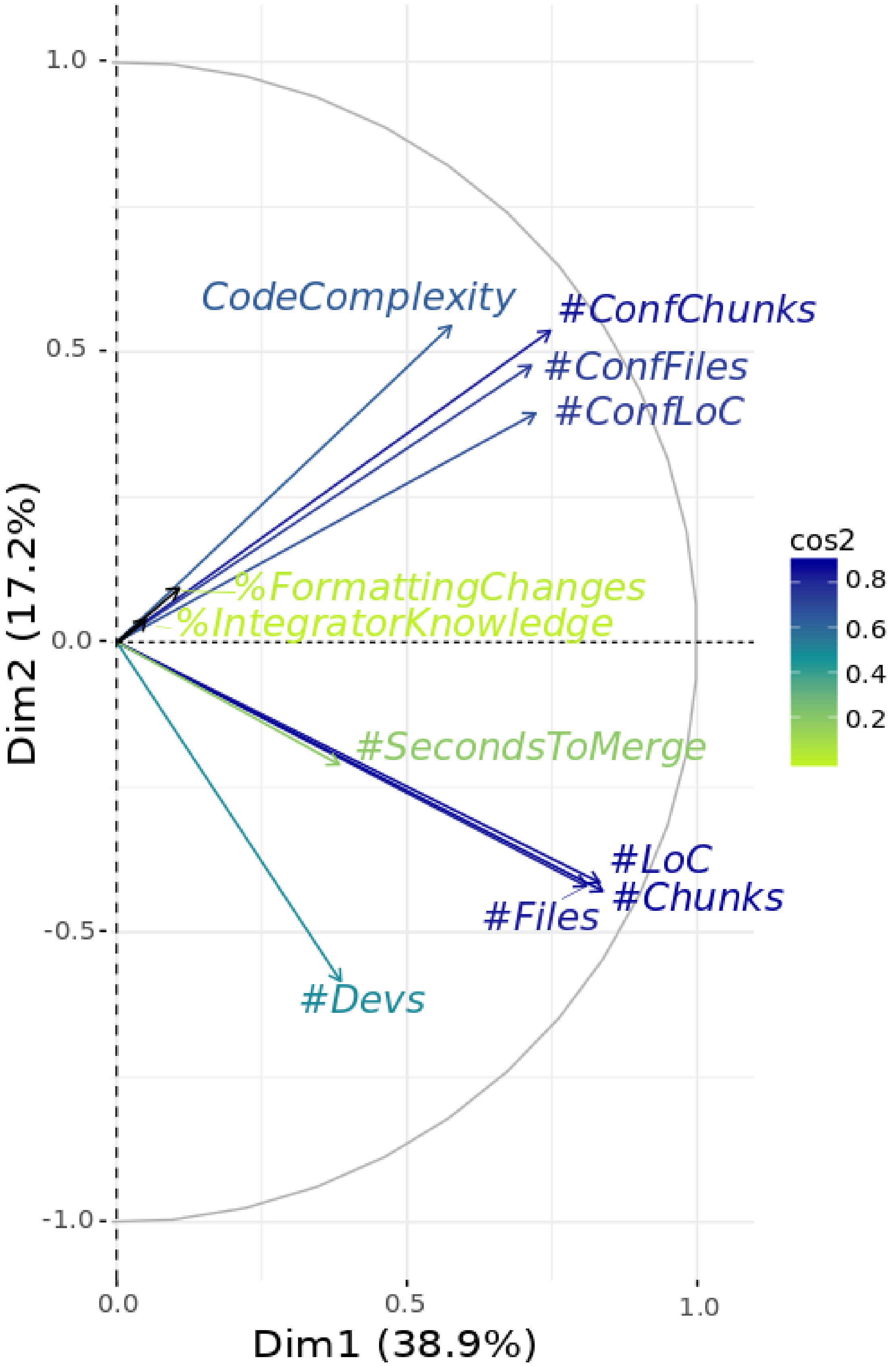
Multiple Regression Model Analysis

Effect-size Analysis

4

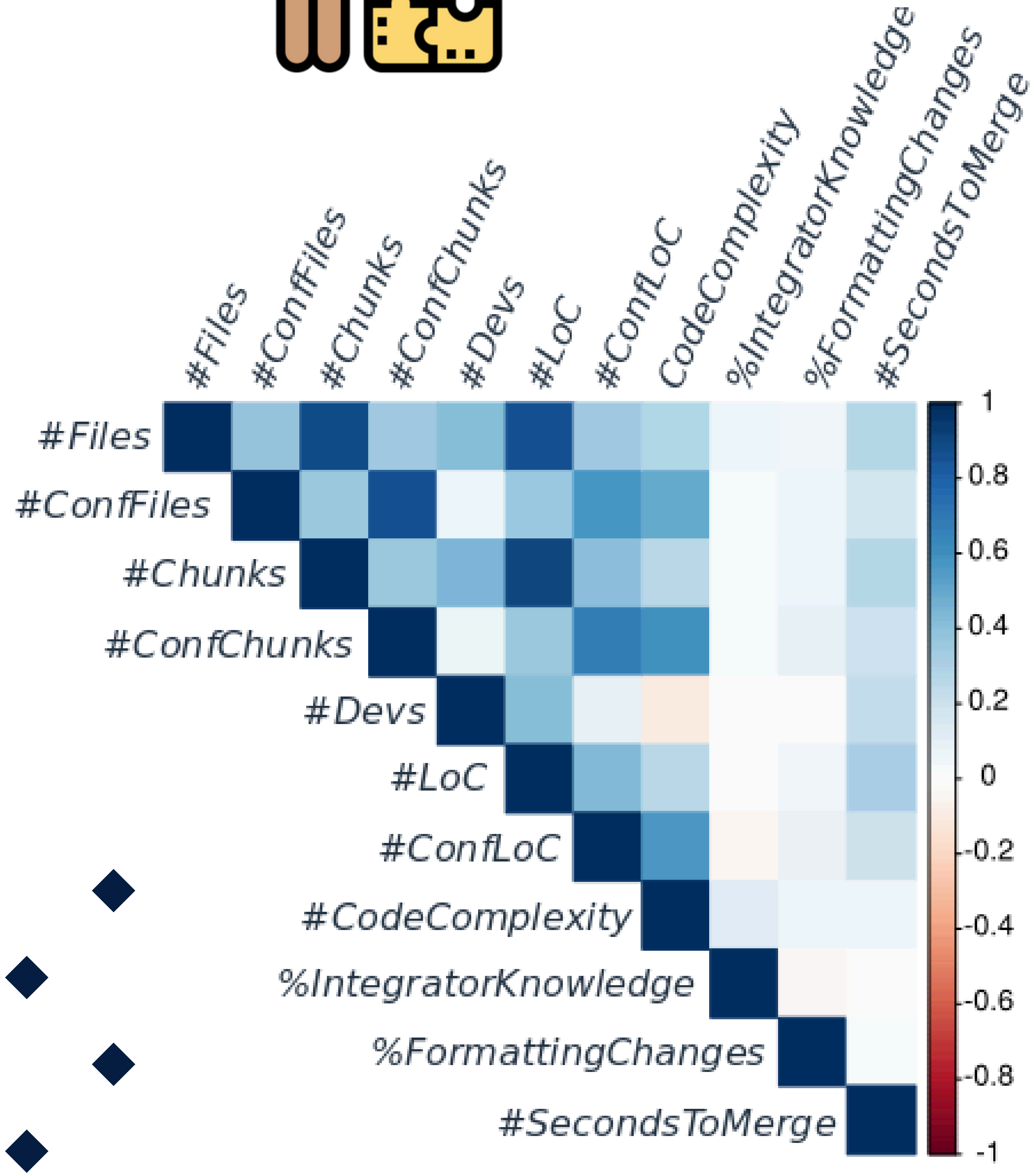
Correlation Analysis

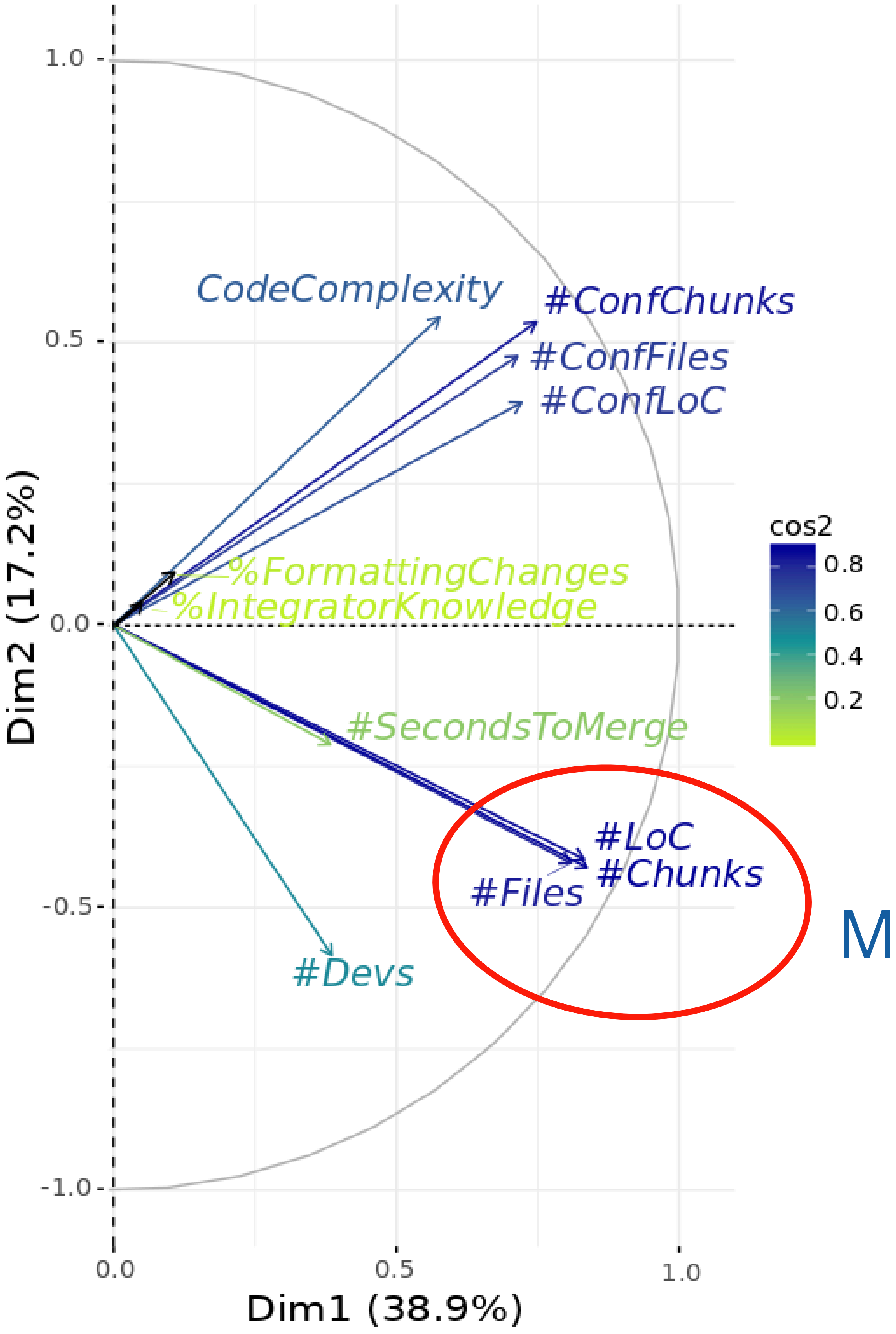




4

Correlation Analysis

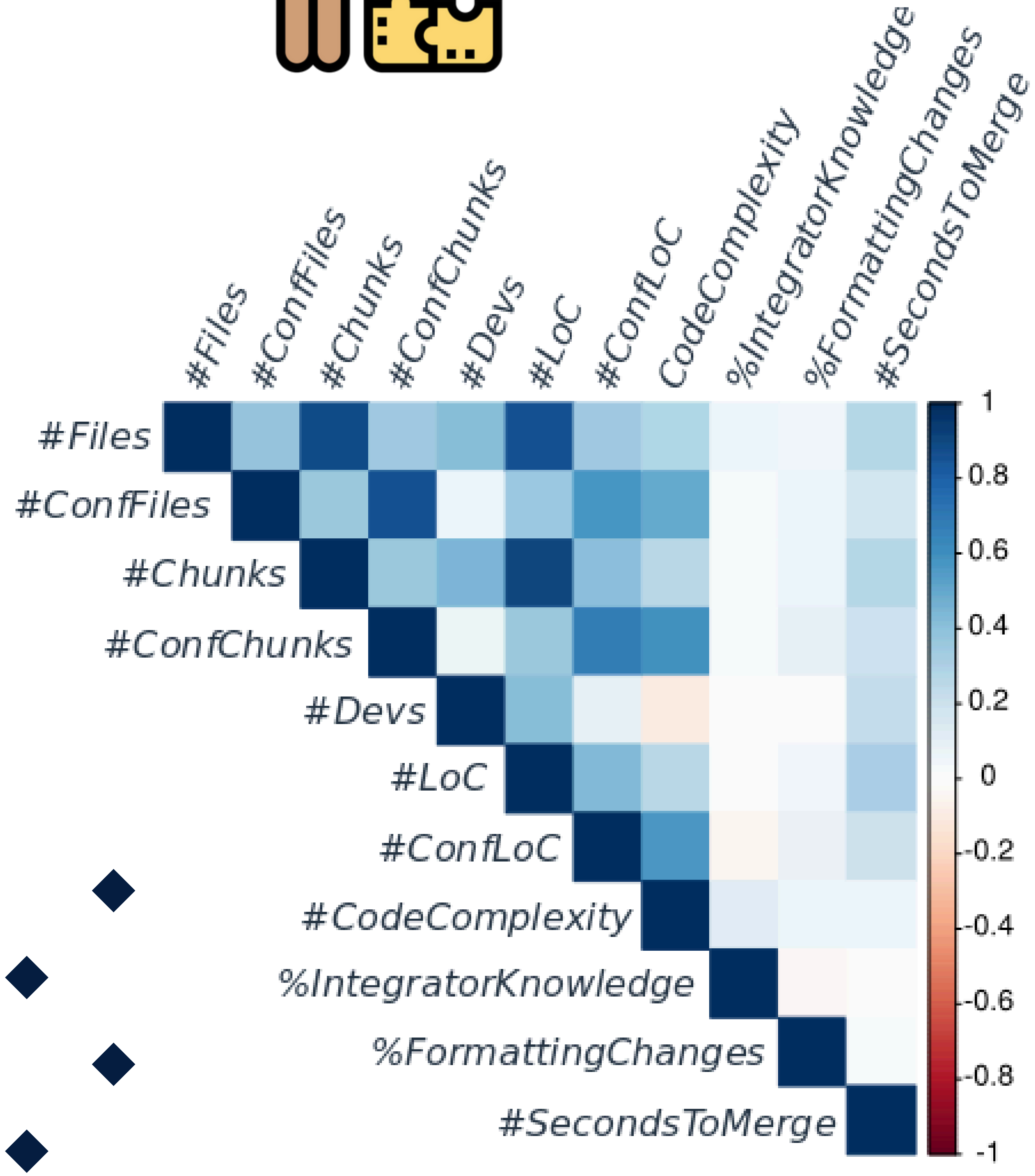


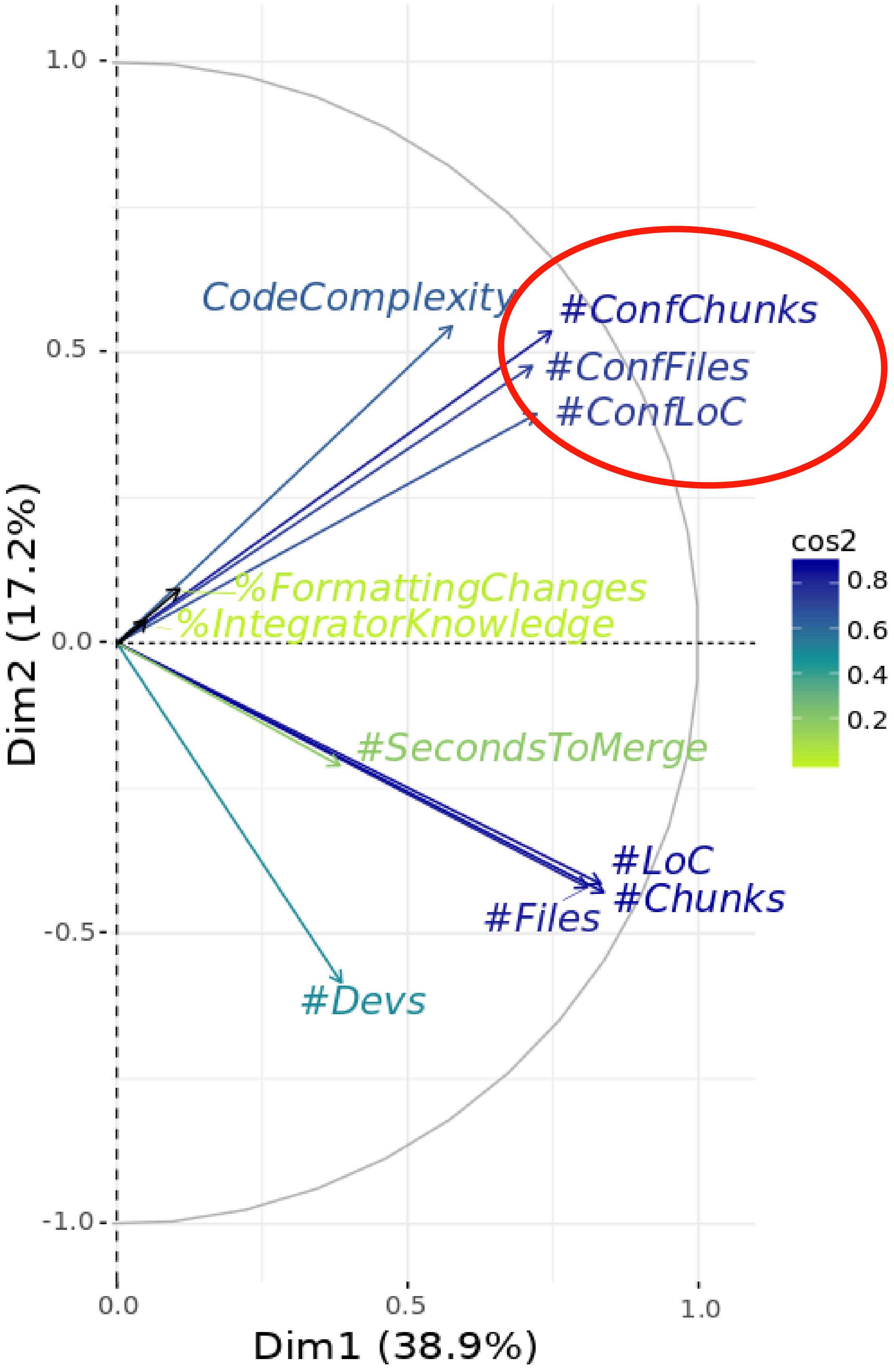


Merge Scenario Size

4

Correlation Analysis

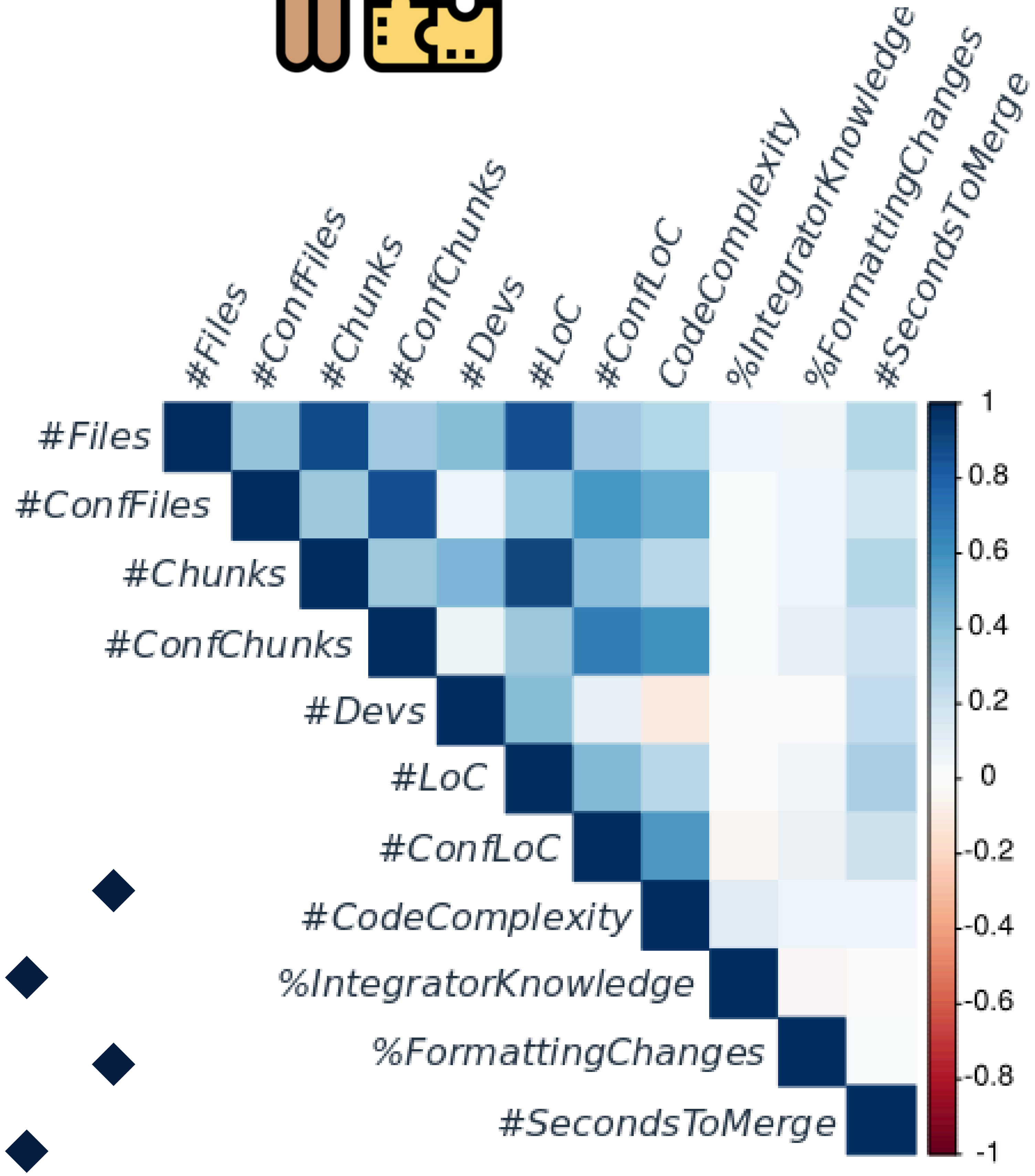


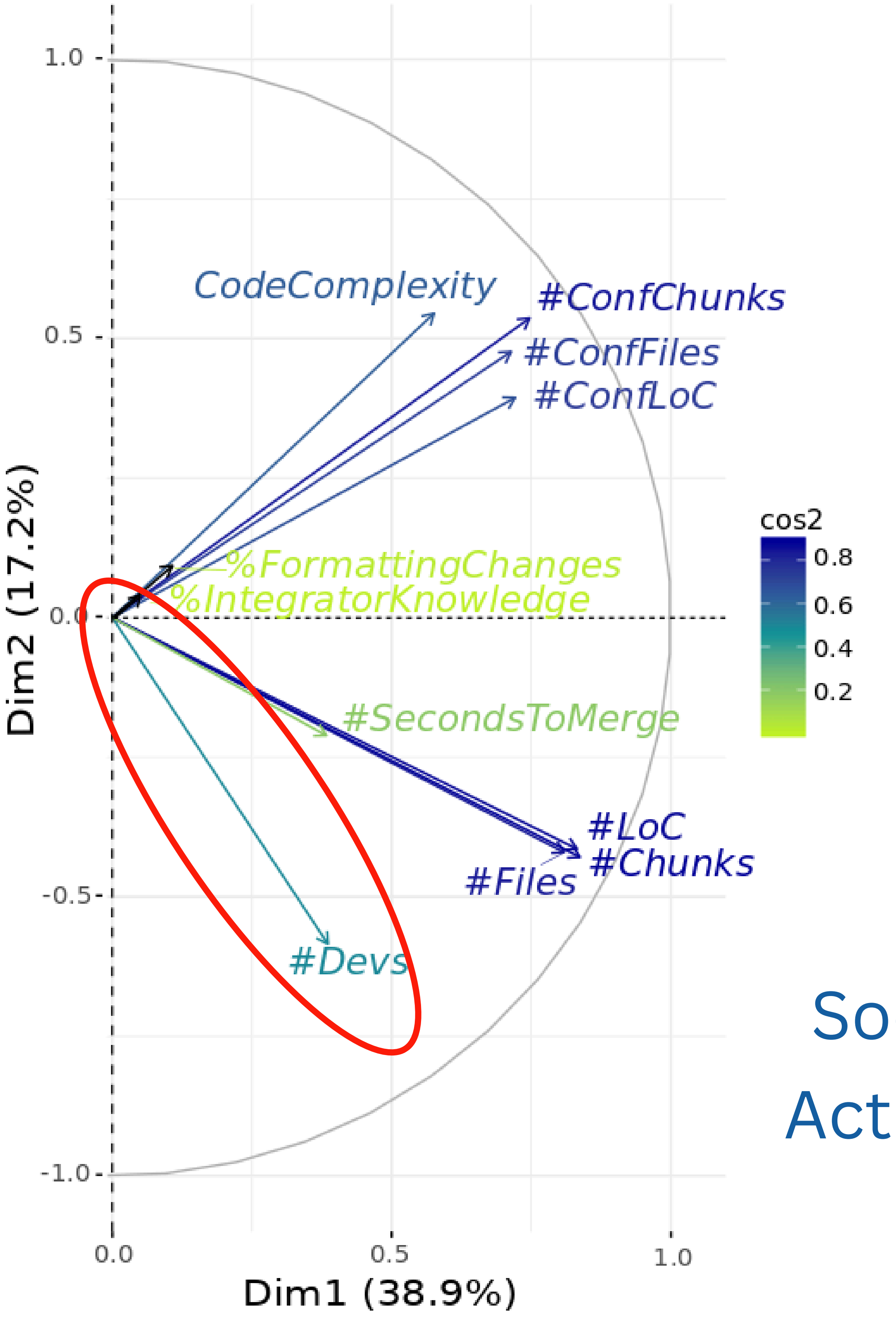


Conflict
Size

4

Correlation Analysis

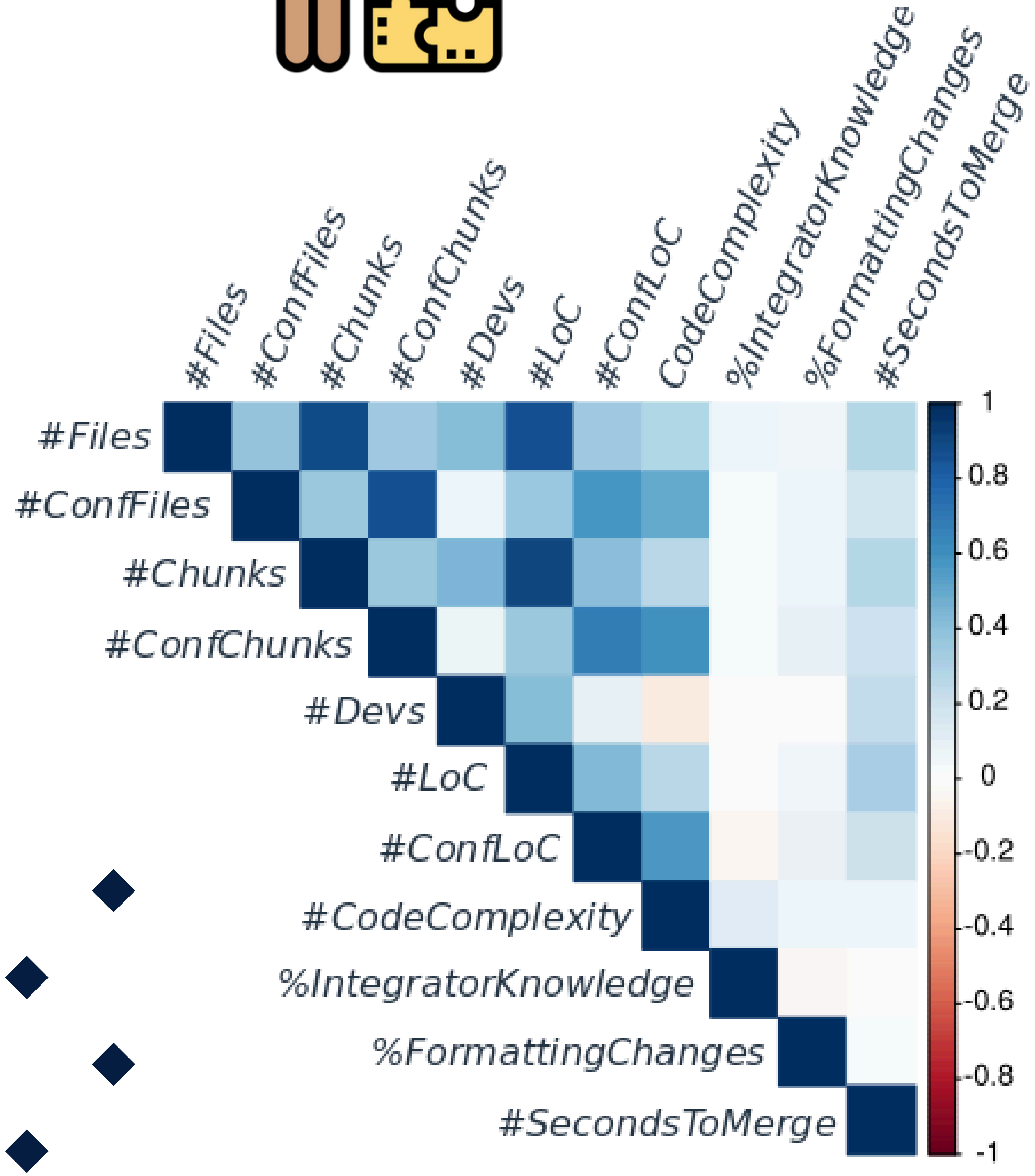


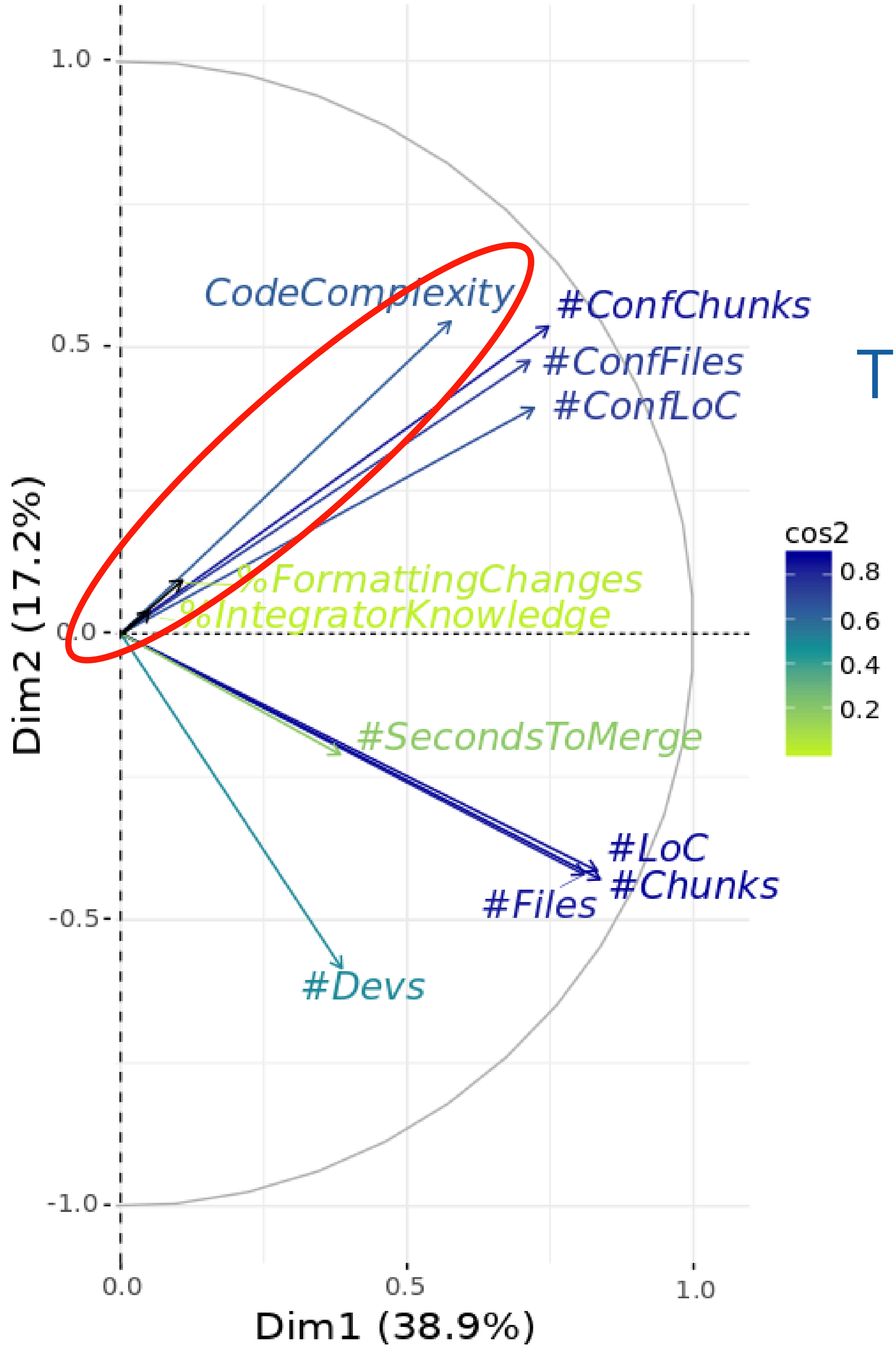


Social Activity

4

Correlation Analysis

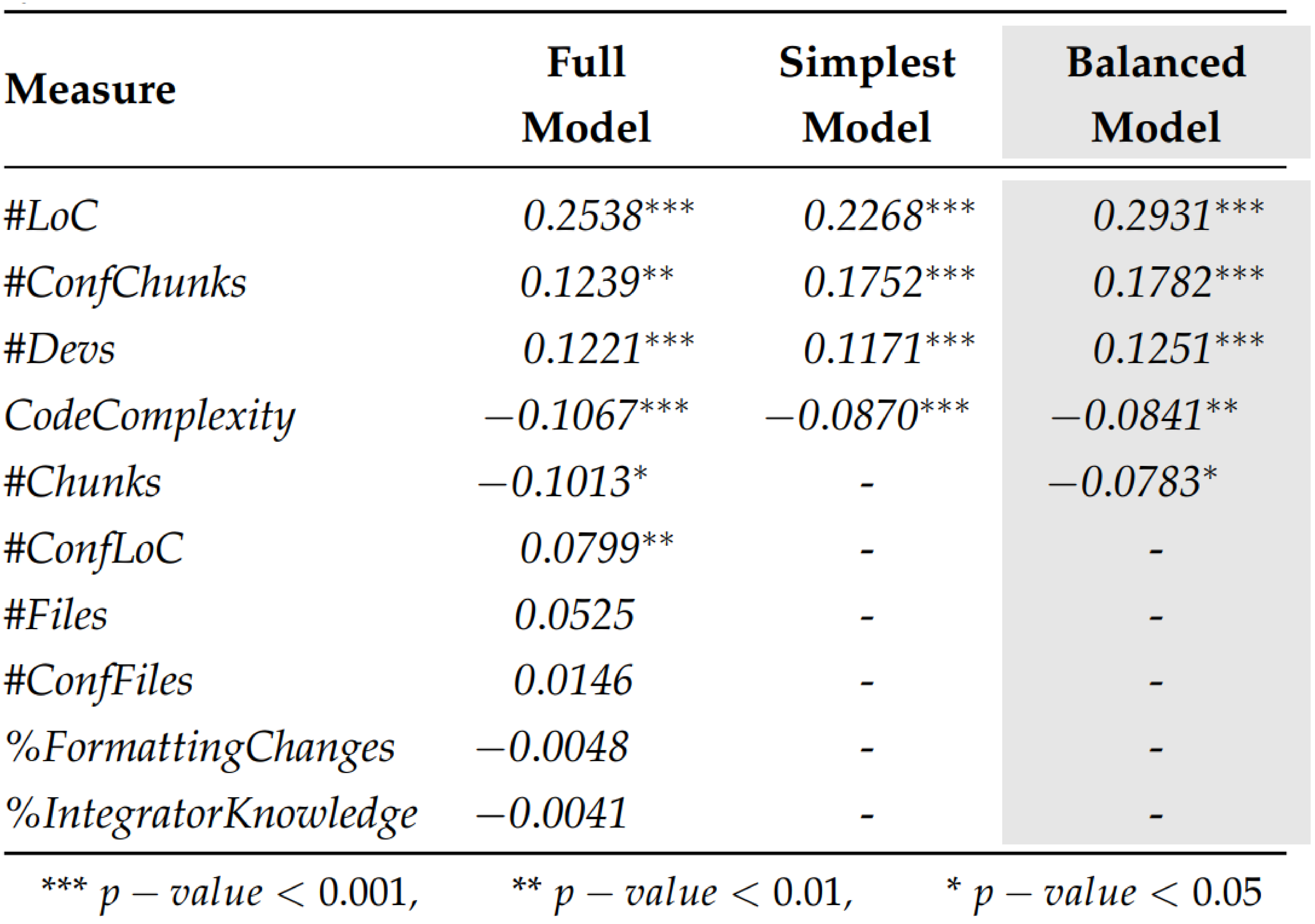




Type of Change

4

Multiple Regression Model Analysis

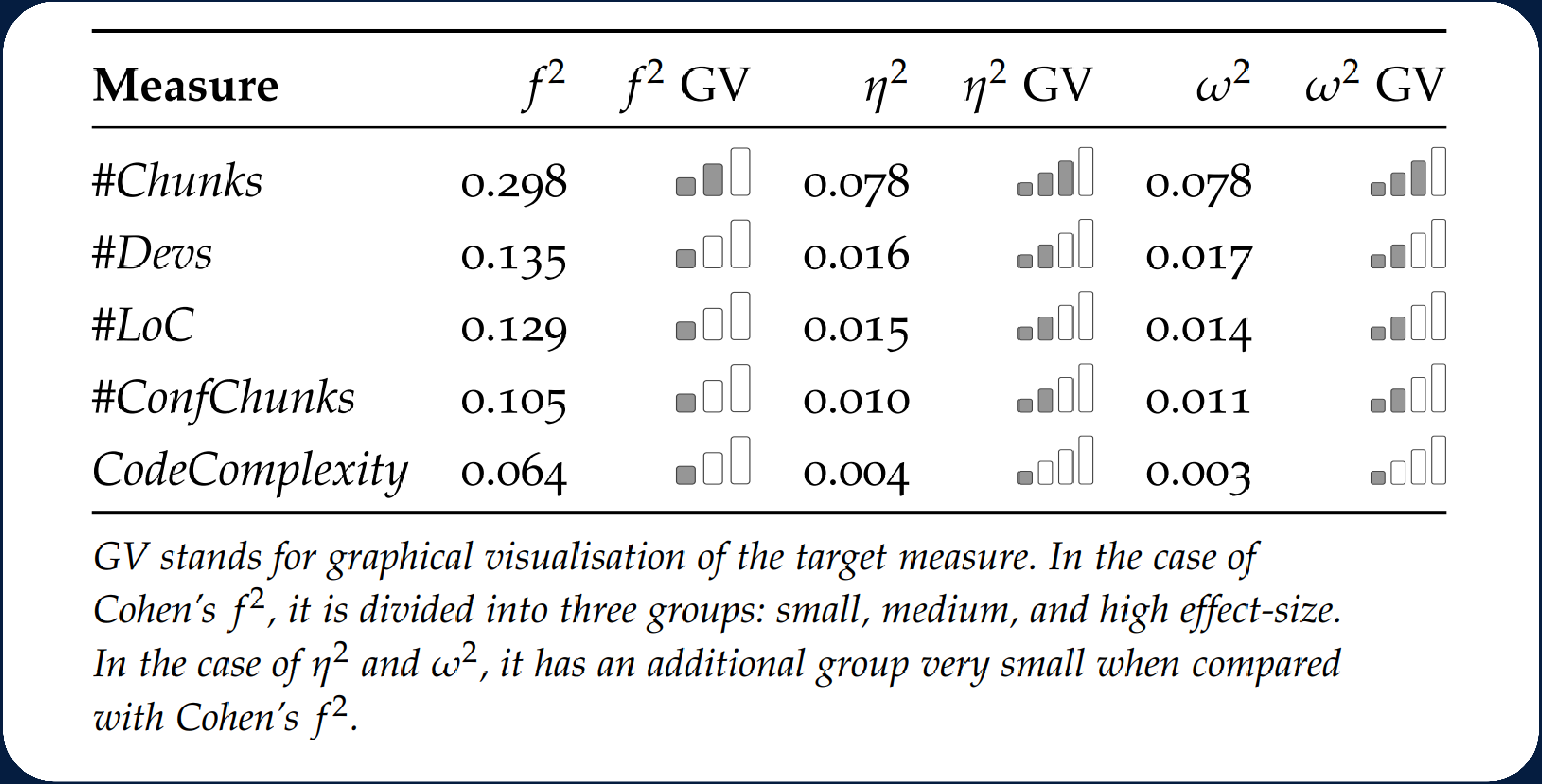


Measure	Full Model	Simplest Model	Balanced Model
#LoC	0.2538^{***}	0.2268^{***}	0.2931^{***}
#ConfChunks	0.1239^{**}	0.1752^{***}	0.1782^{***}
#Devs	0.1221^{***}	0.1171^{***}	0.1251^{***}
CodeComplexity	-0.1067^{***}	-0.0870^{***}	-0.0841^{**}
#Chunks	-0.1013^{*}	-	-0.0783^{*}
#ConfLoC	0.0799^{**}	-	-
#Files	0.0525	-	-
#ConfFiles	0.0146	-	-
%FormattingChanges	-0.0048	-	-
%IntegratorKnowledge	-0.0041	-	-

$^{***}\ p - value < 0.001$, $^{**}\ p - value < 0.01$, $^{*}\ p - value < 0.05$

4

Effect-Size Analysis

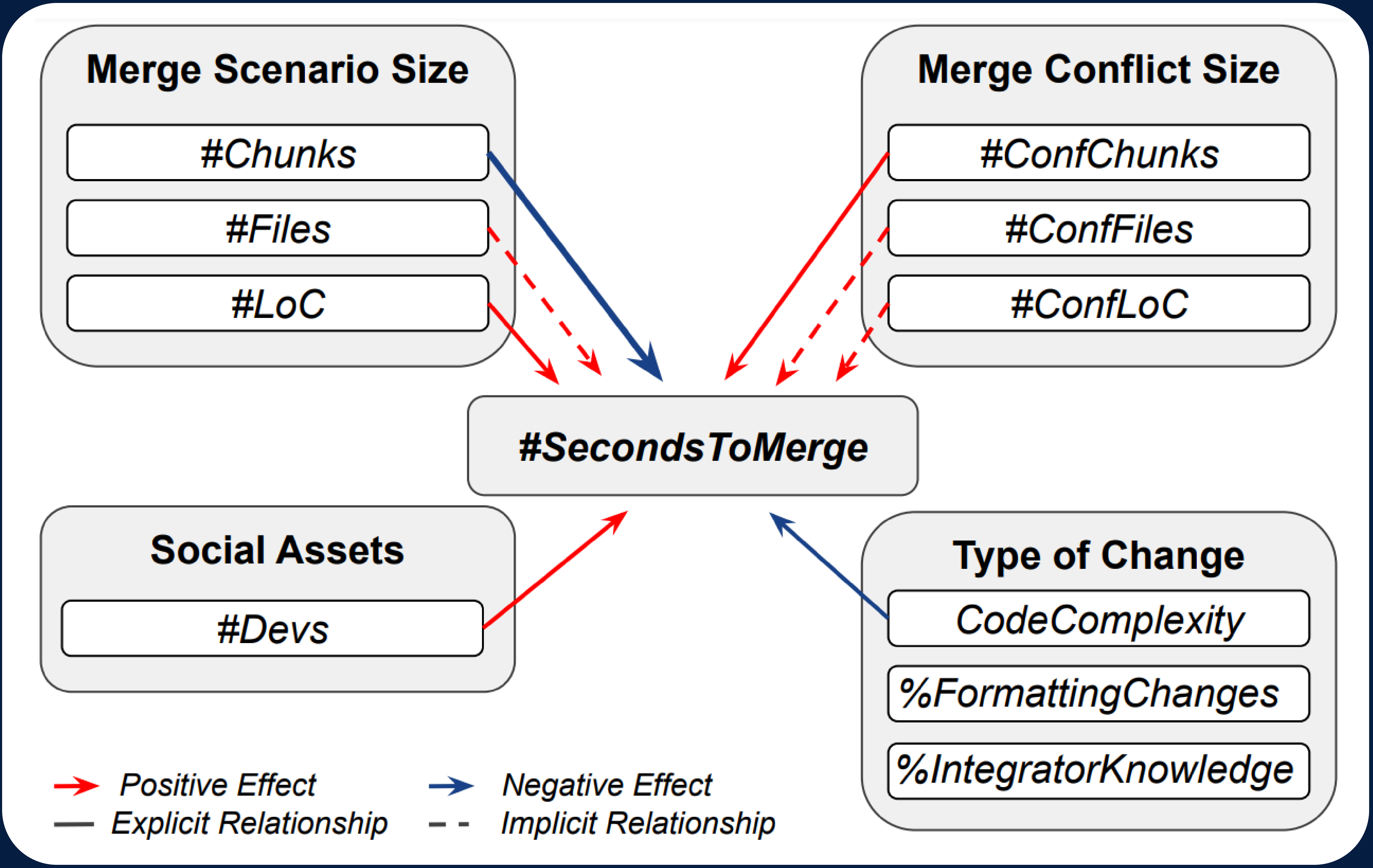


Measure	f^2	f^2 GV	η^2	η^2 GV	ω^2	ω^2 GV
#Chunks	0.298	■■□	0.078	■■■□	0.078	■■■□
#Devs	0.135	■□□	0.016	■■□□	0.017	■■□□
#LoC	0.129	■□□	0.015	■■□□	0.014	■■□□
#ConfChunks	0.105	■□□	0.010	■■□□	0.011	■■□□
CodeComplexity	0.064	■□□	0.004	■□□□	0.003	■□□□

GV stands for graphical visualisation of the target measure. In the case of Cohen's f^2, it is divided into three groups: small, medium, and high effect-size. In the case of η^2 and ω^2, it has an additional group very small when compared with Cohen's f^2.

4

Empirical Study Summary



SURVEY STUDY

4

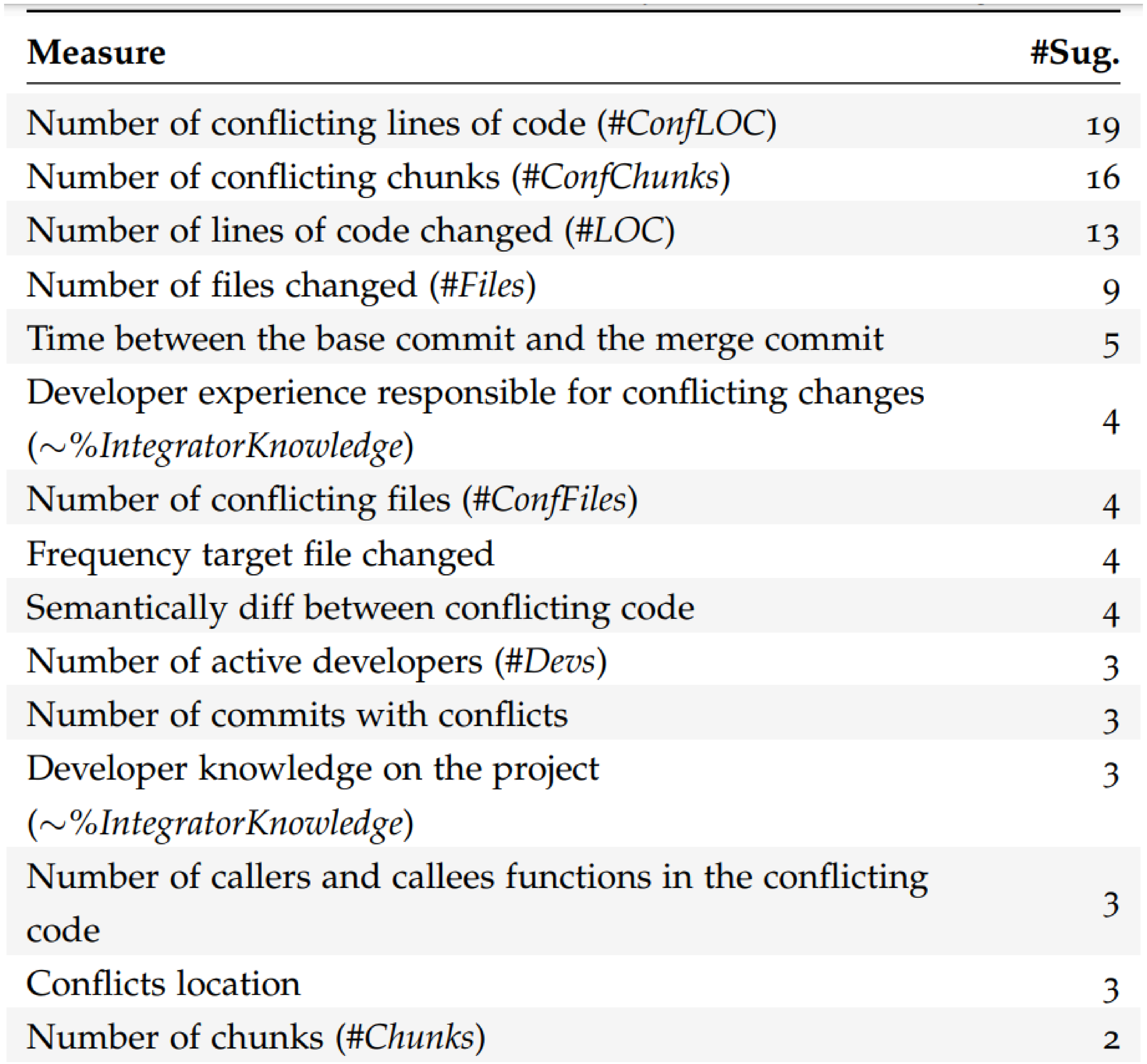
Survey Setup

- Seven-question survey
    - 1st and 7th open-ended
        - Grounded theory (open coding and axial coding)
    - Others close-ended (5-point Likert-type scale)
- Survey topics
    - **Understanding factors** that make merge conflict resolution longer (Q1)
    - **Confirming** empirical study **results** (Q2 - Q6)
    - Getting **remarkable experiences** and **challenges** when resolving merge conflicts (Q7)

4

Factors that Make Conflict Resolution Longer

140 answers and
25 measures

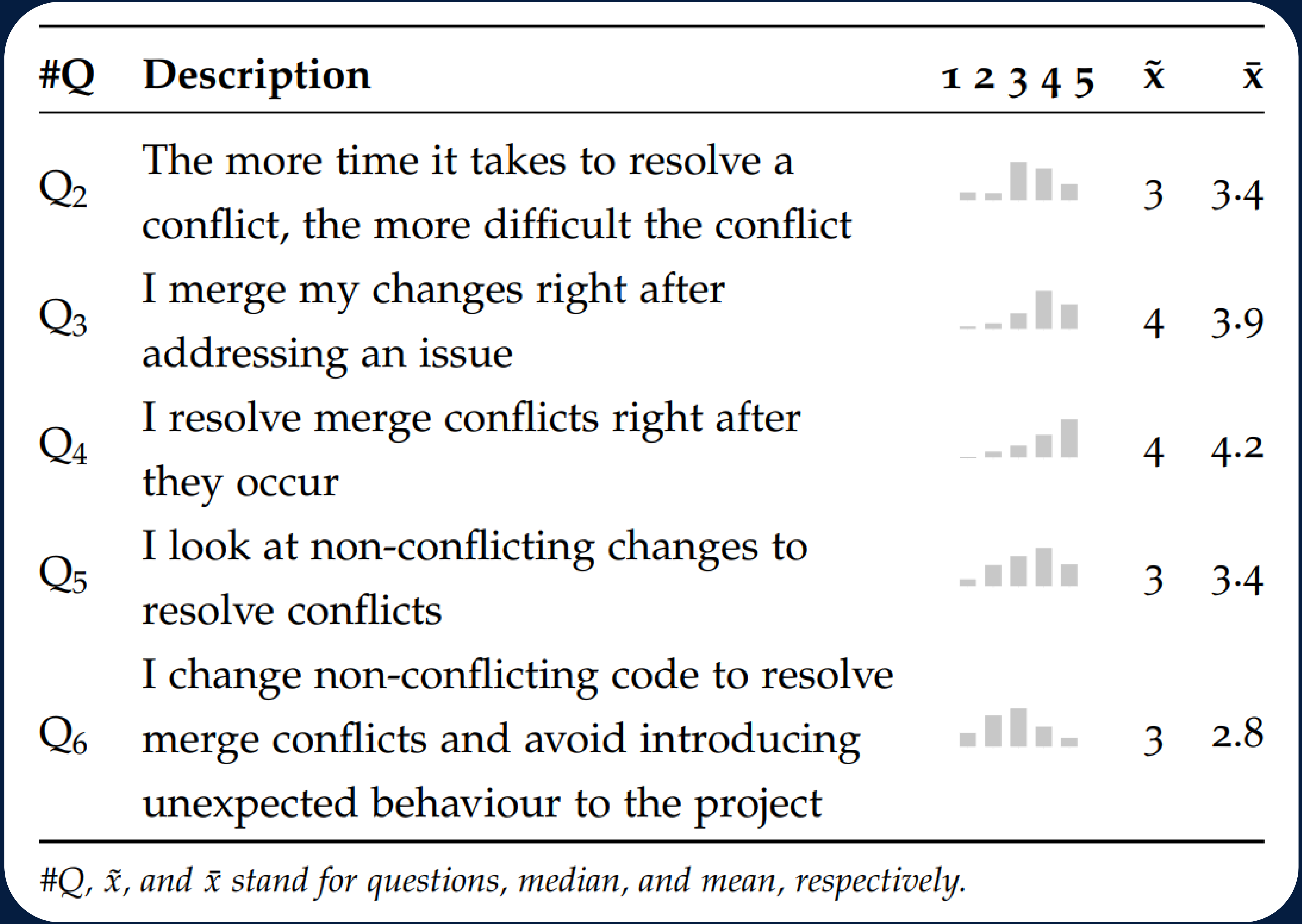


Measure	#Sug.
Number of conflicting lines of code (#*ConfLOC*)	19
Number of conflicting chunks (#*ConfChunks*)	16
Number of lines of code changed (#*LOC*)	13
Number of files changed (#*Files*)	9
Time between the base commit and the merge commit	5
Developer experience responsible for conflicting changes (∼%*IntegratorKnowledge*)	4
Number of conflicting files (#*ConfFiles*)	4
Frequency target file changed	4
Semantically diff between conflicting code	4
Number of active developers (#*Devs*)	3
Number of commits with conflicts	3
Developer knowledge on the project (∼%*IntegratorKnowledge*)	3
Number of callers and callees functions in the conflicting code	3
Conflicts location	3
Number of chunks (#*Chunks*)	2

•••

4

Confirming Empirical Study Results

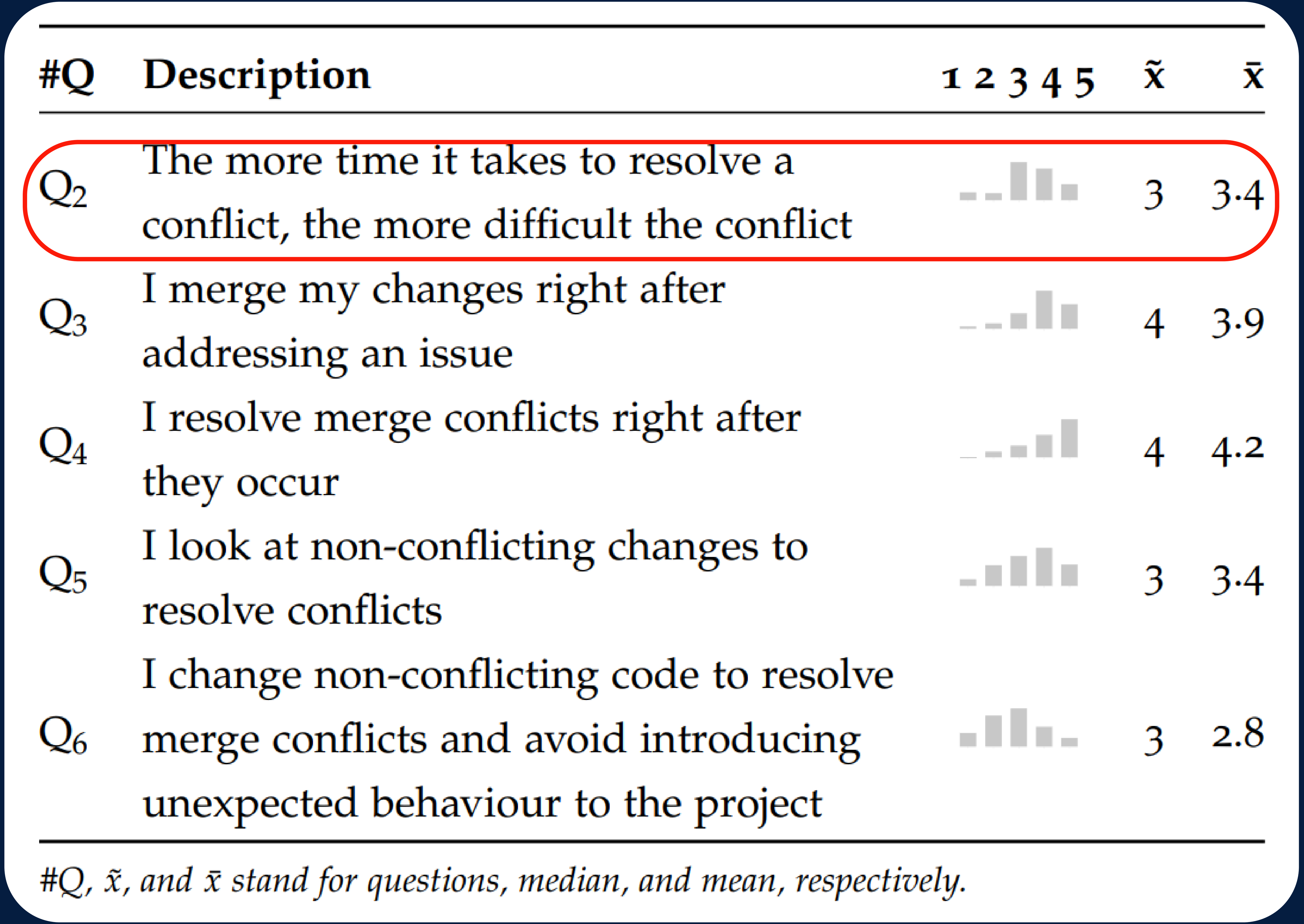


#Q	Description	1 2 3 4 5	x̃	x̄
Q2	The more time it takes to resolve a conflict, the more difficult the conflict		3	3.4
Q3	I merge my changes right after addressing an issue		4	3.9
Q4	I resolve merge conflicts right after they occur		4	4.2
Q5	I look at non-conflicting changes to resolve conflicts		3	3.4
Q6	I change non-conflicting code to resolve merge conflicts and avoid introducing unexpected behaviour to the project		3	2.8

#Q, x̃, and x̄ stand for questions, median, and mean, respectively.

4

Confirming Empirical Study Results

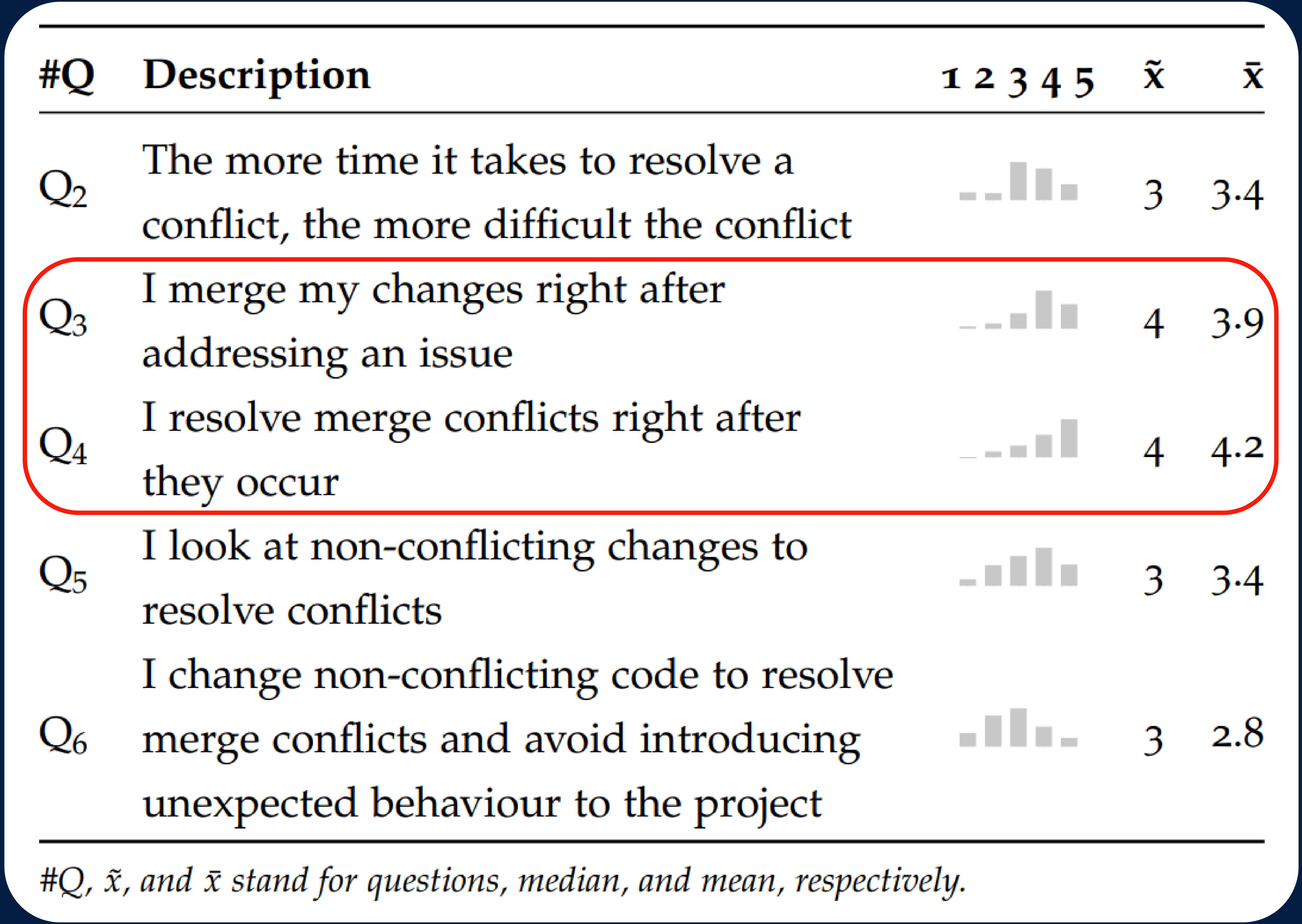


#Q	Description	1 2 3 4 5	x̃	x̄
Q_2	The more time it takes to resolve a conflict, the more difficult the conflict		3	3.4
Q_3	I merge my changes right after addressing an issue		4	3.9
Q_4	I resolve merge conflicts right after they occur		4	4.2
Q_5	I look at non-conflicting changes to resolve conflicts		3	3.4
Q_6	I change non-conflicting code to resolve merge conflicts and avoid introducing unexpected behaviour to the project		3	2.8

#Q, x̃, and x̄ stand for questions, median, and mean, respectively.

4

Confirming Empirical Study Results

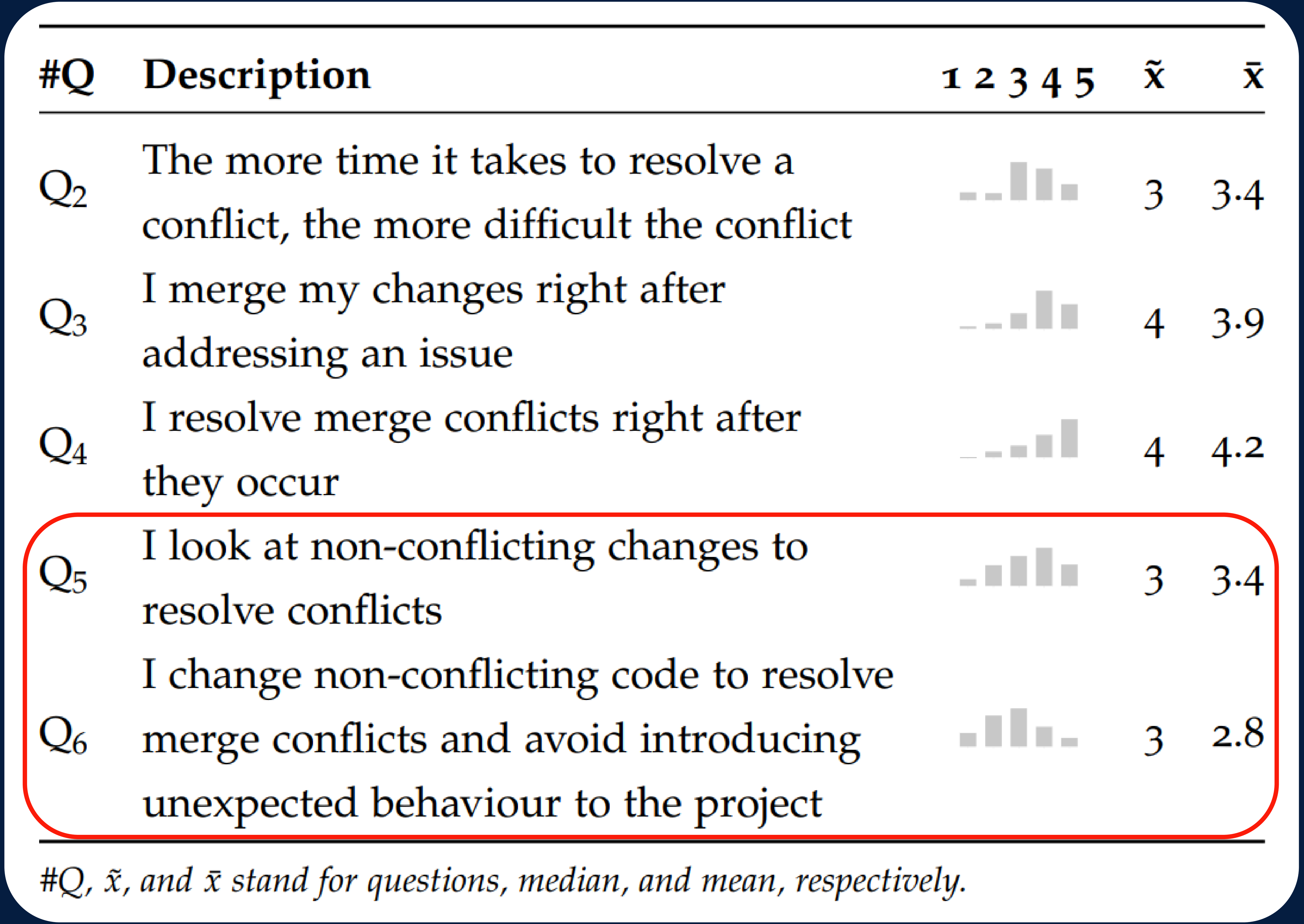


#Q	Description	1 2 3 4 5	x̃	x̄
Q2	The more time it takes to resolve a conflict, the more difficult the conflict		3	3.4
Q3	I merge my changes right after addressing an issue		4	3.9
Q4	I resolve merge conflicts right after they occur		4	4.2
Q5	I look at non-conflicting changes to resolve conflicts		3	3.4
Q6	I change non-conflicting code to resolve merge conflicts and avoid introducing unexpected behaviour to the project		3	2.8

#Q, x̃, and x̄ stand for questions, median, and mean, respectively.

4

Confirming Empirical Study Results



#Q	Description	1 2 3 4 5	x̃	x̄
Q_2	The more time it takes to resolve a conflict, the more difficult the conflict		3	3.4
Q_3	I merge my changes right after addressing an issue		4	3.9
Q_4	I resolve merge conflicts right after they occur		4	4.2
Q_5	I look at non-conflicting changes to resolve conflicts		3	3.4
Q_6	I change non-conflicting code to resolve merge conflicts and avoid introducing unexpected behaviour to the project		3	2.8

#Q, x̃, and x̄ stand for questions, median, and mean, respectively.

4

Challenges on Merge Conflict Resolution

Lack of Coordination

Lack of communication and awareness

Monitor changes at coarse-grained level

Large commits and rare merges

Lack of an overall workflow

Lack of Tool Support

Inappropriate development environment

Inappropriate tools for showing diffs and support merge conflicts resolution

Mismanaging the backlog

Flaws in the System Architecture

Highly coupled code

Technical debt introduction

Lack of Testing Suite or Pipeline for Continuous Integration

Lack of tests and their maintenance

Lack of continuous integration pipeline and its maintenance

4

Challenges on Merge Conflict Resolution

Lack of Coordination

Lack of communication and awareness

Monitor changes at coarse-grained level

Large commits and rare merges

Lack of an overall workflow

Lack of Tool Support

Inappropriate development environment

Inappropriate tools for showing diffs
and support merge conflicts resolution

Mismanaging the backlog

Flaws in the System Architecture

Highly coupled code

Technical debt introduction

Lack of Testing Suite or Pipeline for Continuous Integration

Lack of tests and their maintenance

Lack of continuous integration pipeline and its maintenance

4

Sub-Challenge: Lack of Communication and Awareness

- Create communication channels for all stakeholders and channels (e.g. slack or Microsoft teams) focused on developers or specific components (e.g. backend and frontend developers)
- Fix conflicts as soon as you are aware
- Keep others aware of refactoring changes
- Use adequate tool support to avoid developers working on the same region of code (e.g., GitHub and Bitbucket) and/or tools for managing work (e.g., Jira)

4

Discussions

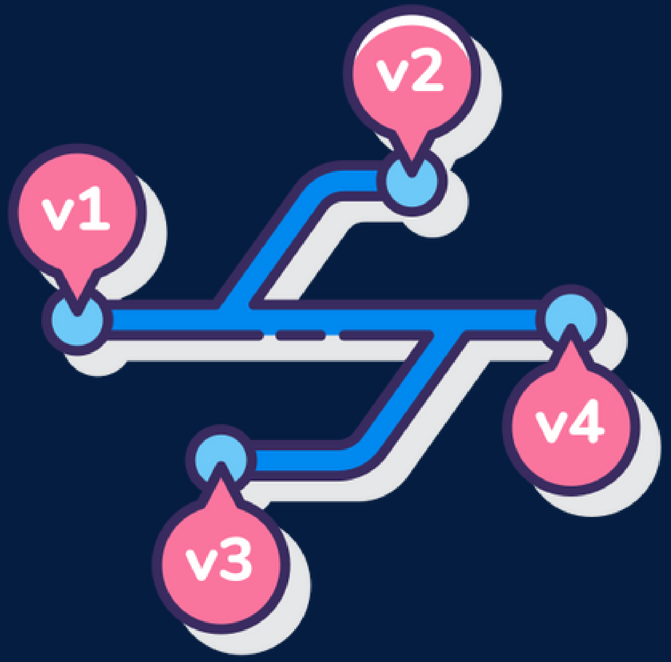
Longest CMS are larger and more complex than the shortest CMSs for most independent variables

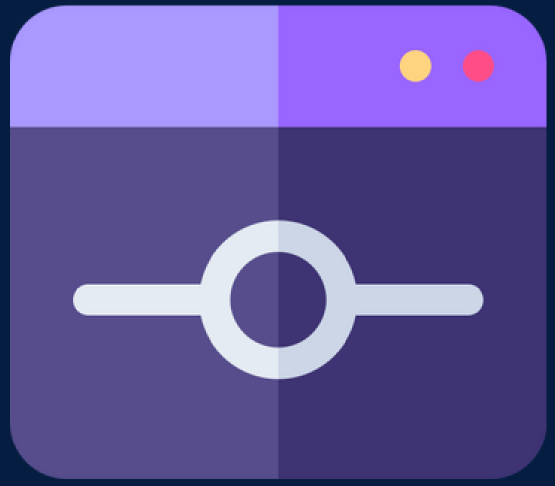
Developers need more time to resolve merge conflicts in **programming language files**, especially when they have a dependency among the code in conflict

4

Discussions

Merge scenario characteristics impact more on the merge conflict resolution time than merge conflict characteristics

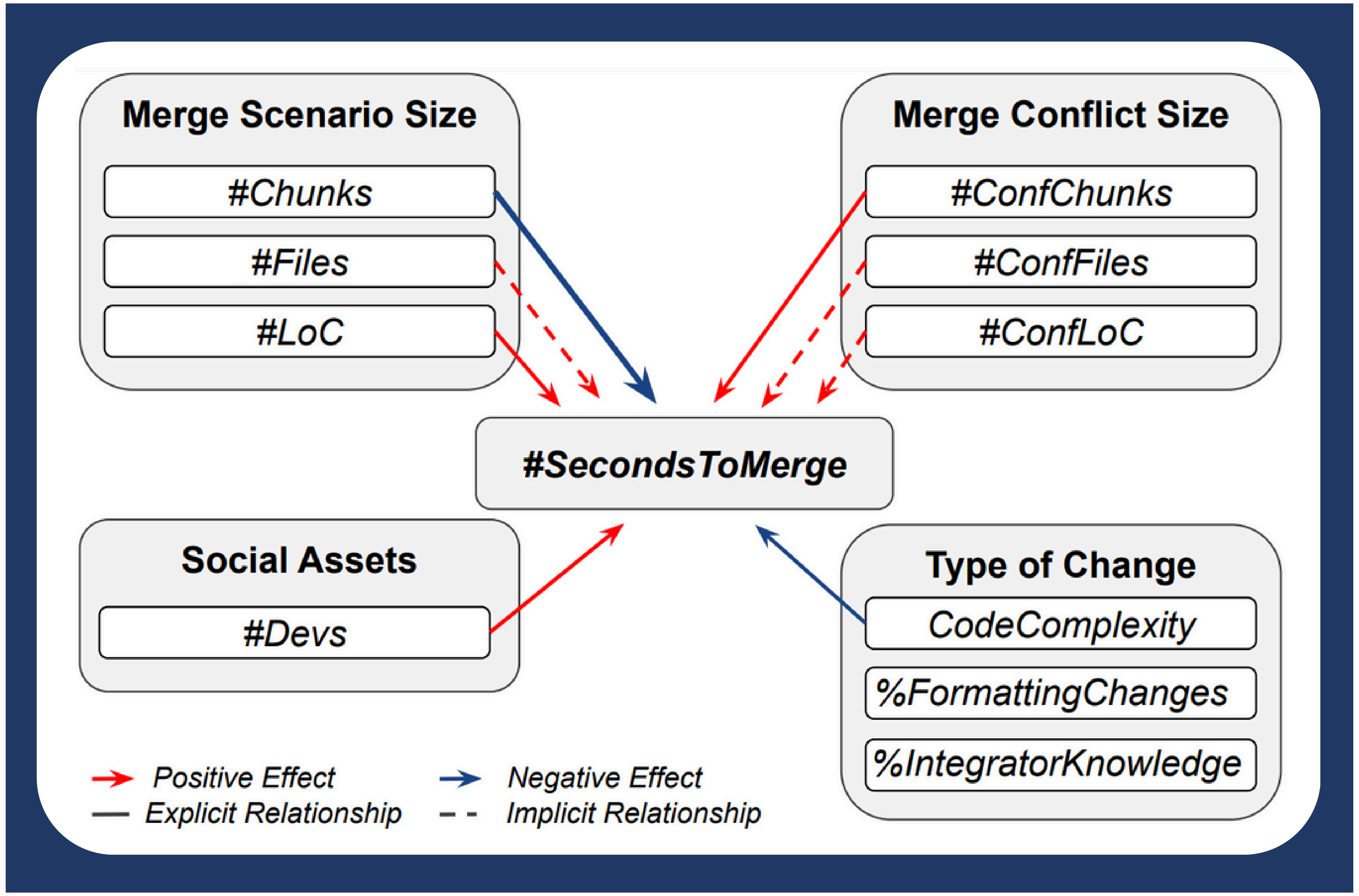




Committing small chunks of code makes the code understanding easier and, consequently, merge conflict resolution faster

4

Study Summary



Merge Scenario Size

#Chunks

#Files

#LoC

Merge Conflict Size

#ConfChunks

#ConfFiles

#ConfLoC

#SecondsToMerge

Social Assets

#Devs

Type of Change

CodeComplexity

%FormattingChanges

%IntegratorKnowledge

Positive Effect

Explicit Relationship

Negative Effect

Implicit Relationship

Lack of Coordination

Lack of communication and awareness

Monitor changes at coarse-grained level

Large commits and rare merges

Lack of an overall workflow

Lack of Tool Support

Inappropriate development environment

Inappropriate tools for showing diffs and support merge conflicts resolution

Mismanaging the backlog

Flaws in the System Architecture

Highly coupled code

Technical debt introduction

Lack of Testing Suite or Pipeline for Continuous Integration

Lack of tests and their maintenance

Lack of continuous integration pipeline and its maintenance

EMPIRICAL STUDIES

1 On the Relation between GitHub Communication Activity and Merge Conflicts

2 Predicting Merge Conflicts Considering Social and Technical Assets

3 Behind Developer Contributions on Conflicting Merge Scenarios

4 Challenges of Resolving Merge Conflicts: A Mining and Survey Study

TAKEAWAYS

Our investigations included several approaches (e.g., developer communication networks and developer roles) **on different merge conflict life-cycles** (e.g., predicting and resolving merge conflicts)

Implication for Researchers

To investigate the **social perspective more often**

To consider **the whole code changes** in a merge scenario

To create **customised models using historical information** and improving the state-of-art of merge conflict prediction

- Merge conflicts are normally introduced by a few developers
- Merge conflicts are recurrently concentrated in only a few files

Implication for Tool Builders

To **use developer roles, files, and the branch developers are touching** information to propose tools to support practitioners avoiding and predicting merge conflicts

To create **solutions incorporated into IDEs** to avoid developers to swictch from one tool to another

To create better **visualizations of code changes and merge conflicts**

Implication for Practitioners

To integrate their code more often **using pull requests**

To create short merge scenarios and with small chunks. It will help not only to avoid merge conflicts, but also to make it simpler to understand and consequently to resolve

To define **policies** to guide themselves on how **to deal with merge conflicts**

INVESTIGATING THE MERGE CONFLICT LIFE-CYCLE TAKING THE SOCIAL DIMENSION INTO ACCOUNT

PhD Candidate: *Gustavo Vale*

EVALUATING AI'S ROLE ON SOFTWARE DEVELOPMENT

Gustavo do Vale

Merge conflicts and technical debt challenge modern software development

AI tools like GitHub Copilot are widely used but not rigorously evaluated for these tasks

GOAL: Assess the quality and usefulness of AI-generated code in real-world scenarios

MERGE CONFLICTS

- What are the most common types?
- Can AI-generated code resolve them effectively?

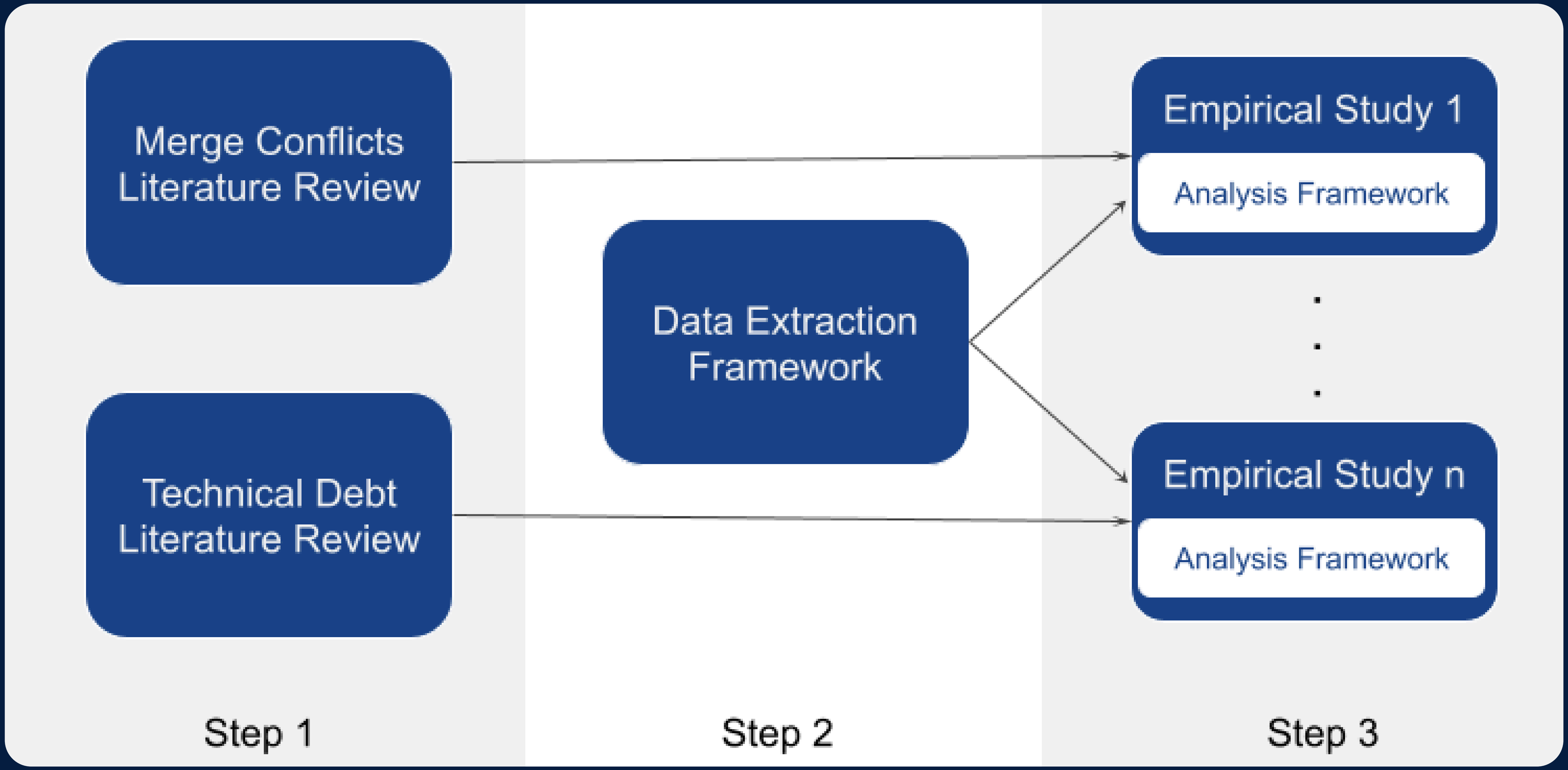
TECHNICAL DEBT

- What are the most frequent and relevant types?
- Can AI tools address these issues adequately?

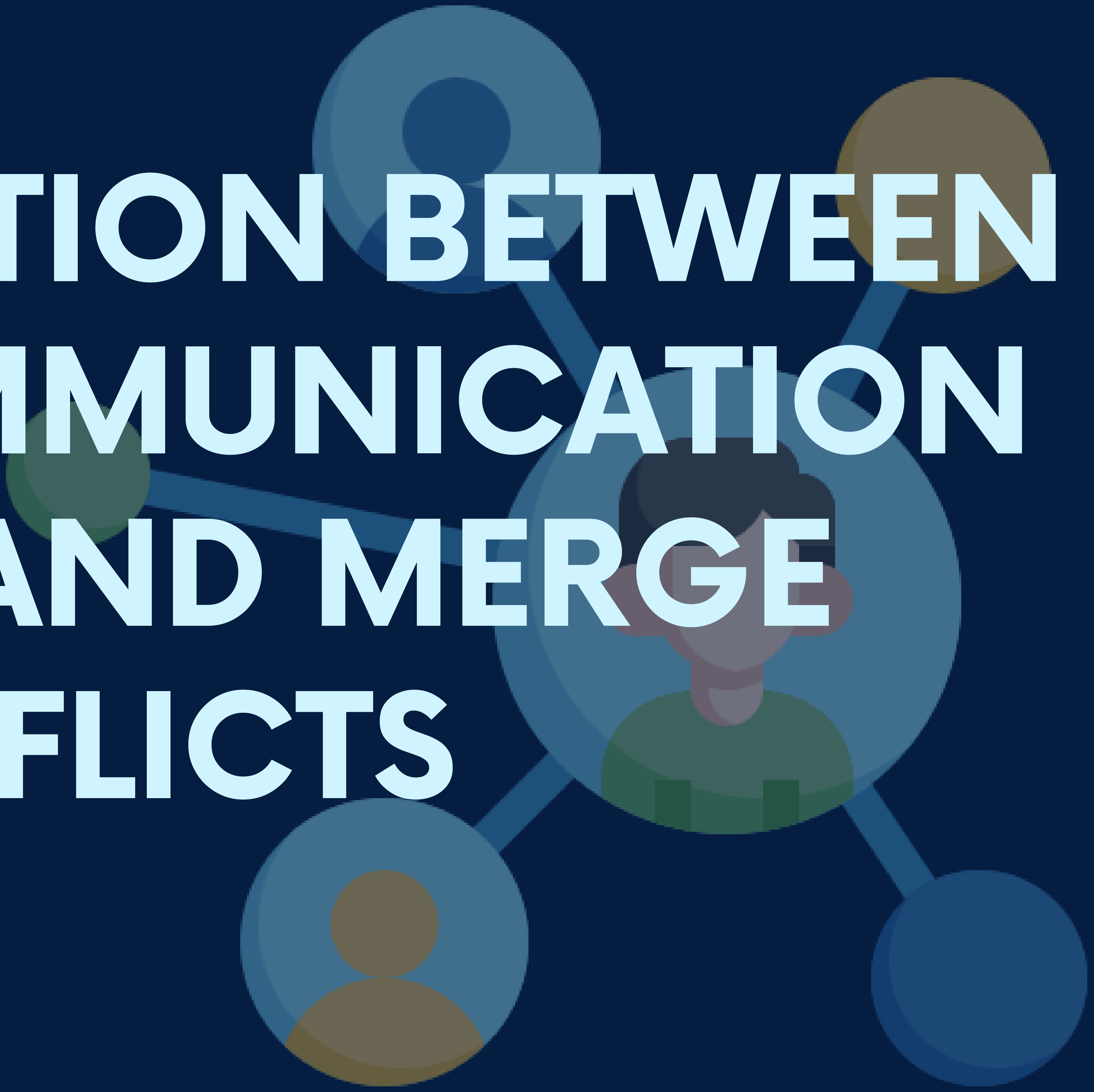
APPLICABILITY IN PRACTICE

- Is AI-generated code effective for:
    - Simple issues (e.g., bug fixes)?
    - Design improvements?
    - Security vulnerabilities?
    - Test coverage?
    - IoT-specific constraints?

METHODOLOGY


Merge Conflicts Literature Review
Technical Debt Literature Review
Data Extraction Framework
Empirical Study 1
Analysis Framework
Empirical Study n
Analysis Framework
Step 1
Step 2
Step 3

ON THE RELATION BETWEEN GITHUB COMMUNICATION ACTIVITY AND MERGE CONFLICTS



1

1

Motivation and Goal

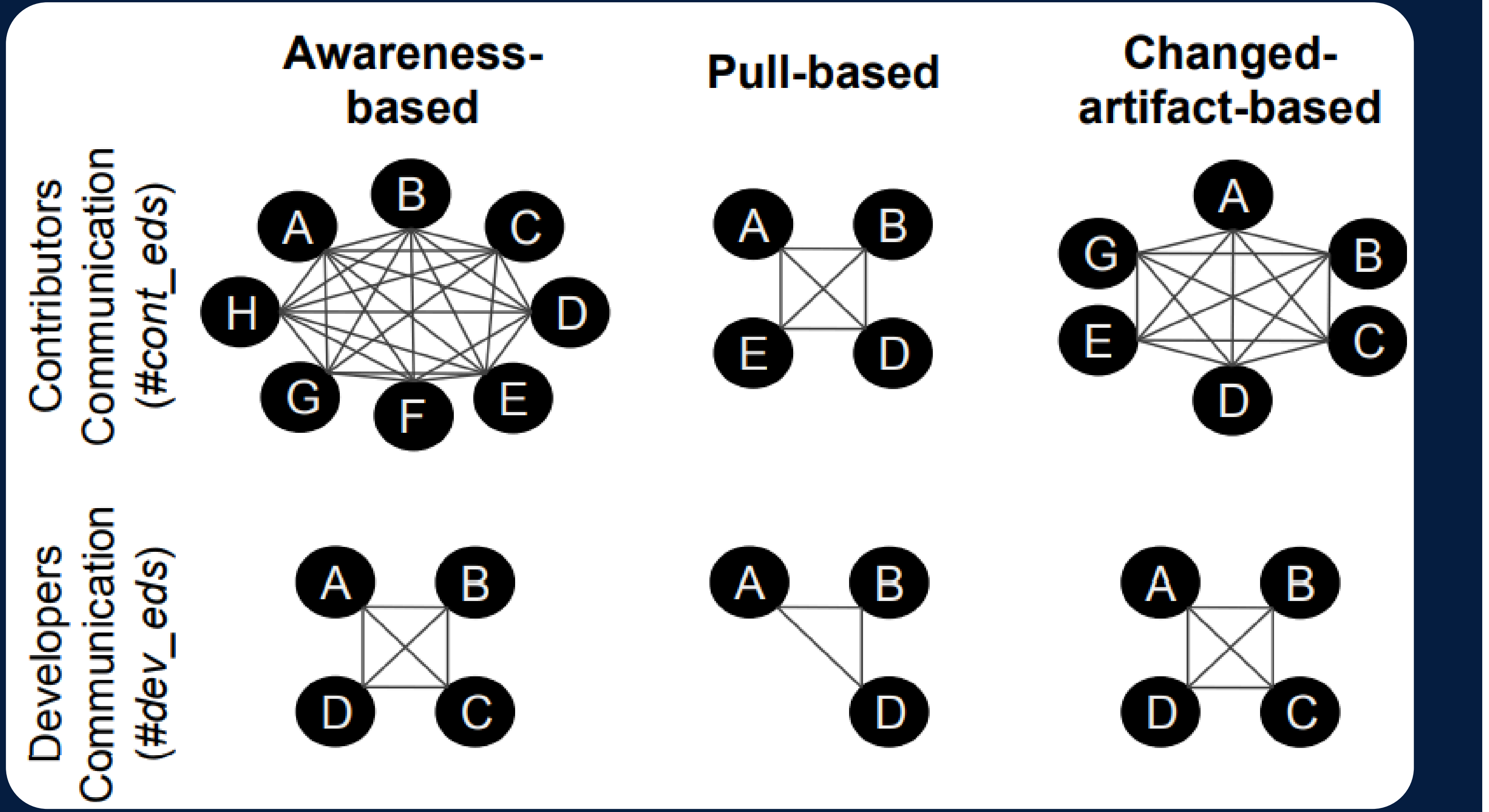
It is believed that proper **communication activity** helps to **avoid merge conflicts**

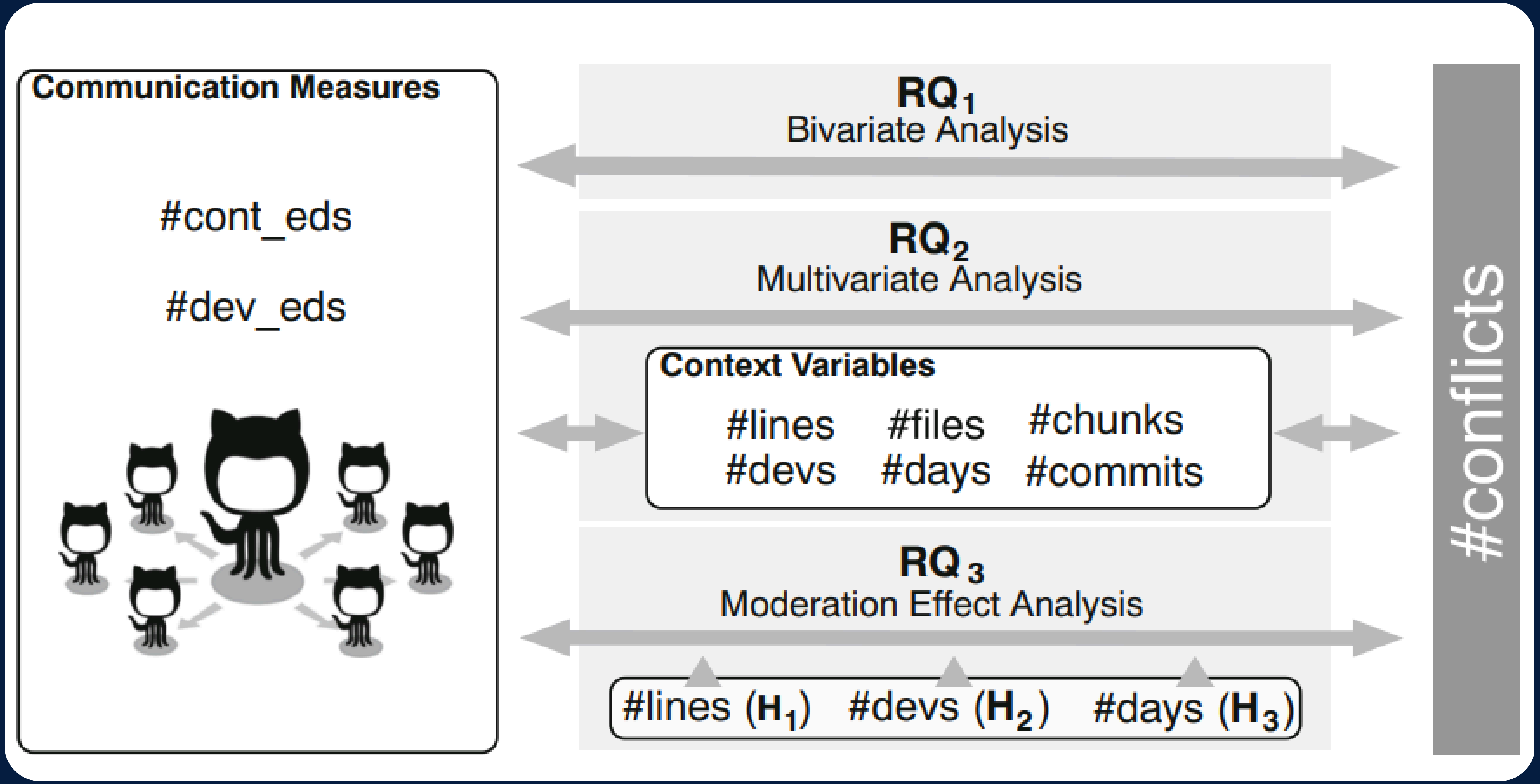
However, in practice, the role of communication activity for merge conflicts to occur or to be avoided **has not been thoroughly investigated**

Communication networks

1

Research Questions


Awareness-based
Pull-based
Changed-artifact-based
Contributors Communication (#cont_eds)
Developers Communication (#dev_eds)
A
B
C
D
E
F
G
H



1

Results

RQ1 - Correlation

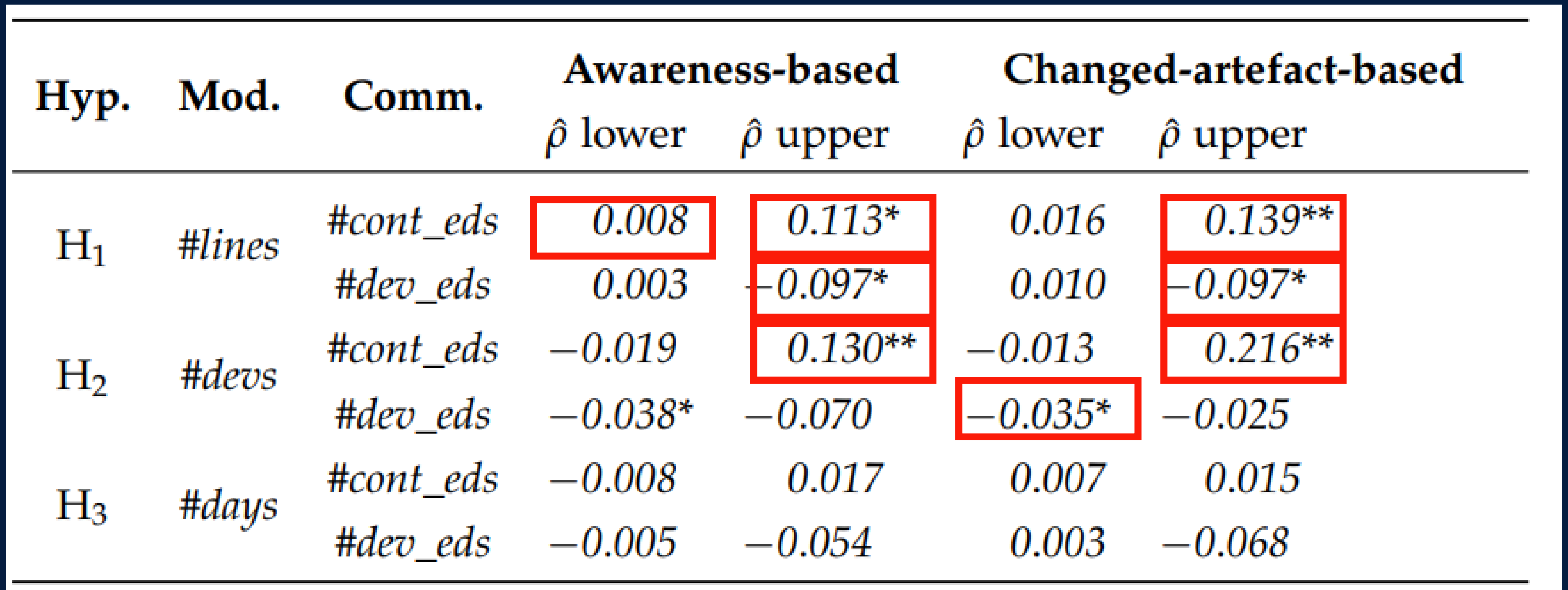
Bivariate correlation analysis shows a significant **weak positive correlation**

In pratical terms, more GitHub communication with more merge conflicts

RQ2 - Correlation Changes with Confounding Factors

Multivariate analysis reveals that there is **no relation** between the communication measures and the number of merrge conflicts

RQ3 - Influence of Merge Scenario Characteristics on the Strength

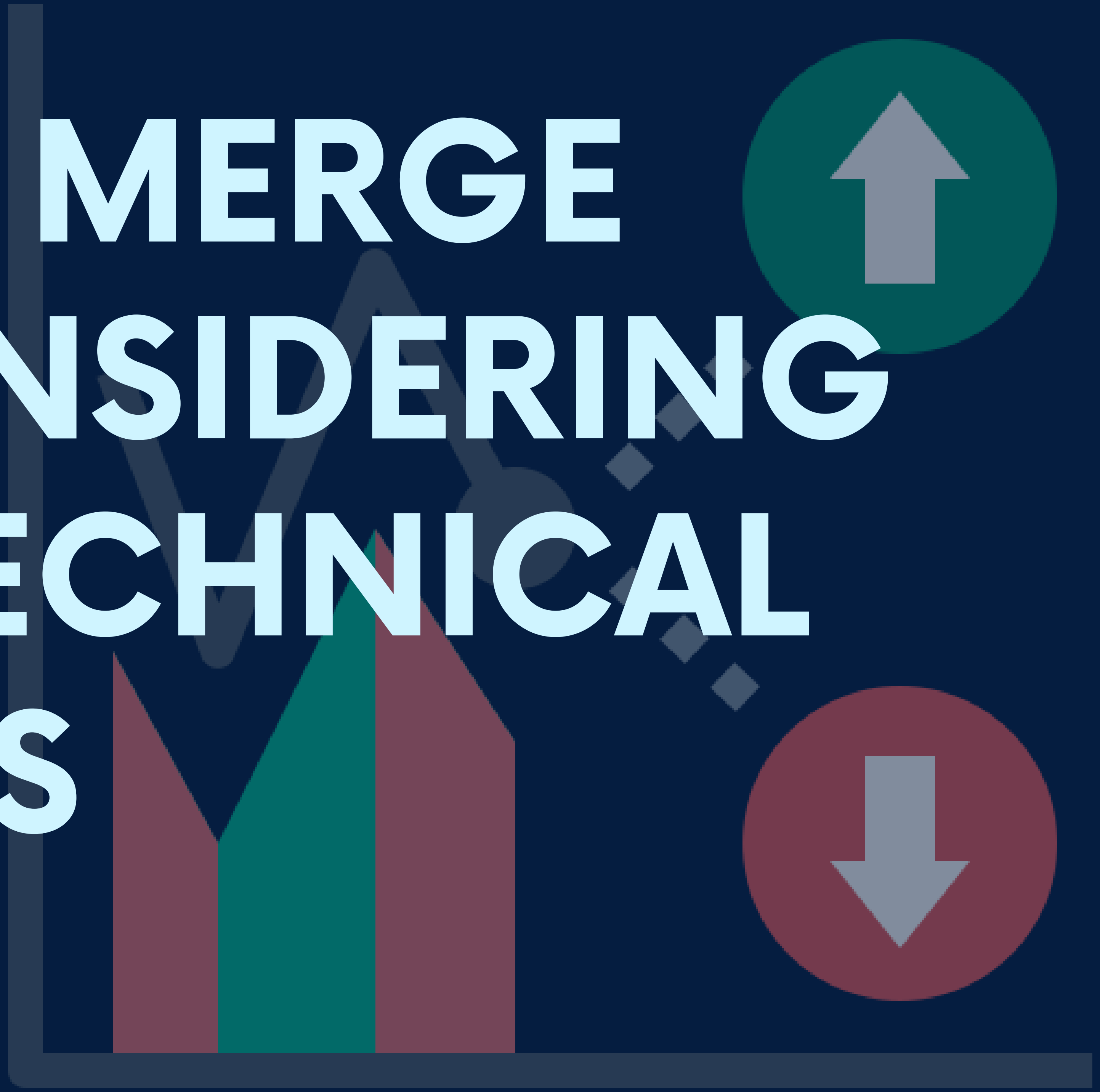


Hyp.	Mod.	Comm.	Awareness-based		Changed-artefact-based	
			$\hat{\rho}$ lower	$\hat{\rho}$ upper	$\hat{\rho}$ lower	$\hat{\rho}$ upper
H_1	*#lines*	*#cont_eds*	0.008	0.113*	0.016	0.139**
		#dev_eds	0.003	−0.097*	0.010	−0.097*
H_2	*#devs*	*#cont_eds*	−0.019	0.130**	−0.013	0.216**
		#dev_eds	−0.038*	−0.070	−0.035*	−0.025
H_3	*#days*	*#cont_eds*	−0.008	0.017	0.007	0.015
		#dev_eds	−0.005	−0.054	0.003	−0.068

Increasing team communication does not influence the occurrence of merge conflicts

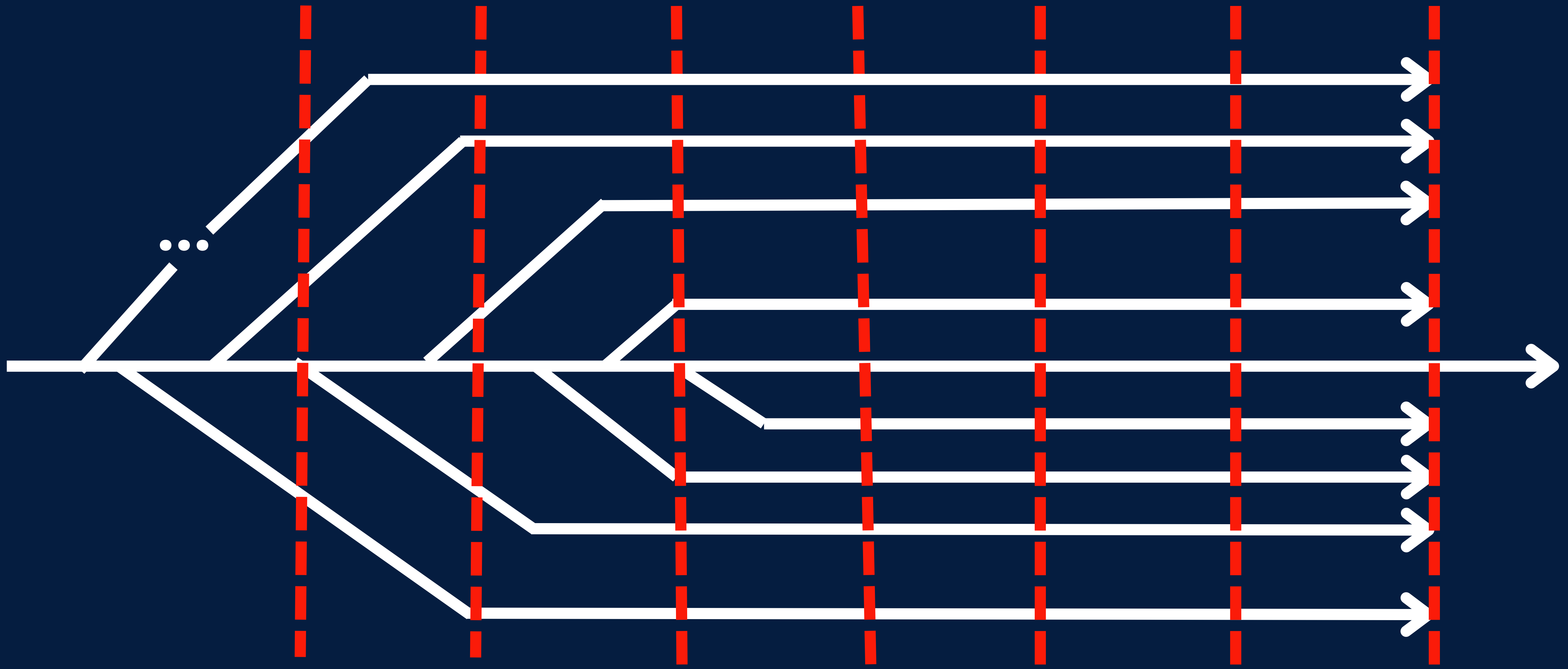
PREDICTING MERGE CONFLICTS CONSIDERING SOCIAL AND TECHNICAL ASSETS

2



2

Motivation and Goal



Constantly pulling and merging can quickly get prohibitively expensive

GOAL - To predict merge conflicts taking the social dimension into account

2

Developer Roles

Top and Occasional contributors

- At project level (global view)
- At merge-scenario level (focused view)

Top contributors collaborate to 80% of changes

2

Specific Goals

RQ1 & RQ2 Which developer roles cause proportionally more merge conflicts (individually and combined)

RQ3 It is feasible to predict merge conflicts using only social measures

RQ4 Combining social and technical assets improve the state-of-the-art of prediting merge conflicts

2

Results - RQ1 & RQ2

RQ1 & RQ2 - Some roles are often related to merge conflicts.

24.6% of merge scenarios that occasional contributors at merge-scenario level touching the source branch are associated to conflicts

32.3% of merge scenarios that top contributors at project level which are occasional developers at merge-scenario level **touching the source branch are associated with merge conflicts**

2

Results - RQ3 & RQ4

Classifiers: Decision tree, Random Forest, and KNN

RQ3 - It is possible to predict merge conflicts with 100% of recall using only social measures

RQ4 - A model with technical measures performs similar to a model with technical and social measures and better than a model with only social measures

BEHIND DEVELOPER CONTRIBUTIONS ON CONFLICTING MERGE SCENARIOS

3

3

Motivation and Goal

There is only a few studies **investigating the involvement of contributors in conflicting merge scenarios**

What is behind developer contributions on conflicting merge scenarios (CMS)?

3

Research Questions

RQ1 - To what extend contributors get involved in CMS?

RQ2 - How often are top contributors involved in CMS?

RQ3 - What are the main characteristics of the changed source files in CMS?

3

Results - RQ1 & RQ2

RQ1 - Only a few developers get involved in more than 10 CMS

- 80% contributors are involved in one or two merge scenarios
- Top contributors often involved in more than 10 CMS
- Half of the contributors have a rate (contributions by conflicts) below 25%

RQ2 - In 42 out of 66 top contributors are also top CMS

- 39.4% of the projects the top contributors participate in >50% of CMS

3

Results - RQ3

Top conflicting contributors commits are responsible for more merge conflicts than the project average in their projects.

- The **coordination of top conflicting contributors** is crucial to the project success
- For most projects **contribution rules** may reduce the emergence of merge conflicts
- The **files often changed are conflict-prone**. Predictions might take advanced of this information