Multi-Perspective Visualization to Assist Code Change Review

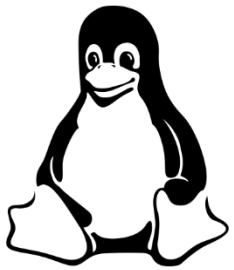
Wang Chen
Wuhan University

Contents

1 Motivation

Introduction

- Code review becomes a very practical approach in software quality assurance
- How to perform code review in timely manner is considered as a big challenge



Related Work

- Some existing tools help code review. Well-known tools include CodeReviewHub, Gerrit, Codecov, CRITICS and etc.
- These tools focus on extracting the code differences, analyzing code coverage, checking code style, facilitating comments input

Code difference
Provided by Codecov

Motivation

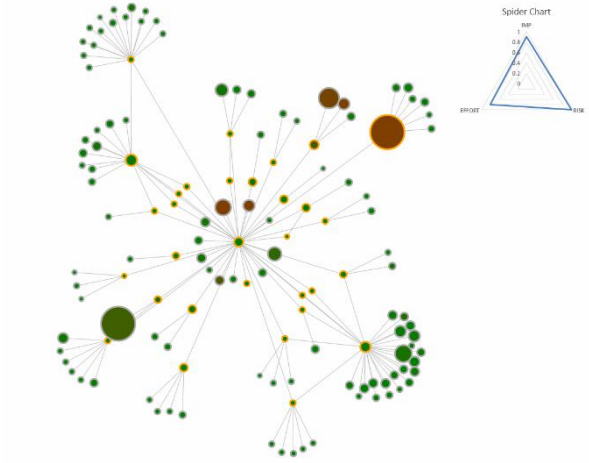
- There can be much more information to be presented by a change review tool
- Therefore, we propose a change review tool, namely, MultiViewer, to serve the purposes.

2 Framework

Overview

MultiViewer will present three useful information, namely effort, risk and impact about change of one commit and visualize these information.

Comprehensive View
provided by MultiViewer



Three Metrics

Effort
Costs of making the changes

Risk
Change’s closeness to bug fixing

Impact
Correlation with other components and influence on the entire system

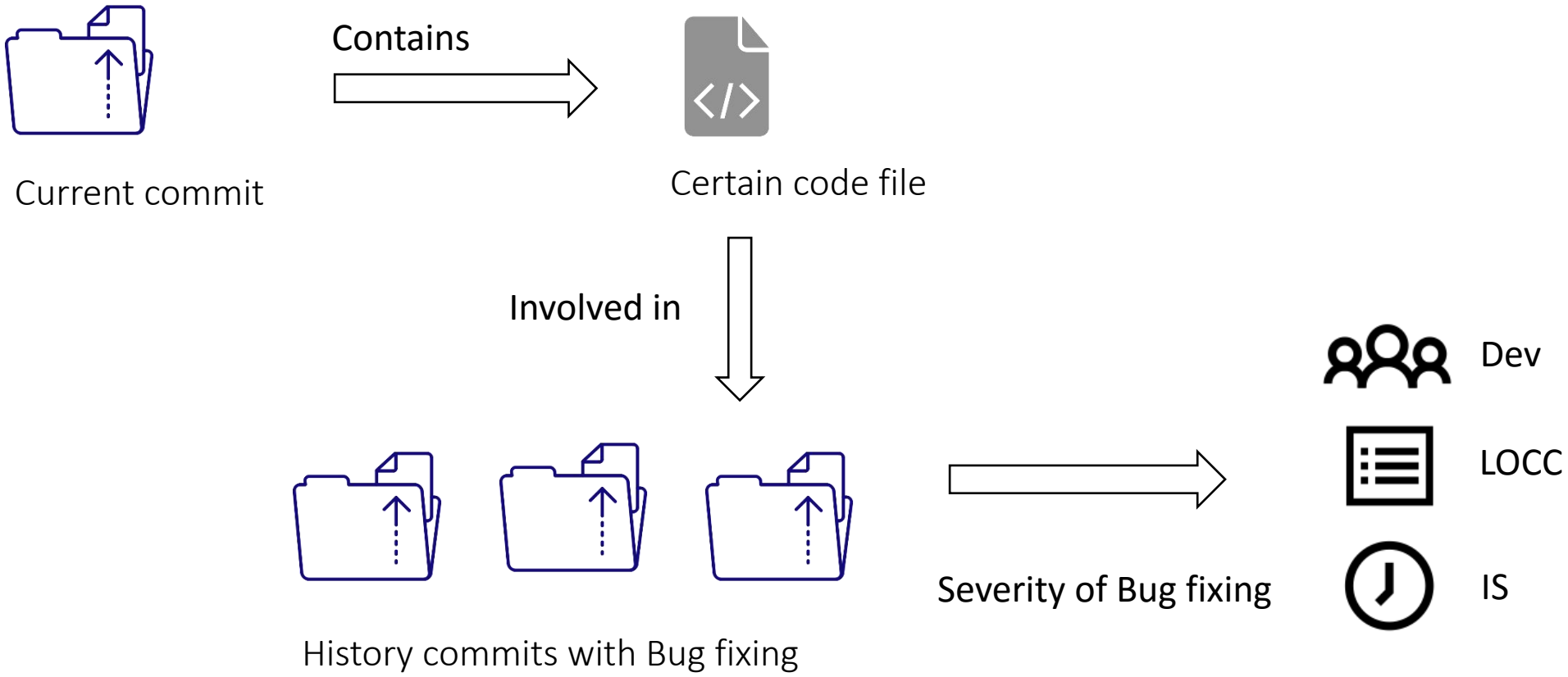
Effort Calculation

Current commit

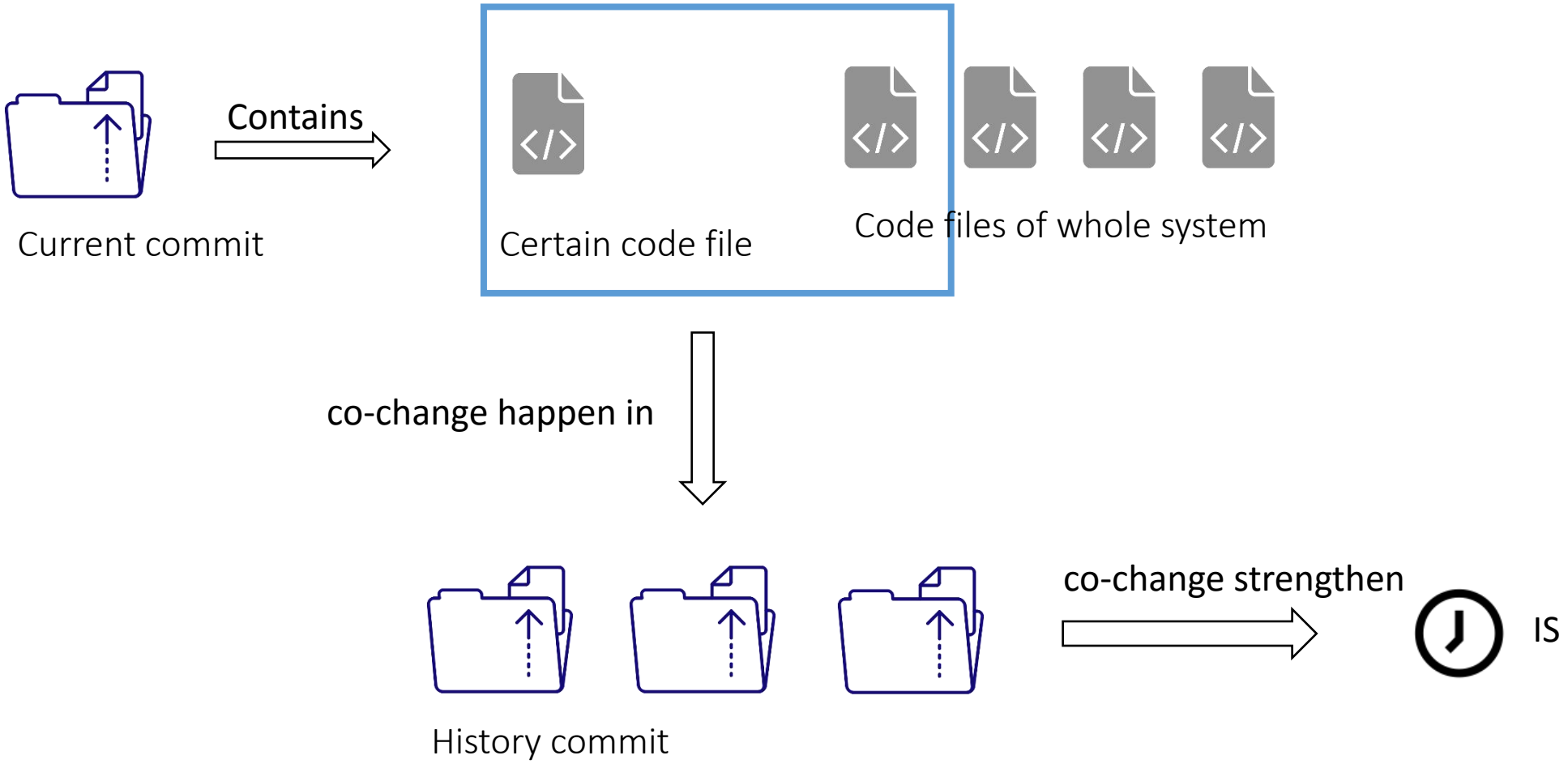
Certain code file

(LOCC) Lines of Changed Code

Risk Calculation



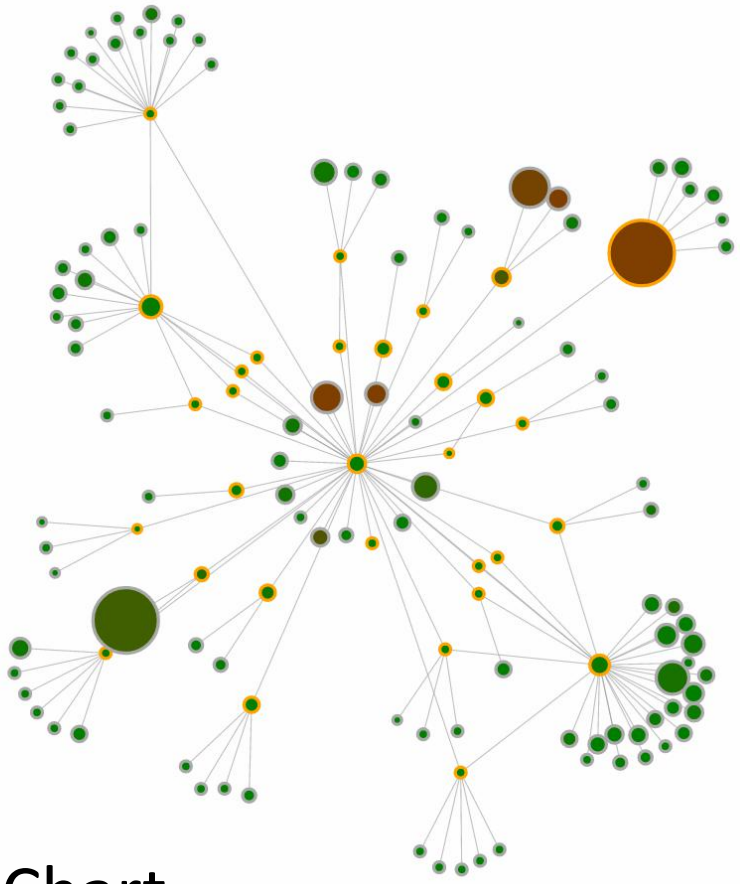
Impact Calculation



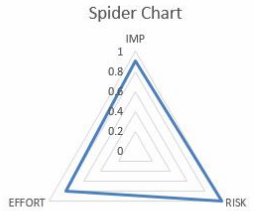
Visualization

- **MultiViewer** provide *Spider Chart* and *Coupling Chart* to help
- The *Spider Chart* provides a preliminary overview of the commit
- *Coupling Chart* focuses on a changeset, visualizing its correlation with other relevant files.

Visualize Example



Coupling Chart



Spider Chart

3 Experiment

Research Question

- **RQ1** : How changes committed by different groups of developers (authors) distinguish from each other with respect to Effort, Risk and Impact

- **RQ2**: How different types of commits distinguish from each other with respect to *Effort*, *Risk* and *Impact*

Experiment Setup

- **Project selection: 10** Java projects on GitHub
- **Project popularity:** Popular and Unpopular

Project groups ▷

Popular			Niche		
Project	Star	Fork	Project	Star	Fork
Buck	4696	722	Jackson-databind	1274	552
Druid	6405	3092	Commons-lang	960	596
Hadoop	3302	3144	Graphhopper	1198	518
Realm-Java	7736	1215	BIMServer	268	203
Spring-Framework	14152	10431	Nutch	1188	850

RQ1

How changes committed by different groups of developers (authors) distinguish from each other with respect to *Effort*, *Risk* and *Impact*.

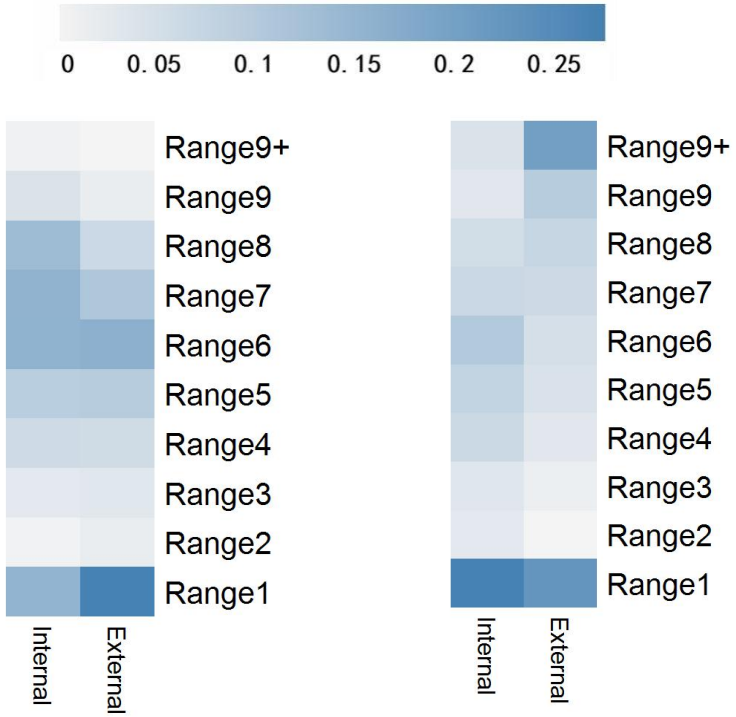
Author Group

- **Author group.** We divided authors into two groups, namely "*Internal*" and "*External*" authors.

All Commit groups ▷

	Project Type	Author Type
G_1	*Popular*	*External*
G_2	*Popular*	*Internal*
G_3	*Niche*	*External*
G_4	*Niche*	*Internal*

Result of Risk



unpopular

popular

◁ Risk comparsion among all Groups of commits

- Internal authors make commits with lower Risk values than External authors
- In Unpopular projects, the conclusion is opposite

RQ2

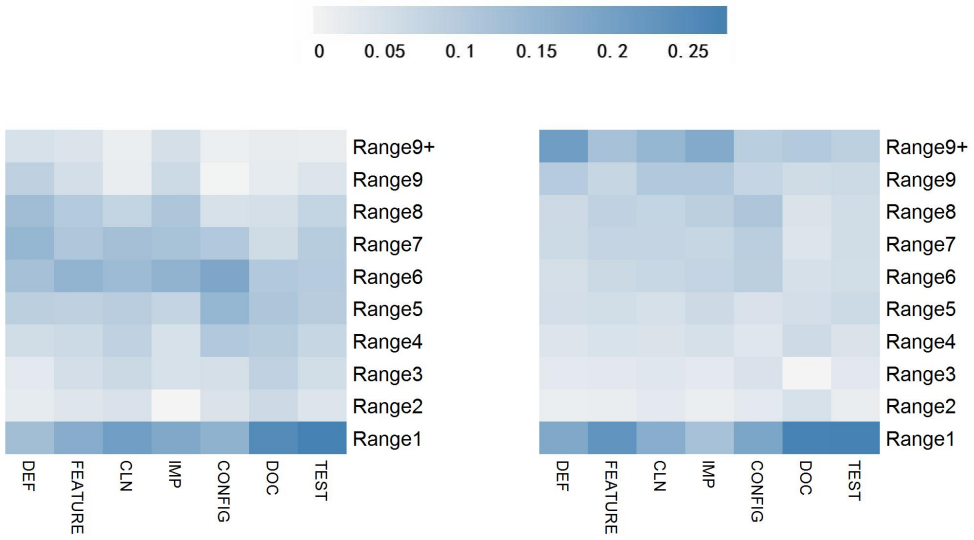
How different types of commits distinguish from each other with respect to *Effort*, *Risk* and *Impact*

Commit Type

- **Commit type:** We considered seven major types of commits and used keyword to recognize the type.

Name for short	Type name	content
CLN	Cleanup	deleting code
IMP	Improvement	improving performance
DOC	JavaDoc	inputing documents
CONFIG	Configuration	configuring the project
DEF	Defect	fixing bugs
FEATURE	Feature	implementing features
TEST	Test	Testing code

Result of Risk



Unpopular

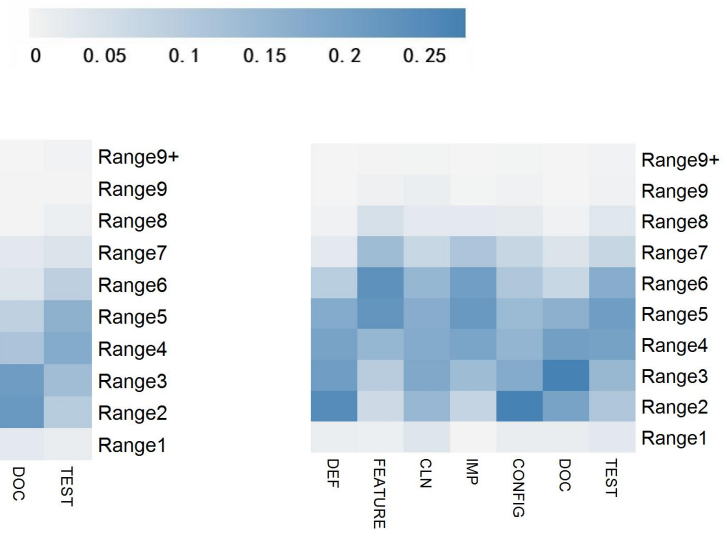
Popular

◁ Risk comparsion

“Defect” is distinguishable with its high Risk

“JavaDoc” and “Test” involve low risk.

Result of Effort



unpopular

popular

◁ Effort comparsion

“Feature” and “Improvement” have high Effort

“Config” and “JavaDoc” commits generally require very low Effort

4 Conclusion

Conclusion

- We defined metrics for code changes: the change effort, risk and impact.
- We also provided a change review assistance tool for GitHub, namely, MultiViewer to visualize such information
- We demonstrated the helpfulness of MultiViewer by showing its ability as indicators to some important project features

Thanks!
Any question?