

# Finding Quality Tradeoffs in Revision Histories

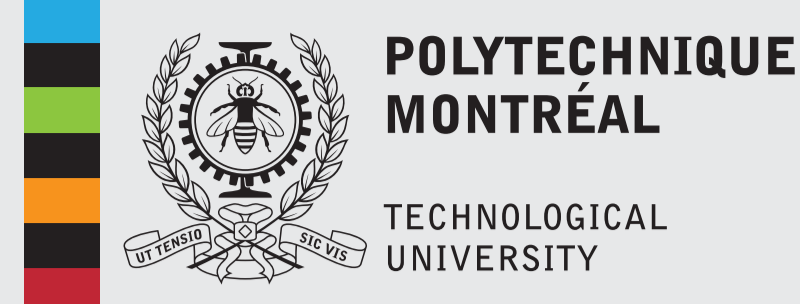
Thomas Schweizer  
thomas.schweizer@umontreal.ca

Vassilis Zafeiris  
bzafaris@aueb.gr

Marios Fokaefs  
marios.fokaefs@polymtl.ca

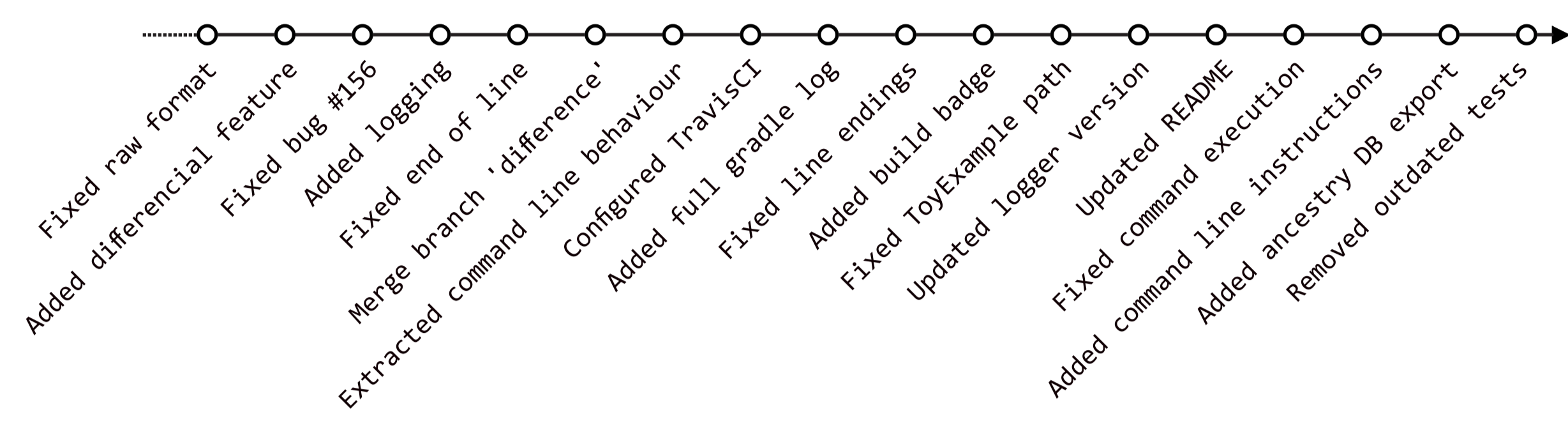
Michalis Famelis  
famelis@iro.umontreal.ca

Université de Montréal



## MOTIVATION

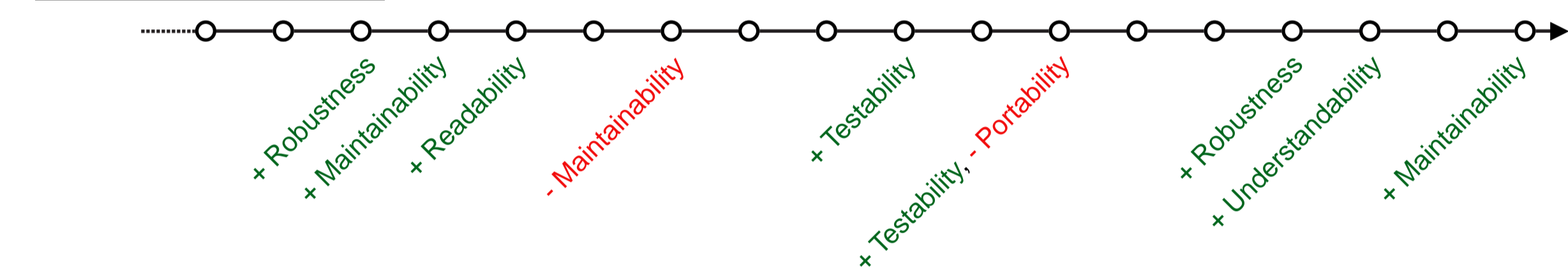
- Contributing to a project requires **contextual design knowledge**.  
Accumulated through development history.
- Obtained by **sifting through** multiple sources of information.  
**Tedious, error-prone, and time consuming** process.



- GOAL**
- Speed up development** by extracting **tacit design knowledge** about **quality concerns** from revision histories.  
Our **assumption** is that when contributors make **tradeoffs** between quality characteristics, they are expressing their **design choices** in the **code**.

- We propose to **annotate coding artifacts** with quality concerns that have **historically** driven their designs.

### Revisions overview



### Annotations for class org.example.FooBar:

- May 2001 - Jan. 2005 / Concerns: Readability
- Jan. 2005 - Aug. 2008 / Concerns: Readability and Robustness
- Aug. 2013 - Jul. 2018 / Concerns: Security and Robustness

## PIPELINE

- Checkout projects from the web
- Compute the metrics for every revision
- Compute the fluctuations of metrics for every artifact
- Compute the quality tradeoffs
- Analyze the quality tradeoffs
- Compile the results in a data sink
- Present the results in IDE or standalone (web) application

## DATASET

- Mined **13 projects** : >**100 000** versions total
- Projects: Ant, ArgoUML, Dagger 2, Hibernate ORM, JEdit, Jena, JFreechart, JMeter, JUnit4, OkHttp, Retrofit, RxJava, Xerces-J
- Integrated a **trusted third party** static analyzer: *SourceMeter*
- Calculated **52 well known metrics** for **each class** in **each version**
- More than a **100GB** in size
- Data** will be **open** upon publication

## BRING YOUR OWN PROJECTS

- We are interested to analyze your projects too!
- Consider submitting them to us!
- We'll get back to you :)

Send us a link to your project at [thomas.schweizer@umontreal.ca](mailto:thomas.schweizer@umontreal.ca)



## FUTURE WORK

- Research robust mapping from metrics to quality
- Improve the detection of tradeoffs and validate it empirically
- Define an unambiguous and measurable definition for Design Intents
- Investigate the relation between design intents and quality tradeoffs
- Incorporate other forms of static analysis (e.g., linters)
- Expand data set
- Expand the dimensions of the fluctuations' classification
- Build IDE plugin

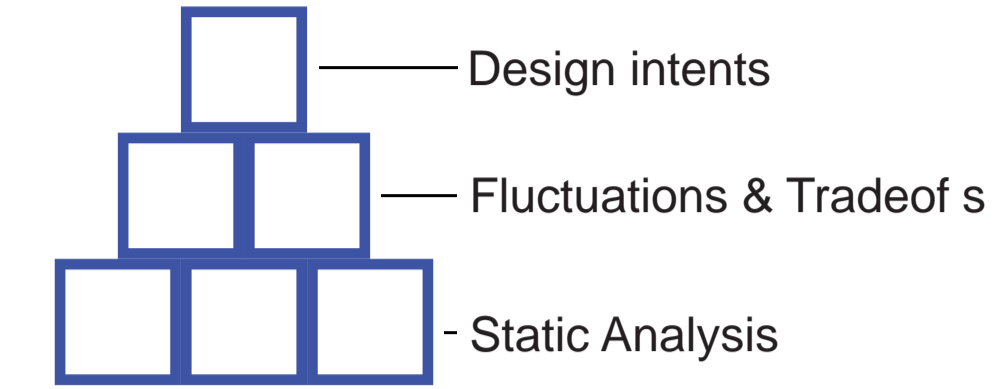
## CREDITS

- [1] Nikolaos Tsantalis et al., "Accurate and Efficient Refactoring Detection in Commit History," 40th International Conference on Software Engineering (ICSE 2018), Gothenburg, Sweden, May 27 - June 3, 2018.
- SourceMeter: [www.sourcemeeter.com](http://www.sourcemeeter.com)
- Icons by 'Freepik' from [flaticon.com](http://flaticon.com)

## EXPERIMENTS

### Strategy

- Division in interchangeable building blocs
- Sanity check with naive implementations
- Qualitative and quantitative empirical validation



### Techniques

- Mixed methods analysis
- Targeted subsets of revisions
- Random sampling

A **metric fluctuation** is the change in one metric between two different versions of the same code artifact.

A **design intent** captures a conscious design choice made by the developer about the structural architecture.

## E1 - MAIN RESULTS

- RQ0:** Do metric fluctuations exist?  
**RQ1:** How can we characterize metric fluctuations?

### RESULTS

- In the majority of revisions, there are no metric fluctuations.
- Metric fluctuations rarely "cancel out".
- Metrics change monotonically in roughly a third of all revisions.
- Quality deteriorates more often than it improves.
- Metric tradeoffs exist in a minority of revisions.

### Classification

- Based on a pilot study with manual validation
- Aimed to characterize the fluctuations of metrics for a revision
- We define a fluctuation with two dimensions:
  - Cardinality** quantifies the amount of changed metrics.  
Values: {ZERO, ONE, MANY}
  - Direction** quantifies the general direction of change.  
Values: {NEUTRAL, IMPROVE, DECLINE, MIXED}

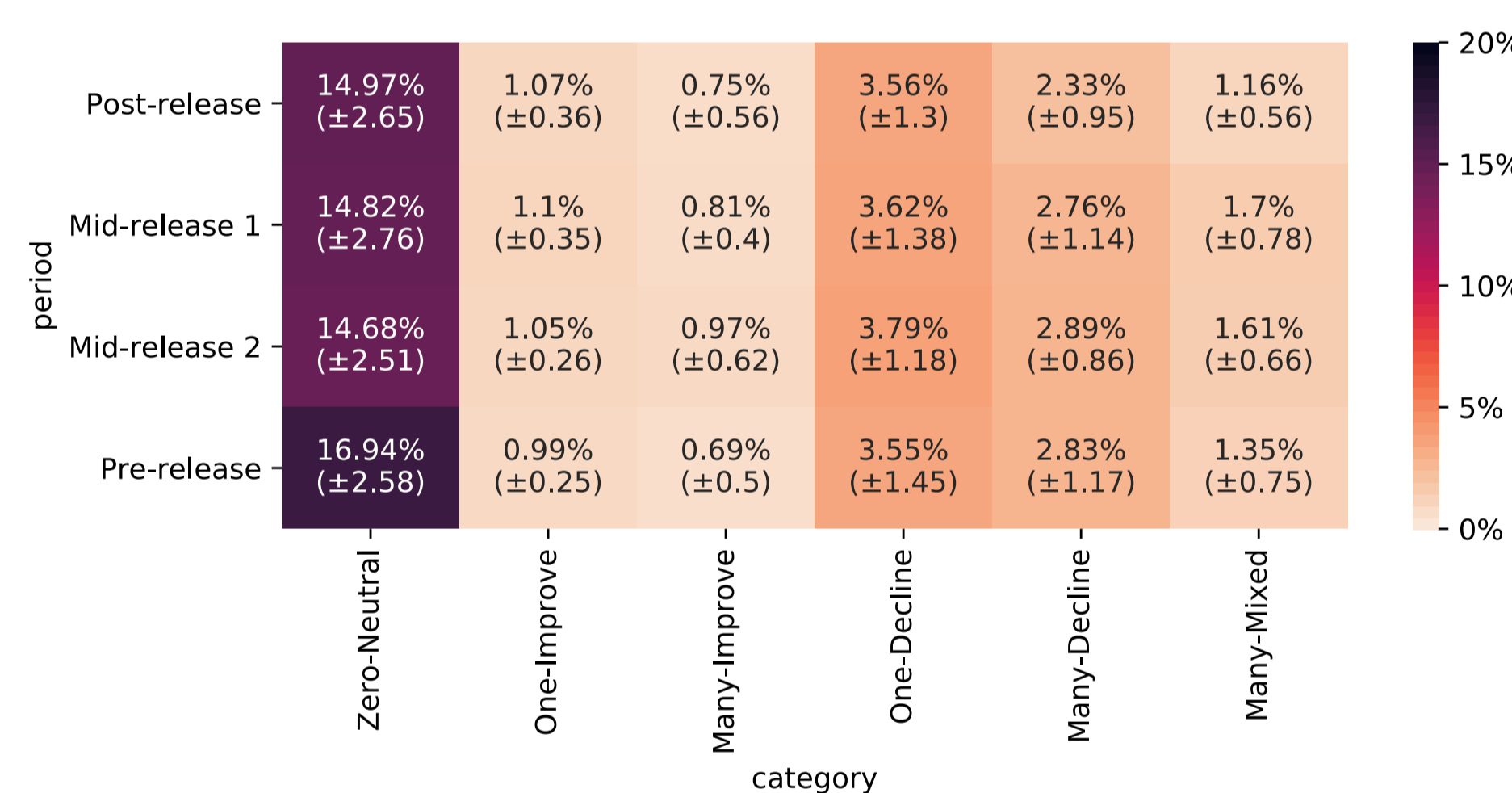
## E2 - CONTEXTS

- RQ2:** Do fluctuations in metrics depend on the development context?

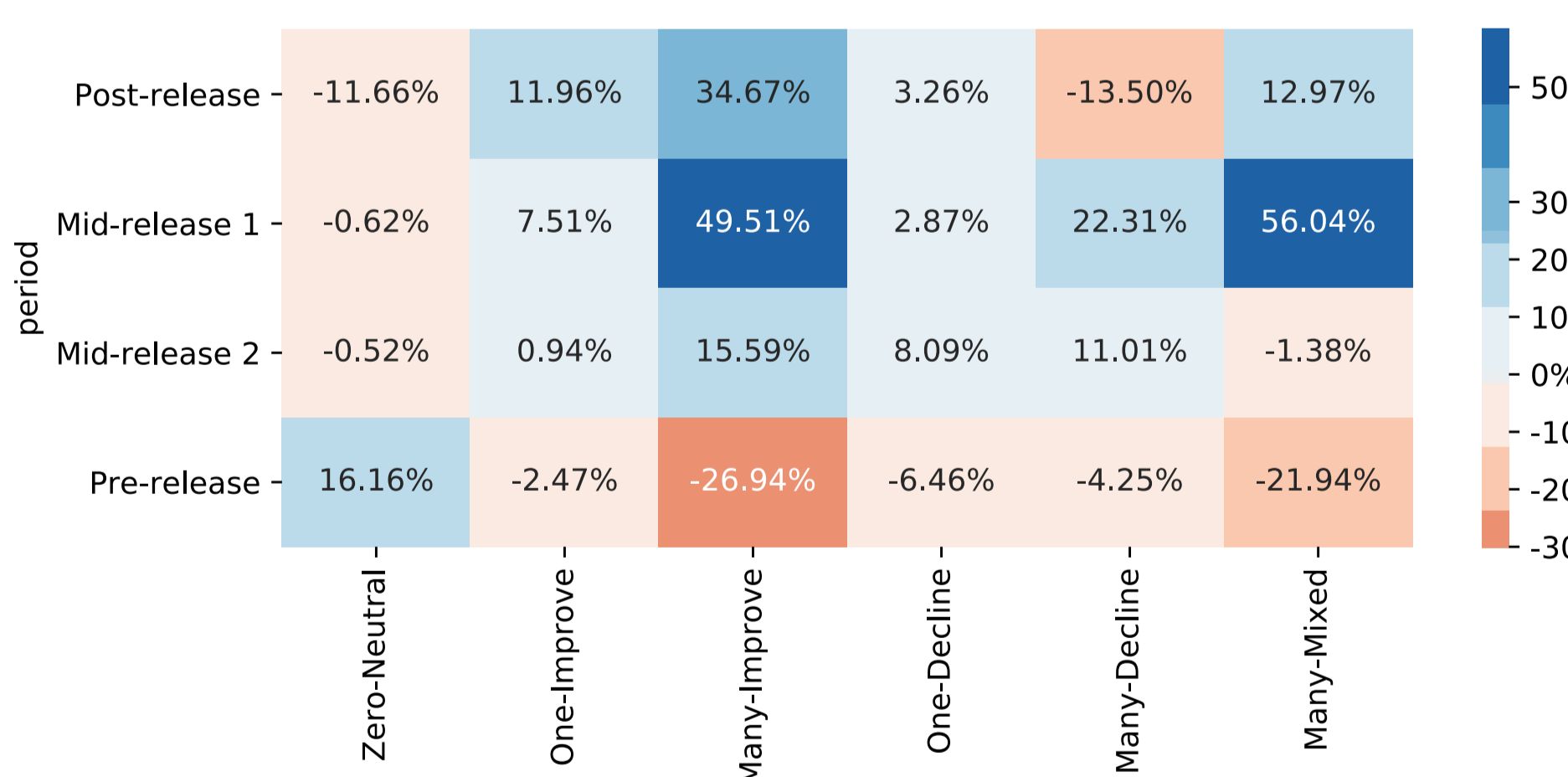
- We investigated three development contexts:
  - Production versus Test code
  - Refactoring
  - Releases
- Refactorings were automatically detected using *RefactoringMiner*[1]
- Releases were extracted manually

### RESULTS

- Tests have even less metric fluctuations than overall production code.
- Refactoring revisions contains more fluctuation activity.
- Refactoring revisions have a significant higher number of internal quality metric tradeoffs.
- Very small effect closer to release dates.
- Interesting differential effects.



(a) Average percentage of project revisions over release periods and metric fluctuation categories.



(b) Percent change of average ratio over successive release periods. Post-release is calculated relative to Pre-release.

Figure 4 : Distribution of metric fluctuations relative to stages in release cycle, averaged over all projects (E2:III).

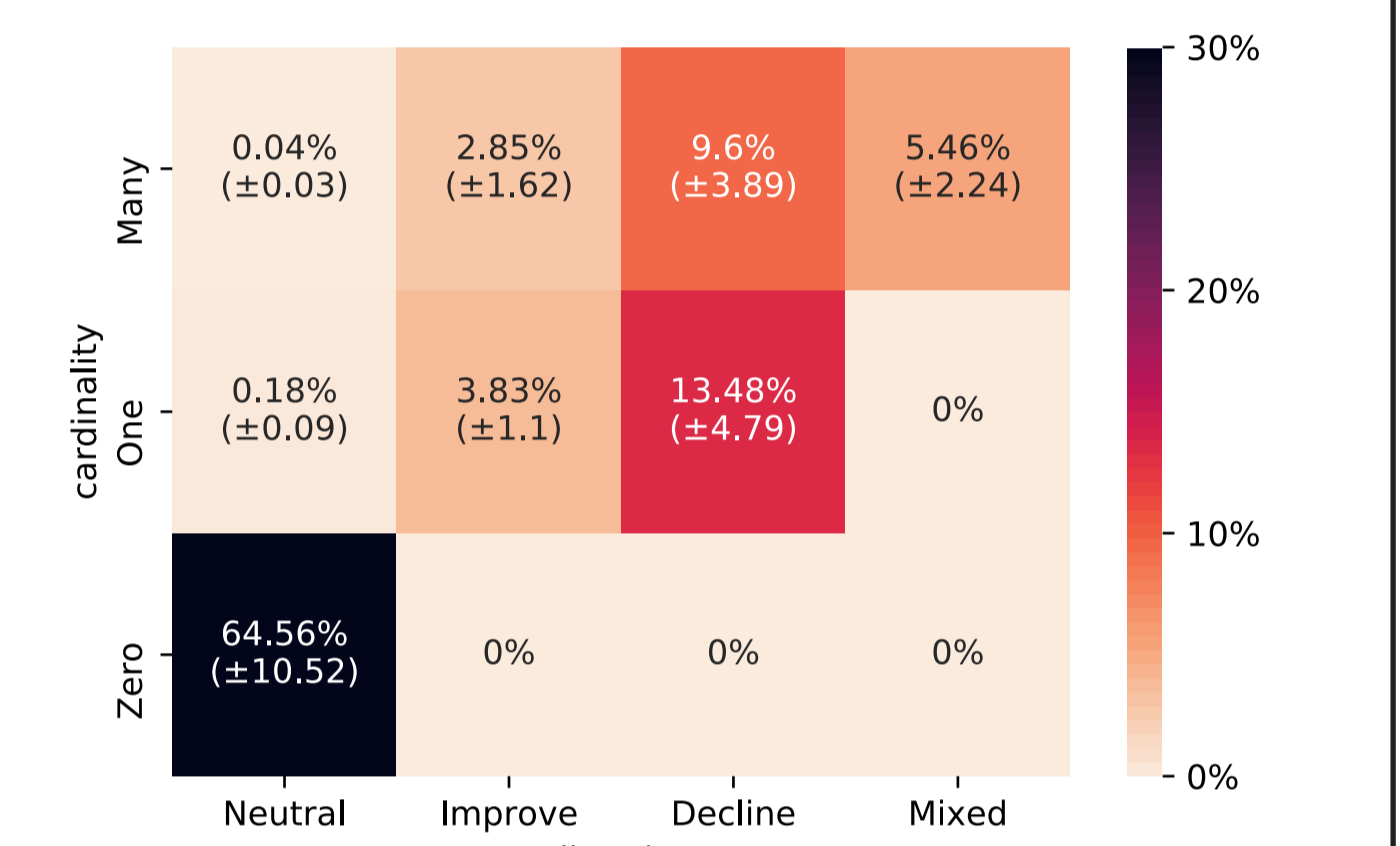


Figure 1 : Distribution of metric fluctuations in production code artifacts, averaged over all projects (E1, E2:I).

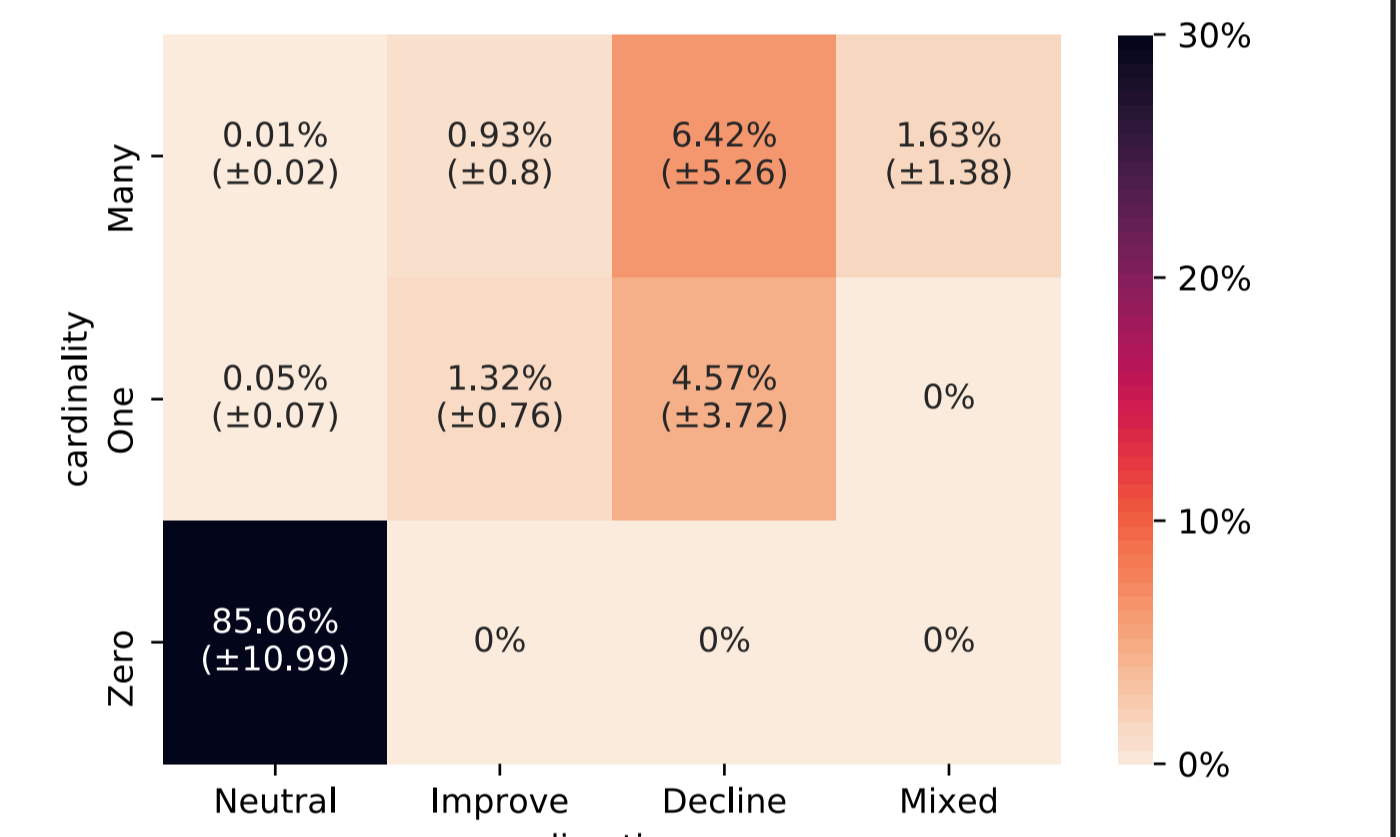
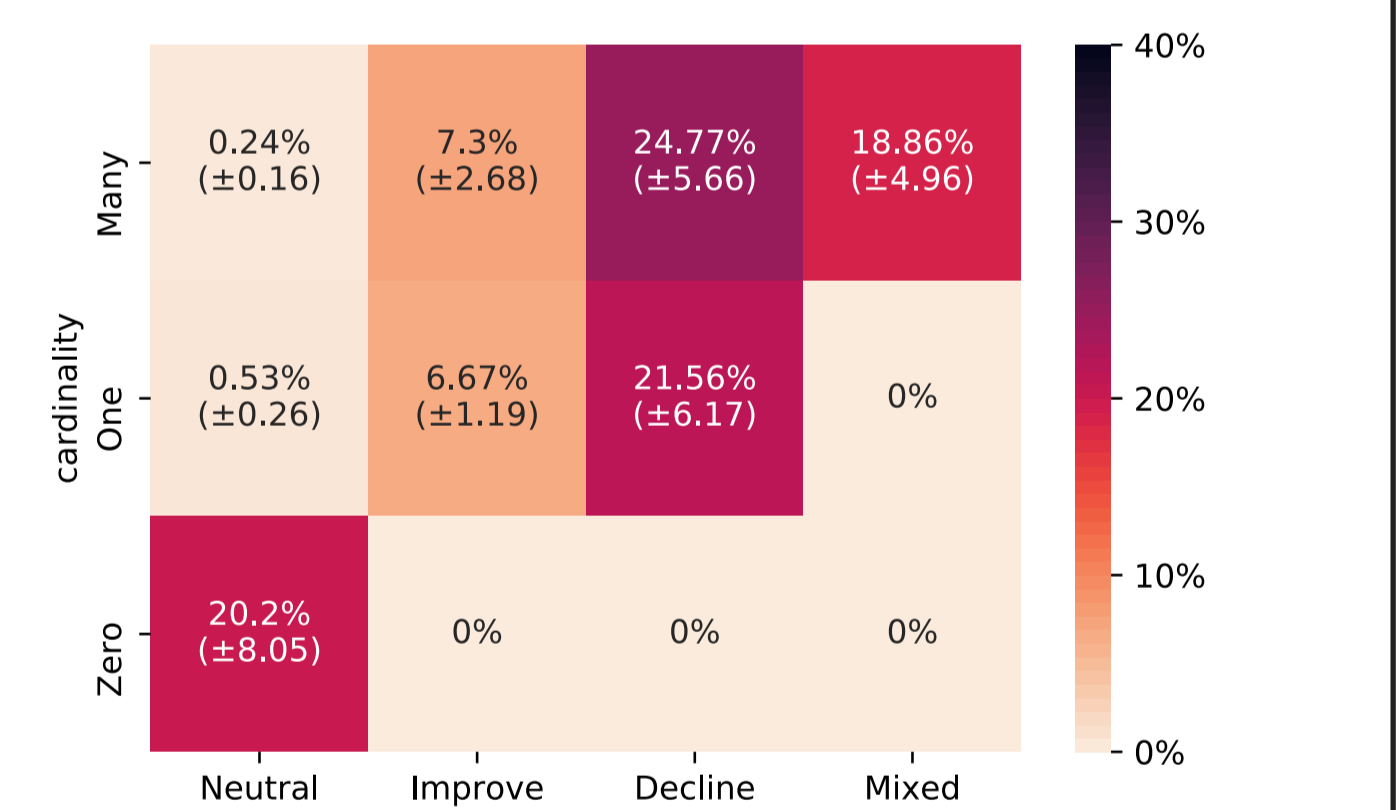
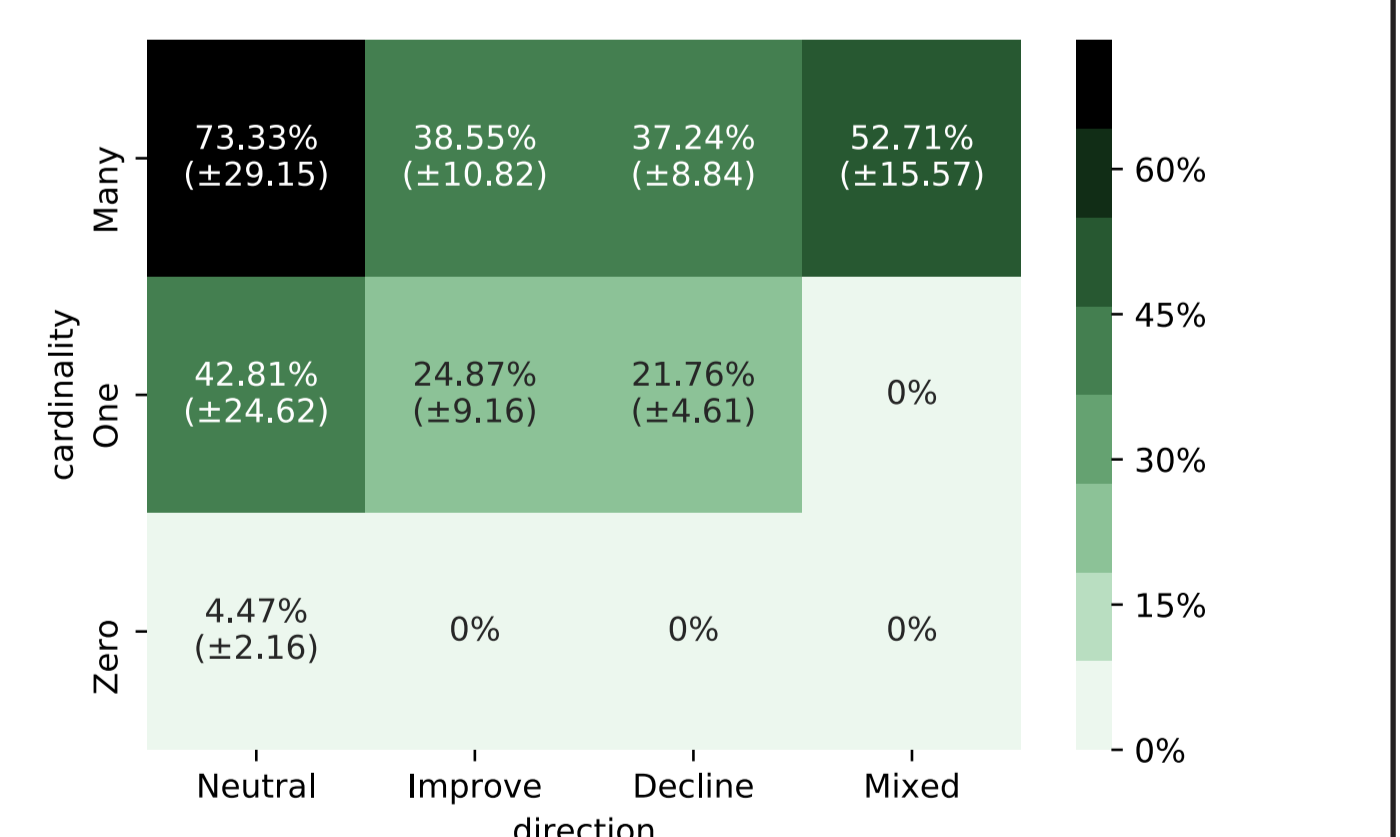


Figure 2 : Distribution of metric fluctuations in test code artifacts, averaged over all projects (E1, E2:II).



(a) Average percentage of metric fluctuations for each category.



(b) Average percentage of RRs with respect to all revisions for each category.

Figure 3 : Metric fluctuation distributions for refactoring revisions averaged for all projects (E2:II).

## E3 - FOCUS ON REFACTORING

- RQ3:** Is refactoring correlated with design intents?

- Refactoring has a big context effect
- Refactoring is widely studied (Solid building block)
- Refactoring is known to be a conscious activity

### RESULTS

- Fluctuations of quality metrics tend to correlate with the presence of design intent.

## CONCLUSION

- Version histories** contain **meaningful** metric fluctuations
- We **identified** a category of fluctuations that has a **high likelihood** to contain design quality tradeoffs
- There exists a **dependency** between metric fluctuations and development context