ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/329639162

EARMO: an energy-aware refactoring approach for mobile apps

Conference Paper · May 2018

DOI: 10.1145/3180155.3182524

CITATIONS
7

READS
44

5 authors, including:

Rodrigo Morales
Concordia University Montreal
17 PUBLICATIONS **170** CITATIONS
SEE PROFILE

Rubén Saborido Infantes
University of Malaga
33 PUBLICATIONS **360** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

6city project View project

moveON View project

All content following this page was uploaded by Rodrigo Morales on 10 April 2019.

The user has requested enhancement of the downloaded file.

ICSE
40th International Conference on Software Engineering
Gothenburg, Sweden
May 27-June 3, 2018

EARMO: An Energy-Aware Refactoring Approach for Mobile Apps

Rodrigo Morales

Ruben Saborido

Francisco Chicano

Foutse Khomh

Giuliano Antoniol

ÉCOLE POLYTECHNIQUE MONTRÉAL

rodrigomorales2@acm.org

@RodMor82

Rodrigo_Morales20

ma

Mobile apps are everywhere

- As of August 2017, there are over 3.5 billion unique mobile internet users. *Source:Statista*
- Users spend on average 69% of their media time on smartphones. *Source: comScore*
- The total number of Android app downloads in 2016 was 90 billion. *Source: App Annie*

Reliability and quality is crucial

Calculator

Google LLC Tools

Everyone

This app is compatible with your device.

★★★★☆ April 8, 2018

18

The only reason that this isn't getting 5 stars is the very long time delay for the "hold to delete" feature. Right now, you have to hold the delete button for about 1.8 seconds to delete everything that you've entered. I feel that 1.8 seconds is a pretty long delay, and something less than 1 second

SASAbus

SASAbus Team Travel & Local

Everyone

This app is compatible with your device.

★☆☆☆☆ September 16, 2016

the last update made everything worse you can't choose your city, the map takes ages to load and all the infos are not updated most of the time.

Previous studies found that anti-patterns have a negative effect on design quality

- Anti-patterns have a negative impact in on code understandability
  Abbes et al [CSMR ‘11]

- Anti-patterns are related to change-proneness
  Khomh et al [ESE ‘12]

Anti-patterns impact energy efficiency

- Mobile phones are constrained in battery life.
- Bad designed apps, can consume battery faster [Gottschalk 2013]

Preliminary Study

- PQ1 Do anti-patterns influence energy consumption?
- PQ2 Do anti-pattern's type influence energy consumption differently?

Study subjects

20 Android apps from a dataset 200 apps

F-Droid

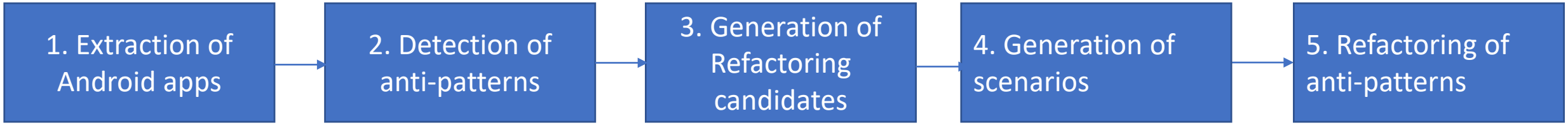
5 object-oriented anti- pattern types

+ 3 Android performance anti-pattern types

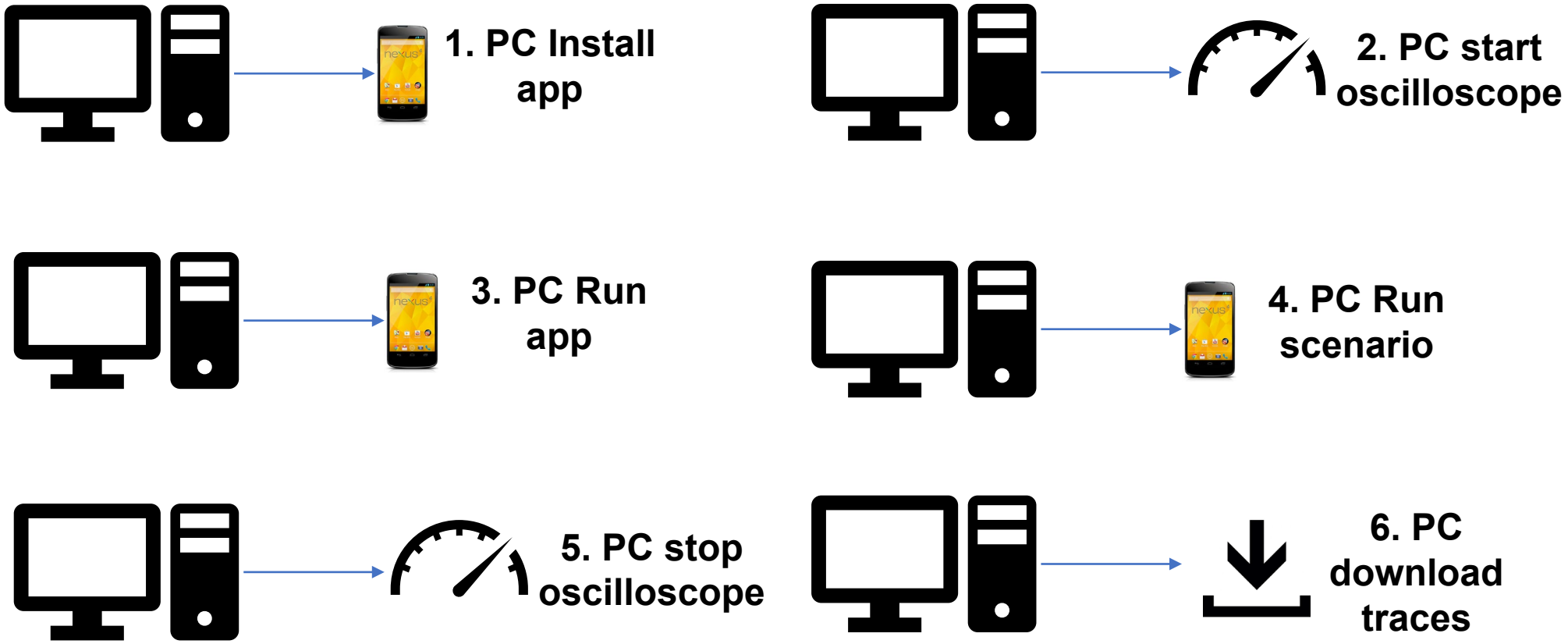
20 Android scenarios

MonkeyRunner

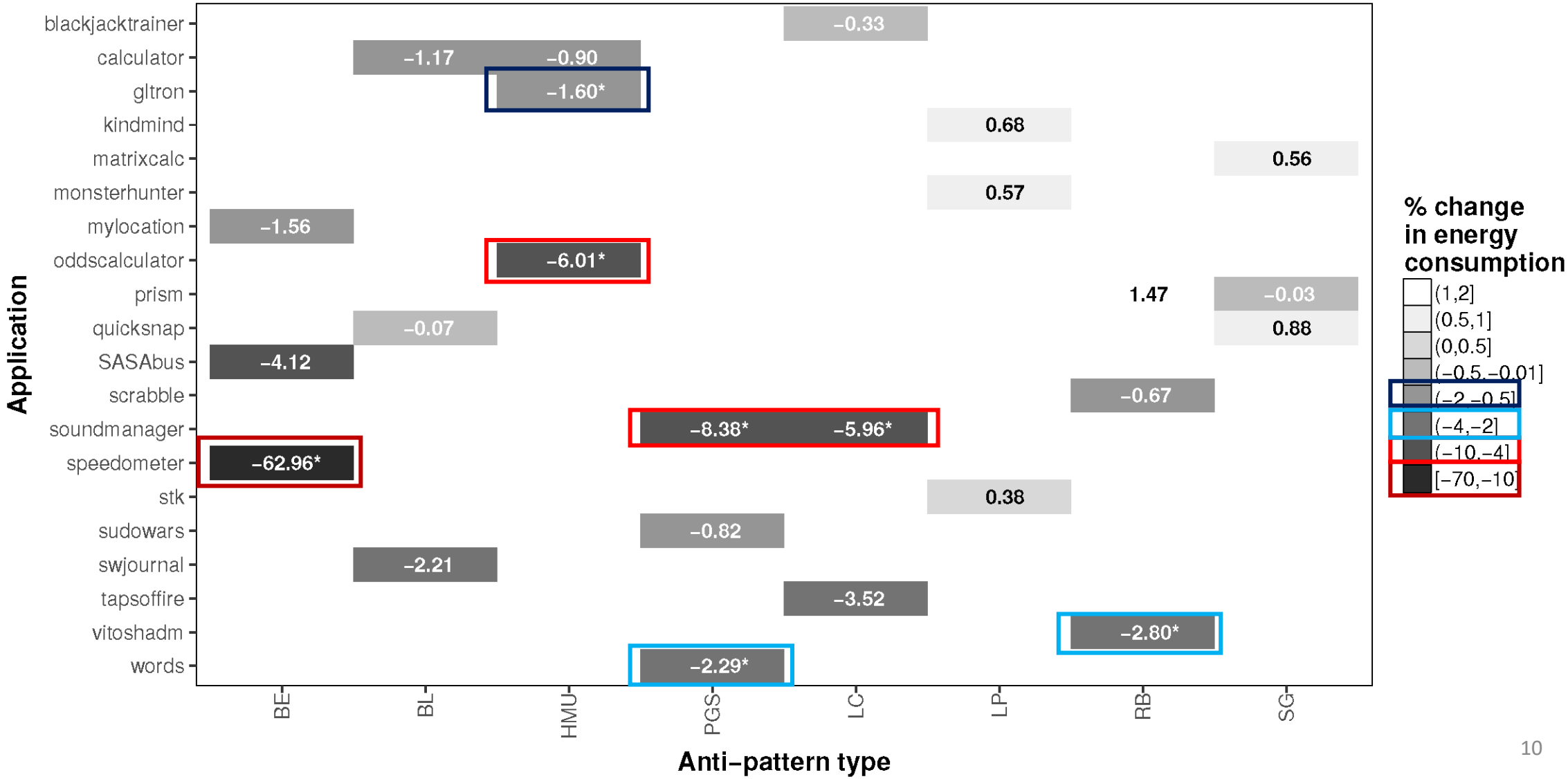
Data collection



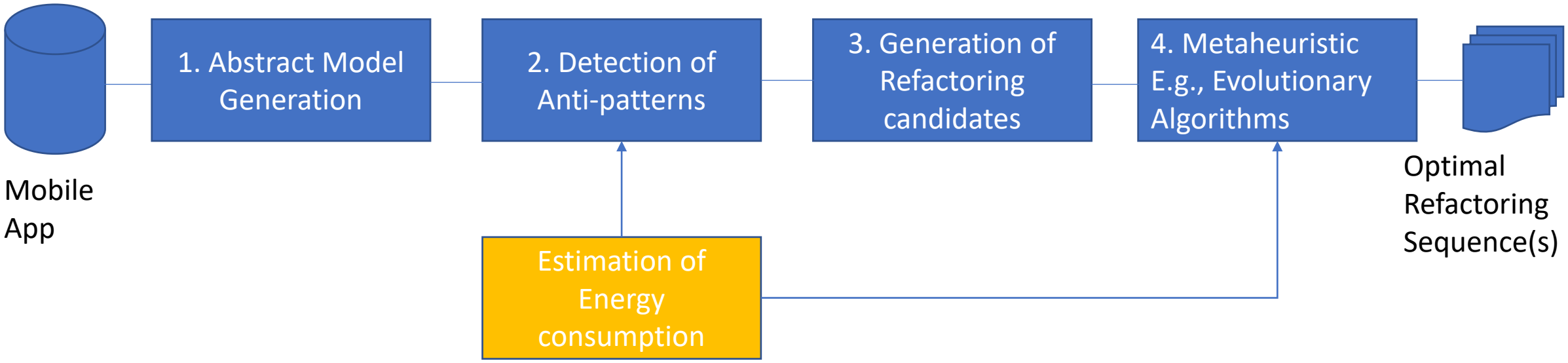
Analysis of the impact of anti-patterns on energy efficiency



Anti-patterns impact on energy efficiency



Refactoring mobile apps while controlling for energy efficiency (EARMO)



Generation of refactoring candidates automatically

1. Identify a source class (class with anti-pattern)
For example: Blob -> Move method

```
1.  class Geometry{
2.      double calcAreaRectangle(Rectangle p1){
3.          return p.Width()*pi.Height();
4.      }
5.
6.      void longParameterlistMethod (int p1, int p2, ... , int p15){}
7.  }
8.
9.  class Rectangle{
10.     private double width;
11.     private double height;
12.     private double Width(){
13.         return width;
14.     }
15.     private double Heigth(){
16.         return heigth;
17.     }
18. }
19.
20. class Shape{...}
```

2. Follow guidelines from the literature to correct anti-patterns

- Move a method to one of its parameter types

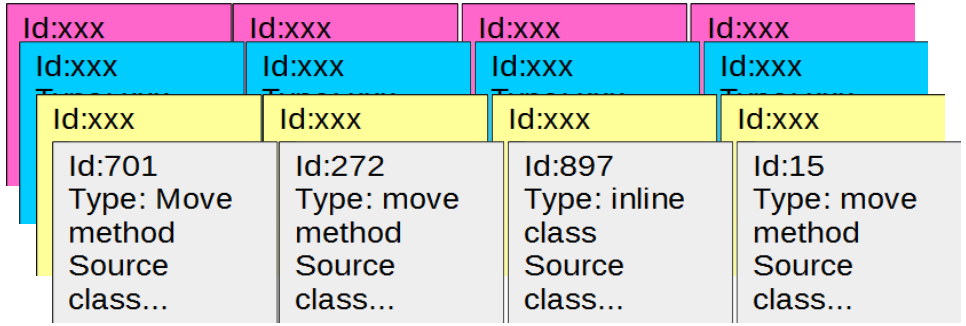
RO	Type	Source class	Method(s)	Target
R1	Move method	Geometry	calcAreaRectangle	Rectangle

Evolutionary algorithm

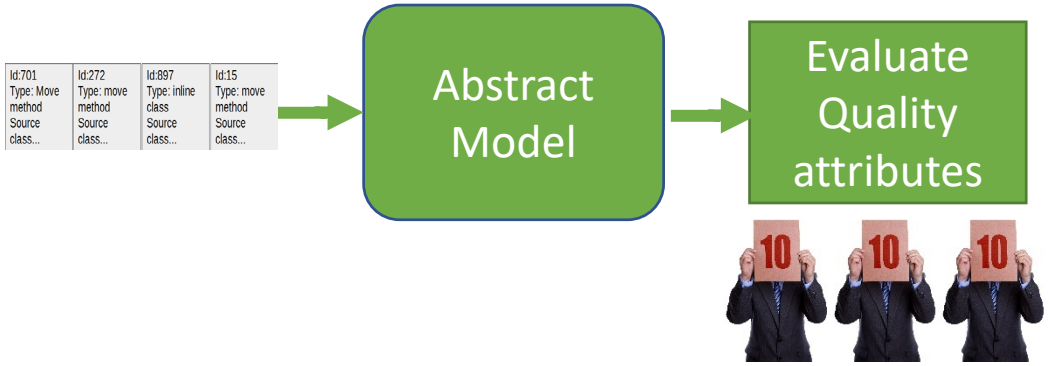
Solution representation

Id:701 Type: Move method Source class...	Id:272 Type: move method Source class...	Id:897 Type: inline class Source class...	Id:15 Type: move method Source class...

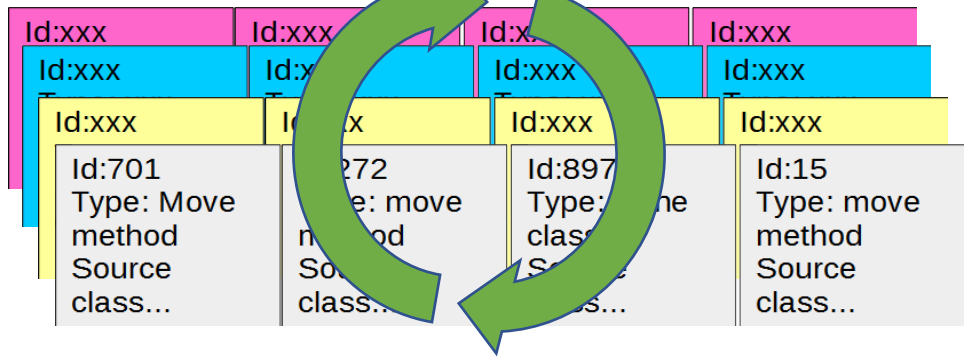
1. Generate initial population



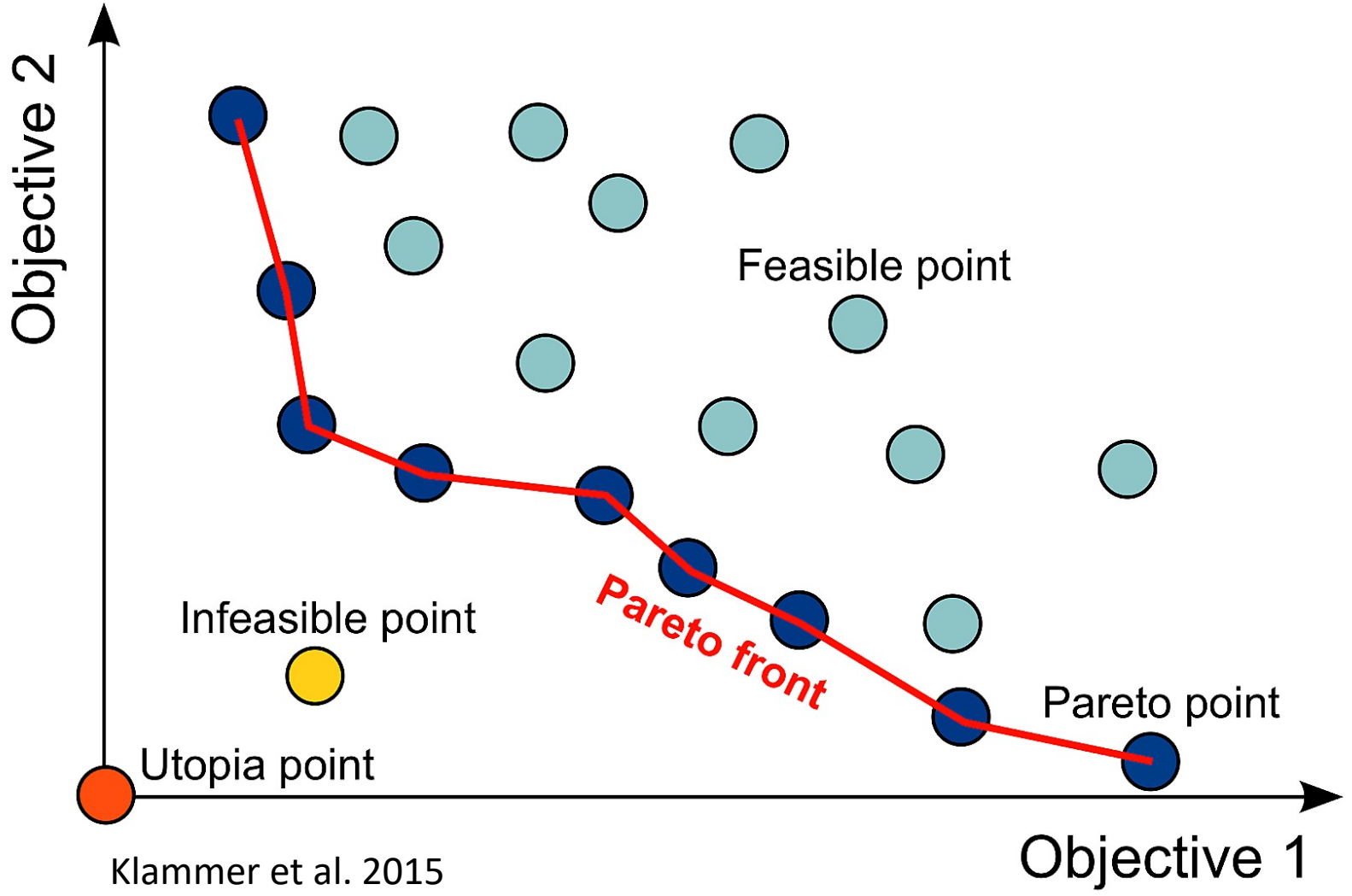
2. Apply each solution in a copy of the original abstract model, and evaluate it .



3. Select the best individuals to pairing and evolving to get the best solutions



Multiobjective Optimization and Pareto optimality

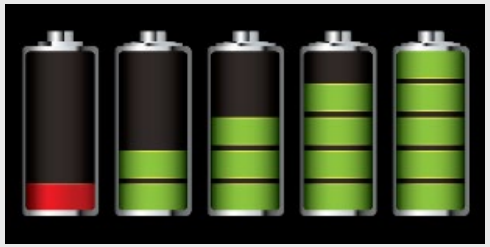


Klammer et al. 2015

Research Questions

RQ1 To what extent can EARMO correct anti-patterns?

RQ2 What is the precision of the energy improvement reported by EARMO ?



RQ3 Can EARMO generate useful refactoring solutions for developers?

Methodology

RQ1 To what extent can EARMO correct anti-patterns?

- *Measure the number of anti-patterns before and after refactoring for each app*

RQ2 What is the precision of the energy improvement reported by EARMO ?

- *Measure the battery life duration before and after refactoring*

RQ3 Can EARMO generate useful refactoring solutions for developers?

- *We conduct a qualitative study with the developers of the apps studied to know their take on the refactorings proposed*

Results

RQ1 To what extent can EARMO correct anti-patterns?

- ***EARMO corrected*** *a median of* ***85%*** *of* ***anti-patterns***

RQ2 What is the precision of the energy improvement reported by EARMO ?

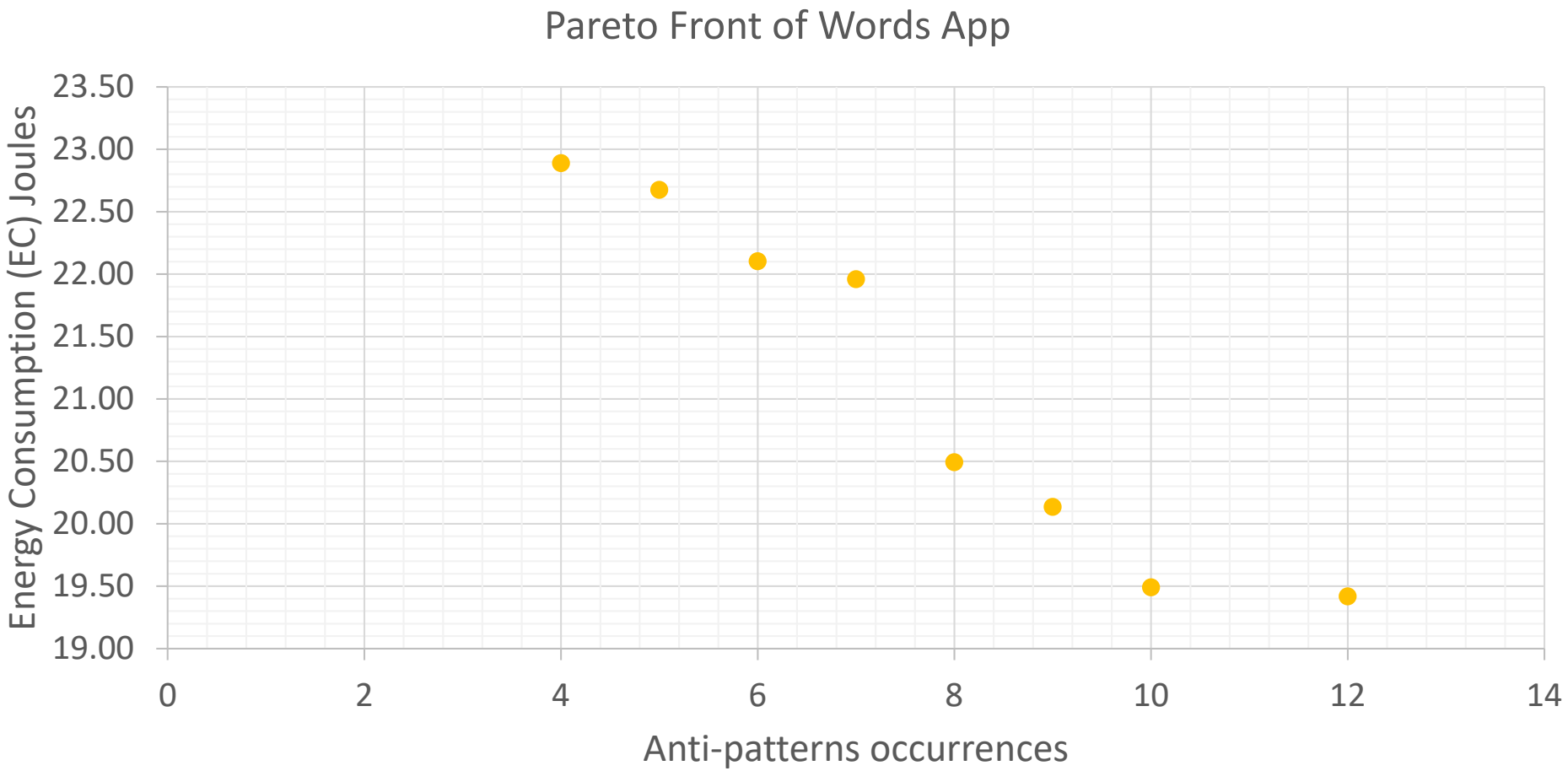
- ***EARMO*** *extended up the battery life of a multimedia app by up to* ***29 minutes***

RQ3 Can EARMO generate useful refactoring solutions for developers?

- ***Developers*** *found* ***68%*** *of the* ***refactorings suggested*** *by* ***EARMO very relevant***

Quality design is improved while controlling for energy consumption

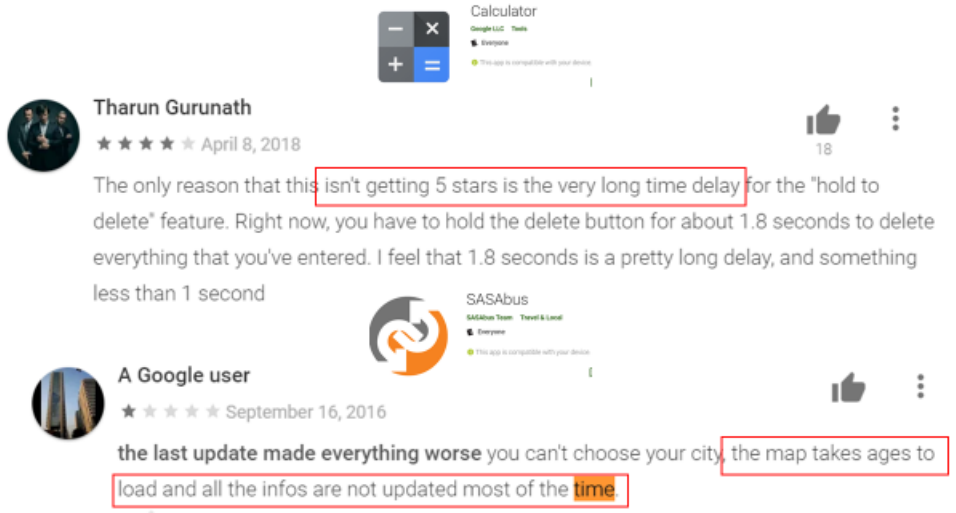
EARMO allows developers to choose the best trade



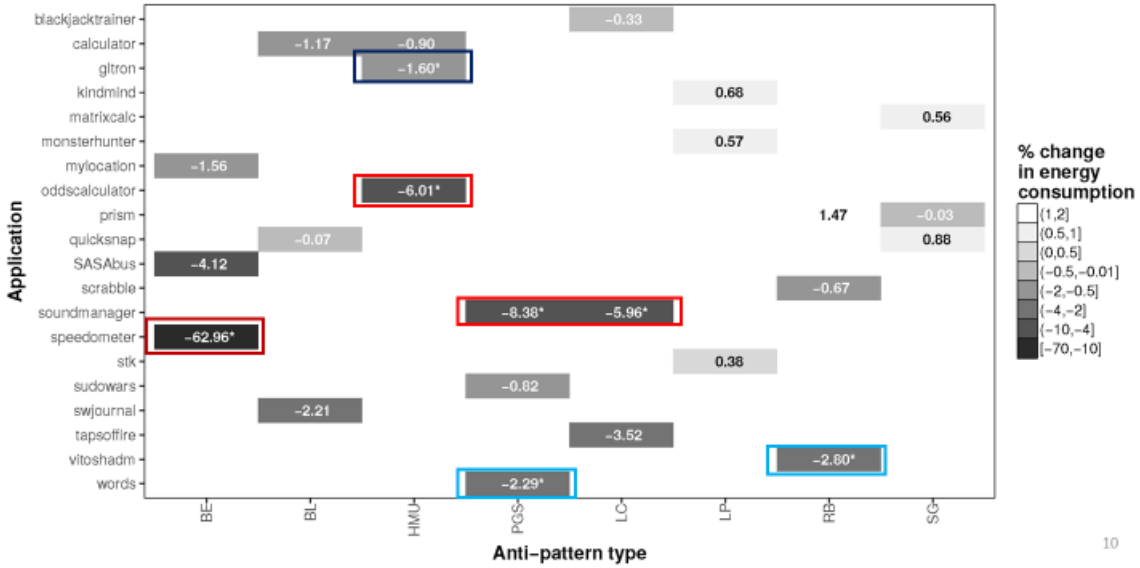
Feedback from Developers

- All developers surveyed considered refactoring to be useful
- The most common refactorings they perform are move method, inline class, extract class, collapse hierarchy and extract interface
- The refactoring type with the highest acceptance from EARMO is *Inline private getters and setters*
- Developers are concerned about the portability or their design (Refactorings like *RHwAM* that requires the use of specific Android API)
- The refactoring tool has to consider the codebase of paid and freeversions. (Speculative generality on free version of Calculator)

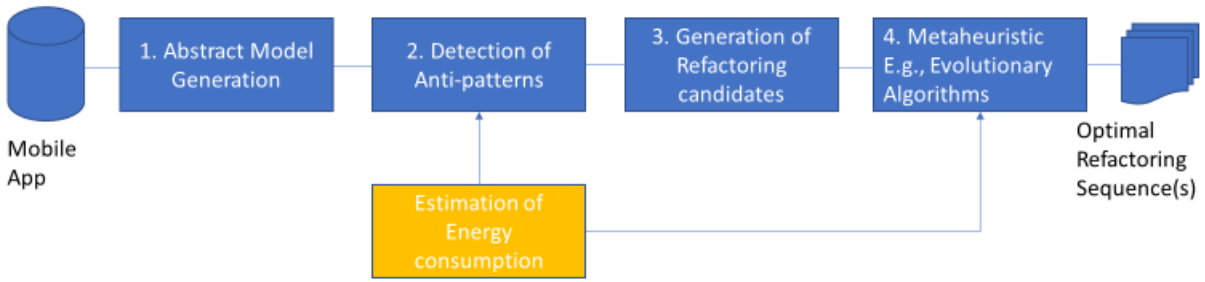
Reliability and quality is crucial



Anti-patterns impact on energy efficiency



Refactoring mobile apps while controlling for energy efficiency (EARMO)



Results

RQ1 To what extent can EARMO correct anti-patterns?

- ***EARMO corrected*** *a median of* ***85%*** *of* ***anti-patterns***

RQ2 What is the precision of the energy improvement reported by EARMO ?

- ***EARMO*** *extended up the battery life of a multimedia app by up to* ***29 minutes***

RQ3 Can EARMO generate useful refactoring solutions for developers?

- ***Developers*** *found* ***68%*** *of the* ***refactorings suggested*** *by* ***EARMO very relevant***

Quality design is improved while controlling for energy consumption

View publication stats