

A Preliminary Study of Android Refactorings

Anthony Peruma

Rochester Institute of Technology

Rochester, New York, USA

axp6201@rit.edu

R·I·T

B. THOMAS GOLISANO
*College of COMPUTING AND
INFORMATION SCIENCES*

May 2019



Maintenance is essential for survival!



3.3 million (and growing) apps on the Google Play Store ^[1]



Consumers are spoiled for choice
The better the app, the higher (and longer) the usage



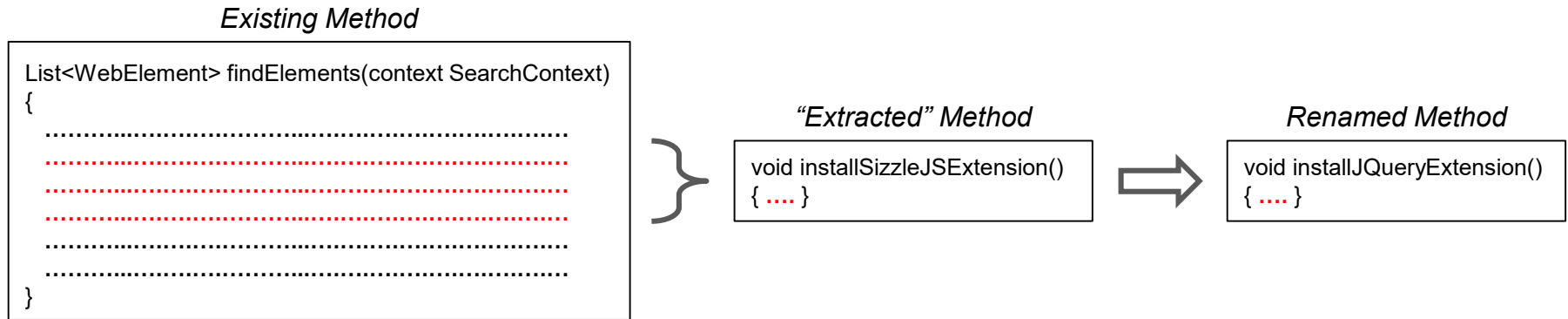
Apps that don't evolve will be left behind



^[1] I. Malavolta, R. Verdecchia, B. Filipovic, M. Bruntink and P. Lago, "How Maintainability Issues of Android Apps Evolve," 2018 ICSME

Code refactoring is an integral part of maintenance

“a disciplined technique for **restructuring an existing body of code**, altering its internal structure **without changing its external behavior**” -- *Martin Fowler*



Refactoring is rework, and rework **hurts productivity** and can be **error prone**

Research goal, objective & questions

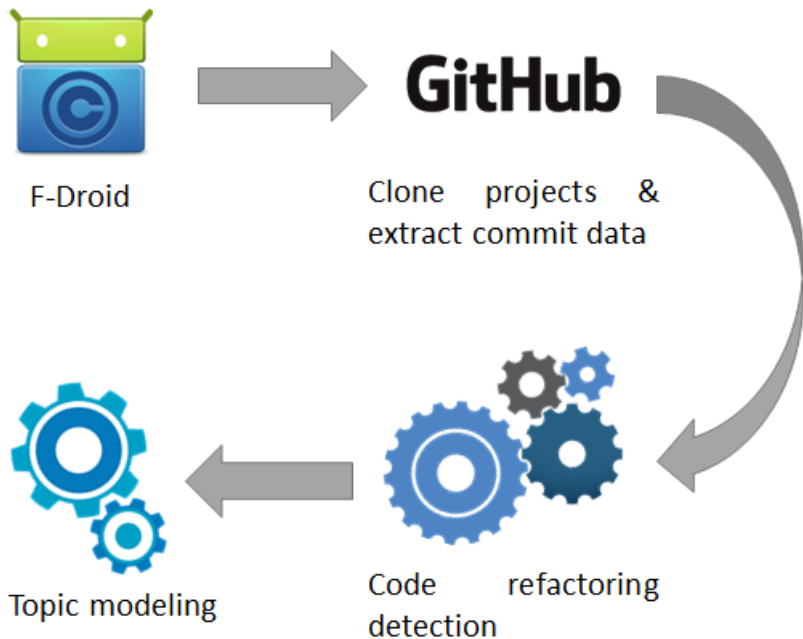
Goal: understand the refactoring habits of app developers and how they differ from traditional system developers

Objective: Better inform/educate app developers on the types of rework they would face; this would lead to better preparedness during app design and implementation

Research Questions:

- What are the most common refactoring operations Android app developers apply on their source code elements?
- What are the reasons that motivate Android developers to refactor their source code?

Methodology



Projects: 1,028

Avg. commits per app: 118.2

Avg. developers per app: 3.8

Avg. refactoring commits per app: 14.9

Avg. refactoring operations per app: 47.8

Avg. refactoring developers per app: 1.4

Tools/Technologies:

Refactoring detection: RefactoringMiner

Topic Modeling: latent Dirichlet allocation (LDA); Python's gensim library

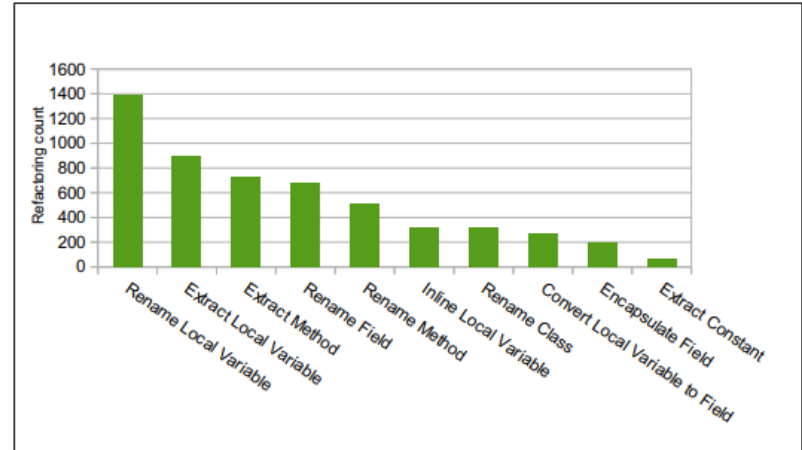
RQ1: Most common refactoring operations

RefactoringType	Count	Percentage
Rename Attribute	18769	12.03%
Rename Method	17072	10.94%
Extract Method	15597	9.99%
Move Class	15209	9.74%
Move Attribute	15043	9.64%
Rename Variable	14375	9.21%
Move Method	9453	6.06%
Rename Parameter	8630	5.53%
Extract Variable	7738	4.96%
Pull Up Method	4977	3.19%

Refactoring Type	Count	Percentage
Renames	65448	41.93%
Non-Renames	90625	58.07%

Elements of UI based datatypes tend to undergo more refactorings - TextView, View, Intent, Context

Popularity of refactorings in traditional Java systems



Stas Negara, Nicholas Chen, Mohsen Vakilian, Ralph E. Johnson, and Danny Dig. 2013. A comparative study of manual and automated refactorings. ECOOP'13

Demonstrates the need to **study Android apps separately from traditional systems**

RQ2: Reasons for code refactoring

LDA based analysis of refactoring commit messages to indicate why app developers refactor their code



Improve code comprehension

Terms *: *renam, code, method*

Example: *"Rename member variables as per README's coding style"*



Change in functionality

Terms *: *updat, add, remov, refactor*

Example: *"Update the progress dialog text when an open port..."*



Resolve defects

Terms *: *fix, bug, issu, improv, chang*

Example: *"Bug fixes to pseudo cursor requery()"*

* Stemmed terms

Other observations:

- Developers knowingly admit that the operations they performed are refactoring activities
Example: "Refactored seralize/deserialize (plain objects): moved to Reference..." when performing an Extract and Move Method operation
- Developers rename source code elements as part of code cleanup activities (e.g., "cleanup from all the re-org") or to comply with coding standards/guidelines ("Rename member variables as per README's coding style")

Summary

A preliminary study on Android apps that:

- Explored the types of refactoring operations performed by app developers
 - Provided a high-level comparison against traditional Java systems
- Highlighted the motivations that drive app developers to refactor their code

Future work:

- As a preliminary study, this work has opened up multiple avenues for future work
- A deep-dive investigation into the renaming habits of app developers

Artifacts from this study are available at: <https://sites.google.com/g.rit.edu/refactoring>

Thank You!

