Study on improving code design and energy efficiency of a mobile app

Dear developer, we are performing an empirical study at Polytechinique Montreal to study the usability of a new approach called EARMO, which proposes refactorings to remove Object-oriented and energy anti-patterns.

We contact you as one of your application (Calculator) was used to test our approach.

The survey will take between 10-15 minutes

The survey is anonymous and the collected data will be kept confidential

Thanks for your contribution

Sincerely,
Rodrigo Morales (rodrigomorales2@acm.org).
Ruben Saborido (ruben.saborido-infantes@polymtl.ca)

*Required

Agreement

You have been selected for this survey because of your potential experience in software development in industry.

Your participation in this survey is voluntary. You can therefore refuse to participate in the survey or refuse to answer some or all of the questions in the questionnaire. At any point before or during the survey, you can withdraw from the survey without justification or penalty whatsoever. In any case, you can mention this survey to your peers but please do not include details about its form and/or content to avoid biasing their participation.

You will not derive any benefits from your participation in this project. You will not receive any financial compensation for your participation in this survey. However, the knowledge acquired thanks to your participation will help us advance the state of the art in software engineering.

Your participation does not entail any additional risk than those to which you are subject in your regular daily activities.

Your participation is necessary to collect ANONYMOUS data regarding your experience and working habits with software refactoring. Your participation will allow us to collect the data to understand the perspective of practitioners about refactoring, and how it is performed in the industries. Your participation will require approximately 20 minutes of your time.

To preserve your anonymity, the questionnaire does not include any nominative or coding information. Your participation will allow us to collect three types of data:

1. Software Refactoring
2. Object-oriented anti-patterns
3. Android anti-patterns

This data will remain strictly confidential within the legal limits. The data will be kept under lock in an office at Polytechnique Montreal. The researchers will use all of the data for the sole objectives of the survey briefly described above.

The anonymous data that will be collected will be used in research papers and help students to complete their M.Sc. and/or Ph.D. theses. No data that could lead to your identification will be published. For monitoring and control purposes, the data could be consulted by an agent of the Research Ethics Board of Polytechnique Montreal or by an agent of a granting agencies. The data will be kept for a 7-year period, after which it will be destroyed.

1. **Unless you indicate otherwise, we would like to reserve the right to summarize or repeat your answers verbatim in our forthcoming paper** *
   Mark only one oval.
   - Accept
   - Reject

2. **Would you like to continue the survey?** *
   Mark only one oval.
   - Yes
   - No *Stop filling out this form.*

Software Refactoring

"A change made to the internal structure of software to make it easier to understand and cheaper to modify without changing its observable behavior"- M. Fowler

3. **1C. Do you perform software refactoring?**
   Mark only one oval.
   - Yes
   - No

4. **1O. If you answer "No" to the previous question, could you explain why?**

5. **2O. Off the top of your head could you name the three refactorings that you perform most often**
   Answer only if it is applicable for you

6. **1M. Refactoring is useful for me when I...**
Answer only if it is applicable for you
Mark only one oval per row.

	Completely Agree	Agree	Neither Agree nor Disagree	Disagree	Completely Disagree
Correct faults					
Improve code design					
Implement enhacements					
Interface with other software					
Adapt apps (so that different hardware, software, or system features can be used)					
Improve energy consumption					

7. **2M. When you perform refactoring...**
Mark only one oval.

- You do it sporadically, when it is needed . i.e., root channel refactoring
- You do it regularly, interleaving refactoring with other tasks. i.e., floss refactoring
- You do not perform refactoring at all

8. **3M. When you find a refactor opportunity, you normally...**
Mark only one oval.

- Refactor the code immediatly
- First study the side-effects, and then perform refactoring
- Discuss with another team member your finding
- Postpone the refactoring for another time (e.g., refactoring session)
- Assign someone to take care of it
- Ignore it

9. **3O. If you selected "Assign someone.." or "Ignore it" to the previous question do you mind explaining why?**

10. **4M. Do you use any refactoring tool support?**
Mark only one oval.

- Yes, I use a tool integrated in my IDE
- No, I manually refactor the code
- Other:

Antipatterns

Anti-patterns are poor design choices that hinder software maintainability, e.g. Blob, Large Class, Long Method, etc. For more info take a look at:
http://sourcemaking.com/antipatterns/software-development-antipatterns
http://sourcemaking.com/refactoring/bad-smells-in-code

In addition to this, there are some Android practices that developers can avoid to improve the performance of their apps. We refer to them as Android anti-patterns in the following.
For more info take a look at
https://developer.android.com/training/articles/perf-tips.html

11. **5M. What is the importance that you assign to the task of removing anti-patterns?**
Mark only one oval.

- ◯ Extremely Important
- ◯ Important
- ◯ Moderately Important
- ◯ Somewhat Important
- ◯ Not very Important

12. **1X. Rank the importance of removing the following anti-patterns**
Please refer to the links provided at the top for more info about the definition of the Anti-patterns
Mark only one oval per row.

	Extremely Important	Important	Moderately Important	Somewhat Important	Not important
Lazy Class	◯	◯	◯	◯	◯
Long Parameter List	◯	◯	◯	◯	◯
Refused bequest	◯	◯	◯	◯	◯
Blob	◯	◯	◯	◯	◯
Collapse Hierarchy	◯	◯	◯	◯	◯
Binding resources too early (Android)	◯	◯	◯	◯	◯
Private getter and setters (Android)	◯	◯	◯	◯	◯
HashMap Usage (Android)	◯	◯	◯	◯	◯

13. **4O. Is there any other anti-patterns that you consider important to correct?**
Yes? Please specify

....

....

....

....

Refactoring suggestions for Calculator App

EARMO (our automated approach) generated a list of refactoring suggestions to apply in your app to remove anti-patterns and improve energy efficiency. We are interested to know your opinion about some of them

After applying the complete list, EARMO removed 88% of anti-patterns and improve the energy consumption more than 7% of your app.

If you are interested, we can provide you the complete list of refactorings and the refactored source code (More details at the end of the survey).

R46 Collapse Hierarchy

We found an instance of Speculative generality anti-pattern in class: com.android2.calculator3.Calculator that extends class com.android2.calculator3.MatrixCalculator.

In fact the body of Calculator is empty. EARMO Suggest to remove the abstract modifier of MatrixCalculator, remove Calculator and update the Android Manifest file and all there references that point to Calculator to point to MatrixCalculator. By doing this, the design complexity is reduced, and we get rid of a class that does not bring any new functionality of the system.

References.
https://sourcemaking.com/refactoring/collapse-hierarchy

14. **Do you accept this refactoring?**
Mark only one oval.

- ◯ Yes (Jump to next refactoring)
- ◯ No

15. **If you select no, could you explain the reason why?**

16. **Is there any modification that you suggest for the the proposed refactoring?**

R194 Inline private getter/setter

We found the method resetGraph() to be called internally in class GraphingCalculator. This method only calls mMiniGraph.zoomReset().

EARMO proposed to called directly mMiniGraph.zoomReset(), in all resetGraph() invocations (2 occurrences), and remove resetGraph() . By doing this we improve the performance by removing an extra virtual method call in the Android Virtual Machine.

Android Developers performance guide.
https://developer.android.com/training/articles/perf-tips.html#GettersSetters

17. **Do you accept this refactoring?**
Mark only one oval.

- ◯ Yes (Jump to next refactoring)
- ◯ No

18. **If you select no, could you explain the reason why?**

19. **Is there any modification that you suggest for the the proposed refactoring?**

R38 Move method

We find the class DisplayOverlay to be a candidate instance of Blob (Large class, with number of method + attributes = 126). Hence, it makes sense to distribute the functionality among other classes. Hence, EARMO proposed to move the method setFade(android.view.View) to class GraphView.

20. **Do you accept this refactoring?**
Mark only one oval.

- ◯ Yes (Jump to next refactoring)
- ◯ No

21. **If you select no, could you explain the reason why?**

22. **Is there any modification that you suggest for the the proposed refactoring?**

R47 Inline private getter/setter

We found the method setmFormulaEditText(com.android2.calculator3.view.FormattedNumberEditText) to be called internally in class DisplayOverlay. This method sets the value for the attribute mFormulaEditText. However mFormulaEditText can be accessed directly in the class avoiding the creation of a virtual method called by the Android Virtual machine.
EARMO proposed to inline this method, following the Android Developers performance guide.
https://developer.android.com/training/articles/perf-tips.html#GettersSetters

23. **Do you accept this refactoring?**
Mark only one oval.

- Yes (Jump to next refactoring)
- No

24. **If you select no, could you explain the reason why?**

25. **Is there any modification that you suggest for the the proposed refactoring?**

R43 Inline private getter/setter

We found the method getTranslateState to be called internally in class DisplayOverlay. This method returns the attribute mState. However mState can be accessed directly in the class avoiding the creation of a virtual method called by the Android Virtual machine.
EARMO proposed to inline this method, following the Android Developers performance guide.
https://developer.android.com/training/articles/perf-tips.html#GettersSetters

26. **Do you accept this refactoring?**
Mark only one oval.

- Yes (Jump to next refactoring)
- No

27. **If you select no, could you explain the reason why?**

28. **Is there any modification that you suggest for the the proposed refactoring?**

R18 Move method

We find the class DisplayOverlay to be a canddiate of Blob class (Number of method + attributes = 126). Hence, it makes sense to distribute the functionality among other classes. Hence, EARMO proposed to move the method isCollapsed() to class GraphView.

29. **Do you accept this refactoring?**
Mark only one oval.

- Yes (Jump to next refactoring)
- No

30. **If you select no, could you explain the reason why?**

31. **Is there any modification that you suggest for the the proposed refactoring?**

R57 Move method

We find the class com.android2.calculator3.BasicCalculator to be a candidate of Blob class (Number of method + attributes = 82). Hence, it makes sense to distribute the functionality among other classes. Hence, EARMO proposed to move the method cleanExpression(java.lang.String) to class com.android2.calculator3.view.FormattedNumberEditText.

32. **Do you accept this refactoring?**
Mark only one oval.

- Yes (Jump to next refactoring)
- No

33. **If you select no, could you explain the reason why?**

34. **Is there any modification that you suggest for the the proposed refactoring?**

About you

35. **6M. What is your age?**
Select an interval
Mark only one oval.

- 18 to 24
- 25 to 34
- 35 to 44
- 45 to 54
- 55 to 64

36. **7M. How many years of developing mobile apps do you have?**
Select an interval
Mark only one oval.

- ◯ <1 year
- ◯ 1-2 years
- ◯ 3-4 years
- ◯ 5-9 years
- ◯ 10 years or more

37. **5O. What is the primary programming language that you use?**
e.g., Java

38. **8M. Which development IDE do you normally use?**
Mark only one oval.

- ◯ Eclipse
- ◯ Android Studio
- ◯ Other:

39. **9M. How many people is involved in the development of Calculator app**
Mark only one oval.

- ◯ 1
- ◯ 2-4
- ◯ >5

40. **6O. Company:**
Your name in Google Play.

41. **10M. Are you interested in receiving the complete list of refactorings suggestions and the refactored code?**
Mark only one oval.

- ◯ Yes
- ◯ No

Powered by