Bachelor Thesis

May 19, 2017

# GUI usability and testing of mobile applications

## Example subtitle

**Lucas Pelloni**

of 18.03.1993, Switzerland (13-722-038)

**supervised by**

Prof. Dr. Harald C. Gall
Dr. Sebastiano Panichella
Giovanni Grano (PhD student)

University of Zurich[UZH]

s.e.a.l.
software evolution & architecture lab

Bachelor Thesis

# GUI usability and testing of mobile applications

Example subtitle

**Lucas Pelloni**

University of Zurich[UZH]

s.e.a.l.
software evolution & architecture lab

**Bachelor Thesis**

**Author:** Lucas Pelloni, lucas.pelloni@uzh.ch

**URL:** https://github.com/lucaspelloni2/BA_PROJ

**Project period:** 08.01.2017 - 08.07.2017

Software Evolution & Architecture Lab
Department of Informatics, University of Zurich

# Acknowledgements

# Abstract

# Zusammenfassung

# Contents

# List of Figures



# List of Tables

# List of Listings

# Chapter 1

# Introduction

## 1.1 Context

## 1.2 Motivation

## 1.3 Motivation Example

## 1.4 Research question

### Subsubsection


**Figure 1.1**: imgs/seal logo

### 1.4.1 Subsection

**Paragraph.** Always with a point.

```
/**
 * Javadoc comment
 */
public class Foo {
   // line comment
   public void bar(int number) {
```

```
        if (number < 0) {
            return; /* block comment */
        }
    }
}
```

**Listing 1.1**: An example code snippet

# Chapter 2

# Related Work

In the following two sections, I summarize the main related works on *automated testing tools for Android apps* and on *the broadly usage of user reviews from app store in Software maintenance activities*. An overview of the recent research in the field can be found in the survey by Martin *et al.* [9].

## 2.1 Automated tools for Android Testing

Unlike traditional software, mobile applications are mainly exercised by user inputs.
In the mobile world, an exremely valid approach to ensure the realiability of these applications is the GUI[1] Testing.
In particular, in this kind of testing, each test case is designed and run in the form of sequences of GUI interaction events.
Depending on their exploration strategy, there are in general three approaches for creating a generation of user inputs on a mobile device [4, 7]: *random testing* [5, 7], *systematic testing* [8] and *model-based testing* [2, 3, 6].

### Fuzz testing

The most widely used tool in practice for testing Android apps with a random approach is *Monkey* [5], the official Android testing command-line tool directly provided by Google.
This tool simply generates, for the specified attached devices, pseudo-random streams of user events into the system, with the goal to stress the *AUT*[2] [5].
The kind of testing implemented by Monkey follows a black-box approach. Despite the robustness, the user friendliness [7] and the capacity of find out new bugs outside the stated scenarios [1], this tool may be inefficient if the *AUT* would require some human intelligence (*e.g.* a login field) for providing sensible inputs [7].
For instance, *Monkey* may cause highly redundant and senseless user events, which may be very time consuming.
Even though it would find out a new bug for a given app, the steps to reproduce the bug may be very difficult to recreate, due also of the randomness of the executed tests.

[1] Graphical User Interface
[2] Application Under Test

## 2.2 Usage of users reviews in Software maintenance activities

this is the part of the usage of users reviews!

# Bibliography


[1] What is monkey testing? types, advantages and disadvantages. .

[2] D. Amalfitano, A. R. Fasolino, P. Tramontana, B. D. Ta, and A. M. Memon. Mobiguitar: Automated model-based testing of mobile apps. *IEEE Software*, 32(5):53–59, 2015.

[3] W. Choi, G. Necula, and K. Sen. Guided gui testing of android apps with minimal restart and approximate learning. *SIGPLAN Not.*, 48(10):623–640, Oct. 2013.

[4] S. R. Choudhary, A. Gorla, and A. Orso. Automated test input generation for android: Are we there yet? (e). In *Proceedings of the 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, ASE '15, pages 429–440, Washington, DC, USA, 2015. IEEE Computer Society.

[5] Google. Android monkey. .

[6] M. Linares-Vásquez, M. White, C. Bernal-Cárdenas, K. Moran, and D. Poshyvanyk. Mining android app usages for generating actionable gui-based execution scenarios. In *Proceedings of the 12th Working Conference on Mining Software Repositories*, MSR '15, pages 111–122, Piscataway, NJ, USA, 2015. IEEE Press.

[7] A. Machiry, R. Tahiliani, and M. Naik. Dynodroid: An input generation system for android apps. In *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*, ESEC/FSE 2013, pages 224–234, New York, NY, USA, 2013. ACM.

[8] R. Mahmood, N. Mirzaei, and S. Malek. Evodroid: Segmented evolutionary testing of android apps. In *Proceedings of the 22Nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, FSE 2014, pages 599–609, New York, NY, USA, 2014. ACM.

[9] W. Martin, F. Sarro, Y. Jia, Y. Zhang, and M. Harman. A survey of app store analysis for software engineering. *IEEE Transactions on Software Engineering*, PP(99):1–1, 2016.