Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 110C (2017) 215–221

Procedia Computer Science

www.elsevier.com/locate/procedia

The 14th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2017)

Towards Automated Capturing and Processing of User Feedback for Optimizing Mobile Apps

Frank Elberzhager[a]*, Konstantin Holl[a]

[a]*Fraunhofer IESE, Fraunhofer Platz 1, 67663 Kaiserslautern, Germany*

Abstract

Mobile devices are nowadays ubiquitous and are heavily used for business and private purposes. Millions of apps exist that support users in multiple ways, e.g., for car navigation, fitness purposes, or messaging in our private lives, but also for business planning purposes and even for controlling whole business processes. Failures in mobile business apps can lead to dramatic consequences in terms of lost revenue, but also in terms of lost trust or even threats for human beings, and thus quality plays a crucial role. On the other hand, as software is nowadays one of the main drivers for innovation, fast delivery of new apps, respectively new functionality, is necessary, i.e., the time to market must often be short. However, in order to understand whether the quality is sufficient, and whether the functionality of the app serves the needs of the users, lean development approaches are emerging and propose the deployment of apps as a minimal viable product (MVP). Here, the app is provided with acceptable quality, but not with every feature, just with the main functionality. Based on such an MVP, early feedback from users is to be collected, which may be related to the quality of the app, but also include wishes and requests for new functionality. In order to analyze and draw conclusions from user feedback, we first have to understand what kind of feedback exists and how it can be interpreted, respectively how valid such feedback is. In this publication, we provide a classification of user feedback for mobile apps, derive feedback channels, and sketch how this can be used in a continuous process to improve mobile apps.

Peer-review under responsibility of the Conference Program Chairs.

Keywords: mobile; mobile apps; user feedback; Opti4Apps

* Corresponding author. Tel.: +49-631-6800-2248; fax: +49-631-6800-9-2248.
E-mail address: frank.elberzhager@iese.fraunhofer.de

1877-0509
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2017.06.087

1. Introduction

Today, software is part of everyone's lives. We have software in our cars, in our refrigerators and washing machines. We encounter more and more smart systems controlled by software, such as smart homes or smart meters. And almost everybody today has a smartphone for private purposes – but smartphones and apps are also more and more becoming a part of the business. They are used in a classical manner, for example, to call colleagues, but also for controlling whole business processes. Here, high quality of the apps used is necessary to avoid negative consequences, such as financial loss or lost reputation due to critical bugs.

Especially the mobile market is showing dramatic increases in growth. New mobile applications are being developed and shipped continuously. For instance, the number of mobile applications available in the Apple App Store continuously rose from 800 applications in the year 2008 to 1.5 million applications in the year 2015[1]. Until 2020, the mobile application market will grow from 70 billion US dollars in 2015 to 189 billion US dollars, according to current market research forecasts[2]. As software is a forcing driver for innovation today, short time to market is often a reasonable business goal for companies, and many companies have to analyze and decide how they can benefit from the mobile market. Furthermore, the mobile market has other specific characteristics. One important one is that users of mobile apps tend to remove an app with bad quality very quickly. In addition, they give negative feedback that influences further potential users to avoid using such an app. Whether they make an online purchase, stream videos and music, or check their bank balance – consumers demand flawless execution of the applications and technologies they utilize. According to a study based on 2,000 subjects[3], even small digital moments have a large impact. In addition, practitioners' portals proclaim that quality, not quantity, is the real mobile application problem[4] and that users have low tolerance for error-prone applications.

The power of the customer is much higher due to direct feedback. However, from another perspective, the good characteristic is that users of mobile apps are often willing to give feedback. The idea is to use such feedback from users as early as possible. For this, we assume that a software product, in our case a mobile app, is developed and deployed to users as a minimal viable product (MVP). An MVP can be understood as a conceptual product that helps developers get an initial impression of what should be implemented (e.g., as a paper prototype), but also as a running product consisting of a minimal set of features at a sufficient level of quality[5]. We follow the second definition, knowing that we have further degrees of freedom in terms of what a minimal set of features and a sufficient level of quality mean. This has to be decided in a concrete context. As mobile apps are often developed in short release cycles, the assumption often fits real development environments.

When such an MVP is delivered to a set of users, we expect to get feedback about the current quality, but also about whether we implemented the right features, or whether new feature requests occur. This means, we directly involve customers (more precisely, the customers' feedback) in new development sprints in order to improve the app with respect to quality, but also with respect to the scope of the features.

Before we can adapt the development methodology to given user feedback, we first have to understand what kind of feedback exists, what this feedback looks like, and which feedback channels are used by users. Our contribution in this paper is first a general overview of the Opti4Apps framework in order to motivate how feedback can improve an MVP mobile app. Second, we classify user feedback according to four dimensions, which improves the understanding regarding such feedback and allows selecting those kinds of feedback that should be considered in an improved development methodology with the goal of improving mobile apps early and fast.

2. Opti4Apps at a Glance

Before we describe the feedback classification, we will first present an overview of our Opti4Apps framework that consumes user feedback, analyzes such feedback, and provides suggestions for improving an MVP mobile app. Opti4Apps stands for *Optimization for Mobile Applications*. The innovative idea is to consider several feedback channels to identify in a valid manner future development directions and quality issues, and to get this feedback as automatically as possible. Figure 1 presents a graphical visualization.

The starting point is a mobile app that is built as an MVP. Such an app provides a reasonable set of implemented features, and may contain certain bugs. The level of quality and the implemented functionality are not defined, but rather need to be determined by the app developers. After the app is deployed, users can use the app. For this, a defined