Universiteit Leiden

ICT in Business

Usage-based measurement of user satisfaction in mobile applications

Name: Lazar Dimitrov
Student-no: s1143220

Date: 09/02/2015

1st supervisor: dr. Hans Le Fever
2nd supervisor: prof. dr. Joost Visser

Usage-based measurement of user satisfaction in mobile applications

Design and validation of a novel approach

Master Thesis

ICT in Business

Lazar Dimitrov

Academic Supervisor: dr. Hans Le Fever

Company Supervisors: dr. Pascal van Eck and prof. dr. Joost Visser

Software Improvement Group

Universiteit Leiden

Table of Contents

List of Figures

List of Tables

Abstract

The market of mobile applications, also called ‘apps’, has seen a dramatic growth in the last few years, quickly evolving into a multi-billion industry. Being an industry with considerably low entry barriers, in which it is theoretically possible for a single app developer to achieve commercial success, it is only natural that in just a matter of years it has turned into a highly saturated and increasingly competitive market. Apps are commonly distributed via app stores, where user-submitted quality ratings are displayed for every app. Maintaining a good app rating is one of the crucial factors determining an app’s visibility, popularity and ultimately – its success amongst the sea of competitors.

However, several major weaknesses in the currently adopted method for rating apps have become apparent and this has created a demand for alternative solutions. Building upon previous research connecting user satisfaction to usage; and capitalizing on the existence of technology allowing for efficient and automated tracking of apps’ usage, this study proposes a candidate solution. Namely, in line with design science principles, a usage-based method for rating satisfaction in apps has been designed and validated.

1. Introduction

1.1. Problem definition

The rating method currently utilized by app stores as a crowd-sourced indicator of user satisfaction is a manually submitted rating of apps on a 5-point scale (5-star rating). This mechanism is not a recent invention and is proven as a successful measure of user satisfaction with physical goods such as books and movies. Despite its prowess when used with those types of products, there has been an accumulation of evidence exposing this method's weaknesses in the context of mobile apps user satisfaction. In addition to inherent problems such as very low response rate (according to [4] the mean percentage of users who rated an app that they have downloaded is only 0.6%) and subjective nature, there are a number of weaknesses, which are distinct to the field of interest.

Realizing the importance of app ratings for developers, a significant number of app raters are using star ratings as a bargaining chip for added functionality. Even though they might be well satisfied with the quality of the existing functionality the app has to offer and enjoy using the app, they deliberately give a low or average rating score. In the comment accompanying their rating, such users openly declare that they would give a perfect rating only if the developer implements a specific functionality they are interested in.

The number of ratings submitted for a given app is among the factors determining the respective app's position in the app stores' ranking charts. In addition, potential app users base their decision on whether to download a specific app in the first place not only on the app's mean rating, but also on the number of individual ratings this average rating is based upon. Being aware of those phenomena and also hoping that increased response rate can statistically negate the effects of subjectivity and intentional abuse by app users, some app developers proactively encourage their users to rate their apps. This practice is however a double-edged sword. Because it relies on prompting users with a popup dialog, some users are annoyed by it. This has led to the creation of a blog (effyr.tumblr.com) dedicated on hunting down such apps and preaching "punishing" the developers for annoying app users by giving 1-star ratings.

On another account, companies such as BestReviewApp.com, BuyAppStoreReviews.com and SafeRankPro.com capitalize on the struggle app developers are experiencing with ratings. These websites offer app developers the ability to buy "genuine" 5-star ratings in order to improve their apps' average rating and of course, their apps' visibility. Such services are not detected by the app stores as fraudulent, because the companies offering them are paying real app users with real mobile devices for actually downloading and rating the respective apps.

Being prone to tampering by users, developers and their competitors, star ratings of mobile apps lose their credibility as the best possible way of measuring user satisfaction. By serving the purposes described in the previous paragraphs, and by being turned into a business on their own, star ratings are deviating from their originally intended purpose.

The purpose of this study is to design a viable alternative to the star rating mechanism and rigorously validating its value as a successful substitute by applying a design science research method.

1.2. Solution proposition

For physical products such as books and videocassettes no practical solution for inexpensive automated tracking of usage duration and usage patterns[1] is known to exist. However, as far as software in general and mobile apps in particular are concerned, information about actual usage times and usage patterns of individual app users is possible to be obtained, aggregated and summarized to meaningful statistical figures, even if the respective app is in use by millions of users.

Usage analytics packages such as Google Analytics and Mixpanel, designed to perform those tasks, have seen adoption by mobile app developers in recent years and are being utilized as a tool for making informed decisions about marketing, managerial and financial matters.

After an extensive investigation of the current technology landscape and consultations with domain experts, usage analytics packages have been identified as a candidate for a suitable

[1] This thesis investigates the idea of replacing user-supplied star ratings by a rating derived from automated tracking of actual app usage.

[2] A *knowledge problem* [25] is a difference between what a researcher knows about the world and what he or she would like to know. Knowledge problems can be solved by asking others, searching the literature, or doing

basis for developing an alternative app rating mechanism. This mechanism will evaluate user satisfaction based on the degree of fit between actual app usage time and usage patterns and predefined benchmark levels of those measures.

For the initial validation phase of the proposed mechanism, the benchmark levels will be derived using a technique called 'micro benchmarking' (explained in Section 3.3.2). However, at a later hypothetical stage when this mechanism is adopted as a ranking method by an app store, that app store will have the power to effectively regulate the benchmark levels of app usage of individual apps.

For example benchmark level for usage time and usage patterns can be decided by app store staff members and be defined per app category. In some app stores such as the Apple App Store every app submitted is exhaustively investigated and tested by app store staff members before it is being approved for publication. Such app stores can simply incorporate an additional step in their app acceptance tests – approval or modification of the benchmark levels for app usage time and usage patterns set by the app developer.

1.3. Research questions

As mentioned in the previous sections, the current method for rating apps implemented by the app stores is far from perfect. A goal of this paper is to develop an alternative, based on actual usage. The logical question then is how appropriate is actual usage as a basis for deriving a user satisfaction rating for mobile apps. Therefore, the main research question this study aims to answer is:

Can user satisfaction of mobile apps be measured based on actual usage rather than asking the user's opinion?

No method for usage-based rating of satisfaction in mobile apps is known to exist. Therefore, an essential task for answering the main research question is to propose a particular way of measuring user satisfaction in apps, which is a design problem[2]. For this

[2] A *knowledge problem* [25] is a difference between what a researcher knows about the world and what he or she would like to know. Knowledge problems can be solved by asking others, searching the literature, or doing research. *Design problems* [26], in turn, are engineering problems, in which the researcher searches for an

reason, this study employs a design science perspective. Specifically, it follows the engineering cycle as presented by Wieringa [2]. Using this cycle, the main research question can be divided into several sub-questions categorized as either knowledge problems (KP) or design problems (DP).

RQ0 (KP): What are the problems of the conventional method for measuring user satisfaction by manually submitted star ratings?

RQ1 (DP): What would the properties and structure of a usage-based user satisfaction measurement mechanism that meets the identified requirements be?

RQ2 (KP): What is the reliability of usage-based measurement of user satisfaction?

RQ3 (KP): To what extent can usage-based measurement solve the inherent problems of opinion polling?

RQ4 (KP): What factors will influence app developers' decision to adopt usage-based measurement?

RQ0÷4 are derived in compliance with the design science principles and each one of them can be mapped to a specific stage of the engineering cycle, defined by Wieringa [2]. **Figure 1** shows the research questions ***RQ0÷4*** in relation to their respective stage in the engineering cycle as well as research methods that are to be applied for each particular stage.

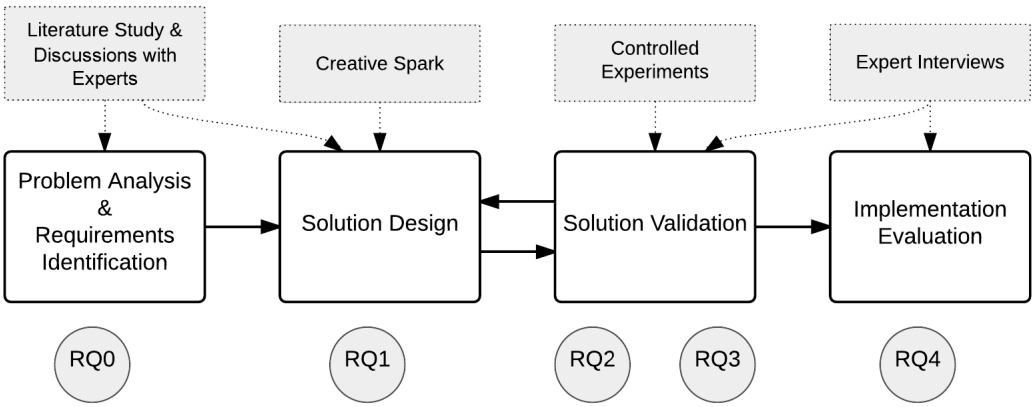

Figure 1: Engineering cycle stages, research questions and research methods

improvement of the world with respect to some goals. The evaluation criteria for answers to both kinds of problems are quite different: *truth* in the case of research problems, *goal achievement* in the case of design problems.

1.4. Research design

The approach selected for undertaking this study and answering the research questions in accordance with the design science principles is outlined in Table 1. Every question is divided into knowledge and / or design problems and actions (research activities) that contribute towards answering the respective research question.

Research Design	**Explanation:** KP = Knowledge Problem / Knowledge Question A = Action / Research Activity DP = Design Problem / Design Question
RQ0 (KP): What are the problems of the conventional method for measuring user satisfaction by manually submitted star ratings? KP: What problems of the conventional method are discussed in the literature? KP: What are the requirements an alternative solution should satisfy? A: Study the existing literature A: Discuss the problems and the requirements with experts	
RQ1 (DP): What would the properties and structure of a usage-based user satisfaction measurement mechanism that meets the identified requirements be? KP: What existing frameworks for user satisfaction known in the literature can be utilized? DP: What measures are necessary to ensure that the solution will satisfy the requirements A: Study the existing literature and discuss with experts the desired properties A: Design and implement the prototype of the solution	
RQ2 (KP): What is the reliability of usage-based measurement of user satisfaction? KP: What is the repeatability of the results? KP: What is the correlation between conventional star-ratings and ratings derived via usage-based measurement? KP: What is the agreement between the two types of measurement? A: Conduct controlled experiments	

A: Perform the appropriate statistical tests on the results from the experiments, in order to solve the abovementioned knowledge problems
RQ3 (KP): To what extent can usage-based measurement solve the inherent problems of opinion polling? KP: To what extent the proposed solution actually meets the identified requirements? A: Conduct interviews with expects A: Provide additional analytical reasoning
RQ4 (KP): What factors will influence app developers' decision to adopt usage-based measurement? KP: What adoption criteria are suitable for assessing the probability of adopting a novel method? A: Study the existing literature A: Conduct interviews with stakeholders, in order to determine the factors contributing to their decision to adopt the proposed solution for measuring user satisfaction.

Table 1: Research design

	Chapters	Chapter 1	Chapter 2	Chapter 3	Chapter 4	Chapter 5	Chapter 6	Chapter 7
Research Questions	Research Question 0	x		x				
	Research Question 1		x	x	x			
	Research Question 2					x		
	Research Question 3					x	x	x
	Research Question 4						x	x

Table 2: Discussion and answers to the research questions. A chapter-oriented overview

The rest of Chapter 1 discusses the academic and business contribution of this study. Chapter 2 informs the reader about existing relevant theoretical models and frameworks. Chapter 3 provides a detailed description of the proposed rating method. Chapters 4 and 5 discuss the design of the controlled experiments, the gathered and calculated data, as well

as the results from its analysis. Chapters 6 and 7 deal with the interview design, results and findings. Finally, Chapter 8 provides a summary of the overall findings and discusses the study limitations and suggestions for future research.

Table 2 is designed to help the reader find where each of the research questions ***RQ0÷4*** is discussed.

1.5. Contribution

Academic relevance

With the mobile app industry advancing in maturity and market significance, its distinctive peculiarities and differences with the broader concepts of information technology and information systems are becoming more and more apparent. Existing research in the field of mobile applications is scarce and focuses predominantly on studying the risks involved with mobile apps (problems related to security of personal information as well as to psychology and behavioural science - such as buying preferences of app users and decision-influencing factors.

Research studies [5, 6] attempt to understand and systematize app usage patterns, but in relation to factors other than user satisfaction.

There are studies [3, 4] that identify problems with the widely adopted app rating and reviewing methods, but no research targeted at validating the viability of any alternative methods seems to have been conducted.

Hopefully this research can spark enthusiasm for scientific validation of other possible solutions of the problem or further research on the discussed solution.

Business relevance

As already discussed the current situation with user satisfaction measurement is not ideal. The widely used methods suffer from inherent subjectivity and low participation rate. In addition these methods are prone to tampering and abuse from app users and app developers. Furthermore, the star rating system is being exploited for purposes other than its original intended purpose (as a marker of user satisfaction). As a result many developers

and users are suffering – the former from unfair competition and the latter – from lack of high quality feedback.

If the proposed alternative solution is found to be viable and capable of eliminating or minimizing the effects of the existing solution's weaknesses, its chances of being widely adopted increase significantly. This can potentially lead to a win-win situation for all parties involved (except for the companies selling 5-star ratings to developers).

2. Theoretical background

This chapter summarizes existing theoretical models and frameworks found to be related to user satisfaction (in general and in the context of software systems and mobile apps in particular) and its connection with actual usage. These theories are taken into consideration when trying to answer ***RQ1: What would the properties and structure of a usage-based user satisfaction measurement mechanism that meets the identified requirements be***. In addition, there is a section (2.3) that discusses adoption criteria for information technology methodology, which are utilized for answering ***RQ4: What factors will influence app developers' decision to adopt usage-based measurement***.

2.1. User satisfaction and relation to actual usage of software applications

The main research question ***Can user satisfaction of mobile apps be measured based on actual usage rather than asking the user's opinion*** implies that user satisfaction could be somehow proportionally related to actual usage of a mobile application. Bailey and Pearson ([12]) define satisfaction in a given situation as "the sum of one's feelings or attitudes toward a variety of factors affecting that situation". Therefore, it is defined as the sum of **m** users' weighted reactions to a set of **n** factors.

$$Satisfaction = \sum_{i=1,j=1}^{i=m,j=n} W_{ij}R_{ij}$$

Where:

Rij **-** The reaction to factor ***j*** by individual ***i***.

Wij - The importance of factor ***j*** to individual ***i***.

This model suggests that satisfaction is the sum of one's positive and negative reactions to a set of factors and allows for placing the feelings of individuals somewhere between a "most positive" and a "most negative" reaction. There are two central requirements for the implementation of this model. First, the set of factors comprising the domain of satisfaction

must be identified. Second, a method for scaling an individual's reaction to those factors must be devised.

As far as the context of mobile applications and their users' satisfaction is concerned, the purpose of this study is not to try to identify and weigh all relevant factors, but instead, to study the reliability of automatically measurable usage statistics as an indicator for user satisfaction. In other words, this research is exclusively focused on the relationship between user satisfaction in mobile apps and factors related to actual usage that are automatically measurable, while other possible satisfaction influencers remain beyond its scope.

Actual usage frequency and patterns might provide valuable insights on how satisfied a particular user is. For example, user session duration can directly or indirectly reflect the level of user satisfaction. However, depending on the category of the particular app and its main purpose, longer user session duration could be an evidence for either a relatively satisfied or a relatively unsatisfied user.

For example, for applications that are relatively small in terms of complexity and functionality and are time and performance-critical, such as a calculator app for instance, short usage time is actually desirable. Users of such applications expect the app to perform its calculations and deliver its content / results as fast as possible, thus letting them waste as little time as possible using the app and focus on other tasks.

On the contrary, for apps that are used primarily in one's free time, such as games or magazine apps, longer user sessions will most probably be indicative of higher user satisfaction levels. If they find the content or the gameplay highly engaging and immersive, users could spend hours using the respective app.

Just like satisfaction is influenced by multiple factors with varying importance, usage of a software application is defined by several indicators (user session duration being one of them), which in the context of satisfaction in a certain app can also have varying importance.

Another well-known framework is the one developed by DeLone and McClean. DeLone and McLean's Systems Success Model ([7]) has seen a wide adoption by other researchers, who have contributed to its validation and enrichment. Information systems use and user satisfaction are two of the six inter-related dependent variables contributing to information systems success. The model is presented on **Figure 2**.

They conceptualize use as: "frequency of use, depth of use, duration of use, appropriateness of use, system dependence, actual use, and self-reported use, among others" and argue that use and user satisfaction are closely inter-related. Positive experience with use inevitably leads to greater user satisfaction and similarly, increased user satisfaction raises the user's intention to use which leads to increased actual usage of the information system. Other studies ([9], [10], [11]) which adhere to the ISSM model of DeLone and McLean examine and prove the association between use and user satisfaction.

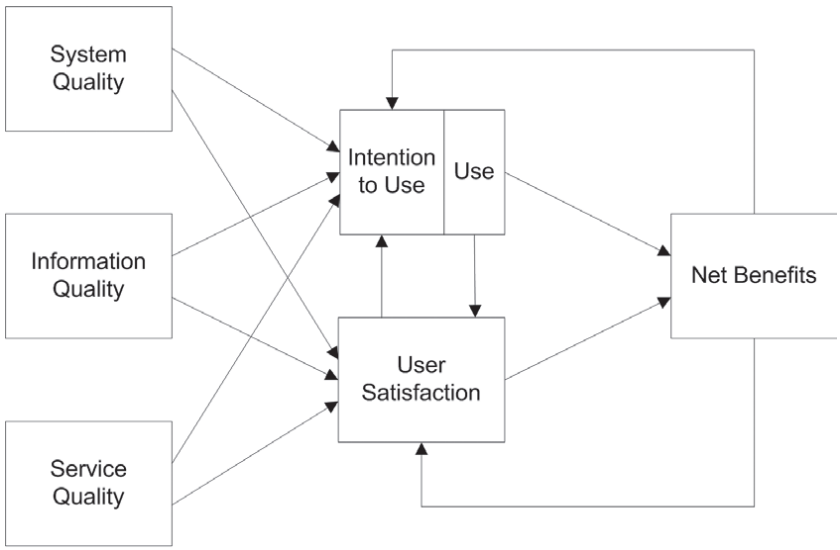

Figure 2: DeLone and McLean Information Systems Success Model, [7]

When systemizing all research adhering, validating and contributing to their ISSM model, DeLone and McLean have come up with the following list of use determining factors (or metrics) [7], [20]:

- Frequency of use
- Time / duration of use
- Number of accesses
- Usage pattern
- Dependency
- Appropriateness of use

In the context of mobile applications, all of these metrics are relevant, but the last two (dependency and appropriateness of use) are hardly measurable without interviewing a representative enough percentage of the population of app users. As far as appropriateness

of use is concerned, even such a technique might suffer from insufficient reliability, because of the variance in user experience levels and the probability that for example, a significant part of the users of a particular app are not experienced enough to provide an adequate enough estimation of how appropriately they are using the app.

2.2. SERVQUAL or Gaps Model of service quality

SERVQUAL, also called Gaps model for evaluating service quality, which was authored by Zeithaml, Parasuraman and Berry in the 80s of the 20th century, is widely adopted as a tool for performing gap analysis of the quality of a service provided by a certain company against the customer expectations for the particular service ([21], [22]). In this model service quality is defined as the gap (the difference) between expected and perceived levels of service performance (GAP5 in **Figure 3**).

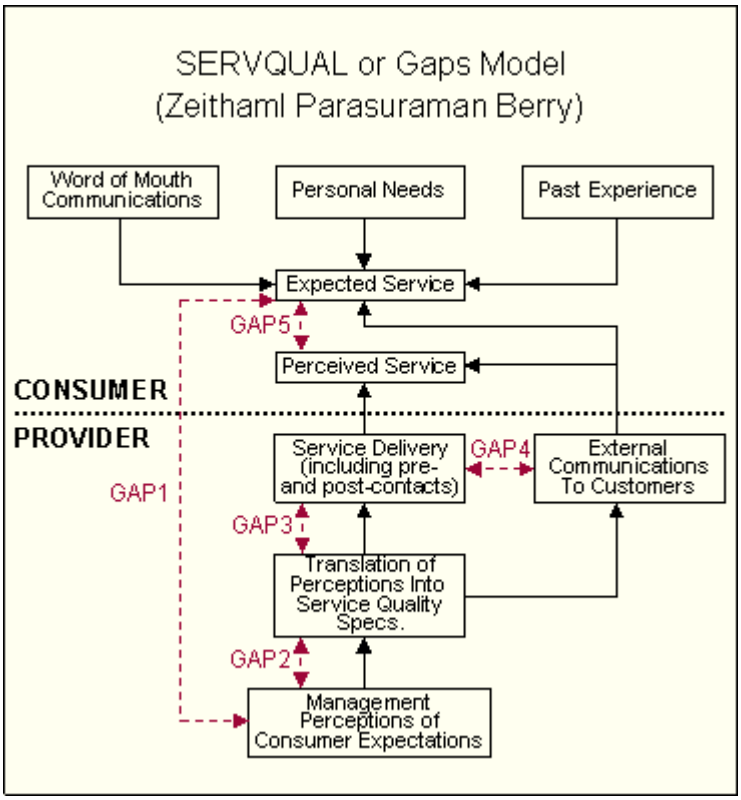

Figure 3: SERVQUAL / Gaps Model

It could be said that the primary goal of any given mobile application is to deliver a certain service to its users. For example, an application could enable the user to solve a particular problem by providing a set of functionalities and tools and this is essentially a service.

Therefore, concepts from a model, set around service quality, as perceived by consumers, could be utilized for measuring user satisfaction in mobile apps.

Despite the fact SERVQUAL is widely embraced, there have been numerous reasons for criticizing it over the years (summarized by Francis Buttle [21]). One particular criticism is that "SERVQUAL is inappropriately based on an expectations-disconfirmation model adopted in the customer satisfaction literature rather than an attitudinal model of service quality. In this literature, customer satisfaction (CSat) is operationalized in terms of the relationship between expectations (E) and outcomes (O). If O matches E, customer satisfaction is predicted. If O exceeds E, then customer delight may be produced. If E exceeds O, then customer dissatisfaction is indicated." Iacobucci et al ([23]) argue that there are conceptual and operational differences between service quality and customer satisfaction and conclude that the constructs "have not been consistently defined and differentiated from each other in the literature", but suggest that the two constructs may be connected in a number of ways.

This criticism, however poses no obstacle for using the model in creating a model for measuring user satisfaction in mobile apps (described in Section 3.2), which provides the basis for the solution proposed in this study. In this model each usage metric is rated, based on the difference between expected and observed values of the respective metric.

2.3. IT methodology acceptance criteria

In regards to ***RQ4: What factors will influence app developers' decision to adopt usage-based measurement***, Riemenschneider et al. investigate the applicability of five theoretical models for individual acceptance of information technology tools in the context of acceptance of methodologies in [13].

The authors argue that in comparison with adopting an IT tool, adopting a method is usually more mandatory than voluntary and tends to be more radical than incremental. Across the five models (**Table 3**) they have identified four determinants of usage intentions as significant for the context of methodology acceptance in at least one model: usefulness (all

five models), subjective norm (TAM2[3], TPB[4] and MPCU[5]), voluntariness (TAM2 and PCI[6]), and compatibility (PCI).

The four constructs found to be significant by Riemenschneider et al. can be used as criteria for assessing the developers' acceptance of the alternative app rating method proposed in this study. These criteria and their respective definitions are laid out in **Table 4**.

	Theoretical Models				
Construct	**TAM**	**TAM2**	**PCI**	**TPB**	**MPCU**
Usefulness	Usefulness	Usefulness	Relative Advantage	Attitude	Job Fit
Ease of Use	Ease of Use	Ease of Use	Complexity		Complexity
Subjective Norm		Subjective Norm		Subjective Norm	Social Factors
Affect					Affect
Voluntariness		Voluntariness	Voluntariness		
Compatibility			Compatibility		
Result Demonstrability			Result Demonstrability		
Image			Image		
Visibility			Visibility		
Perceived Behavioural Control - Internal				PBC - Internal	
Perceived Behavioural Control - External				PBC - External	Facilitating Conditions
Career Consequences					Career Consequences

Table 3: IT tool acceptance models and constructs, Riemenschneider et al. [13]

[3] TAM – Technology Acceptance Model, F.D. Davis, 1989
TAM2 – Technology Acceptance Model 2, V. Venkatesh and F.D. Davis, 2000
[4] TPB – Theory of Planned Behavior, I. Ajzen, 1985
[5] MPCU – Model of Personal Computer Utilization,
[6] PCI – Perceived Characteristics of Innovation, Moore and Benbasat, 1991

Criterion	Definition
Perceived Usefulness	The extent to which users believe that a method improve their job performance
Compatibility	The extent to which a method is compatible with existing norms or past experiences of potential users
Subjective Norm	The extent to which users think that people important to them (for example: colleagues, mentors, managers) would encourage them to adopt a method
Voluntariness	The extent to which users think that they will adopt a method voluntarily

Table 4: IT methodology acceptance criteria, Riemenschneider et al. [13]

2.4. Conclusion

In conclusion, the interrelation between usage and user satisfaction of software systems suggested by DeLone and McLean in their ISSM model has been examined and confirmed by other studies ([9], [10], [11]). This provides credibility to the notion that user satisfaction in mobile apps (which are essentially software systems) could be measured based on an examination of their usage patterns. Most of the usage determining factors identified by DeLone and McLean are automatically measurable and therefore – suitable as integral elements of an automatic system for rating user satisfaction, based on actual usage. In addition, Bailey and Pearson's formulation of satisfaction and the gap between expected and perceived service in the SERVQUAL model by Zeithaml, Parasuraman and Berry are key concepts in the algorithm utilized by the alternative rating method proposed in this research. The algorithm is described in detail in the next chapter (Chapter 3). Finally, the work of Riemenschneider et al. which based on previous research, summarizes the criteria relevant for assessing adoption probability for IT methodologies will be used for determining the major influencers for adoption of the proposed alternative rating solution.

3. Proposed alternative solution for measuring user satisfaction

This chapter is among the chapters focused on answering ***RQ1: What would the properties and structure of a usage-based user satisfaction measurement mechanism that meets the identified requirements be***. First of all, Section 3.1 summarizes the requirements that the proposed solution should fulfil. Secondly, unlike Chapter 2, which looks at what existing frameworks for user satisfaction can be utilized for proposing an alternative method for rating it, sections 3.2, 3.3 and 3.4 explain the hypothesis for the proposed solution's underlying model - a summation model for measuring user satisfaction, based on automatically measurable usage metrics (which are listed and explained in Section 3.5) that was developed.

3.1. Identified requirements for an alternative solution

The main goal of this project is to propose a viable alternative method for rating user satisfaction in mobile apps. In order to achieve this, the underlying model of the proposed solution needs to satisfy a number of requirements. Before undertaking the process of developing a prototype of the alternative method, a series of short preliminary discussions with app developers and other domain experts have been conducted. The aim of these discussions was two-fold. First – to confirm the conclusions inferred from previous research ([3, 4]) and Internet sources [16, 17, 18, 19], that the major problems with the currently adopted conventional method for rating user satisfaction by manually submitted star ratings (discussed in detail in Section 1.1) are: rather low response rates, subjectivity and high tampering susceptibility. Thus providing an answer to ***RQ0***. And secondly, to identify all the requirements that an alternative solution is supposed to meet.

The list of identified requirements as an outcome of these discussions is available in **Table 5**.

ID	Requirement	Description
R1	Automation	The process for deriving the user satisfaction rating must require as little user and administrative input as possible
R2	Universal applicability	The rating method must be suitable for rating user satisfaction in broad range of mobile applications varying in purpose and

		intended usage patterns
R3	Flexibility	The mathematical model that the method builds upon must support parameterization, in order to allow for rating different kinds of applications with similar levels of accuracy
R4	High response rates	When adopted, the rating method must ensure response rates significantly higher than the ones observed in conventional rating of user satisfaction
R5	High level of protection against tampering	The design of the rating method is expected to provide protection against tampering that is significantly higher than the one offered by the conventional method (possibilities for tampering must be largely eliminated)

Table 5: Requirements for an alternative user satisfaction rating method

3.2. Model for rating user satisfaction in mobile apps

As mentioned in Section 2.1, just like according to Bailey and Pearson ([12]) satisfaction is influenced by multiple factors with varying importance; user satisfaction in applications is defined by several contributing factors, which in the context of satisfaction in a certain application can also have varying importance.

Therefore, when trying to come up with a model for rating user satisfaction in mobile apps, based on usage metrics, it might make sense to employ the weighted sum model template, just like Bailey and Pearson did for their satisfaction model. Therefore, in this thesis, the model defined by the following formula is used as a starting point. In order to try to answer ***RQ2***, the reliability of this model is tested (in Chapter 5).

$$\boldsymbol{User\ satisfaction_i} = \sum_{j=1}^{j=n} W_j R_j$$

Where:

$\boldsymbol{User\ satisfaction_i}$ = user satisfaction rating of mobile app ***i***

$\boldsymbol{R_j}$ = the rating for usage metric ***j***

$\boldsymbol{W_j}$ = the importance (weight) of usage metric ***j*** for the level of user satisfaction in mobile app ***i***

n = the total number of satisfaction related usage metrics

It is hypothesized that the metrics selected for the model (they are discussed in detail in Section 3.5) are of such nature, that they contribute to a unidimensional[7] construct (user satisfaction).

Nowadays, for mobile apps, it is possible to measure the value of each usage metric that allows automatic gathering (the majority of metrics), for every single user using the app in a given time frame. As discussed in Chapter 4, this can be accomplished with several readily available usage analytics platforms, such as Google Analytics. Using such platforms, it is for example possible to say how many distinct users have used the app between 10:30 and 17:30 on 01.01.2014 and also for how long each user has interacted with the app within that time frame.

However there are usage metrics related to user satisfaction that are challenging or even impossible to measure automatically. Such a metric is retention rate, or the ratio between users who do not uninstall the app and continue using it and the total number of users who have downloaded the app. At the time of preparing this report, the most popular smartphone operating systems (iOS, Android and Windows Phone) as well as their corresponding app stores do not support automatic detection of the event in which a user of a certain app uninstalls the respective app. Usage analytics packages such as Google Analytics are unable to differentiate between a user who has uninstalled the application and one who has not opened the app for several days.

The usage metrics selected for the model for mobile apps user satisfaction and the reasons for their selection are discussed in more detail in Section 3.5.

3.3. Alternatives for calculating R_j

Several different approaches can be used for determining the rating for each individual usage metric ($\boldsymbol{R_j}$). The different alternatives, as well as their strengths and weaknesses are discussed in the following subsections.

[7] Level of unidimensionality – the degree to which a set of independent variables actually measure a single common variable

3.3.1. Alternative 1

Probably the simplest way for calculating the rating for each usage metric ($\boldsymbol{R_j}$) is to take the app having the highest value for the particular usage metric and use that value as "the best", use the zero value as "the worst", map the interval between "the best" and "the worst" value to a 1 to 5 star scale and rate each app, depending on its value of the respective usage metric.

Such a rating technique might be economical to implement and easy to comprehend, because it seems very logical, but it has several weaknesses. First of all, it is not universal (cannot be applied for all usage metrics for all kinds of applications). As already explained, higher values of certain usage metrics are not always positively correlated with higher levels of user satisfaction. Secondly, this technique is subject to high probability of successful tampering by using fake or emulated users. The tampering success level will depend only on the number of simultaneously active fake users the cheating entity can secure and the duration for which the fake users can be "using" the app, both factors being positively correlated with the cheating success rate. Therefore this alternative for calculating $\boldsymbol{R_j}$ does not provide adequate levels of requirements R2 and R5 satisfaction.

Speaking about cheating success probability the question of averaging the value of each usage metric among all users of the app present for the measurement time frame comes up. If the value of the metric is distributed exponentially or in another non-linear way among users; and if there are cheating attempts, this most probably will be the case; the arithmetic average (the mean of the distributing) will be of little value. The median value will be much more representative as a statistical measure and will ensure higher robustness against tampering attempts, provided that the number of emulated users is sufficiently smaller than the number of real users.

3.3.2. Alternative 2

One alternative to the abovementioned technique for deriving the ratings for individual usage metrics, which is supposedly better in terms of universal applicability, reliability and anti-tampering robustness, involves once again averaging the value of the measured metric among users, but providing a rating that evaluates the proximity of this average value to an

expected or benchmark value for that metric. This implies that the resulting rating will be less than perfect, both in situations in which the measured value exceeds the expected value and situations in which the measured value is below the expected one (**Figure 4**). And the bigger the distance between both values is, the lower the resulting rating will be.

This technique is used in the prototype of the usage-based satisfaction rating system. The individual metrics' ratings are calculated in the interval [0.5 – 5.5] in order to allow for aggregating the produced ratings in equally sized 1-star slots. [0.5 – 1.5) corresponds to 1 star, [1.5 – 2.5) – 2 stars, [2.5 – 3.5) – 3 stars and so on. A bell-shaped smoothing curve, imitating normal distribution, as the one shown on **Figure 4** is applied for deriving the individual ratings. The 'standard deviation' used for the smoothing is set to $\frac{2}{3}$ E_i (E_i is the 'expected' value for usage metric i). As a result, if for example the 'expected' value for a given metric is 100, and the actual value is 33.3 or 166.7, this will lead to a rating of 3.49 stars. If the actual value equals 0 or 200, the rating will be 2.08 stars and if it is ~330 and bigger, the rating will be 0.5 stars.

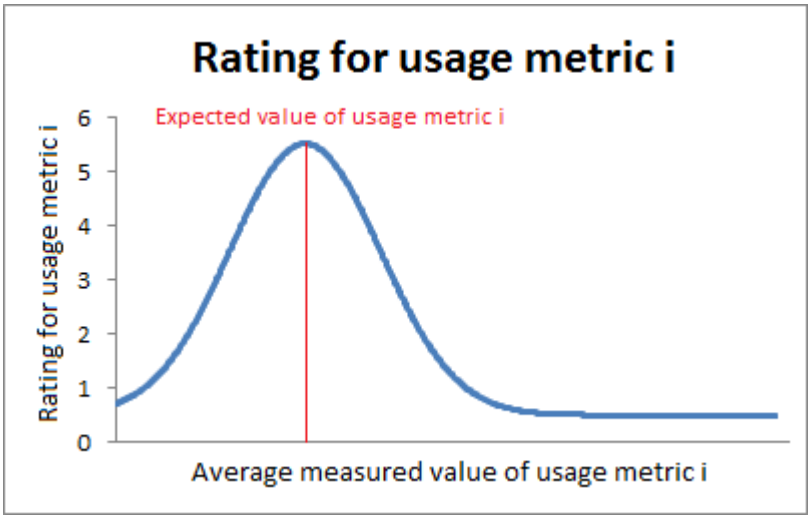

Figure 4: Rating for usage metric i

In a hypothetical future state of affairs in which the alternative rating system is adopted by the app stores, the system will have to provide them with the means for realistic estimation of the expected or reference value of each usage metric. The approach for obtaining a reference value for each metric named "micro benchmarking" and utilized in this thesis is explained in detail in the following subsection (3.3.2.1).

3.3.2.1. Micro benchmarking – a technique for deriving reference values for the usage metrics

Categorization of applications

Micro benchmarking relies on comparison of usage metric values exclusively among very similar apps. This technique will require dividing all existing and future applications in small enough categories or sub-categories. As a result of this division, all apps falling under a certain category have to share a single purpose and therefore - the same expected usage patterns. Such categorization will be much granular than the one currently in place at the app stores and unlike it, will most probably require a hierarchy consisting of two or more levels of sub-categories. An example for a small enough category of apps would be "clock apps" which would most likely be a subcategory of "tools".

Making the changes in app stores category system user-visible will most probably be undesired, because of the unnecessary increase of complexity for users. But of course, it is possible to let the two categorizing systems coexist by keeping the old division as user-facing, while making the new division back-end exclusive.

Because all apps in a single category will be highly similar in terms of purpose and range of user-accomplishable tasks, it will be possible to use a single expected / benchmark value for each usage metric for all applications in the category.

Determination of the "best" app in a category

As far as realistically estimating the expected or benchmark level of the usage metrics is concerned, a good approach would be to determine the best performing application in the category and use its usage metrics values as a reference for calculating the ratings for the usage metrics for the remaining apps.

Then the question of how to determine the "best" application in a category comes up. Probably the easiest way would be to simply look at the number of downloads each app has at that moment of time and select the most downloaded app as the best on the grounds of the positive correlation that exists between the user satisfaction in an certain app and the user preference to favour it instead of its direct competitors.

Critique of the “best” in a category principle and improvements

Unfortunately, It is difficult, if not impossible to say with certainty what percentage of the user base has come to prefer an app after downloading and trying multiple alternatives. In addition, there is no guarantee that if a vastly superior alternative materializes at a later time for example, large numbers of previously highly satisfied users will not be tempted to try or even switch to the novel solution. Therefore the number of users that have been recently active (for example active in the past two weeks) is more reliable than the number of downloads as a dynamic measure of user satisfaction.

From another point of view, people are inherently change-resistive in general and will be reluctant to switch from something they highly enjoy using and are very familiar with to an alternative they have no experience with, regardless of how well it is received. Similarly to adopting new technology, a certain adoption curve is in place for every new application. That is why a situation in which a well-established application with a large number of currently active and very satisfied users competes with a relatively new application, currently in the initial stages of its adoption curve with a relatively small number of active users, who are however exceptionally satisfied with it is highly possible.

In order to account for the presence of adoption curves and to minimize their destabilizing impact on the accuracy of measuring user satisfaction with the number of currently active users, this number can be adjusted with a factor reflecting how long each application has been present on the market. Because exponential growth with comparable rate is observed both for smartphone adoption and for development of new mobile applications, the theoretical probability of a single user deciding to use a certain app instead of its competitors, provided that all apps in the category are theoretically equally good, remains relatively stable over time. Therefore the number of currently active users can be linearly adjusted (for example divided by) with the number of days that have passed since the initial release of the application.

Of course this technique is limited in its ability to reflect the changes in user satisfaction over the lifecycle of the application. For example a dramatic overhaul of the user interface could bring a marked increase of user satisfaction. This is why by default, the Apple app store displays only ratings and user reviews submitted for the latest available version of a particular app. For the same reason, when testing the hypothesis laid out in Section 3.2 by

applying the micro benchmarking rating technique on usage data from live applications, adoption curve adjustment has not been performed and instead, a distinction between different app versions has been made.

3.3.3. Alternative 3 – basic user satisfaction score

While the previously discussed alternatives for deriving individual metrics ratings are based around comparison between apps and setting reference values, for each usage metric based on that comparison, it is possible to define a user satisfaction construct that is calculated independently for each rated application and simply reflects all of the actual usage metrics values demonstrated by each app. This construct is called basic (internal) user satisfaction score and like the rating derived with micro benchmarking, adheres to the general hypothesis that user satisfaction can be measured by summing usage metric ratings (in this case transformed usage metrics actual values). The value of the basic user satisfaction score of application i for the measurement period k is defined as:

$$Basic\ User\ Satisfaction\ Score_{ik} = \sum_{j=1}^{n} Wbuss_j V_{jk}$$

Where:

V_{jk} = the actual value of usage metric j for the measurement period k

$Wbuss_j$ = the basic user satisfaction score weight for usage metric j

Both alternative 2 and alternative 3 for deriving individual metrics ratings are utilized in this research and experiments in which both types of satisfaction ratings are calculated for live applications have been conducted (the experiments and their results are discussed in chapters 4 and 5). The purpose of these experiments is to try to validate the hypothesis defined in Section 3.2.

The following section explains what values has been used for the importance factors (weights) for the micro benchmarking ratings and the basic user satisfaction score throughout the experiments and why.

3.4. Importance factors (weights) - W_j

As mentioned earlier, an important goal of the user satisfaction model is to be universally applicable (R2). Despite the fact that it is logical for all apps in a micro benchmarking category to share the same expected / benchmark levels for the usage metrics, some usage metrics could be relevant only to some of the apps competing in a certain category. The importance of each usage metric ($\boldsymbol{W_j}$) will be constant among all apps in a category, however metrics that are irrelevant for a certain app will have $\boldsymbol{W_j = 0}$ for that particular app.

It is assumed that the importance of individual metrics for the overall user satisfaction in an app can vary depending on the category of the app. However the process of determining realistic weights for the usage metrics for different categories of apps is beyond the scope of this research project, because it would require processing the usage data of significantly larger number of mobile apps belonging to various categories. If the app stores adopt the rating system, it would make sense for them to define different weights for the usage metrics, based on on-going monitoring of the usage of all apps offered in the stores.

For the purpose of the experiments, for the micro benchmarking derived satisfaction rating calculations, all the participating metrics are equally weighted.

$$W_1 = W_2 = W_3 = \ldots = W_n$$

W_j = the importance (weight) of usage metric j

n = the number of participating usage metrics

And the sum of the all the weights equals 1 and this means that the combined user satisfaction rating derived via micro benchmarking varies between 0.5 and 5.5 stars like the individual metrics ratings (see subsection 3.3.2).

$$\sum_{j=1}^{n} W_j = 1$$

As far as the basic user satisfaction score is concerned, the need for a different way of weighing becomes apparent. This construct is essentially a sum of the usage metrics values. In order to compensate the fact that not all of the metrics are measured by using a single

unit (the list of usage metrics is available in the next section) and also to equalize the contribution of each metric towards the final basic user satisfaction score, a transformation to the measured metric values is applied. The intervals, in which all of the usage metrics values have fluctuated during the experiments' duration, are transformed to the interval of fluctuation of the usage metric 'Users'. Thus, a unique weight for each metric is derived. The weight of metric j for the basic user satisfaction score is:

$$Wbuss_j = \frac{\max(M_1)}{\max(M_j)} S_j$$

Where:

$\max(M_1)$ = the maximum observed value of the usage metric 'Users'

$\max(M_j)$ = the maximum observed value of the usage metric j

S_j = the sign of contribution of metric j to the basic user satisfaction score (can be either positive (1) or negative (-1))

S_j = 1 for $j \in [1, 7]$

S_j = -1 for $j \in [8, 9]$

It has been determined that, at least for the apps participating in the experiments, all metrics besides '*Exceptions per session*' and '*Exceptions per hour' are positively correlated with user satisfaction and therefore for all of them* S_j *equals 1. For* '*Exceptions per session*' and '*Exceptions per hour',* S_j equals -1.

3.5. Usage metrics

As stated before, the goal of this chapter is to acquaint the reader with the properties and structure of usage-based user satisfaction measurement mechanism that meets the requirements listed in Section 3.1. Now that the hypothesis that user satisfaction can be measured by summing the individual ratings of satisfaction-related usage metrics has been presented and the alternatives for obtaining the ratings for the individual metrics – discussed in the previous sections, the stage is set to introduce and discuss the actual usage metrics that have been selected to participate in the hypothesized model.

In order to meet the requirement of significantly higher participation rate than the conventional rating system (R4), the alternative solution for rating user satisfaction relies entirely on usage data that is automatically collectible by utilizing currently available technology. This decision results in added benefits – because the solution builds up on existing and widely available technology, this greatly reduces prototype development times and costs. In addition, because the proposed solution is highly automated (requirement R1 is satisfied), its adoption and maintenance costs are also significantly smaller in comparison with systems requiring a higher level of non-automated interaction.

The prototype of the solution (discussed in Chapter 4) uses data collected with Google Analytics as the basis for deriving its usage metrics. In Section 4.2 it is explained why Google Analytics was preferred instead of other existing alternatives. The selection of usage metrics chosen to participate in the user satisfaction model is limited by the capabilities of Google Analytics, which are extensive and satisfactory in the context of mobile apps. All of the metrics are logically related in one way or another to user satisfaction and it is possible to draw parallels between many of them and the use metrics identified by DeLone and McLean [7], [20].

In the remainder of this section, the nine metrics selected for the proposed solution are listed and explained. The list of selected metrics with their brief descriptions and calculation explanations is available in **Table 6**.

Usage Metric	**Explanation**
Users	The absolute number of unique users who have interacted with the app at least once during the measurement period
Percent of returning users	The percentage of all users interacting with the app during the measurement period who have also interacted with it in a previous measurement period
Sessions per user	The average number of continuous interactions

	(sessions) per active user in the measurement period Calculated by dividing the total number of user sessions that have occurred during the measurement period by the total number of users active during the same period
Session duration (seconds)	The average duration of a user session in seconds Calculated by dividing the combined duration of all user sessions that have occurred during the measurement period by the total number of user sessions
Screen views per session	The average number of unique screen views performed by an active user during a single user session Calculated by dividing the total number of unique screen views performed by users throughout the measurement period by the total number of user sessions measured during the same period
Purchases per user	The average number of in-app purchases made by an active user Calculated by dividing the total number of unique in-app purchases made by all users during the measurement period by the total number of active users
Purchases per hour	The average number of in-app purchases made by an active user during an hour of interaction with

	the app Calculated by dividing the total number of unique in-app purchases made by all users during the measurement period by the combined duration of all user sessions (in hours)
Exceptions per session	The average number of unhandled exceptions (crashes) encountered by an active user during a single user session Calculated by dividing the total number of registered unhandled exceptions by the total number of user sessions
Exceptions per hour	The average number of unhandled exceptions (crashes) that have occurred during an hour of interaction with the app Calculated by dividing the total number of registered unhandled exceptions by the combined duration of all user sessions (in hours)

Table 6: Usage metrics

3.5.1. Users

In a hypothetical situation of a perfect completion between two competing apps – both apps are very similar in purpose and types of tasks performable by users, both of the developers have dedicated equal marketing budgets for promoting their respective app and both apps are released on the market simultaneously; the number of active users should be applicable as a direct differentiator of user satisfaction. The app with which users are more satisfied should exhibit a higher number of active users.

The conditions of perfect competition predetermine that most users will initially try both applications and eventually settle with the one they enjoy using more than the other and effectively continue being active users of the app their more satisfied with.

As discussed in Section 3.3, micro benchmarking, the technique for deriving target values for the usage metrics, which relies on comparison among all directly competing apps, selects the application exhibiting the highest number of active users for the measurement period as the "best" in its category. According to the definition of this technique, the "best" app is going to receive maximum ratings for the metrics: *Users*, *Percent of returning users*, *Sessions per user*, *Time between sessions*, *Session duration* and *Screen views per session.* The respective metric ratings for the other competing apps are going to depend on the distance observed (how much higher or lower the metrics values exhibited by a competing app are than those exhibited by the "best" app. The reason for this decision is that although the abovementioned metrics are generally positively correlated with satisfaction, in certain situations and for certain categories of apps, this might not be the case. (See the following subsections for more details).

3.5.2. Percent of returning users

As mentioned in previous sections, current smartphone software technologies do not allow for automated measurement of retention rate – the percentage of users who do not uninstall and continue using an application they have previously installed.

Percent of returning users does not have exactly the same meaning as retention rate (see **Table 6**). It could be said that *Percent of returning users* has an advantage over retention rate – the former is actually able to reflect a continued user engagement with the respective app – it is based on a figure measuring the actual active usage. In contrast, as far as retention rate is concerned, the user may have not uninstalled a certain app and still never use it. In fact, some applications that come preinstalled by the handset vendors or network operators are non-removable (non-uninstallable). This phenomenon diminishes the reliability of retention rate as a measure of satisfaction, because it is always possible that users unsatisfied with the non-removable apps are enjoying alternative ones and never return to using the former.

A drawback of both *retention rate* and *percent of returning users* is that they are obviously also dependent on the number of users who install the application for the first time. This figure is hardly related to user satisfaction and reflects user expectations instead.

3.5.3. Sessions per user

This metric is a direct indicator of user engagement, which in turn is a component of user satisfaction. There should be a positive correlation between the two – more satisfied users are expected to actually use the application more often. However some types of applications are supposedly normally used more often than others on average. This is one of the reasons why segmenting all applications in concrete enough categories and applying the micro bechmarking technique is necessary prerequisite for using *sessions per user* as an adequate indicator of satisfaction.

Within Google Analytics all distinct user interaction events that have occurred within a distance in time less or equal to 30 minutes are counted towards a single user session (and effectively each subsequent event extends the duration of that session, while the first one is considered as the start of the session).

3.5.4. Session duration

Session duration could be different from *sessions per user* in terms of meaning, but it also reflects the level of user engagement.

Unlike *sessions per user*, which should always be positively correlated with satisfaction, there are certain categories of apps for which longer user sessions speak for not particularly satisfied users. Such types of applications are, for example ones that are supposed to improve productivity by allowing users to complete their tasks at hand as fast as possible, without sacrificing effectiveness.

3.5.5. Screen views per session

As the previous two metrics *screen views per session* could indicate the level of user engagement. For the three most popular smartphone operating systems (iOS, Android and

Windows Phone), screens can be considered as the highest-level building blocks of an application and most applications have several distinct screens. There is one notable exception – type of Android apps called “widgets” can have no screens in the classical sense. Upon placing these applications on the home screen of an Android device, their user interface becomes just a part of the whole user interface of the home screen providing the user with additional functionality.

While using an application the user navigates between its distinct views within the duration of a single session. It is difficult to say whether this metric is positively or negatively correlated with satisfaction. The answer to this question could be highly dependent on the category of the particular app. The user could be highly satisfied and because of that to view all of the available screens while using the app, or they could be wandering among the sea of available screens unable to easily find a specific functionality they are looking for. In both situations the value of this metric will be high, but the sign of correlation with user satisfaction will be different.

From another point of view, the average level of complexity exhibited by the apps in a category could be the main influencer of this metric (it should be normal for an app with a lot of screens to have a lot of *screen views per session*).

Despite the ambiguity of this usage metric and its questionable ability to account for the level of user satisfaction, from a scientific point of view, it will be interesting to compare its values among apps within the same category that rank highly and poorly in terms of user satisfaction.

3.5.6. Purchases per user and Purchases per hour

These metrics account for the willingness of users to spend additional money on items (in-app purchases) within an app they have already been using. This desire is positively correlated with satisfaction and it is not uncommon that users spend money on in-app purchases in order to express their gratitude and satisfaction to the developer of the app.

Purchases per user could be more meaningful for applications that have a single in-app purchase that is non-exhaustive, such as an in-app purchase that once bought by the user, unlocks all possible additional functionality forever. In such a case it is very close in meaning

to conversion rate, or the percentage of all users who have bought the paid version (the single in-app purchase).

Purchases per hour is more suitable for applications with a variety of available in-app purchases, which are predominantly consumable (exhaustive). An example for such an upgrade is additional fuel for a car racing game – once the additionally purchased fuel is depleted (consumed) the user has to purchase the same in-app purchase again if they want to continue playing.

Depending on the presence, number and nature of the in-app purchases available in a particular app, either one, both or none of the two importance factors - $\boldsymbol{W_{Purchases\ per\ user}}$ and $\boldsymbol{W_{Purchases\ per\ hour}}$ could be equal to zero. (See Section 3.2 for the meaning of the importance factors)

Because both of the purchase-related metrics are unconditionally positively correlated with satisfaction, when using the micro benchmarking technique, the reference values for these metrics are going to be equal to the highest ones exhibited among the apps competing in an app category. Therefore, as far as the purchase-related metrics are concerned, the "best" app in the category is not determined by looking at the number of the apps' active users.

3.5.7. Exceptions per session and Exceptions per hour

Application crashes are always unwanted and undoubtedly their presence will affect satisfaction and usability negatively. Still they could be tolerable by some users, depending on the frequency of their occurrence.

The unhandled exceptions (crashes) are of course not a direct component of user behavior and usage patterns, but nevertheless they are triggered by or during a user's interaction with the app in question. In addition, the fact that information about their occurrence is easily collectible and their obvious relation with satisfaction is a good enough reason to include these metrics in the model for rating user satisfaction.

Exceptions per session is more suitable for tracking unconditional crashes (crashes that occur in every single user session, regardless of external factors), while *Exceptions per hour* are

more suitable for measuring exceptions that occur on a seemingly random basis (such exceptions are usually triggered by rare combinations of preconditions).

Because the exception-related metrics are always impacting satisfaction negatively (user satisfaction and presence of unhandled exceptions are always negatively correlated), when using micro benchmarking, the "best" performing app in an app category in terms of exceptions is not the app with the highest number of active users. Instead, the target values for *Exceptions per session* and *Exceptions per hour* are going to be equal to the lowest ones exhibited by an app in the category.

3.6. Conclusion

In conclusion, taking into account existing theoretical frameworks and the list of requirements for an alternative user satisfaction rating method, it is hypothesized that a single measure for user satisfaction in mobile apps could be defined as the weighted sum of individually rated, automatically measurable usage metrics. In addition, it is proposed that the individual ratings for each metric can be obtained via a comparison among directly competing apps. The validity of this hypothesis is tested with quantitative research (controlled experiments, discussed in Chapter 4 and Chapter 5) and qualitative research (interviews, discussed in Chapter 6 and Chapter 7).

4. Prototype system for rating user satisfaction and live experiments

4.1. Introduction

As mentioned in Chapter 1, an important goal of this research is to test the hypothesis laid out in Section 3.2 (that user satisfaction in mobile apps can be measured by weight-summing the individual ratings for several usage metrics[8]) by conducting live experiments with real-world mobile apps.

Testing the hypothesis with usage data from real apps requires building a prototype system that tracks the usage of the observed apps and continuously rates user satisfaction in them by using the proposed model for rating user satisfaction (the model is explained in Section 3.2). The aim of the current chapter is to acquaint the reader with details about the prototype system that has been developed, the reasoning behind key decisions for its design, as well as to provide information about the conducted live experiments. The next chapter (Chapter 5) acts as a summary of the experiments' results and the key findings.

In order to significantly reduce the necessary development effort and to allow the research project to fit in the usual time frame for a master's thesis project, it has been decided to use existing tools and technology as much as possible, instead of developing the entire prototype from scratch. The following section describes an existing usage measurement platform (Google Analytics), which was selected as a basis for developing the prototype system.

4.2. Selecting Google Analytics as a basis

In the past few years the market has seen the rise of a number of web-based services offering mobile app usage tracking. One of them – Google Analytics – has undergone several major revisions and has reached maturity. Unlike some of its competitors, it was originally developed for tracking web sites and this provides two advantages. First of all, it has a large

[8] The usage metrics selected for the user satisfaction model are described in detail in Section 3.5.

established community of web site developers using it and secondly, the transition from using the web site tracking service to using the app tracking one is straightforward.

In addition, unlike competing solutions, Google Analytics provides a comprehensive set of APIs[9], specifically designed for allowing third-party developers to extend Google Analytics' existing functionality. This API covers a function of vital importance for the development of a prototype user satisfaction rating system – it allows for simultaneous automated and systematic (in regular time slots) tracking of the usage patterns and behaviours of practically unlimited number of mobile apps, developed by different app developers.

Unlike Google Analytics, many of its alternatives, such as Mixpanel and KISSmetrics are more personalized solutions that require extensive customization, specific for every single tracked application. What is more, these solutions focus mainly on tracking user-produced actions or events, such as button clicks (highly application specific), while disregarding the collection of some more general types of usage data, such as session duration. Both of these circumstances hamper their applicability for a universal solution.

Because of the advantages Google Analytics has over competing platforms (third-party extendibility and focus on usage metrics that are universally traceable among apps, this platform has been selected as a raw usage data provider for the prototype of the system. As a result all applications that have integrated the Google Analytics app tracking solution (as of October 2014 Google Analytics app tracking is offered for Android, iOS and Windows Phone) could potentially be tracked and user satisfaction in them – rated with the proposed prototype system. However, in order to be granted access to a particular app's usage data, the prototype needs the app's owner's explicit permission.

However, Google Analytics is not without drawbacks. Despite its flexibility, it is not possible for third parties to extend or alter the core functionality of the usage tracking service (that is not possible with any of Google Analytics' direct competitors either). Therefore the selection of usage metrics that the prototype can employ for its user satisfaction rating algorithm is limited by the list of raw usage data units already offered by Google Analytics. Despite the fact that this limitation implies a somehow compromised solution, after careful examination of the possible usage metrics, it was concluded that Google Analytics allows for an adequate

[9] API – Application Program Interface - a set of routines, protocols and tools for building software applications

list of usage metrics related to user satisfaction. One notable exception is however 'retention rate' which is not directly measurable with Google Analytics. Combining the two available metrics: 'Users' and 'Percent of returning users', however, produces a construct that is close in meaning to 'retention rate' (see subsection 3.5.2 for more details). The complete list of usage metrics selected for the user satisfaction model is discussed in detail in Section 3.5 and the model itself – in sections 3.2, 3.3 and 3.4.

4.3. Prototype system for rating user satisfaction

The prototype of the system is built as a scalable cloud-based web application. Although for the purpose of this study the usage of a limited number of mobile apps is tracked, the prototype's architecture and implementation would allow for simultaneous tracking of hundreds or thousands of apps without additional optimizations. The user interface allows participating app developers to:

1. Register and provide access to their Google Analytics data
2. Visualize and regularly monitor the values of the 9 satisfaction related usage metrics
3. Visualize and regularly monitor the resulting overall user satisfaction rating as well as the rating for each individual metric

Screenshots of the prototype system demonstrating the latter two activities are available in Appendix 1 (**Figure 11** and **Figure 12**).

In the background, on a daily basis, at a defined time, the web application:

1. Queries Google Analytics for the raw usage data of each participating app, corresponding to the previous day
2. Calculates the respective values of all usage metrics. **Table 6** in Section 3.5 describes how raw usage data, coming from Google Analytics, is transformed in order to produce the usage metrics
3. Based on comparison of the usage metrics' values among the participating apps, calculates a rating for each metric for each application as well as an overall user satisfaction rating for each app

4. Stores the raw usage data, the metrics' values and the corresponding ratings in a database, in order to allow for convenient data visualizations and execution of statistical tests

4.4. Experiments description

To test the hypothesis introduced in the previous chapter, experiments using the prototype described above have been conducted. In these experiments, the prototype tracks the usage and calculates the user satisfaction ratings of 2 applications on a daily and weekly basis between the 8th of May 2014 and the 5th of October 2014. The rating calculations are based around the hypothesis for a usage-based user satisfaction rating algorithm laid out in Section 3.2. More specifically, two types of user satisfaction ratings are calculated (satisfaction rating derived via micro benchmarking (described in subsection 3.3.2) and basic user satisfaction score (described in subsection 3.3.3)).

The comparison of usage data exclusively among apps belonging to a single category is an essential prerequisite for obtaining user satisfaction ratings based on the micro benchmarking technique. For that reason, the tracked apps belong to a single category - both of the participating applications can be broadly categorized as 'app widget'[10] apps or 'app widget-related' apps for Android-based mobile devices. However, this is not required for deriving the basic user satisfaction score, because its calculation is entirely dependent on data belonging to the application being rated (it does not depend in any way on the usage of competing applications).

It should be noted that the category of apps selected for the experiments might turn out to bee 'too broad' in the long term, as different types of app widget apps could be used quite differently, depending on the type of content they serve. Further research is necessary in order to answer with certainty the question whether further segmentation within the category of app widget and app widget-related apps is necessary.

By downloading and using either one of the participating applications, its users effectively give their consent to the respective app's privacy policy. Both apps' privacy policies inform

[10] App widget Android apps are mobile applications for Android-based mobile devices, which include application views that can be embedded in other applications (such as the home screen application) and whose content is periodically updated.

the users that usage data is being collected and that the data collected contains no identifiable information. The absence of such information guarantees that security of sensitive private user details remains intact and prevents establishing a link between a unique individual and certain app usage patterns.

The conduction of the experiments has not interfered with the normal life cycle of the participating apps – the update schedule of the apps has not been affected by the experiments.

The following section is dedicated to providing brief background information about the applications selected for the experiments. Each application is described in its own subsection.

4.5. Participating applications – general information

4.5.1. Application 1 (Clock widget)

The first application is a clock app widget whose initial version has been released in May 2012 and has reached relative maturity by the start of the experiment. It has seen rapid development throughout the second half of 2013 and the first quarter of 2014 – several new versions have been released in that period, with each subsequent one introducing improvements and adding new functionality. Due to this development, the app has seen a sharp increase in popularity – on each day in the experiment's duration the presence of more than ten thousands new users have been observed. The app includes a single one-time in-app purchase, which once bought by the user, unlocks additional functionality and removes advertisements forever.

Usage tracking for this application has started with the release of a new (major) version of the app on 8th of May 2014[11] and during the experiment one additional version (mostly bug-fixing oriented) has been released – on 29th of May 2014.

[11] According to the Google Play app store, as of 1st of June 2014 49.29% of all active users are using a version of the app that includes usage tracking code, as of 30th of September 2014, this figure is 81.81%

Because this application is distributed via the Google Play app store, data for star ratings submitted by its users is accessible and included in this research (the comparison between conventional and alternative user satisfaction ratings is summarized in Chapter 5).

4.5.2. Application 2 (Launcher)

This application is a home replacement[12] application for Android whose latest version has been released in mid 2013 in an Android enthusiast web forum. Earlier versions have been released in previous years, but due to the amount of rework involved in each subsequent major version, the major versions can safely be considered separate products.

Probably partly due to the nature of distribution (outside the mainstream Google Play app store), this app has a significantly smaller (than application 1), but very loyal user base. A slight decline in active users has been observed for the past several months and perhaps this is caused by the relatively prolonged lack of updates and the fierce competition in the specific market segment.

This application contains no in-app purchasable unit and therefore lacks any data for the purchase related usage metrics. For that reason those usage metrics are completely excluded from the alternative rating calculations.

The owner of the application has integrated Google Analytics and has been tracking the usage data since 2013 (way before the start of the experiments).

Because this application is not available in any app store, no mechanism for user-submitted star ratings for it exists. Hence, this application cannot contribute to the comparison of the alternative satisfaction rating method with the conventional (star) rating method.

4.6. Conclusion

To sum up, the prototype of the system builds upon Google Analytics to create a scalable web application that rates mobile applications by using the algorithm outlined in Section 3.2.

[12] Home replacement Android app – an application used as an alternative home screen application. Such applications hold app widgets and app shortcuts

With the help of the prototype, experiments with two apps have been conducted, for which user satisfaction ratings using the micro benchmarking technique and basic user satisfaction scores have been calculated.

The complete analysis of the data collected for the usage metrics of the tracked applications, as well as the calculated user satisfaction ratings and basic user satisfaction scores, comparison with conventional star ratings received for the duration of the experiments is available in Chapter 5. That chapter also discusses of the validity and reliability of the hypothesis laid out in Section 3.2, as well as the replicability of the experiments and the resulting findings.

5. Experiments data analysis

5.1. Introduction

As discussed in Chapter 4, with the help of the prototype of the proposed usage-based rating system, experiments with two applications have been conducted between the 8th of May 2014 and the 5th of October 2014. The purpose of the experiments is to track the usage, calculate usage-based user satisfaction ratings of the participating apps and ultimately, after analysing the results and performing statistical tests, try to answer ***RQ2: What is the reliability of usage-based measurement of user satisfaction***.

The following section focuses on summarizing the usage data collected throughout the experiments. In addition, the weekly fluctuations of the metrics are provided in tabular form.

5.2. Usage data summary

Application 1 - general usage information:

Total number of users (between the 8th of May 2014 and the 5th of October 2014): 2420272

Average number of daily active users: 44242 (minimum: 24123, maximum: 58579)

Average number of new users per day: 16039 (minimum: 13863, maximum: 24120)

Application 2 - general usage information:

Total number of users (between the 8th of May 2014 and the 5th of October 2014): 25535

Average number of daily active users: 2361 (minimum: 2237, maximum 2513)

Average number of new users per day: 147 (minimum: 113, maximum 188)

Visual representations of the daily and tabular representation of the weekly values of all usage metrics participating in the usage-based user satisfaction rating algorithm (described in Chapter 3) for the observed applications are available in Appendix 2.

The following section includes visual representations of the daily values of: the user satisfaction rating calculated by using micro benchmarking (described in subsection 3.3.2) and the basic user satisfaction score (described in subsection 3.3.3) for both of the participating applications as well as the conventional star rating of application 1 submitted by users in the Google Play store.

5.3. Conventional and alternative user satisfaction ratings

The satisfaction ratings data is divided in subsections corresponding to each tracked application.

5.3.1. Application 1

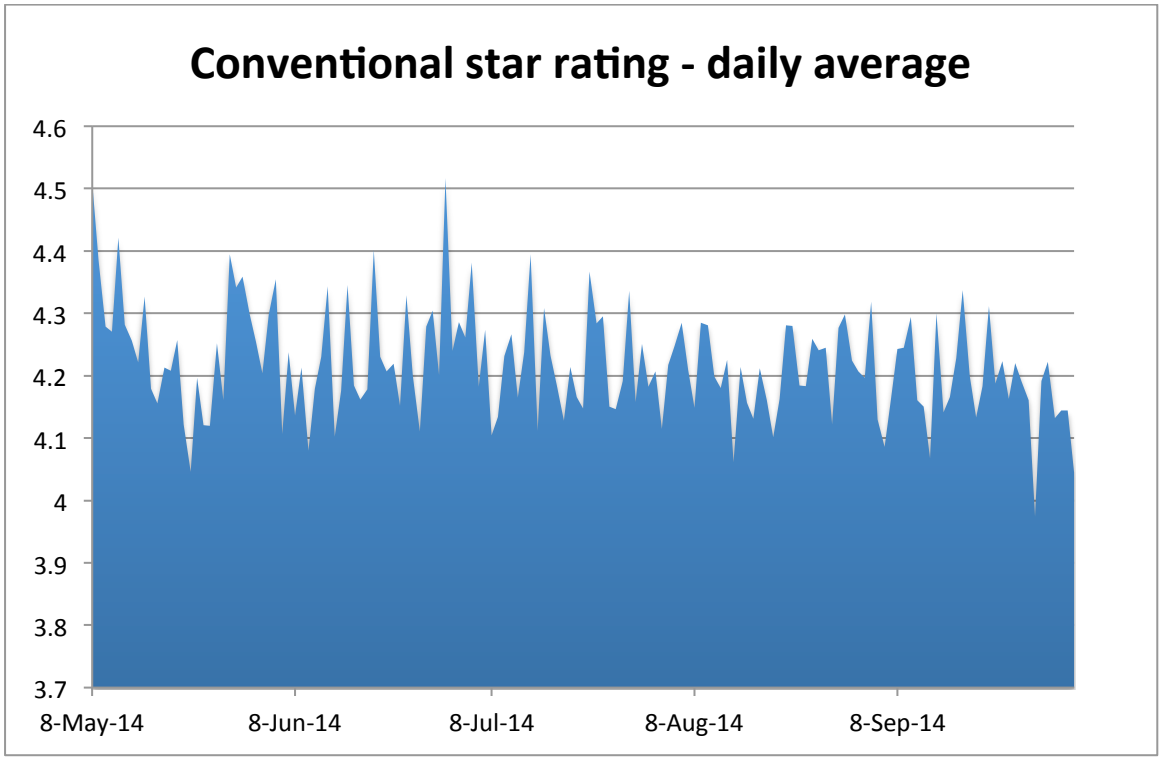

Figure 5: Application 1 - Conventional star rating, daily average

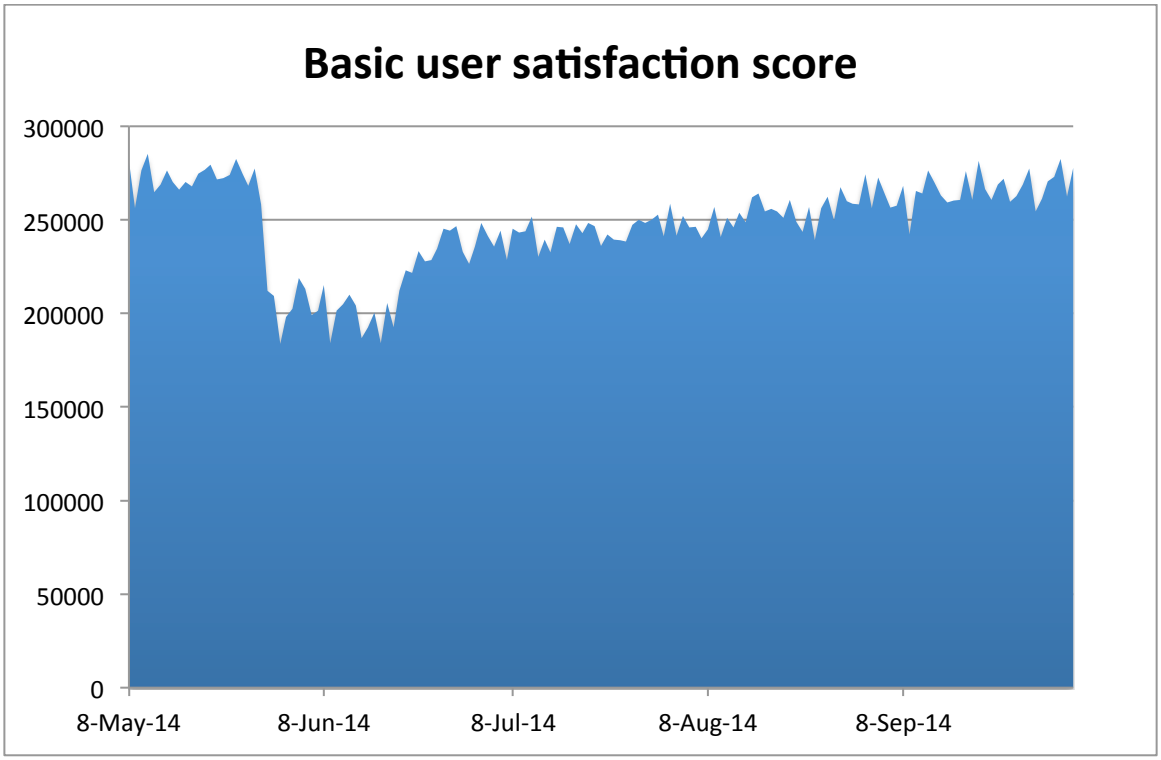

Figure 6: Application 1 - Basic user satisfaction score

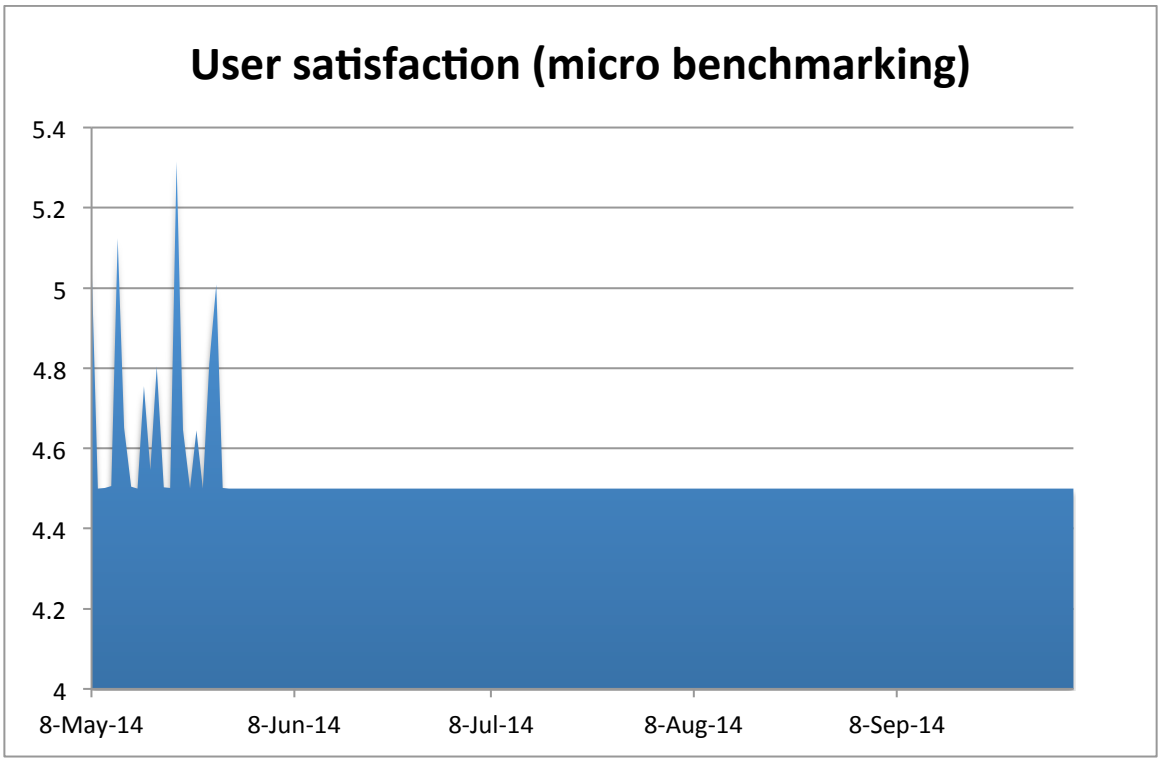

Figure 7: Application 1 - User satisfaction rating (micro benchmarking)

5.3.2. Application 2

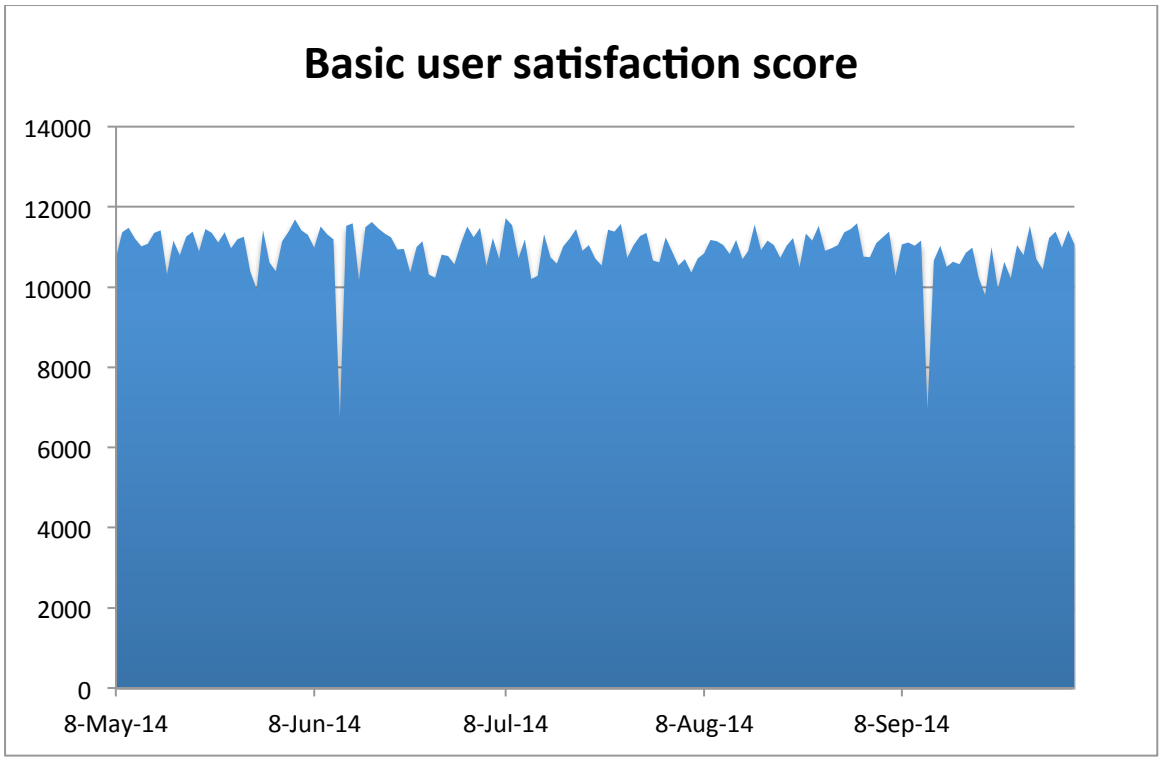

Figure 8: Application 2 - Basic user satisfaction score

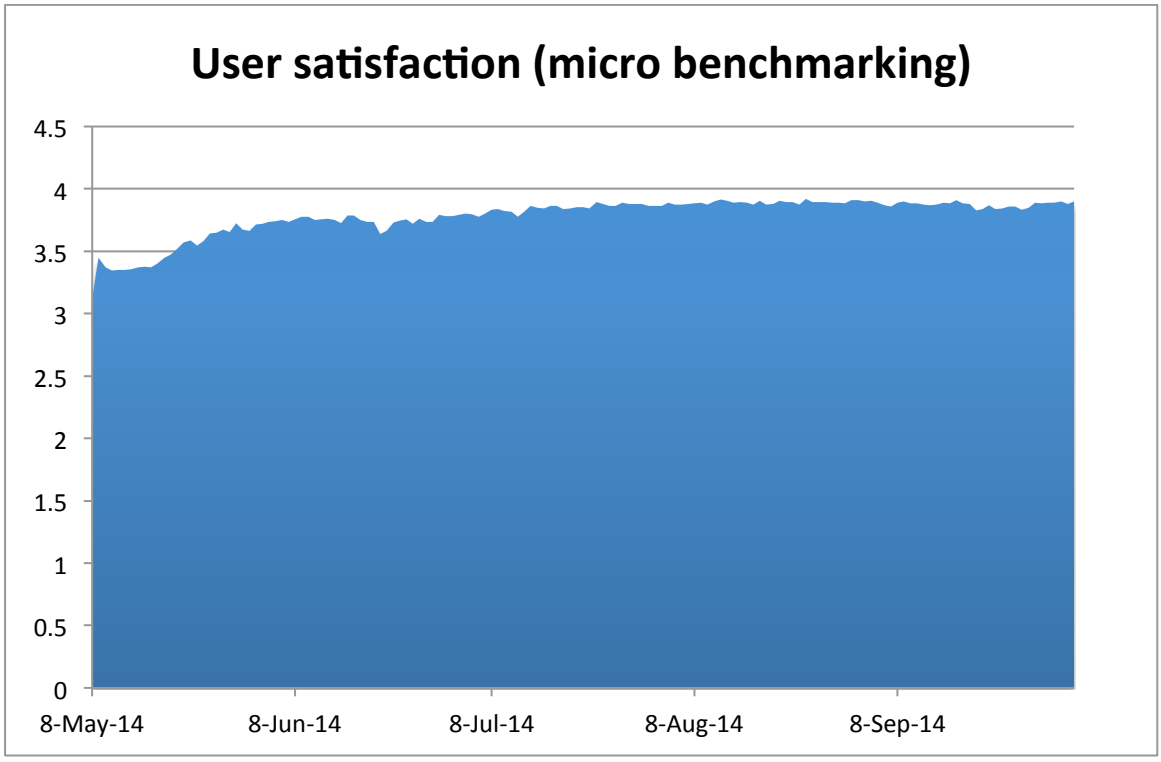

Figure 9: Application 2 - User satisfaction rating (micro benchmarking)

5.4. Data analysis

The main goal of the experiments with live applications is to test the validity of the hypothesis for a model for rating user satisfaction in apps based on usage metrics (presented in sections 3.2, 3.3 and 3.4) with quantitative data and to try to provide an answer to ***RQ2: What is the reliability of usage-based measurement of user satisfaction***.

Because both the user satisfaction rating derived with micro benchmarking and the basic user satisfaction score are based around a summation model that produces a single dependent variable (construct) as an arithmetic sum of transformed independent variables, it has been decided that Cronbach's Alpha test [27] is to be performed on the two types of calculated ratings, in order to test the reliability[13] of the proposed alternative rating method as well as the unidimensionality of the two constructs.

In addition, for application 1, the derived alternative user satisfaction ratings are compared with the conventional star ratings, because data for the latter it is available for that application. Besides basic correlation tests between conventional and alternative satisfaction ratings, a Bland-Altman test [28] is performed in order to test the agreement between them.

The results from the abovementioned statistical tests as well as comments about the distributions of the usage metrics are presented in the remaining subsections of this chapter.

5.4.1. Application 1: Usage metrics data analysis

Looking at the data presented in subsection 10.2.1, it can be noted that application 1's number of active users has gradually increased from around 24000 to around 58000, while the percent of returning users has increased from 0% to around 70% and has levelled off once this value has been reached. The latter phenomenon can be explained with the fact that when usage tracking for this app has started (on 8th of May 2014) and for quite a while after that, the majority of the active users had not upgraded to the latest version of the app yet and therefore their usage of the app is not accounted for, because earlier versions had not included usage-tracking code.

[13] Dr. Joost Schalken-Pinkster has recommended the Cronbach's Alpha test as "suitable statistical" test for testing the reliability of the basic user satisfaction score and the satisfaction rating derived via micro benchmarking

The release of a subsequent version of the app (on 29th of May 2014) has resulted in a sharp increase in the number of unhandled exceptions, because this version has introduced a new relatively frequently appearing software bug. This new release also has lead to a temporary increase in the number of sessions per user (by 5.5%) and the number of purchases per user (by 25%). However it cannot explain the sudden increase in the average session duration (by more than 30%) and the number of screen views per session (by 7.5%) that has occurred about three weeks later.

5.4.2. Application 2: Usage metrics data analysis

The data in subsection 10.2.2 shows that throughout the duration of the experiment with application 2, that application has gradually lost about 4% of its initial daily active user base (2500), although this figure has partially recovered in August and September 2014. The percent of returning users has fluctuated regularly between 92% and 95%.

A seemingly unexplainable 8% drop in the average number of sessions per user per day has been observed around 17th of June 2014. This event also coincides with an increase in the average user session duration by about 10%.

Screen views per session have regularly fluctuated with +-10% around the value of 2. Meanwhile the amount of unhandled exceptions has remained relatively stable, but isolated occurrences of values about 4.5 times higher than the average are observed on 12th of June and 12th of September 2014.

5.4.3. Application 1: Statistical tests results

Cronbach's Alpha test of the basic user satisfaction score of application 1

Cronbach's Alpha has been calculated for the basic user satisfaction score of application 1 with several different sets of conditions – daily and weekly calculation of the basic user satisfaction score, several different sets of participating usage metrics, two different experiment durations (08.05.2014 – 05.10.2014 and 01.06.2014 – 05.10.2014). The condition sets with the corresponding Alpha values are provided in **Table 7**.

The second experiment duration (01.06.2014 – 05.10.2014) has been added in order to try to tackle the negative effects (such as the severe increase in unhandled exceptions) from releasing a newer version of Application 1 on the 30th of May 2014. It is clearly visible in **Table 7** that when data for both versions of the app is included and the exception-related metrics participate in the basic user satisfaction score calculation, Alpha values are very low (even negative).

		Included usage metrics			
Experiment duration	Measurement period	All 9 usage metrics	All metrics, besides purchase-related	All metrics, besides purchase-related & exception-related	All metrics, besides exception-related
08.05.2014 to 05.10.2014	1 day (daily)	-0.27	-0.13	0.54	0.43
	1 week (weekly)	-0.70	-0.57	0.66	0.46
01.06.2014 to 05.10.2014	1 day (daily)	0.64	0.79	0.52	0.23
	1 week (weekly)	0.71	0.84	0.63	0.33

Table 7: Application 1 – Cronbach's Alpha (Basic user satisfaction score)

On the contrary, when data collected before the 1st of June 2014 is discarded, Cronbach's Alpha values are quite high for basic user satisfaction scores calculated using all 9 usage metrics and even higher (above 0.7) when the in-app purchase-related metrics are excluded from the calculation. Such Alpha values are generally considered an evidence for high reliability and unidemensionality of the tested construct.

It can be speculated that the slightly lower Alpha values for the situations when the in-app purchase-related metrics are present in the calculation are due to the fact that user willingness to pay for in-app purchase units is more loosely related to user satisfaction than the rest of the usage metrics. The data shows (see **Figure 18** in Appendix 2) that despite the popularity of application 1, no more than 1% of its users buy the only offered in-app purchase. In addition, at the moment of preparing this document, Google Play still does not offer paid apps and in-app purchases to users in some countries [29], therefore users of the app residing in those countries are unable to buy the in-app purchase even if they intended to.

Cronbach's Alpha test of the user satisfaction rating derived from micro benchmarking

Cronbach's Alpha has been calculated for the alternative user satisfaction ratings of application 1, derived from comparison between the participating apps (micro benchmarking) for the duration (08.05.2014 – 05.10.2014) and the resulting **Alpha value is 0.44**. When the test is performed for the shorter period (01.06.2014 – 05.10.2014), Alpha is basically equal to zero, because in that period the rating from micro benchmarking as well as the sub-ratings for the usage metrics are virtually constant. Discarding the in-app purchase related metrics, as in the case of testing the basic user satisfaction score barely affects the Alpha value, because the variance in these metrics is equal to zero.

Alpha value of 0.44 is too low to prove the reliability and unidimensionality of the satisfaction rating derived from micro benchmarking. The reasons for the disappointing results could be:

1. The rather small number of applications participating in the experiments – resulting in hardly any change in the sub-ratings for many of the usage metrics
2. The selected category of apps for the experiments is too broad – the difference in the values of some of the metrics between the participating apps is so large, that it automatically results in extremely low or extremely high sub-ratings for those metrics

Agreement between the conventional satisfaction ratings and the basic user satisfaction score

In order to compare the basic user satisfaction and the conventional star rating, the correlation between the two measures for application 1 has been calculated for a daily and weekly basis. The calculations have been made for two measurement durations (08.05.2014 – 05.10.2014 and 01.06.2014 – 05.10.2014) and the results are shown in **Table 8**.

Experiment duration	Measurement period	Included usage metrics	
		All 9 usage metrics	All metrics, besides purchase-related
08.05.2014 to 05.10.2014	1 day (daily)	-0.11	-0.12
	1 week (weekly)	0.13	0.16
01.06.2014 to 05.10.2014	1 day (daily)	-0.17	-0.12
	1 week (weekly)	0.15	0.25

Table 8: Application 1 - Correlation between conventional rating and basic user satisfaction score

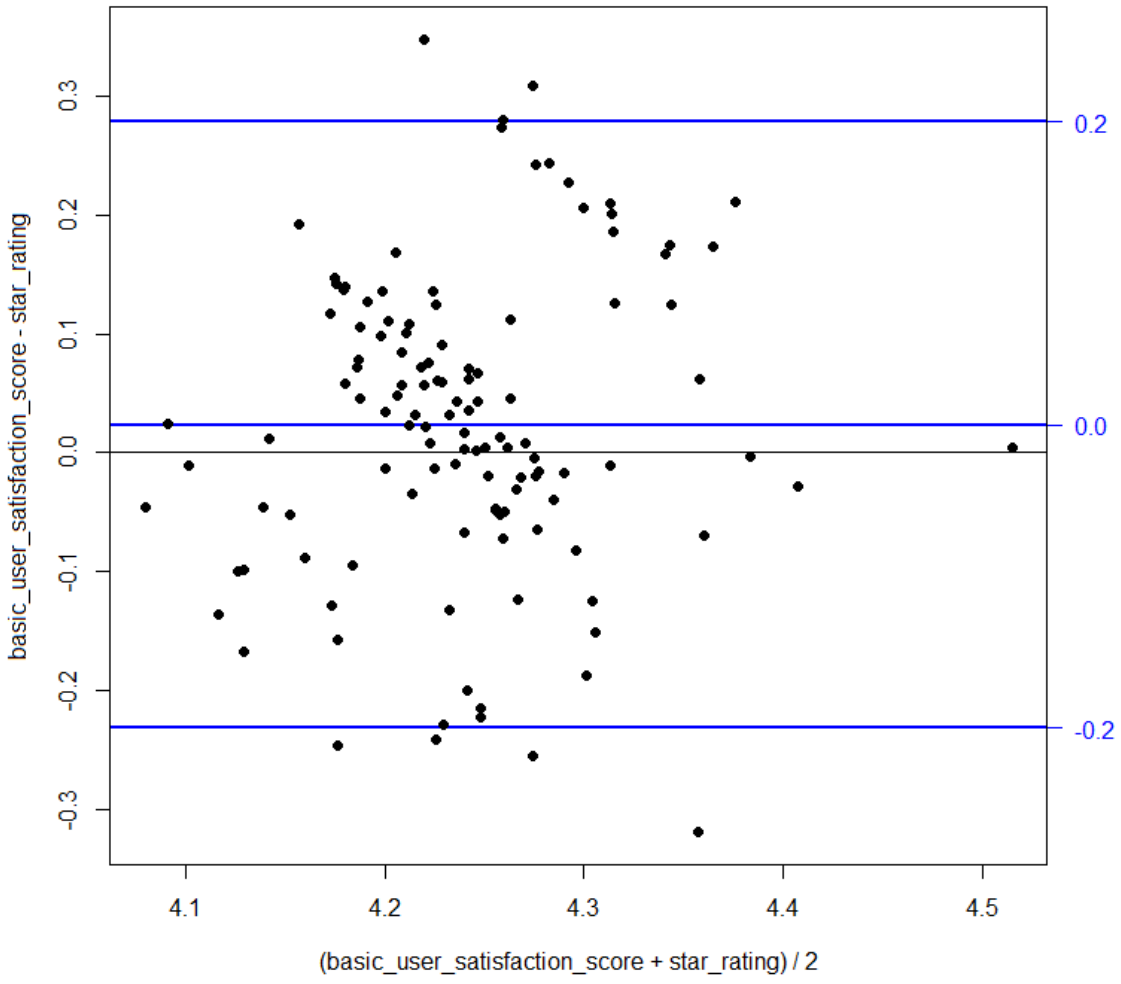

Figure 10: Application 1 - Basic user satisfaction score versus conventional star rating - Bland-Altman plot

Both constructs are rather weakly correlated. The highest measured positive correlation (0.25) is present for weekly measurement in the period 01.06.2014 – 05.10.2014. The weak correlation is not necessary a bad thing – the manually submitted ratings for application 1 are suffering from at least one of this method of rating's inherent problems (discussed in Section 1.1). Out of the 2420272 unique users active between 08.05.2014 and 05.10.2014, just 3685 have submitted a rating in the app store.

Besides correlation, the agreement between conventional star ratings and the basic user satisfaction score has been determined for application 1 by using the Bland-Altman test. The resulting Bland-Altman plot is show on Figure 10.

Because the basic user satisfaction score is virtually an uncapped measure (its value can grow to hundreds of thousands or more (see subsection 3.3.3 for its definition)), for this test its calculated values have been transformed from the range [183579 – 285356] to the observed range of the conventional ratings ([3.97 – 4.52]).

As it can be seen on Figure 10, in more that 95% of the cases, the transformed basic user satisfaction score is either higher than the corresponding star rating by no more than 0.3 stars or lower by no more than 0.25 stars and on average it is about 0.05 stars higher than it.

5.4.4. Application 2: Statistical tests results

Cronbach's Alpha test of the basic user satisfaction score of application 2

Cronbach's Alpha has been calculated for the basic user satisfaction score of application 2 with several different sets of conditions – daily and weekly calculation of the basic user satisfaction score; two different sets of participating usage metrics[14]; two different experiment durations (08.05.2014 – 05.10.2014 and 17.06.2014 – 05.10.2014). The second duration was added in order to diminish the influence of a sudden unexplainable change in the average session duration and average number of sessions per user since 17th of June 2014. It should also be noted that the dates of 12th of June and 12th of September 2014 have been deliberately excluded from the calculation. They are considered outliers because of the

[14] Neither of the two sets of participating usage metrics include the in-app purchase-related metrics, because this application contains no in-app purchases

exceptionally high number of exceptions occurring on these dates. The condition sets with the corresponding Alpha values are provided in Table 9.

		Included usage metrics	
Experiment duration	Measurement period	All metrics, besides purchase-related	All metrics, besides purchase-related & exception-related
08.05.2014 to 05.10.2014	1 day (daily)	0.60	-0.91
	1 week (weekly)	0.58	-0.59
17.06.2014 to 05.10.2014	1 day (daily)	0.66	0.34
	1 week (weekly)	0.64	-0.34

Table 9: Application 2 - Cronbach's Alpha (Basic user satisfaction score)

The observed values of Cronbach's Alpha are relatively high (although slightly below 0.7), especially for the duration 17.06.2014 - 05.10.2014. Such Alpha values contribute to the hypothesis that the basic user satisfaction score is a unidimensional construct.

Cronbach's Alpha test of the user satisfaction rating derived from micro benchmarking

Cronbach's Alpha has been calculated for the daily calculated alternative user satisfaction ratings of application 2 derived from micro benchmarking in the category for the duration (08.05.2014 – 05.10.2014) with resulting **Alpha value = -0.05** and for the duration (17.06.2014 – 05.10.2014) – **Alpha = 0.42**.

Alpha value of 0.42 is too low to prove the reliability and unidimensionality of a satisfaction rating derived from micro benchmarking. The reasons for the unsatisfactory results, as mentioned in subsection 5.4.3 could be the insufficient number of apps in the experiments and too broad boundaries of the selected app category.

5.5. Conclusion

In conclusion, from the results of the conducted experiments it can be inferred that the reliability of the basic user satisfaction score as a measure for satisfaction is high. The same conclusion cannot be made about the rating derived from micro benchmarking, at least not from the conducted experiments alone. The basic assumption behind this idea – that all, or

sufficiently high number of apps in a category have to be compared in order to produce a satisfaction rating is not accomplished – the number of conducted experiments is simply too low. In addition subsequent research seems to be necessary for determining the app category boundaries, which are essential for the method of micro benchmarking.

The following chapters (6 and 7) focus on the interviews conducted with app developers in order to determine the factors influencing app developers' decision to adopt the proposed alternative rating method as well as to assess the extent to which the novel rating method would be superior to the conventional rating method, in terms of participation rates, subjectivity and tampering susceptibility.

6. Adoption probability assessment – interviews

6.1. Introduction

Within the context of the engineering cycle, as defined by Wieringa [2], ***RQ2: What is the reliability of usage-based measurement of user satisfaction*** *and* ***RQ3: To what extent can usage-based measurement solve the inherent problems of opinion polling*** belong to the solution validation stage, while ***RQ4: What factors will influence app developers' decision to adopt usage-based measurement of user satisfaction*** relates to the implementation evaluation stage (**Figure 1**). While quantitative research has been applied for answering ***RQ2*** (discussed in detail in chapters 4 and 5), the qualitative approach has been selected for ***RQ3*** and ***RQ4***. More precisely, a series of interviews with app developers has been conducted in order to obtain answers to those questions.

This chapter is designed to present the interview design and provide the reasoning for its selection.

6.2. Qualitative research

The qualitative approach has been selected for research questions ***RQ3*** and ***RQ4*** for the following reasons. First of all, the decision to either adopt or not a certain methodology is influenced by subjective factors. Such a decision is rather personal and could reflect the decision maker's beliefs, emotions and feelings towards the methodology. Secondly, the quantitative data, collected throughout the experiments described in chapters 4 and 5 is of little use when attempting to answer ***RQ3***. An actual assessment of the impact of adopting usage-based measurement of satisfaction in apps on participation rates, subjectivity and tampering susceptibility would only be possible once a significant percentage of apps have been continuously rated using the new method. Because this goes beyond the scope of the current study, it has been decided that expert interviews could instead serve valuable insights for ***RQ3***.

6.3. Interview design

For preparing the interview questions, the findings of Riemenschneider et al. regarding criteria for acceptance of a novel information technology methodology [13] (see Section 2.3) are closely followed. The authors' work focuses on investigating five existing IT tool acceptance models and their applicability for methodology acceptance. The criteria found to be significant for methodology acceptance include: perceived usefulness, compatibility, subjective norm and voluntariness. These constructs are explained in **Table 4** in Section 2.3 and act as basis for preparing the interview design.

The interview design consists of three distinct parts:

1. Introductory (general) questions,
2. Explanation and demonstration of the developed prototype rating system and feedback (including focus on the 'Perceived usefulness' construct and juxtaposition between the traditional and the alternative method for user satisfaction rating)
3. Questions related to other acceptance influencers (including focus on the constructs 'Compatibility', 'Subjective norm' and 'Voluntariness')

The interview design is available in **Table 10**.

Adoption of usage-based measurement of user satisfaction – Interview design
Introductory (general) questions - For how long have you/your company been developing mobile apps? - Which platforms are you developing apps for? (iOS - iPhone, iOS - iPad, Android, Windows Phone) - How popular are your apps? How many downloads and active users do your app(s) have? - What per cent / how many of your users have rated your app(s) and are you actively asking your users to rate them from within the app(s)? - How satisfied are they with the app(s) on average (what is the average star rating) and what are in your opinion the reasons for this rating? - Do you believe the star rating of your app(s) is representative enough of how the bulk of your user base feels about your app(s)?

- In your opinion, how important is the star rating for the overall success of an app and your app(s) in particular?

- To what extent do you think the star rating system is fair?

Explanation and demonstration of the system and feedback

- Are you familiar with and are you using Google Analytics or another analytics suite in order to increase your awareness of how your users interact with the app(s)?

(if necessary, explain in detail)

(Explain the main idea behind this project, including the expected advantages of the proposed new rating method over the conventional star rating system;

Explain how the prototype works and how it is going to (hypothetically) work if widely adopted by app stores and developers;

Demonstrate on a laptop the basic workflow that an app developer follows when using the prototype system;

If necessary, explain and demonstrate in detail the process of fulfilling the main prerequisite - integrating the Google Analytics library and adding the tracking source code;

Explain that in a hypothetical state of wide adoption, the app store owners / Google / Apple will be interested in simplifying or even completely eliminating this step)

- In your opinion, to what degree can this new methodology solve the inherent problems of manual star ratings (subjectivity, low response/participation rates, susceptibility to tampering and abuse)?

-To what extent, can it hypothetically coexist with the conventional rating system? Can/should the star ratings be eventually phased out?

- Do you expect the adoption of the new system to introduce a different kind of problems? Please elaborate.

- Do you foresee any benefits ensuing from this adoption?

- Can you provide a rough estimation of the additional effort and respective costs you / your organization will have to incur in order to adopt the new rating system?

- In your opinion, are the expected benefits, if any, enough to justify the additional costs?

Acceptance influencers-related questions
- To what extent will the new method be compatible with your existing app development process? - Do you expect any major (or minor) disruptions in the workflow? - What difficulties (pitfalls) (if any) do you foresee in the adoption process? - If the adoption of the system is not enforced by the app stores, or if they are unwilling to facilitate the adoption process in any way, will this affect your willingness to adopt the system? How? - If a number of major names in app development (such as Facebook for example) adopt this system for their apps, will this influence your willingness to adopt the system? - If other app developers / companies you have (close) relations with or directly compete with adopt the system, will this affect your decision about the system? - If the app stores embrace the idea of the system and enforce its adoption by app developers, will this lead to negative attitude towards the system and its adoption?

Table 10: Interview design

6.4. Interviews – research sample

The interviews have been conducted with both company-employed and freelance developers from the Netherlands and Bulgaria. In terms of experience - all of the selected interviewees have at least three years of experience in mobile app development. This requirement guarantees that the subjects have not only dealt with the conventional method for rating apps, but have also developed an informed opinion about it. In addition, such experience duration would suggest that the interviewees would have had the time to not only become adept at using the core tools and technologies and applying best practices, but also familiarize themselves with secondary services like usage analytics platforms. On average their experience is 5.5 years, but several have more than ten years of experience.

	Company-employed	Self-employed / Freelance
The Netherlands	3	3
Bulgaria	4	4

Table 11: Interviewed app developers

In total 14 interviews have been conducted between 22nd of September 2014 and 4th of November 2014 – 6 in the Netherlands and 8 in Bulgaria. The distribution between individual and company employed interviewees is shown in Table 11. At the moment of preparing this paper in the Netherlands there are 62+ companies engaged in mobile app development, while in Bulgaria those are 54+. There is at least one such company that has offices in both countries. It is more difficult to estimate the number of active freelance developers, but it is safe to assume that in both countries they are already more than 1000.

When selecting the sample (size and particular subjects) the judgement (or purposeful) sampling technique [30] for qualitative research has been applied, because of the abovementioned considerations when narrowing down the research sample.

6.5. Conclusion

To sum up, a qualitative research approach, namely a series of interviews with app developers, has been selected to answer ***RQ3*** and ***RQ4***. The interviews include: a demonstration of the prototype of the proposed usage-based satisfaction measurement system, feedback and comparison with the conventional star rating method, and ultimately, based on adoption criteria suggested by Riemenschneider et al. [13], assessment of the factors that would influence the developers' willingness to adopt the novel user satisfaction rating method.

The following chapter (7) presents the results from the conducted interviews.

7. Interview results

7.1. Introduction

This chapter discusses the results from the interviews conducted with app developers. Section 7.2 focuses on the introductory part of the interview design. Section 7.3 discusses: the interviewees' impressions from the demonstration of the prototype of the proposed rating system, their feedback for usage-based measurement of satisfaction in apps in general, as well as their opinions on how it compares against the conventional rating method based on the latter's inherent weaknesses. Finally, Section 7.4 presents the developers' opinions on the factors (such as compatibility, voluntariness and subjective norm) that would have influence on their willingness to adopt the new rating method.

7.2. General information

Among all interviewees iOS is the most popular platform to develop apps – 85% (12 out of 14) are developing apps for iPhone and / or iPad. For Android and Windows Phone this figure is 64% and 29% respectively.

78% of the interviewed believe that the star rating (its average value as well as the number of users who have rated the respective app) is of significant importance for the overall success of an app. *"Before downloading an app, I always look at the rating and the reviews, but I always looks at every bit of information that is out there"* says an interviewee from a Dutch app development company. Another one says: "*The more stars we have, the higher the number of potential users interested in our apps*". Google has recently added an option for users to filter out apps with average rating of less than 4 (out of maximum 5) stars when searching for an app in the Play Store [31], thus reducing the chances of lower rated apps to be noticed at all.

Almost 80% of the participants perceive app ratings as important, however the majority of them question the manually submitted star rating's representativeness, fairness and usage as intended. Many of the interviewees had started discussing one, two or all three of the conventional rating method's inherent problems (low response rates, tampering susceptibility and subjectivity) even before they were asked for their opinion on the subject.

None of the developers reports star rating participation rates of their apps to be higher than 1% (less than 1 per 100 users have rated a particular app of theirs) and 93% of them are aware that star ratings are quite often fabricated by app developers. "*Yes people cheat! Because they can…*" stated one.

7.3. Feedback for the proposed alternative rating system

First of all, it should be noted that all of the interviewed developers are familiar with and actively using one or more analytics platforms for their apps. Among the ones they have mentioned are: Google Analytics, Mixpanel, Flurry analytics, Hockey, KISSmetrics and Splunk.

In general they find using the prototype of the usage-based user satisfaction rating system straightforward and intuitive and all of them imply that a combination of usage metrics' values could be used as an indication of satisfaction. However 29% (4 out of 14 interviewees) are sceptical about applying the micro benchmarking technique for deriving the reference values for the usage metrics (explained in subsection 3.3.2.1 and used in the prototype), mostly because: *"Current categories in the app stores are too broad, in my opinion. And I am not sure the app stores will be willing to further categorize every app out there, just to produce more precise ratings"*. Meanwhile, another 14% declare they are simply unsure how adequate micro benchmarking in particular would be. *"I don't know, it might be a good technique, but to be able to say that with certainty, we need lots of data"* says one of the interviewed developers. Overall however, the majority (79%) feel optimistic about the effectiveness of usage-based assessment of user satisfaction in apps in general, as well as its accuracy.

7.3.1. Usage metrics discussion

None of the participants disputes the current selection of metrics for the prototype. However, some suggest that two additional metrics, namely: *retention rate* (defined in Section 3.2) and *churn rate* are also important indicators of user satisfaction. *Churn rate* is defined as the number of users who cease using an app during a specified time period divided by the total number of users for the same period and is basically equal to 1 – retention rate. However, as discussed in Section 3.2, no reliable method for calculating both of these metrics in an automated manner currently exists. Furthermore, subsection 3.5.2

provides reasoning why the metric *percent of returning users*, which is included in the prototype, is supposedly better than these two metrics. Additionally, two of the interviewees explicitly state that according to them the usage metrics included in the rating algorithm are not necessarily equally important for the overall level of user satisfaction.

Asking the developers to what extent they believe that the proposed novel method for rating satisfaction in apps could alleviate the problems inherent to the manually submitted star ratings has provoked somewhat mixed reactions, as discussed below.

7.3.2. Usage-based versus conventional satisfaction rating: response rates

First of all, a significant improvement in terms of response rate is expected by 71% of the interviewed, because of the fact the rating process would become completely automatic. However it is suspected by some that this increase might not be sustainable in the long run. They argue that most probably a large percentage of app users currently do not realize that many app developers are already collecting usage statistics for their apps, even though when downloading the respective app, users are supposedly giving their consent to the app's privacy policy (which is required to inform them about the practice of collecting usage statistics). A hypothetical public announcement of ubiquitous adoption of usage-based ratings might provoke a sudden uproar from users. Those of them currently unaware of app usage tacking are likely to voice a negative opinion. As a result, app stores will most probably require that each application includes a clearly identifiable user option to turn off usage data collection altogether, thus reducing the response rate for the alternative user satisfaction rating.

On another note, 14% of the interviewees hint at another hindrance for response rate, namely the fact that in some markets a significant share of smartphone users do not have an active Internet connection all the time. According to a report by the Nielsen Company [32], in the first half of 2012, 43% of Brazilian smartphone users had a data plan, while for Russia, India and Turkey this figure was respectively: 43%, 57% and 49%. It is reported that in Russia and India, more often than not, smartphone users are relying on WiFi connectivity, which is not available everywhere, or "pay-as-you-go" data pricing, which has limited data capacity.

However, it should be noted that Google Analytics (as well as most probably other analytics platforms) would cache locally all the usage data the device could not send to the server, because of lack of Internet connectivity and transmit it at a later time, when connectivity becomes available. If each recorded usage event were time stamped, it would be possible to periodically recalculate usage-based ratings of apps concerning periods in the past, once usage data corresponding to those periods is eventually submitted to the system.

7.3.3. Usage-based versus conventional satisfaction rating: subjectivity

In terms of subjectivity, 78% of the developers argue that the new rating solution will improve on the old one and 64% believe it should make a profound difference in the right direction. Those who share this position, argue that users who enjoy using a particular app, but give it a low rating for unrelated reasons (such as the reasons discussed in Section 1.1), would not be given the opportunity to do so anymore. Two of the interviewees on the opposite side of the spectrum say that users who dislike an app for subjective reasons[15] are likely to influence the usage-induced rating, if not more strongly, then at least to a similar extent they influence conventional ratings of apps. They argue that such users, motivated by subjective reasons, would most probably uninstall or stop using the app they dislike altogether and as a result – the value of the usage metrics like *percent of returning users* and *retention rate* will decline, which will lower the overall usage-based rating of the app.

7.3.4. Usage-based versus conventional satisfaction rating: tampering susceptibility

As far as the new rating method's superiority over the currently adopted method in terms of tampering susceptibility is concerned, it can be said that the majority of the interviewed developers (79%) feel uncertain. They mention two distinct types of potential threats to tampering protection that would stem from adopting usage-based measurement of satisfaction.

[15] These particular developers give the following examples for subjective reasons to dislike an app: not enough knowledge in order to use the app accordingly, unwillingness to invest sufficient time in getting familiar with a more complex app

Firstly, one of them states that: *"It is possible to write an automated piece of software that simulates user behaviour and run it on hundreds or thousands of simulated devices, especially if you are simulating Android devices. On the iOS platform this is less likely to succeed"*. Secondly: "*A widespread misuse of usage tracking code* (purposely misplaced analytics API method calls) *is highly likely to occur*". For example, the developer can call the method that reports to the rating system that the user has opened a certain screen of the app and the method reporting the closure of the screen in a loop and even add a randomly changing period of inactivity between each call of the first method and each call of the second one.

The effectiveness of the first type of tampering is to a varying extent diminished by the fact that an exact match between simulated user presence and behaviour and those resulting in perfect usage-derived satisfaction rating would be hardly achievable (the values of the usage metrics leading to a perfect score are constantly changing and this change is influenced by external factors[16], which are difficult to compensate for by simulated usage). Another more radical solution for preventing fraud with simulated devices would be to devise a way that lets the rating system distinguish between real and simulated mobile devices, by using unique hardware identifiers, for example. However, of course, cheaters might somehow discover a way to circumvent such a protection.

As far as the problem of developers' practice of putting usage tracking related method calls at wrong places in the app's source code or using them in a wrong way is concerned, there are two possible solutions, both of which will require that the usage-based satisfaction measurement method is adopted by app stores. The first approach involves extensively checking the mobile application (binary) file after a developer uploads it for distribution in the respective app store. This check is going to verify that the developer is using all usage tracking methods according to prescription. Such a check cannot be performed for the source code of the app (the developer is not submitting the source code, only the compiled application file), unless the check is built into the software development environments used for writing and compiling the applications. The second approach is more radical and would require the smartphone operating system vendors' involvement. This approach will basically

[16] If the micro benchmarking technique (explained in Chapter 3) for deriving usage-based satisfaction ratings is applied, an app's rating depends not only on the app's own usage metrics values, but also on those corresponding to all other apps it is directly competing against.

eliminate the need for app developers to add usage-tracking code to their apps, because usage tracking will be handled at lower, operating system level, instead of the application level. Companies such as Google and Apple could embrace the idea of embedding the app usage tracking logic within their operating systems at some point in the future, in case they decide to adopt usage-based measurement of user satisfaction.

7.3.5. Other feedback for the proposed alternative rating system

When asked whether the novel satisfaction rating method should coexist with the currently adopted one or if the new one should replace the old one altogether, the results are as follows. 64% of the interviewees believe that both solutions can and should coexist for a while so that *"results from both can be compared"*, while 21% think that the new method should directly replace conventional star ratings if proven to be superior in terms of *"accuracy"* and *"fairness"*. One of the developers believes that the two types of rating measure two distinct aspects of satisfaction (from the perspective of user psychology) and for that reason, the rating methods should coexist. According to him the conventional rating represents *"perceived satisfaction"* and the usage-based rating – *"actual satisfaction"*.

In terms of perceived benefits, expected to stem from usage-based rating's adoption, the developers state: *"increased developer awareness of what functionalities* [in the apps] *are frequently used and enjoyed by the users"* and *"the fact that a process that is current manual and takes away from the user's time and attention could become automated is enough of a benefit by itself"*.

As far as anticipated problems are concerned, the developers once again state the expected new kinds of rating tampering threats (discussed in the previous subsection) as well as the highly probable privacy related issue of users unwilling to accept that their behaviour on their smartphones (the way they use their apps) is being tracked. This potential problem is discussed in subsection 7.3.2.

Based on their experience with existing usage analytics packages and the expected analogy between them and the new system for usage-based satisfaction measurement, in terms of complexity of integration, the prevailing majority of interviewees (86%) believe the effort and cost of adopting and integrating the new system to be "negligible" and "insignificant".

Furthermore, 93% of them agree or strongly agree with the statement that the benefits that they expect to result from adopting the new system are significant enough to justify the additional costs linked with the adoption and integration process.

7.4. Acceptance influencing factors besides *perceived usefulness*

7.4.1. Compatibility

In terms of compatibility with their established app development process, all of the interviewees think the new rating method is going to be "highly", "substantially" or "sufficiently" compatible. 43% of them expect only minor disruptions in the workflow resulting, for example, from: "the learning curve for studying the new usage tracking API". 14% express apprehension of more significant disruptions, such as: *"in the beginning, integrating the new system might lead to our apps crashing or showing <Application not responding> messages on Android devices produced by quirky manufacturers. The system needs to be extensively tested for compatibility with such devices. Quite often these companies break core Android functionalities when developing their devices* [Android is largely open source and Android device makers are free to change certain aspects of the operating system running on their devices]*"*.

7.4.2. Voluntariness

The developers' willingness to adopt the new rating method is somewhat negatively correlated with the voluntariness of this decision. 14% of the interviewed will be happier to adopt the method, if app stores enforce the adoption, rather than if it they make the adoption optional. The reasoning for this is: *"If the app stores enforce the new rating system, that means they have tested it and have proven it is a good solution. That is certainly going to boost my confidence in it* [the usage-based rating system] *too"*. However, the majority (64%) state that whether the adoption is enforced, or not, would bring hardly any difference on their decision to adopt the system. Similarly, only 21% report that an app store's decision to facilitate the adoption process (provide developers some form of assistance) after the adoption has been hypothetically enforced by that app store, could somehow further influence their willingness to adopt.

7.4.3. Subjective norm

When looking at the effect of third parties' decision to adopt the alternative satisfaction rating method could have on the interviewees' own willingness to adopt it, the results are as follows: 36% expect some positive influence by direct competitors' decision to adopt (it is going to act as a form of assurance or accelerator of the adoption process). The remainder do not expect any influence at all. 29% expect to be positively influenced by contracts' or other fellow developers' decision to adopt, while the rest expect no difference. In case a big name company (in the rank of Facebook or Twitter) decides to adopt the new system, this might affect 29% of the interviewees somewhat positively and 7% somewhat negatively.

7.5. Conclusion

The decision to pursue the qualitative approach (conducting interviews with mobile app developers) in an attempt to answer ***RQ3*** and ***RQ4*** has resulted in valuable insights on the proposed alternative (usage-based) user satisfaction rating method.

In relation to ***RQ3***, usage-based measurement of user satisfaction in apps is expected to be significantly better than manually submitted star ratings in terms of subjectivity and participation rate. However the level of improvement in the latter factor could be somewhat diminished by concerns for decline in user privacy. As far as protection against tampering is concerned, there is a new challenge to overcome (misuse of usage tracking source code). Without achieving this, the new rating method could hardly improve on the currently adopted one.

Regarding ***RQ4***, the most important factor influencing app developers' willingness to adopt the new rating method appears to be *perceived usefulness*. The majority of the interviewees (79%) believe usage-based measurement of satisfaction would be more effective and perhaps more accurate than opinion polling. This alone is enough of an incentive for them to want to adopt or at least try the new rating method. *Compatibility* is in general also a very important factor. In this case there is almost no concerns about compatibility issues – usage-based satisfaction rating is expected to be highly compatible with app developers' established work processes and only minor problems in the adoption process are expected. *Voluntariness* of adoption and *subjective norm* appear to be factors of relatively weak

influence on developers' willingness to adopt. An adoption enforced by app stores as well as competitors' or fellow developers' decision to adopt might slightly boost one's own willingness to adopt.

8. Conclusion

The purpose of this study was to design and validate an alternative to the currently adopted method for measuring user satisfaction in mobile apps (manually submitted star ratings), which was found to be flawed. In compliance with design science principles and more specifically, the engineering cycle defined by Wieringa [2], the research process consisted of the following stages:

- Identification of the problems of the conventional rating method and the requirements for an alternative
- Design of a usage-based user satisfaction measurement mechanism that meets the identified requirements
- Validation of the proposed alternative satisfaction measurement method
- Evaluation of the alternative solution

Each of these stages, the corresponding research questions and their answers are discussed below.

8.1. Identification of the problems of the conventional rating method and the requirements for an alternative

Based on a review of the available literature sources [3, 4, 16, 17, 18, 19] and discussions with experts, three substantial problems inherent to the conventional rating method were identified, namely: rather low response rate, subjectivity and high tampering susceptibility, thus proving an answer to ***RQ0: What are the problems of the conventional method for measuring user satisfaction by manually submitted star ratings***. In addition, the following requirements for an alternative usage-based user satisfaction measurement solution were suggested: automation, universal applicability, flexibility, high response rates, high level of protection against tampering. These requirements are explained in **Table 5** in Chapter 3.

8.2. Design of a usage-based user satisfaction measurement mechanism that meets the identified requirements

This stage aimed at answering ***RQ1: What would the properties and structure of a usage-based user satisfaction measurement mechanism that meets the identified requirements be.*** A hypothesis was built upon existing theoretical frameworks such as: the ISSM model [7, 8, 20], Bailey and Pearson's formulation of satisfaction [12] and SERVQUAL [21] and taking into consideration the list of requirements for an alternative user satisfaction rating method identified in the preceding stage. It states that a measure for user satisfaction in mobile apps can be defined as the weighted sum of individually rated, automatically measurable, satisfaction-related usage metrics. Several alternatives for calculating the individual ratings for each metric were suggested, but two of them (micro benchmarking (subsection 3.3.2) and basic user satisfaction score (subsection 3.3.3)) were selected as the ones best satisfying the identified requirements. Afterwards a prototype system was developed that embodies the hypothesis for usage-based satisfaction measurement and these two techniques. It takes advantage of an established mobile analytics platform (Google Analytics). Serving as a proof of concept, this prototype was utilized in the stages of solution validation and evaluation.

8.3. Validation of the proposed alternative satisfaction measurement method

With the help of the developed prototype, experiments with two real-life applications were conducted. The aim of these experiments was to collect quantitative data (raw usage data, calculated alternative ratings and submitted conventional ratings) for the duration of four months, in order to try to answer ***RQ2: What is the reliability of usage-based measurement of user satisfaction***, as part of the proposed solution's validation. The statistical analysis of the gathered data led to the conclusion that the basic user satisfaction score is highly reliable as a measure of user-satisfaction. However, the same conclusion was not reached about the rating derived by applying the micro benchmarking technique.

As another step in the solution validation, qualitative data was collected via expert interviews, in an attempt to answer ***RQ3: To what extent can usage-based measurement solve the inherent problems of opinion polling***. Analysis of the data led to the following

conclusion: the novel solution is expected to achieve higher objectivity and much higher participation rate than the conventional satisfaction rating, even though concerns for user privacy might somewhat weaken the improvement in the latter. Another conclusion was that advancement in protection against tampering would be challenging – it is achievable only if measures to prevent misuse of usage-tracking source code and possibly measures for discerning between real and simulated mobile devices are taken. These measures are going to require additional, possibly significant investments.

8.4. Evaluation of the alternative solution

The second goal of the conducted expert interviews was to reach an answer to ***RQ4: What factors will influence app developers' decision to adopt usage-based measurement***. The data collected led to the conclusion that the most important decision-influencing factor in this case is *perceived usefulness,* followed by *compatibility*. *Voluntariness* and *subjective norm* appear to be significantly less important.

In general almost 80% of the interviewees are willing to try the alternative rating method and perhaps adopt it. This is mostly because they believe it could be better than the currently adopted in terms of effectiveness and accuracy and also because they find it highly compatible with their established work processes and expect very few issues to arise throughout and after the adoption.

8.5. Limitations

It should be noted that the number of conducted experiments for rating real life apps, based on their usage, is significantly lower than originally planned. In order to participate in one of the experiments, an app has to be tracked by Google Analytics and its developer should opt in for sharing the usage data with the prototype of the usage-based satisfaction rating system.

The first issue here is that not every developer is already using Google Analytics, even though it is a very popular analytics platform. Furthermore, the incentive for such developers to not only integrate Google Analytics within their apps, but also to release the updated versions of the apps soon after the integration and share their usage data with a

third party, simply in order to take part in a research project is rather small. Because it is quite farfetched to expect that many app developers would be willing to go through all of these steps, the vast majority of developers approached (26 in total) were known to already be using Google Analytics for tracking the usage of their apps.

Unfortunately, the measures taken to overcome the first issue have done nothing against the second one, namely the apparent unwillingness of app developers to share usage data of their apps with third parties (for research purposes). Out of 26 only 3 developers expressed willingness to participate by providing access to their usage data and one of those 3 changed their mind eventually.

The low participation rate's negative impact on external validity is especially strong in the case of the satisfaction rating derived via micro benchmarking. The core principle of that technique – that all, or representative enough number of apps in a single category are to be compared against each other in order to produce a satisfaction rating is hardly satisfied.

In addition, it is difficult to make assumptions about the repeatability of the results from the experiments. Even though the proposed solution has been designed with universal applicability and flexibility in mind (requirements R2 and R3 in **Table 5**) it is unknown if conducting the experiments with different kinds of apps exhibiting different usage patterns will produce similar results.

Furthermore, the credibility of the results from the expert interviews could be somewhat compromised by the research sample size. The sample size has been affected by the limitation in time and resources, which is in place due to the fact this study is done as part of a master thesis project. In addition, even though the number of active app development companies in the Netherlands and Bulgaria is more or less known, the sizes of the populations of company-employed and self-employed app developers are unknown. For that reason, the sample might not be representative enough, or the results might exhibit a bias towards either of the two groups. It should be noted however, that, at least in the conducted interviews, no significant difference in opinion between the two groups has been observed.

A representative of one of the major app stores has been approached for an interview, in order to assess that app store's willingness to adopt the proposed alternative solution. However that person has demonstrated no interest in participating in such an interview.

8.6. Recommendations for future research

As discussed in Section 3.4, for the purpose of this study the importance factors (W_j) corresponding to each usage metric participating in the rating algorithm are set to be equal. This has been done for the sake of simplicity and because of lack of sufficient time and resources for additional research. Further research (with various kinds of mobile applications) is needed in order to conclude how important for user satisfaction each usage metric actually is. It is currently assumed that the importance is going to vary among different app categories, but this assumption should be backed by scientific evidence.

Another possible direction for future research would be to design and validate different (non-summation) models for satisfaction based around satisfaction-related usage metrics.

It would also be interesting to further study the relationship and causality between usage and satisfaction in the context of mobile applications and other information systems.

9. References

[1] K. Peffers, T. Tuunanen, M. A. Rothenberger, S. Chatterjee, A Design Science Research Methodology for Information Systems Research, Journal of Management Information Systems 24. 3, 2007

[2] R. Wieringa, A Unified Checklist for Observational and Experimental Research in Software Engineering (Version 1). Technical Report TR-CTIT-12-07, Centre for Telematics and Information Technology, University of Twente, Enschede, 2012

[3] Rajesh Vasa , Leonard Hoon , Kon Mouzakis , Akihiro Noguchi, A preliminary analysis of mobile app user reviews, Proceedings of the 24th Australian Computer-Human Interaction Conference, p.241-244, November 26-30, 2012, Melbourne, Australia.

[4] Henze, Niels, Martin Pielot, Benjamin Poppinga, Torben Schinke and Susanne Boll, My App is an Experiment: Experience from User Studies in Mobile App Stores, International Journal of Mobile Human Computer Interaction (IJMHCI) 3 ,2011.

[5] Q. Xu, J. Erman and A. Gerber etc. Identifying Diverse Usage Behaviors of Smartphone Apps. In IMC, 2011.

[6] A. Tongaonkar, S. Dai, A. Nucci, and D. Song. Understanding Mobile App Usage Patterns Using In-App Advertisements. In International Conference on Passive and Active Measurement (PAM), 2013.

[7] William H. Delone , Ephraim R. McLean, The DeLone and McLean Model of Information Systems Success: A Ten-Year Update, Journal of Management Information Systems, v.19 n.4, p.9-30, Number 4/Spring 2003.

[8] DeLone, W.H., and McLean, E.R., Information systems success: The quest for the dependent variable, Information Systems Research, 3, 1 ,1992.

[9] Igbaria, M. and Tan, M., The consequences of the information technology acceptance on subsequent individual performance, Information & Management. 32. 3, 1997

[10] Gelderman, M., The relation between user satisfaction, usage of information systems and performance, Information & Management, 34. 1, 1998

[11] Torkzadeh. G. and Doll, W.J., The development of a tool for measuring the perceived impact of information technology on work. Omega — The International Journal of Management Science. 27. 3, 1999

[12] J.E. Bailey, S.W. Pearson, Development of a tool for measuring and analysing computer user satisfaction, Management Sciences 29. 5, 1983

[13] C. Riemenschneider, B. Hardgrave, and F. Davis, Explaining software developer acceptance of methodologies: a comparison of five theoretical models, IEEE Transactions on Software Engineering, vol. 28, no. 12, pp. 1135–1145, 2002

[14] Yin, R. K., Case Study Research (3rd Ed.), Thousand Oaks, CA: Sage Publications, 2003

[15] P. Baxter, S. Jack, Qualitative Case Study Methodology: Study Design and Implementation for Novice Researchers, The Qualitative Report 13. 4, 2008

[16] C. Thomas, Flappy Bird's Smoke & Mirrors – Is Something Fishy Going On? Using Bots For App Store Rankings, 2014, http://www.bluecloudsolutions.com/blog/flappy-birds-smoke-mirrors-scamming-app-store

[17] D. Takahashi, Inside Apple's Crackdown On App Ranking Manipulation, 2012, http://www.businessinsider.com.au/apples-crackdown-on-app-ranking-manipulation-2012-7

[18] https://twitter.com/jimyounkin/status/408779815244664833

[19] Eff Your Review, 2014, http://effyr.tumblr.com

[20] DeLone, W.H., and McLean, E.R., Information Systems Success: The Quest for the Independent Variables, Journal of Management Information Systems, vol. 29, no. 4, pp. 7 – 62, 2013

[21] Francis Buttle, SERVQUAL: review, critique, research agenda, Vol. 30 Iss: 1, pp.8 – 32, 1996

[22] J.Joseph Cronin Jr.a, Michael K Bradyb, G.Tomas M Hult, Assessing the effects of quality, value, and customer satisfaction on consumer behavioral intentions in service environments, Journal of Retailing, vol. 76, no. 2, pp. 193–218, 2000

[23] D. Iacobucci, K.A. Grayson and A.L. Omstrom, "The calculus of service quality and customer satisfaction: theoretical and empirical differentiation and integration", in Swartz, T.A., Bowen, D.E. and Brown, S.W. (Eds), Advances in Services Marketing and Management, Vol. 3, JAI Press, Greenwich, CT, pp. 1-68, 1994

[24] Google Mobile App Analytics, https://developers.google.com/analytics/devguides/collection/mobile

[25] R. J. Wieringa and J. M. G. Heerkens, The Methodological Soundness of Requirements Engineering Papers: A Conceptual Framework and Two Case Studies. Requirements Engineering, 11(4), pp.295-307, 2006.

[26] R. J. Wieringa, N. A. M. Maiden, N. R. Mead, and C. Rolland, Requirements Engineering Paper Classification and Evaluation Criteria: A Proposal and a Discussion. Requirements Engineering, 11(1), pp.102-107, 2006

[27] L. Cronbach, Coefficient alpha and the internal structure of tests, Psychomerika, 16, pp.297-334, 1951

[28] J.M. Bland, D.G. Altman, Measuring agreement in method comparison studies, Stat Methods Med Res, 8, pp.135-160, 1999

[29] Google Play Store, Paid App Availability https://support.google.com/googleplay/answer/143779

[30] M.N. Marshall, Sampling for qualitative research, Family Practice, 13, 522-525, 1996

[31] Google Adds Optional Rating Filter To Play Store Searches That Limits Results To Apps With 4 Stars Or More, October 2014, http://www.androidpolice.com/2014/10/19/google-adds-optional-rating-filter-to-play-store-searches-that-limits-results-to-apps-with-4-stars-or-more

[32] The mobile consumer: A global snapshot, February 2013, http://www.nielsen.com/content/dam/corporate/uk/en/documents/Mobile-Consumer-Report-2013.pdf

10. Appendices

10.1. Appendix 1 – Prototype screenshots

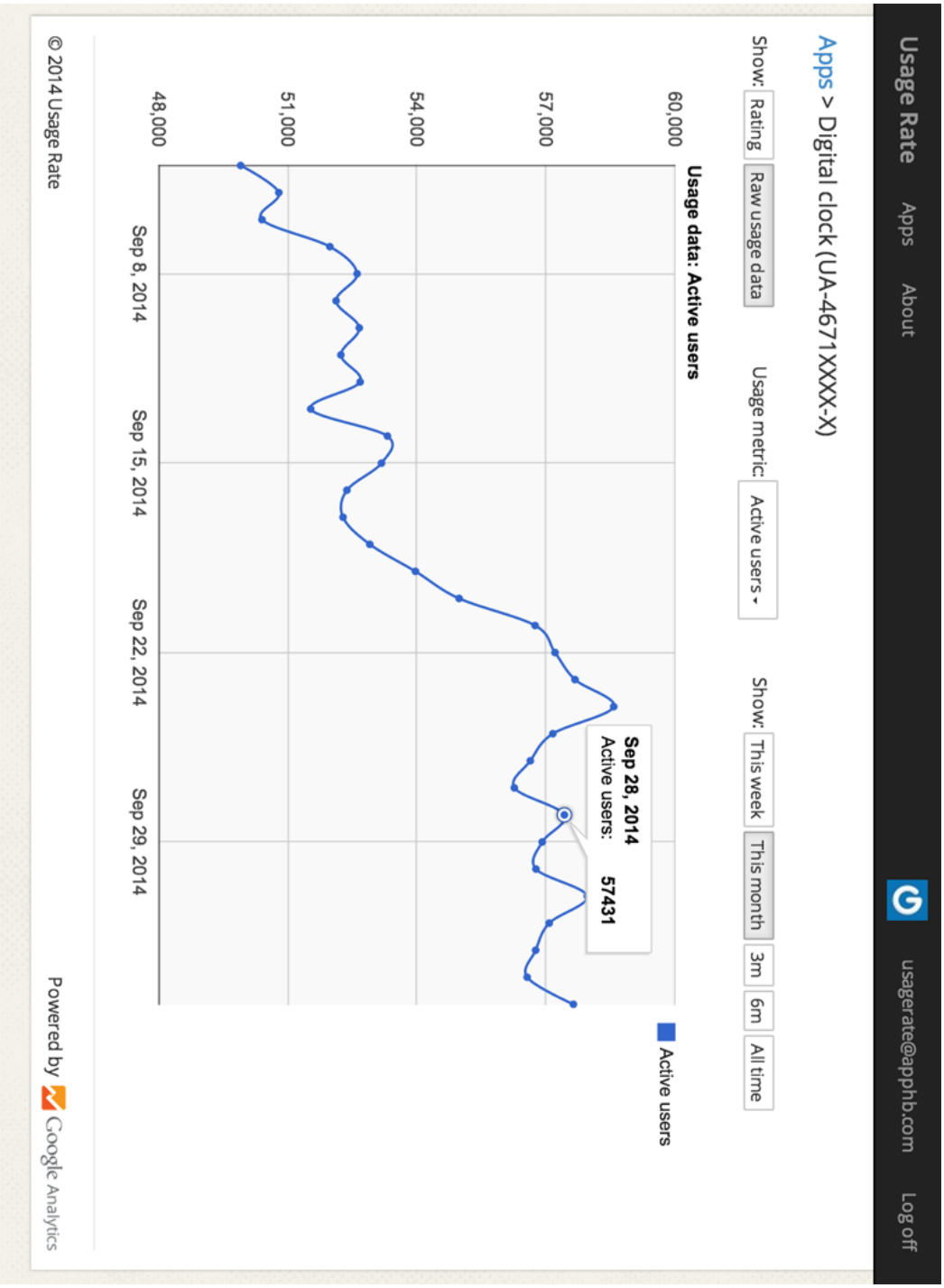

Figure 11: Application 1 – usage metric "Users" – daily values

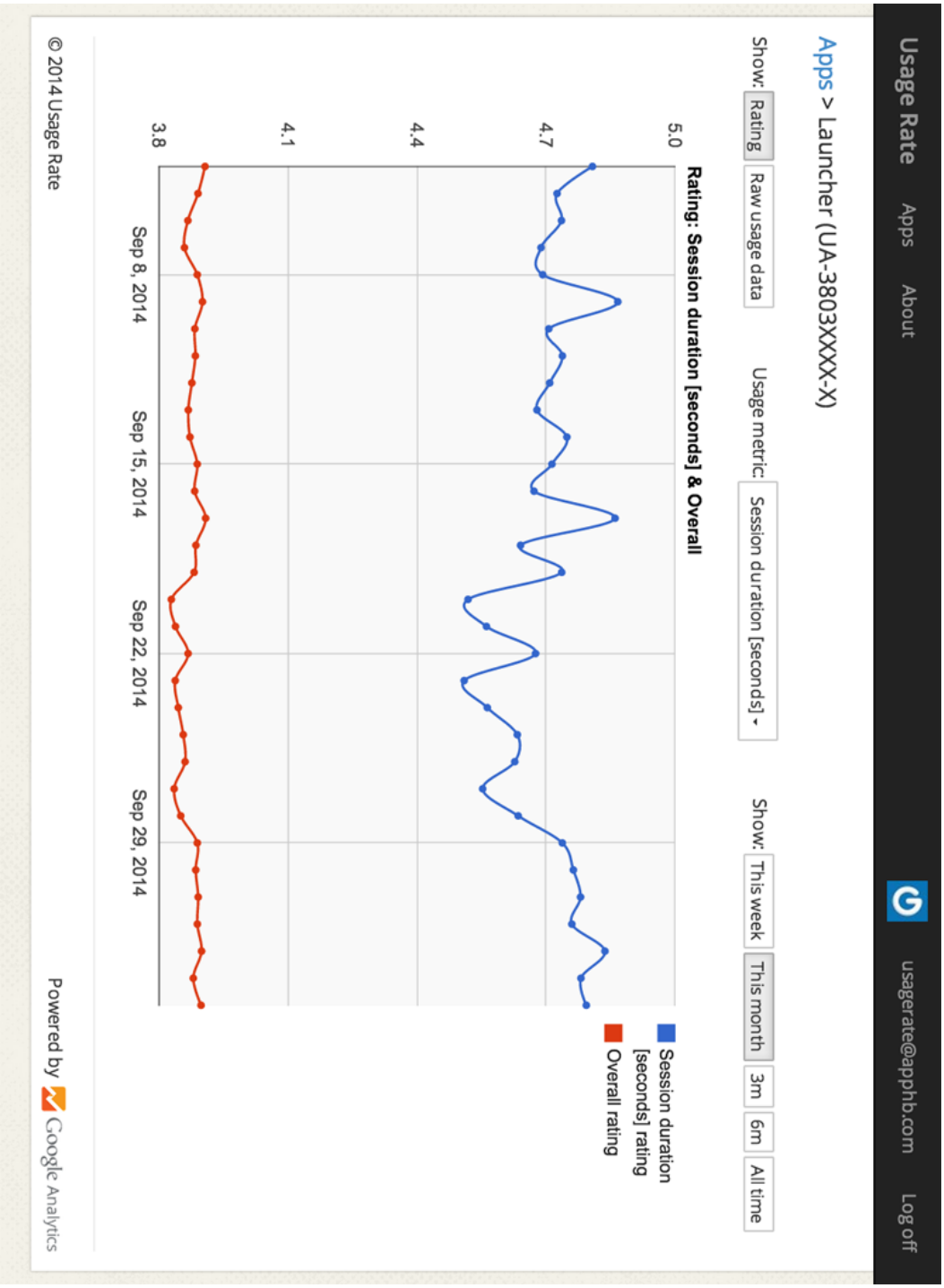

Figure 12: Application 2 – rating for “Session duration” and overall user satisfaction rating, calculated via micro benchmarking – daily values

10.2. Appendix 2 – Experiments, usage metrics data

10.2.1. Application 1

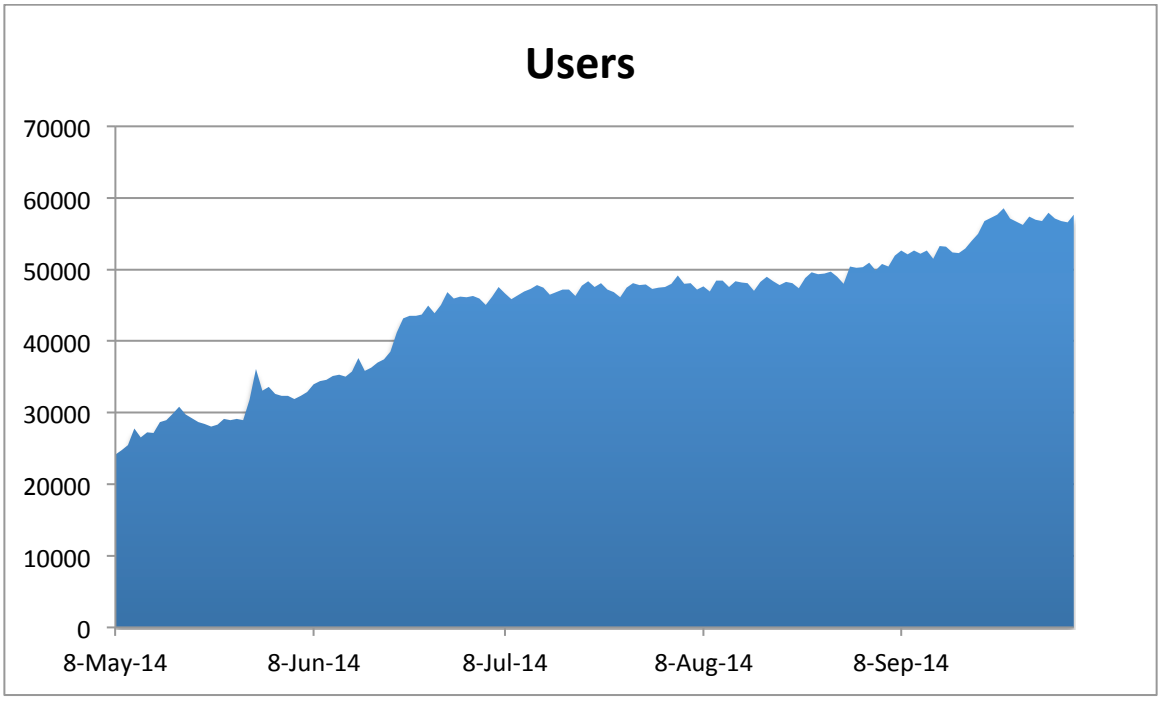

Figure 13: Application 1 – Users

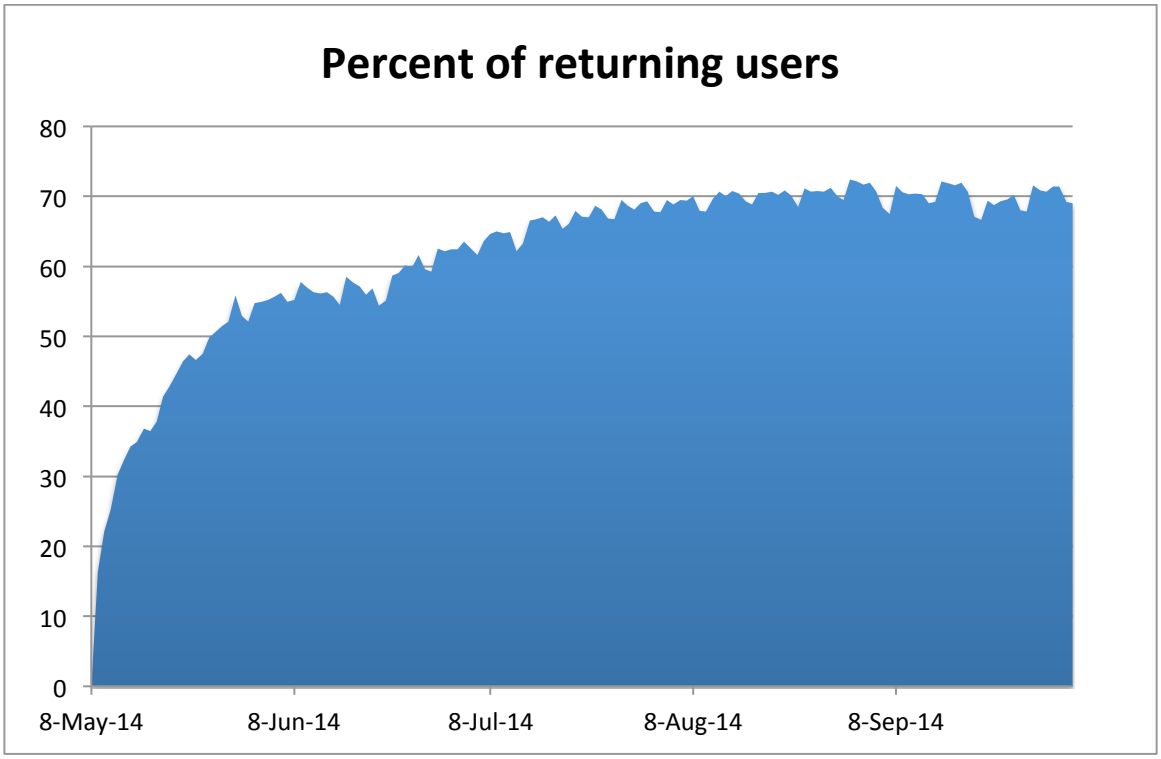

Figure 14: Application 1 – Percent of returning users

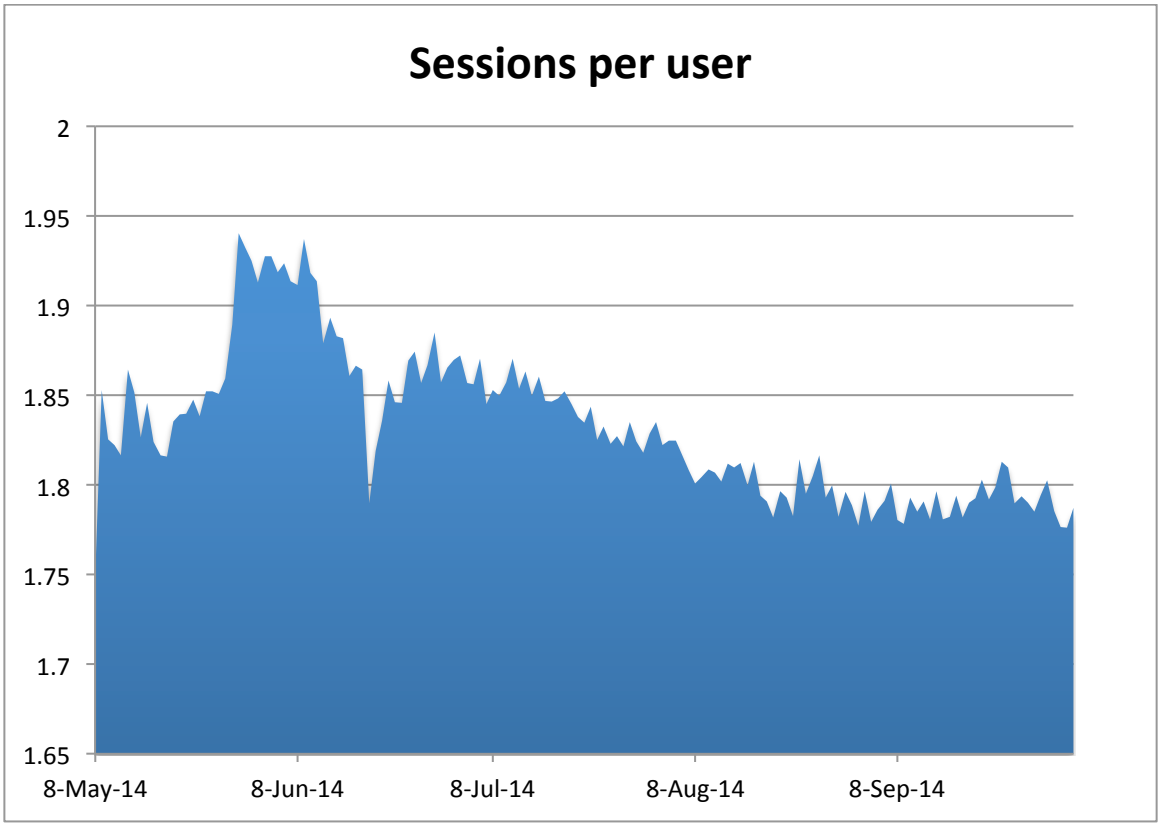

Figure 15: Application 1 - Sessions per user

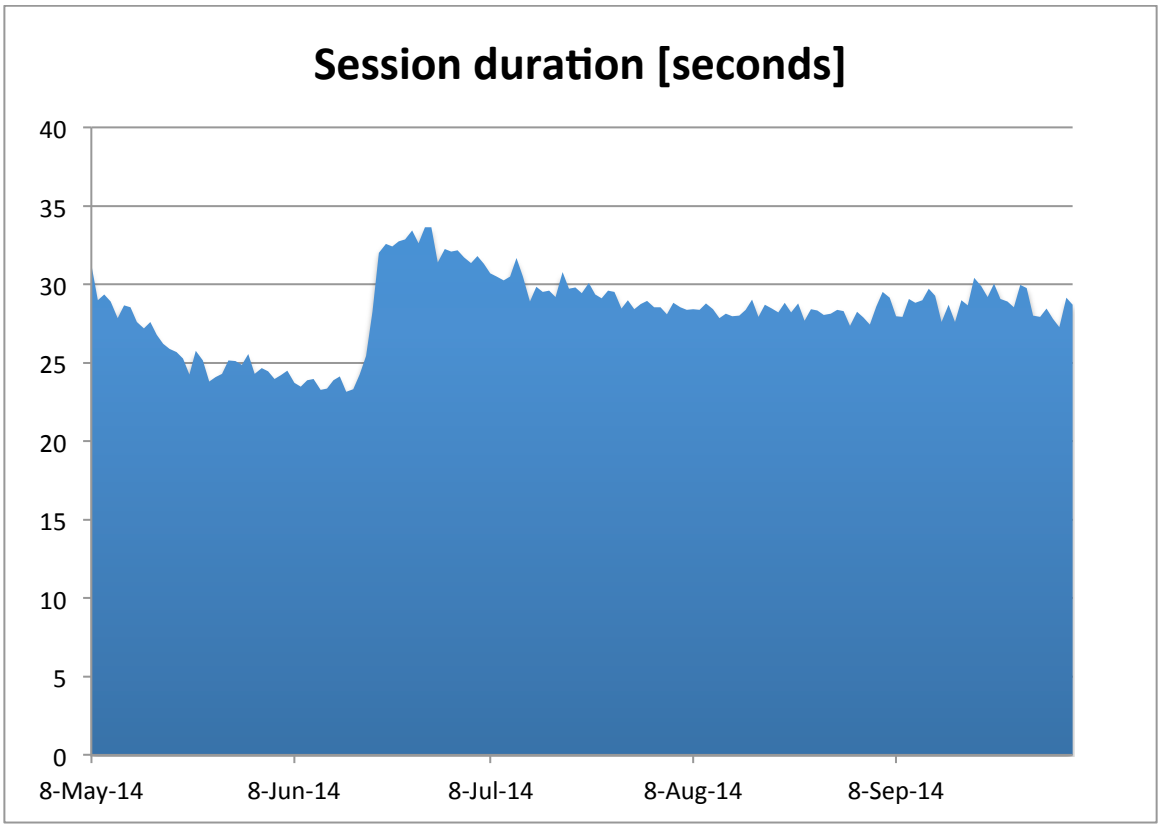

Figure 16: Application 1 - Session duration, seconds

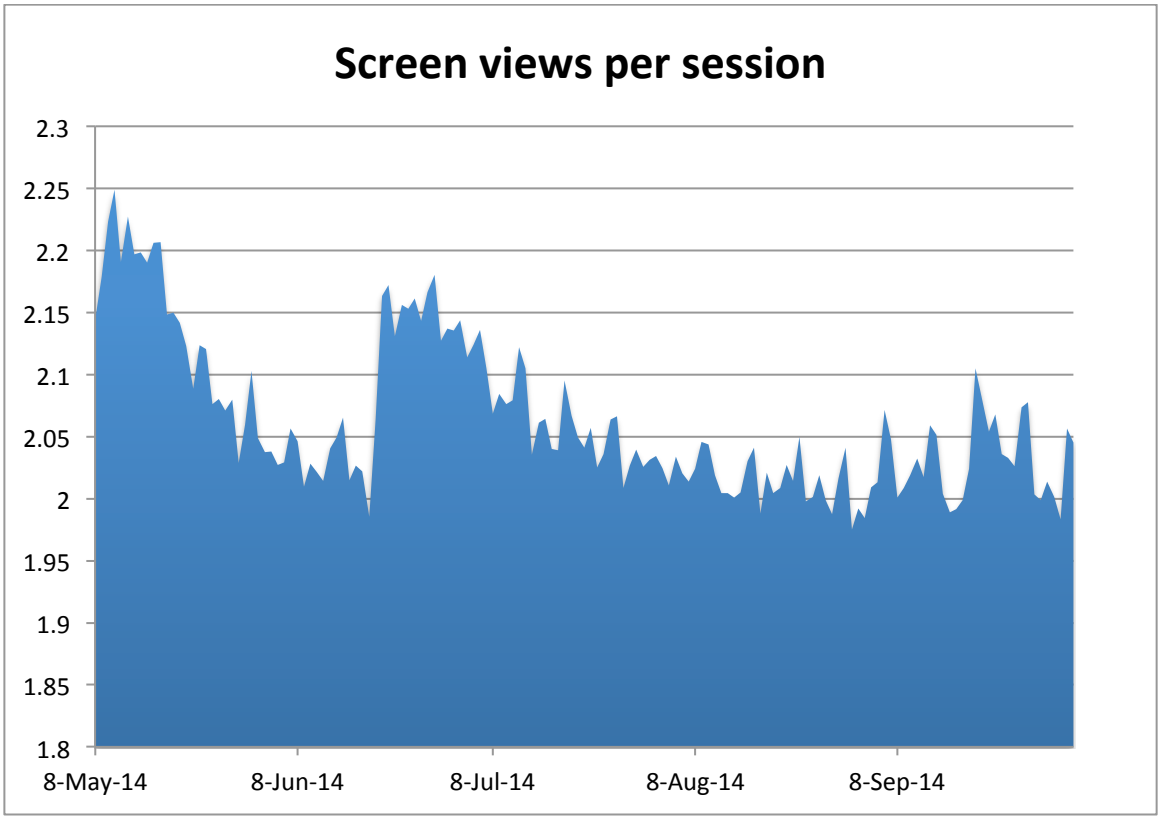

Figure 17: Application 1 - Screen views per session

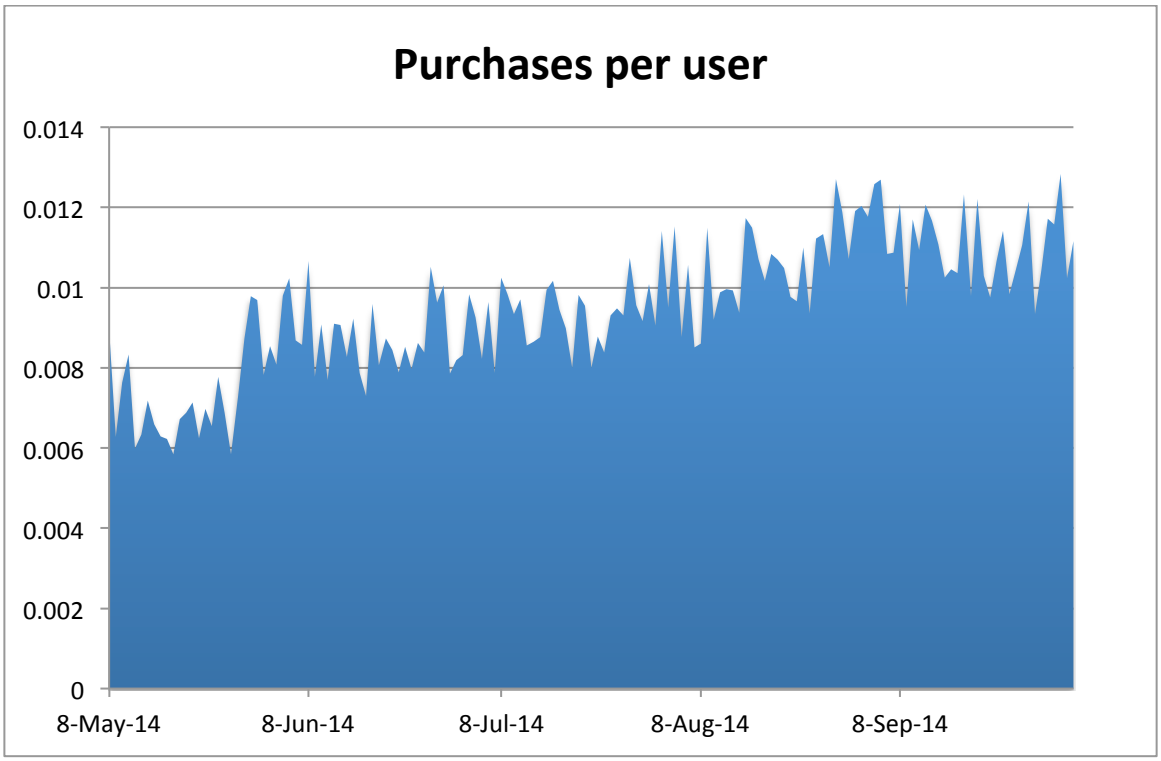

Figure 18: Application 1 - Purchases per user

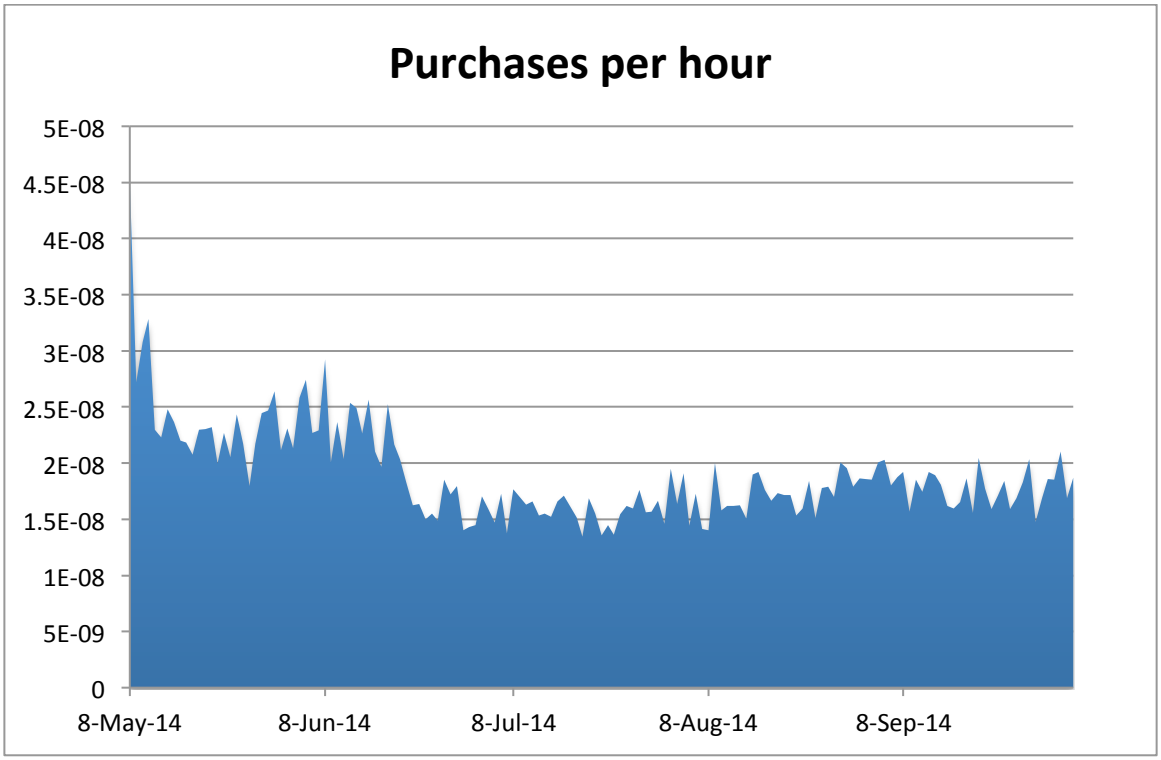

Figure 19: Application 1 - Purchases per hour

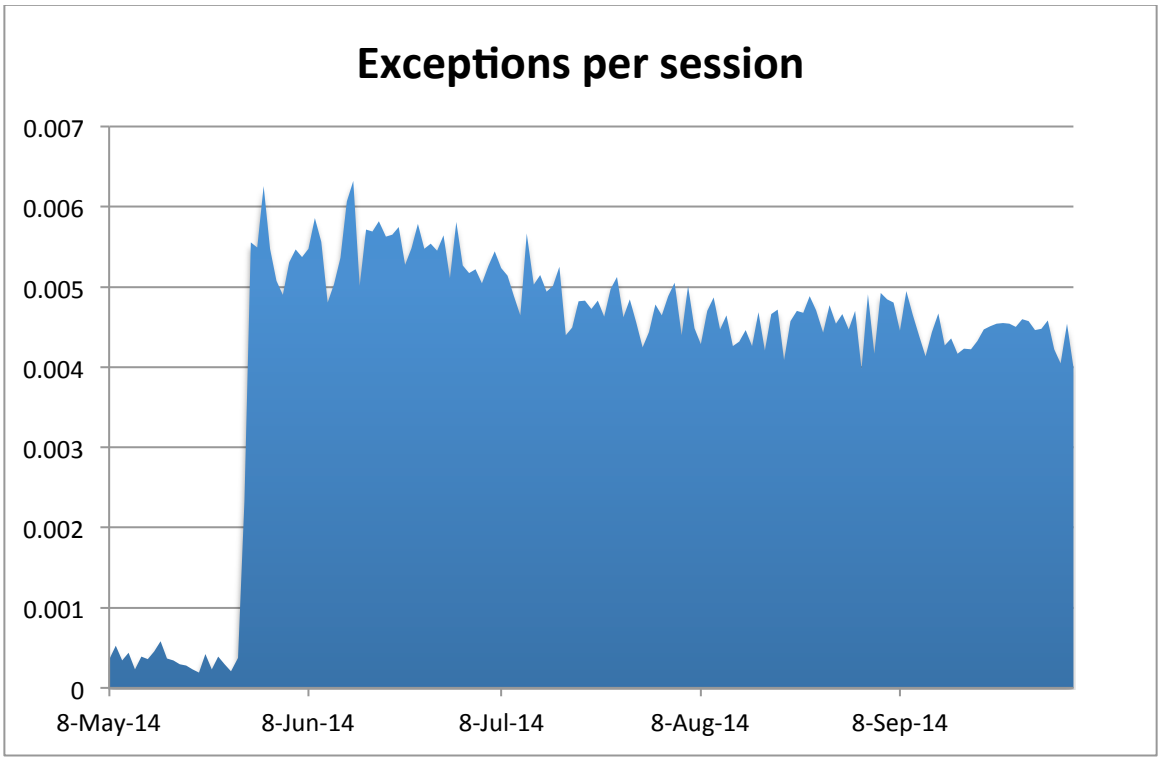

Figure 20: Application 1 - Exceptions per session

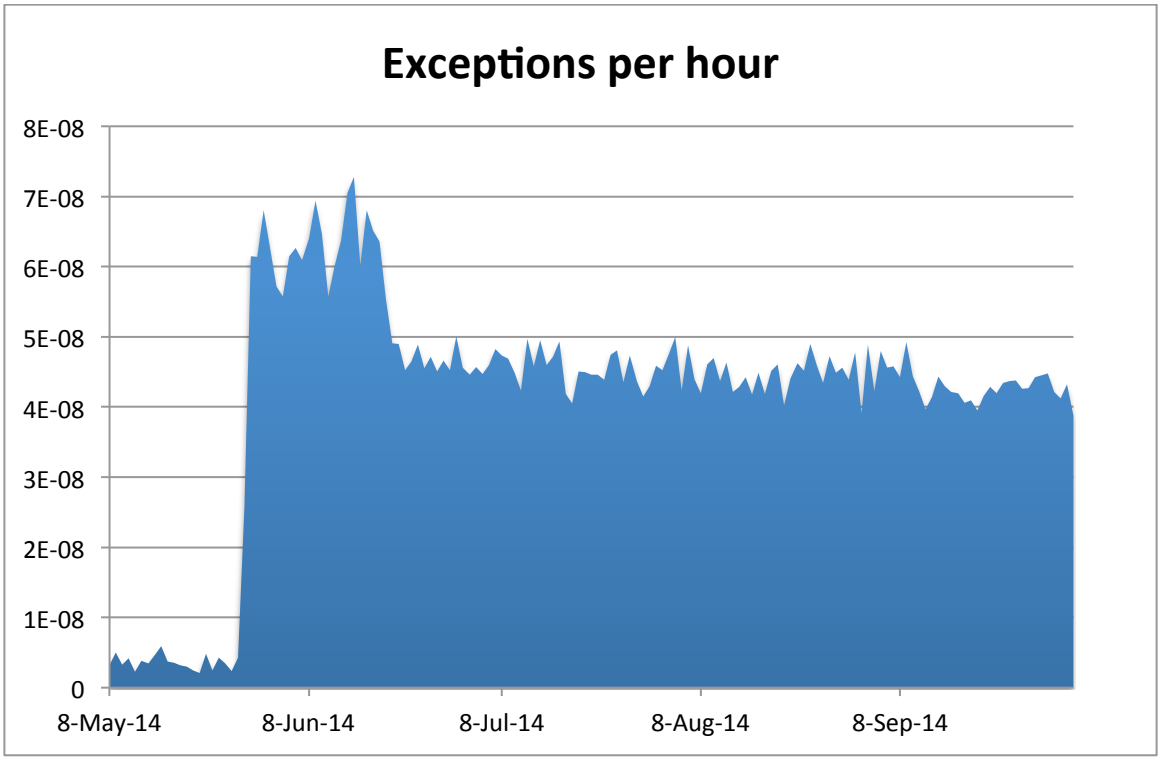

Figure 21: Application 1 - Exceptions per hour

Week start - Monday	Week end - Sunday	Users	Percent of returning users	Sessions per user	Session duration [s]	Screen views per session	Purchases per user	Purchases per hour	Exceptions per session	Exceptions per hour
5-May-14	11-May-14	85389	0.00	2.17	29.59	2.20	0.007823113	3.38669E-08	0.000426667	4.00521E-09
12-May-14	18-May-14	148530	12.58	2.46	27.72	2.20	0.006345979	2.25875E-08	0.000391237	3.91991E-09
19-May-14	25-May-14	143586	23.14	2.58	25.47	2.13	0.006895364	2.24024E-08	0.000291469	3.17932E-09
26-May-14	1-Jun-14	152525	30.65	2.75	24.74	2.07	0.008054681	2.27666E-08	0.0031701	3.55929E-08
2-Jun-14	8-Jun-14	154171	33.73	2.84	24.26	2.04	0.009230348	2.46251E-08	0.005297159	6.06619E-08
9-Jun-14	15-Jun-14	166980	35.01	2.82	23.72	2.03	0.008616162	2.32422E-08	0.005579597	6.53517E-08
16-Jun-14	22-Jun-14	182428	35.66	2.72	27.22	2.07	0.008272771	1.99613E-08	0.005613314	5.72731E-08
23-Jun-14	29-Jun-14	204539	38.78	2.84	33.06	2.16	0.009103323	1.6505E-08	0.005523777	4.64167E-08
30-Jun-14	6-Jul-14	208412	42.09	2.88	31.83	2.13	0.00876597	1.53959E-08	0.005269442	4.5984E-08
7-Jul-14	13-Jul-14	211683	44.21	2.88	30.78	2.09	0.009153647	1.6004E-08	0.005148907	4.64723E-08
14-Jul-14	20-Jul-14	208858	47.21	2.92	29.64	2.06	0.009305363	1.579E-08	0.004867323	4.56156E-08
21-Jul-14	27-Jul-14	210099	48.72	2.89	29.55	2.05	0.008975311	1.49593E-08	0.004817155	4.52803E-08
28-Jul-14	3-Aug-14	211240	50.33	2.89	28.66	2.03	0.009940149	1.656E-08	0.004629341	4.48691E-08
4-Aug-14	10-Aug-14	212811	51.17	2.86	28.48	2.03	0.009825054	1.63798E-08	0.004688805	4.57384E-08
11-Aug-14	17-Aug-14	211767	52.37	2.87	28.25	2.01	0.010450025	1.70708E-08	0.004443881	4.36974E-08

18-Aug-14	24-Aug-14	213951	52.94	2.83	28.45	2.02	0.010389868	1.68609E-08	0.004515116	4.40865E-08
25-Aug-14	31-Aug-14	218140	53.40	2.85	28.19	2.01	0.011096901	1.78908E-08	0.004636507	4.56946E-08
1-Sep-14	7-Sep-14	224106	53.58	2.83	28.30	2.01	0.011784722	1.89721E-08	0.004621658	4.53567E-08
8-Sep-14	14-Sep-14	232772	53.00	2.82	28.81	2.03	0.011296956	1.81662E-08	0.004525135	4.36225E-08
15-Sep-14	21-Sep-14	238385	52.99	2.83	28.85	2.03	0.010833482	1.73526E-08	0.004293572	4.13376E-08
22-Sep-14	28-Sep-14	254302	51.20	2.84	29.35	2.05	0.010781367	1.75616E-08	0.004543683	4.29983E-08
29-Sep-14	5-Oct-14	251920	53.40	2.84	28.18	2.01	0.011047136	1.78997E-08	0.004332108	4.26999E-08

Table 12: Application 1 - Usage metrics, weekly values

10.2.2. Application 2

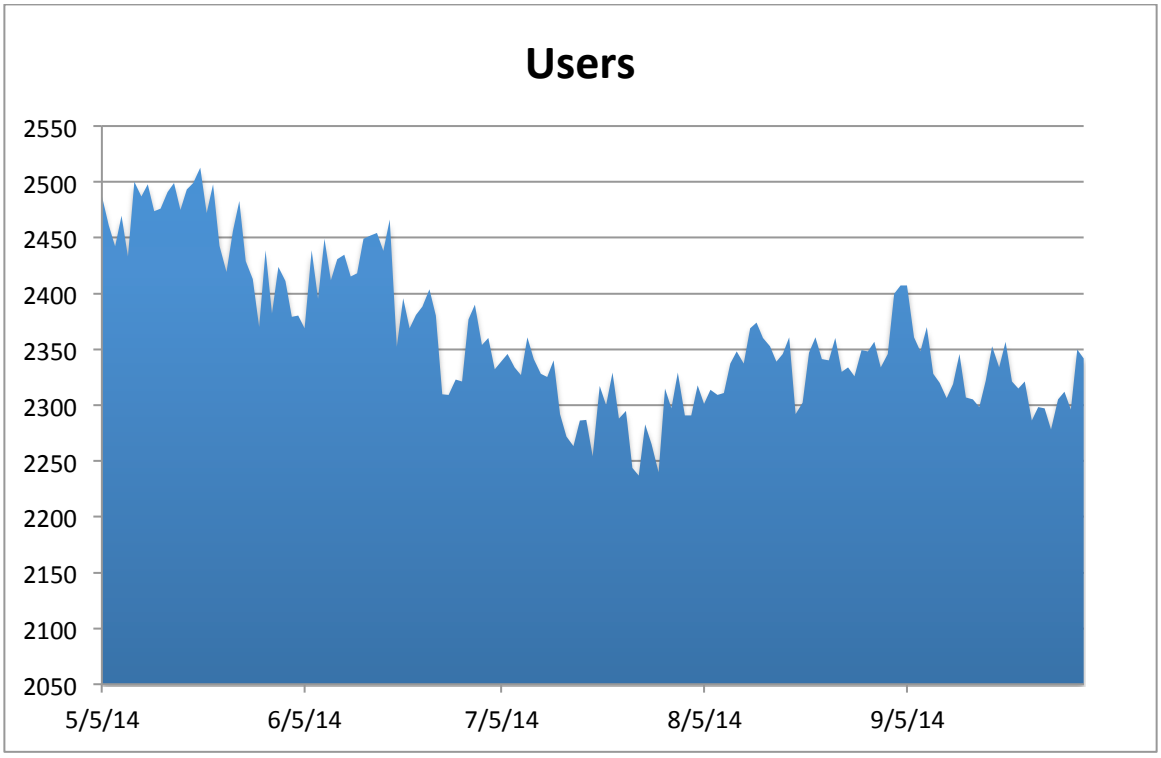

Figure 22: Application 2 – Users

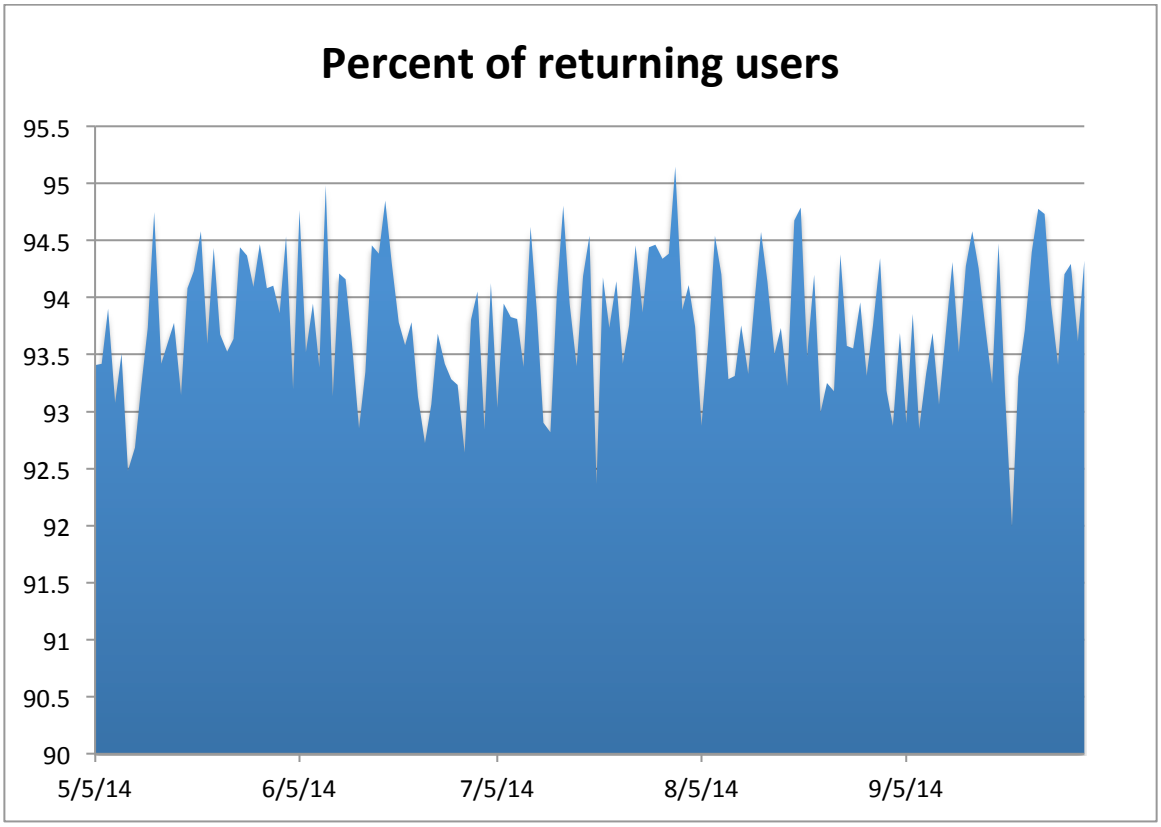

Figure 23: Application 2 - Percent of returning users

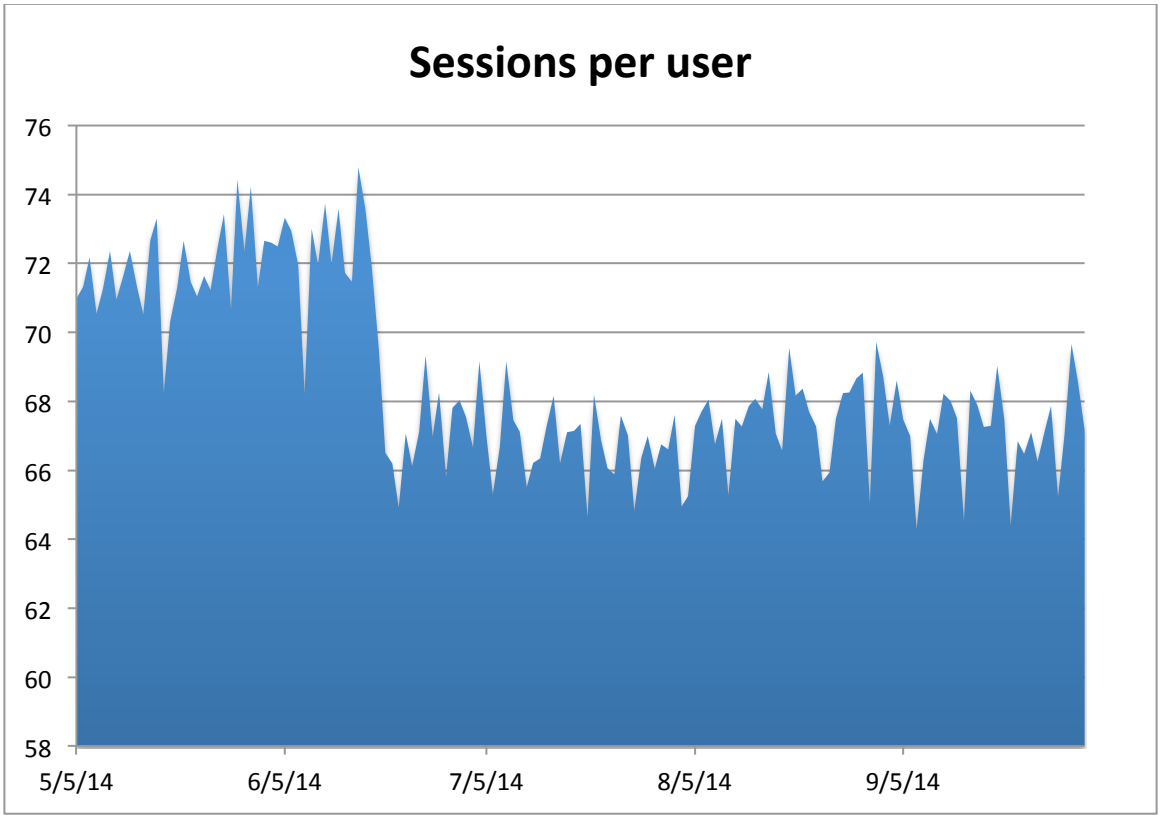

Figure 24: Application 2 - Sessions per user

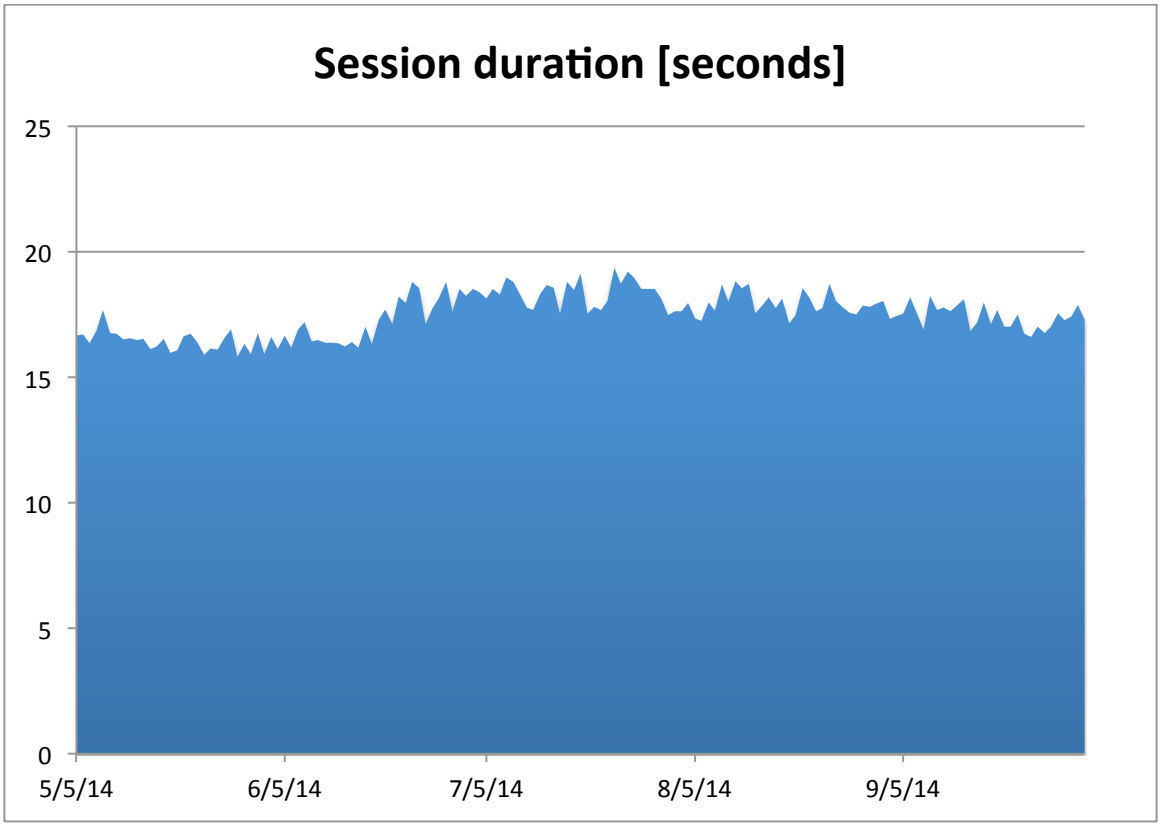

Figure 25: Application 2 - Session duration, seconds

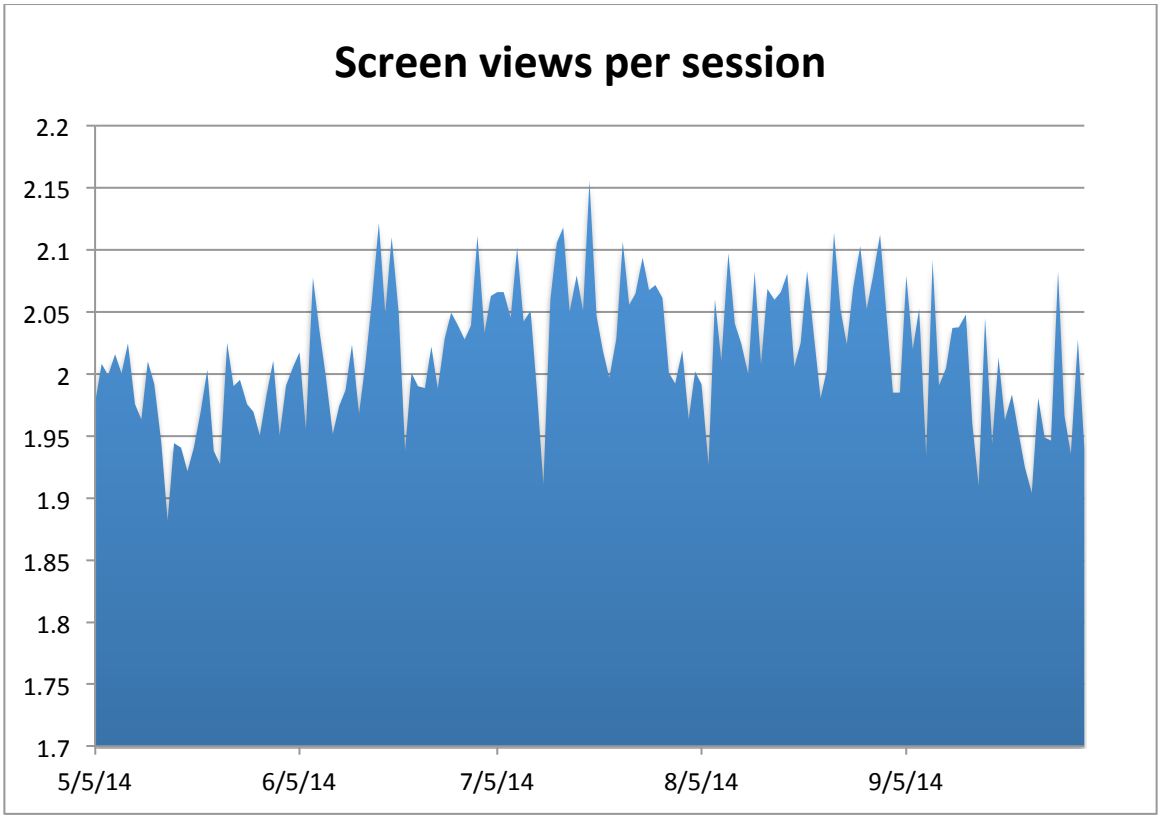

Figure 26: Application 2 - Screen views per session

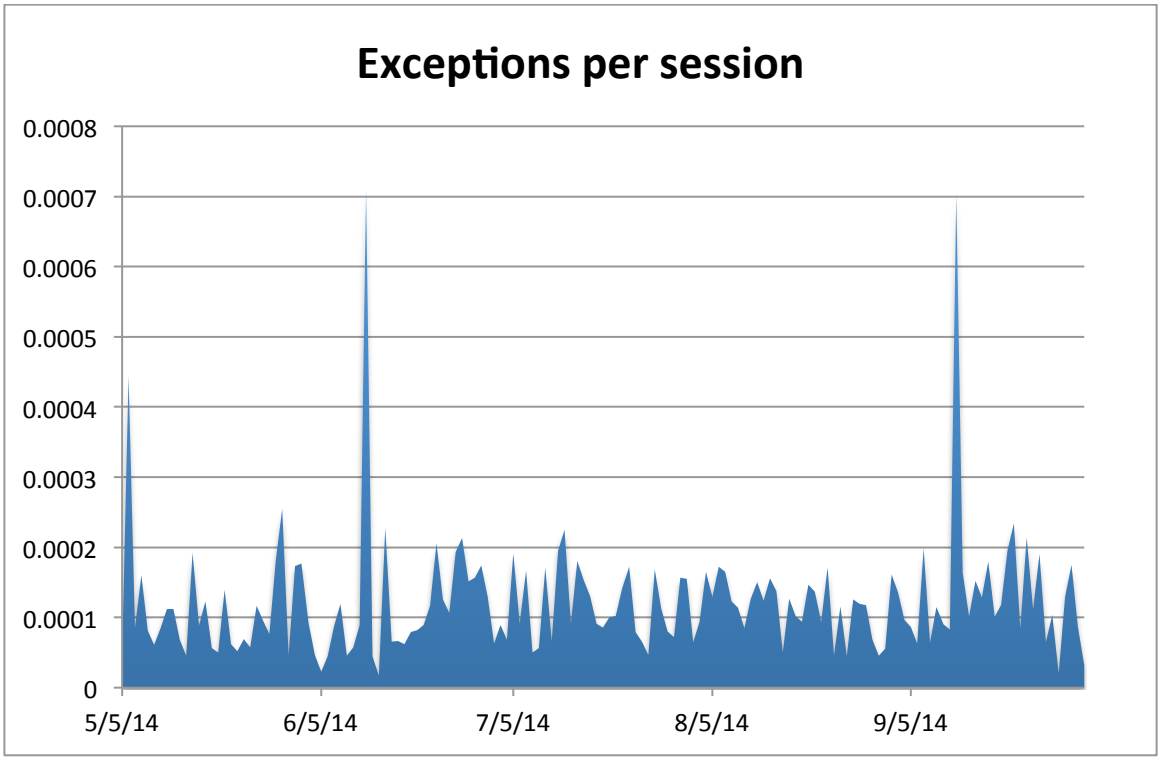

Figure 27: Application 2 - Exceptions per session

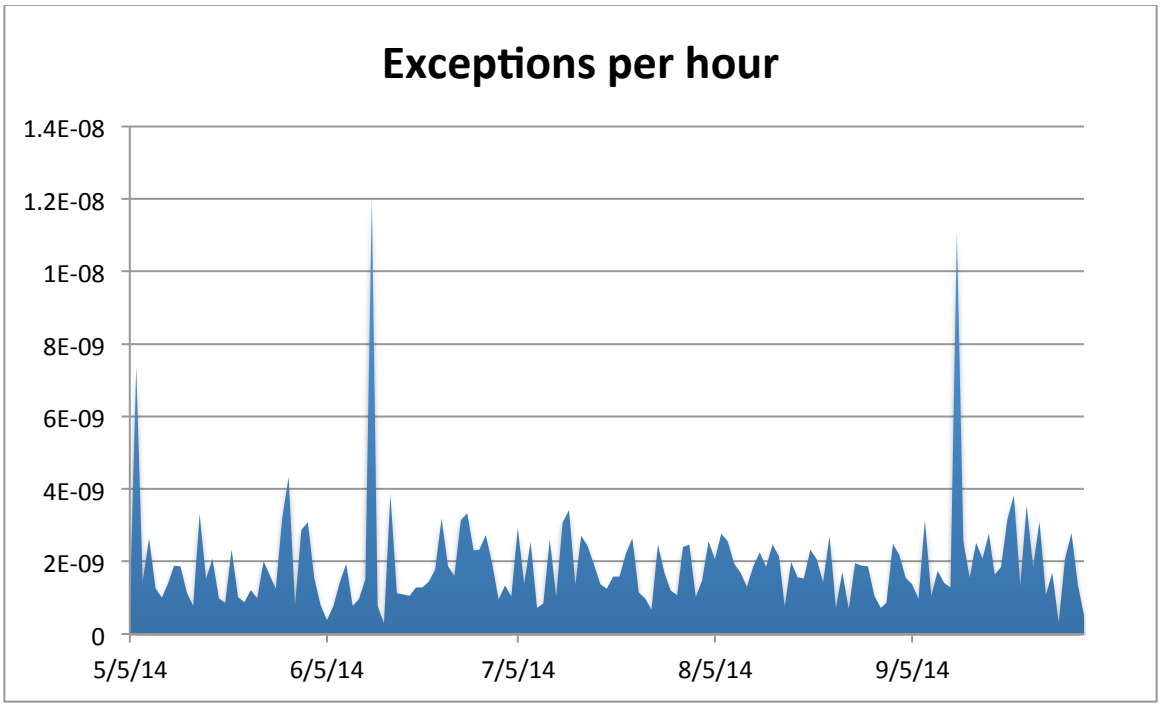

Figure 28: Application 2 - Exceptions per hour

Week start - Monday	Week end - Sunday	Users	Percent of returning users	Sessions per user	Session duration [s]	Screen views per session	Exceptions per session	Exceptions per hour
5-May-14	11-May-14	3859	69.58	319.58	16.82	2.00	0.000137846	2.27644E-09
12-May-14	18-May-14	3812	71.07	326.21	16.42	1.95	0.00010615	1.7959E-09
19-May-14	25-May-14	3766	72.54	327.89	16.27	1.96	6.9645E-05	1.18939E-09
26-May-14	1-Jun-14	3673	73.13	335.23	16.34	1.98	0.000134818	2.2924E-09
2-Jun-14	8-Jun-14	3695	72.15	327.86	16.51	2.00	8.41976E-05	1.41677E-09
9-Jun-14	15-Jun-14	3711	71.36	332.31	16.38	1.99	0.000168668	2.86036E-09
16-Jun-14	22-Jun-14	3590	72.62	327.12	17.09	2.05	7.91925E-05	1.28725E-09
23-Jun-14	29-Jun-14	3615	69.18	305.66	18.16	2.01	0.000164709	2.51934E-09
30-Jun-14	6-Jul-14	3569	69.91	311.35	18.28	2.06	0.00011519	1.75001E-09
7-Jul-14	13-Jul-14	3529	70.36	310.22	18.32	2.03	0.000132449	2.00878E-09
14-Jul-14	20-Jul-14	3414	71.41	312.64	18.39	2.09	0.000118987	1.79717E-09
21-Jul-14	27-Jul-14	3439	71.82	309.56	18.53	2.05	0.000110843	1.6618E-09
28-Jul-14	3-Aug-14	3348	73.18	318.24	18.06	2.02	0.000105119	1.61683E-09
4-Aug-14	10-Aug-14	3486	70.45	311.28	17.84	2.02	0.000136392	2.12341E-09
11-Aug-14	17-Aug-14	3481	70.93	320.80	18.21	2.04	0.000124472	1.89895E-09
18-Aug-14	24-Aug-14	3481	70.87	317.44	17.84	2.03	0.000113122	1.76095E-09
25-Aug-14	31-Aug-14	3557	70.85	311.22	17.89	2.07	9.1236E-05	1.4163E-09
1-Sep-14	7-Sep-14	3601	69.48	311.60	17.72	2.04	0.000113182	1.77419E-09
8-Sep-14	14-Sep-14	3523	70.88	309.99	17.74	2.02	0.000189543	2.96786E-09
15-Sep-14	21-Sep-14	3545	70.75	309.61	17.26	1.97	0.000157619	2.53632E-09
22-Sep-14	28-Sep-14	3428	72.02	313.23	17.02	1.96	0.000112688	1.83879E-09
29-Sep-14	5-Oct-14	3464	72.49	317.30	17.64	2.00	9.91708E-05	1.56165E-09

Table 13: Application 2 - Usage metrics, weekly values