Available approaches to heuristic evaluation of smart-phone applications

University of Oulu
Faculty of Information Technology and Electrical Engineering / Information Processing Sciences
Bachelor's Thesis
Tomi Lämsä
Date

Abstract

Usability heuristics are a lightweight tool for finding usability problems in a piece of software. These heuristics can be used for any kind of software, from desktop applications to web sites to games and to smart-phone applications. With the rise of smart phones and applications developed for them there has been a need to update the approach of heuristic evaluation. This is since the evaluation methodologies should consider the limitations and new possibilities brought by smart-phone applications.

In this bachelor's thesis different these issues are discussed, and along this several lists of usability heuristics were identified and presented. In addition to listing out the current research on smart-phone application heuristics, these lists of heuristics were compared against the issues that are specific to smart-phone applications. Research questions in this thesis are as follows:

RQ1: What guidelines / usability heuristics can be found that specifically target mobile device user interfaces?

RQ2: How do these heuristic lists consider the difficulties faced by smart-phone applications?

Keywords
Usability, heuristic evaluation, smart-phone applications, apps, Android, iOS

Supervisor
PhD, University lecturer Mikko Rajanen

Contents

1. Introduction

The subject of this study is to expand the research on mobile application usability heuristics. The purpose is to find lists of heuristics that can be directly applied during smart-phone application development to improve the usability of those applications.

Usability in general means how easily an application can be used by a user to achieve his or her goals. It can also include aspects related to the attractiveness of the application. Usability is interesting since as creators and implementers of applications we usually wish for our creations to be used, and without good usability this is less likely to happen. This is further discussed in chapter 2.

Heuristic evaluation is a light-weight tool that can be used to improve the usability and accessibility of applications, so it was chosen as the focus of this study. It seems that traditional heuristic evaluation is problematic when applied to applications developed for mobile devices (Po, Howard, Vetere, & Skov, 2004). The authors continue that since heuristic evaluation, which does not consider the context of use, can be problematic on mobile applications, for which the context of use is important.

Other sources also agree with this sentiment, mentioning that the Nielsen's traditional usability heuristics were created for desktop and web environments, not for devices with smaller screen sizes that are often used in varying contexts (Joyce & Lilley, 2014). Joyce & Lilley (2014) continue that usability heuristics developed for smart phone applications should be more focused towards the world of mobile applications, and that the heuristics should not be named similarly as Nielsen's heuristics to help usability experts familiar with Nielsen's heuristics to settle into the different do different domain. It seems that these claims might have some truth, since in an article comparing a set of heuristics developed for mobile applications scored higher in finding usability issues and in confidence of use in professional environment (Joyce, Lilley, Barker, & Jefferies, 2016). It should be mentioned however that in this study another set of heuristics developed for mobile devices scored lower than the traditional Nielsen's heuristics, so it seems that Nielsen's heuristics still can be useful.

Smart-phone applications are pieces of software that are installed on a user's phone. These applications can be installed either through a marketplace or by downloading the installable package on one's device and installing it that way. For both Android and iOS there is a vibrant 'official' market place that has many applications available, developed either by companies, individuals or organizations. The devices also come with pre-installed applications placed there by the device manufacturer, such as Samsung for their Android phones and by Apple for the iPhones. These smart-phone applications are often used while on the move, for example while sitting in a bus or walking outside, that influences what issues should be looked at when considering the usability of these applications.

There are several ways to improve the usability of an application, and heuristic evaluation seems to be one of the easier ways to do this. Considering that not all smart-phone applications are developed by organizations with a lot of resources, it would make sense there to be available good lists of heuristics for the use of smart-phone application developers.

There are many different usability heuristics lists available through research, and the goal of this study is to 1) map out ones that are specific to smart-phone applications and 2) how these heuristic lists consider certain difficulties encountered by applications running in smart-phones, which are discussed in chapter 2.3. Main contribution of this research is a list of different pieces of research presenting these heuristic lists, and a checklist that shows which of these heuristics address which of the difficulties, and if some heuristic lists have gaps in the coverage of those difficulties.

In chapter 2 we look at what is heuristic evaluation, why it is focused on in this research and what research states of smart-phone applications and their heuristics. Further, in chapter 3 the different lists of heuristics identified in research are presented, chapter 4 discusses the differences between these heuristic lists and how they address the mobile-application specific challenges, and finally chapter 5 presents conclusions.

2. Prior Research and background

Usability for mobile devices has been of interest for the research community for a while already. Bertini, Gabrielli & Kimani (2006) researched the usability aspects of mobile computing devices, though their research was made before the emergence of smart phones, as the first iPhone was released in 2007. Billi et al. (2010) discuss about the challenges of usability and accessibility of mobile computing devices and the new challenges and possibilities such devices offer. Kuparinen, Silvennoinen & Isomäki (2013) discuss about usability heuristics for mobile map applications (MMA). They state that the traditional Nielsen's list of heuristics does not cover some aspects specific only to map applications, such as "location awareness, mobility and interruptions". There seems to be a need in the world for heuristic lists meant for different domains.

There are many ways to define usability, for example these three different sources:

1) "The capability of the software product to be understood, learned, used, and attractive to the user, when used under specified conditions." (International Organization for Standardization [ISO], 2000)

2) "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use." (International Organization for Standardization [ISO], 1998)

3) "The ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component." (IEEE, 1990)

The first definition seems to be the one that is closest to what is wished for a smart-phone application. The second and third one omits attractiveness of the application altogether, while the second one also limits the scope to 'specified context of use' and the third one seems to take the approach that the application is a system that is to be operated rather than a more casually used application as smart-phone applications at a times can be. It was also discussed by Joyce et al. (2016) that first-time impressions of an application are important, and as humans depend on visual senses, attractiveness plays a part in that.

There are different ways to evaluate usability of a system, and three of the most common are cognitive walkthrough, heuristic evaluation and thinking-aloud study. These three techniques are used to identify usability problems in computer systems – problems that cause users to slow down, cause troubles or impede on the working practices of a user. Cognitive walkthrough has the usability evaluators describing a typical user, defining a task for this user and constructing a "correct path" to accomplish this task. After this planning phase, the evaluator will execute the evaluation by trying to accomplish the task as if he or she was the user described, seeing in each step if the user would be able to advance and if he or she would get the proper feedback from the system. In think-aloud technique the researcher finds a potential end-user for the system, defines tasks for these users to accomplish and familiarize themselves with the system and the environment. After this the evaluators will perform the evaluation, possibly with a third person helping to facilitate the situation. During the evaluation, the user is suggested to think aloud what he or she is thinking while attempting to do the task, and the evaluator gathers usability problems and highlights on the system based on this feedback. In heuristic evaluation, the usability expert or experts evaluate a mock-up of a system based on a list of recognized usability principles, and the evaluation is simple enough that any computer expert should be able to perform it. (Hertzum & Jacobsen, 2001)

Hertzum & Jacobsen (2001) mention that heuristic evaluation's informality is considered a necessity in getting computer experts to perform it. This informality would seem to fit well to the fast-moving world of mobile application development where large-scale laboratory or field testing with users can take up much resources and can be easier for people such as developers to accomplish with little previous knowledge of usability issues. Lima Salgado & Freire (2014) state that usability of many mobile applications is still a challenge.

Considering what Hertzum & Jacobsen (2001) and Lima Salgado & Freire (2014) stated, heuristic evaluation of smart-phone applications is looked at in this research. This practice is further discussed in chapter 2.1, while the chapter 2.2 focuses on heuristic evaluation of smart-phone applications and chapter 2.3 discusses of smart-phones and the applications developed for them.

2.1 Heuristic evaluation

Heuristic evaluation means that a person looks at a user interface (UI) and tries to see what positive and negative things can be found in it (Nielsen & Molich, 1990). The authors continue that some rules can be used when looking at the interface to give guidance on what is and is not desirable. They found out that by using this method, evaluators consisting of computer science students and "industrial computer professionals" could identify from 51% to 20% of known usability problems in four different systems. The list of guidelines used in this research was published by the same authors in another piece of research (Molich & Nielsen, 1990). The authors developed this list based on their own experiences, and according to them most of usability problems should fit into these categories. This list of guidelines is also presented in **Table *1***. Choosing a correct heuristic for the evaluation is critical to the results of the evaluation (Rusu, Rusu, Quiñones, Roncagliolo, & Rusu, 2018).

Table 1 Usability Heuristics by Molich & Nielsen (Molich & Nielsen, 1990)

Heuristic name	**Description**
Simple and Natural Dialogue	Dialogues should not contain irrelevant or rarely needed information. Every extraneous unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility. All information should appear in a natural and logical order.
Speak the User's Language	The dialogue should be expressed clearly in words, phrases, and concepts familiar to the user rather than in system-oriented terms.
Minimize the User's Memory Load	The user's short-term memory is limited. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate. Complicated instructions should be simplified.
Be Consistent	Users should not have to wonder whether different words, situations, or actions mean the same thing. A particular system action-when appropriate-should always be achievable by one particular user action. Consistency also means coordination between subsystems and between major independent systems with common user populations
Provide Feedback	The system should always keep the user informed about what is going on by providing him or her with appropriate feedback within reasonable time.
Provide Clearly Marked Exits	A system should never capture users in situations that have no visible escape. Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue.
Provide Shortcuts	The features that make a system easy to learn-such as verbous dialogues and few entry fields on each display-are often cumbersome to the experienced user. Clever shortcuts-unseen by the novice user-may often be included in a system such that the system caters to both inexperienced and experienced users.
Provide Good Error Messages	Good error messages are defensive, precise, and constructive. Defensive error messages blame the problem on system deficiencies and never criticize the user. Precise error messages provide the user with exact information about the cause of the problem. Constructive error messages provide meaningful suggestions to the user about what to do next.
Error Prevention	Even better than good error messages is a careful design that prevents a problem from occurring in the first place.

Table 1 above shows the classic list of usability heuristics by Molich & Nielsen. The heuristics are quite general but have been used by many evaluators over the years.

It seems that usability heuristics have their role. While evaluating a mobile guide, it was found that usability heuristics can identify issues related to general usefulness of such a guide, which other usability approaches (rapid reflection, field evaluation and laboratory evaluation) were not able to identify. Using multiple methodologies at evaluating can be useful, however fast evaluation methods should be utilized in early phases of product development when the product is still in quicker iteration cycles. (Kjeldskov et al., 2005)

General usability heuristics developed for a desktop computer environment might not reveal usability problems specific to mobile devices (Yanez Gomez, Cascado Caballero, & Sevillano, 2014). This is confirmed by Rusu et al. (2018), as they studied students doing heuristic evaluation on a single application with different sets of heuristics. They report that the participants of the study felt that application domain-specific heuristics or heuristics aimed at mobile applications better cover the usability aspects of the evaluated application as opposed to using more general Nielsen's heuristics. Othman, Sulaiman & Aman (2018) also discuss that specific lists of heuristics are more useful than more generalized ones, and they specifically mention that in their study, it was more difficult to properly categorize the issues when using a more generalized list of heuristics rather than a specific one.

Even if there are apparent upsides to utilizing usability heuristics and there is a lot of research to be found of heuristic evaluation, it is not entirely certain how much it is used in the industry. A recent study discussed that heuristic evaluation is underutilized in gaming industry in northern Europe and north America (Rajanen & Rajanen, 2018). This disparity is of interest, since it would make sense that industry would be the party interested in utilizing usability heuristics as it should make their products better overall.

2.2 Heuristic evaluation of smart-phones and their applications

The research of mobile usability heuristics seems to have been focused on two aspects of the devices, the usability heuristics of the mobile devices themselves and heuristics for the applications developed for those devices. The heuristics for the mobile devices themselves include heuristics that take into consideration the physical aspects of the devices, such as ease of carrying / holding the device and robustness of the device itself, as mentioned in heuristics by Billi et al. (2010). Heuristics made for the applications (or apps) that are developed for different operating systems that run on smart phones (such as Android or iOS) are only concerned about the application itself, not the operating system surrounding it or the physical device the user is holding in his hand. Example of heuristics developed for applications are the list of heuristics by Joyce & Lilley (2014), which is further discussed in chapter 3.5.

There have been some papers related to this topic in addition to the ones above, Heo, Ham, Park, Song, & Yoon, (2009) wrote about the heuristic evaluation of mobile phones where they presented three different aspects to look at when considering usability of a mobile phone: logical user interface, physical user interface and graphical user interface. Yanez Gomez, Caballero & Sevillano (2014) made a list of heuristics that considers user-interface specific aspects of the mobile device, such as menus, navigation issues and system feedback.

2.3 Mobile devices and their applications

Description of a handheld device from early 2000's defines them as such: 1) the device must operate without cables (with exception of recharging or synchronization with a desktop computer), 2) it must be easily usable with hands only when no surface for it is available 3) one can add applications or use have access to the Internet with it (Weiss, 2003). These criteria seem to still be applicable to smart phones today.

Billi et al., (2010) mention that mobile devices have some limitations when compared to desktop environments: smaller screen sizes, limited ways of input, bandwidth and connectivity issues, limited battery life and computing resources and heterogeneity of devices. They also mention that the limitations require new ways of usability and accessibility evaluation.

Also, according another source, several constraints must be addressed when developing for mobile devices. First is the limited input / output capability of these devices, including small screen sizes and limited network connectivity. Second is the varying contexts of use, usage might range from few seconds to minutes and there might be many different distractions from the surrounding environment. Third, the tasks differ from those usually done with desktop devices and the authors classify following kinds of tasks: searching, browsing, communication, money transactions (such as buying a product), playing games and finally just killing time. Fourth, web-site usage must be planned for mobile-usage in mind, the sites should include similar elements and structuring of information for all kinds of use cases (be it mobile or desktop computer). Fifth, limited resources such as processing capability and battery usage need to be considered. Sixth, there is a wide variety of different kinds of devices and different user groups have different wishes (such as perceived privacy, acceptance of technology and capacity of personalization) a designer needs to be aware that she needs to address many ways to interact with the application being developed. Because of the differences between mobile devices, this is made more pronounced than it might be in other environments. (Yanez Gomez et al., 2014)

Finally, applications developed for smart phones can be divided to two categories: native and web applications. Yanez Gomez et al., (2014) mention that native application's source code is compiled, and it's run on the device 'natively', while a web application is usually built with JavaScript, HTML and CSS, just like web sites are developed, and is interpreted in some special component inside the UI ("Mobile Application Development: Web vs. Native"). It can however be argued that this distinction has no bearing on the usability aspects of the application, as for example Yanez Gomez et al., (2014) do not differentiate between native and web applications in their study, and I would argue that for an user this distinction has no meaning as long as the application is easy to use.

3. Usability heuristics for mobile applications

Research on mobile application usability heuristics is not very broad, as Inostroza, Rusu, Roncagliolo, & Rusu, (2013) state. They report that they could not find any usability heuristics aimed directly at touch-screen based mobile devices. The authors also mention that when using a narrowly-focused list of heuristics, such as mobile application heuristic list would be, one can find more usability issues. Salazar, Lacerda, von Wangenheim & Barbalho (2012) also mention that because of differences between mobile and desktop environments, expertise gained in the other one might not be applicable in the other (as cited in de Lima Salgado & Freire, 2014). It would then seem that using heuristics aimed at smart-phone applications would be very beneficial, but still in their study de Lima Salgado & Freire (2014) found that almost half of the studies they looked at used heuristics that are Nielsen & Molich's list or very near to that.

Usability heuristic lists aimed at smart-phone applications might then be useful for the parties developing said apps. In the following subchapters I have gathered some relevant lists that could be found in research. The found lists are in chronological order by the paper publication year.

The papers containing the lists of heuristics were searched from Google Scholar, with search words such as "mobile application usability heuristics" and "smart phone usability heuristic". The list presented in this chapter is by no means completely exhaustive, and there are likely other lists of mobile application heuristics that were not discovered when the search was done. The lists included were selected by reading the publication telling of the list and seeing if the authors stated that the list was meant for either smart-phone applications or mobile applications. The subsequent subchapters discuss the lists and the publications that described the lists. The subchapters are ordered by the year the publication was made.

3.1 Heuristics by Bertini, Gabrielli & Kimani

Bertini, Gabrielli & Kimani (2006) discussed about the peculiarities of mobile computing devices. They suggested that only focusing on task performance and efficiency is not suitable for such devices, in contrast to it being well applicable to desktop computing, where the tasks are usually well established and predictable. Based on these observations, they developed their own set of heuristics.

The list of mobile usability heuristics developed by Bertini et al. (2006) and is visible in **Table *2***. Their list of heuristics was developed before the emergence of smart-phones in a time when it was not as common to install third-party applications on one's device.

They developed their list of heuristics by first analysing the usability issues related to mobile computing, after this the authors individually looked at the traditional Nielsen's usability heuristics and considered which heuristics were not relevant for mobile usability and which needed to be modified. After this the authors individually compared their list of selected heuristics to the work done by the other two authors, and finally the authors consolidated a final list of heuristics for which they then got feedback from HCI experts and revised to make the final list.

Table 2 Usability heuristics by Bertini, Gabrielli & Kimani (2006)

Heuristic name	Definition
H1: Visibility of system status and losability/findability of the mobile device	Through the mobile device, the system should always keep users informed about what is going on. Moreover, the system should prioritize messages regarding critical and contextual information such as battery status, network status, environmental conditions, etc. Since mobile devices often get lost, adequate measures such as encryption of the data should be taken to minimize loss. If the device is misplaced, the device, system or application should make it easy to find it back.
H2: Match between system and the real world	Enable the mobile user to interpret correctly the information provided, by making it appear in a natural and logical order; whenever possible, the system should have the capability to sense its environment and adapt the presentation of information accordingly.
H3: Consistency and mapping	The user's conceptual model of the possible function/interaction with the mobile device or system should be consistent with the context. It is especially crucial that there be a consistent mapping between user actions/interactions (on the device buttons and controls) and the corresponding real tasks (e.g. navigation in the real world).
H4: Good ergonomics and minimalist design	Mobile devices should be easy and comfortable to hold/ carry along as well as robust to damage (from environmental agents). Also, since screen real estate is a scarce resource, use it with parsimony. Dialogues should not contain information which is irrelevant or rarely needed
H5: Ease of input, screen readability and glancability	Mobile systems should provide easy ways to input data, possibly reducing or avoiding the need for the user to use both hands. Screen content should be easy to read and navigate through notwithstanding different light conditions. Ideally, the mobile user should be able to quickly get the crucial information from the system by glancing at it.
H6: Flexibility, efficiency of use and personalization	Allow mobile users to tailor/personalize frequent actions, as well as to dynamically configure the system according to contextual needs. Whenever possible, the system should support and suggest system-based customization if such would be crucial or beneficial.
H7: Aesthetic, privacy and social conventions	Take aesthetic and emotional aspects of the mobile device and system use into account. Make sure that user's data are kept private and safe. Mobile interaction with the system should be comfortable and respectful of social conventions.
H8: Realistic error management	Shield mobile users from errors. When an error occurs, help users to recognize, to diagnose, if possible to recover from the error. Mobile computing error messages should be plain and precise. Constructively suggest a solution (which could also include hints, appropriate FAQs, etc). If there is no solution to the error or if the error would have negligible effect, enable the user to gracefully cope with the error.

As can be seen from the table 2 above, many of the issues discussed in there are for the mobile-specific use case, but there are also more generic ones that can also be mapped to original list by Nielsen & Molich. There are some heuristics that, while being named differently, are discussing of the same issues. For example, “Good ergonomics and minimalist design” from Bertini, Gabrielli & Kimani maps to “Simple and Natural Dialogue” by Nielsen & Molich, “Visibility of system status and losability/findability of the mobile device” to “Provide Feedback” and “Realistic error management” to “Provide Good Error Messages” and “Error Prevention”. However, the rest of the issues in the list seem to not have a direct relation to any particular heuristic by Nielsen & Molich, and some, but not all, of them are related to the heuristics being specifically for mobile use case. The authors do not mention what skill level the user of this list should have, but since they gathered feedback by enlisting usability experts, this list would seem to be more aimed at such evaluators.

3.2 Heuristics by Billi et al.

Billi et al. (2010) proposed a list of heuristics for use of evaluating the accessibility and usability of mobile computing. They state that their list is different from standard heuristics in that it considers mobile usage and context, whereas the standard heuristics assume a static desktop context of use. According to them the size of the device is a major difference between static desktop computers and mobile devices. They continue that on desktop environments it is normal to just dump all information that a user might need on the screen, whereas on a mobile device this is impossible due to the limited screen size. They mention that this is also more work for mobile device designers, since they should consider carefully what to show and what to hide from the user in the task the user is doing. The methodology developed by the authors should also consider contextual information of the user. (Billi et al., 2010)

According to the authors, the methodology presented considers that users interact with mobile devices / applications in much more open-ended way than they do with traditional applications, and that the interaction is less task-oriented. Thus, the methodology is more “activity-centred” rather than being “task-focused”. The heuristics were developed by basing them partly on the Web Content Accessibility Guidelines and by mapping existing desktop computer usability guidelines to existing knowledge of differences of usage between desktop computers and mobile devices / applications. (Billi et al., 2010)

Table 3 Heuristics by Billi et al. (2010)

Heuristic name	Description
H1: Visibility of system status and losability/findability of the mobile device.	By means of the mobile device, the system should always keep users informed about what is going on. Moreover, the system should prioritize messages regarding critical and contextual information, such as battery status, network status, environmental conditions, etc. Since mobile devices often get lost, adequate measures, such as encryption of data, should be taken in order to minimize the loss. If the device is misplaced, the device, system or application should make it easy to recover it.
H2: Match between the system and the real world	This enables the mobile user to interpret correctly the information provided, by presenting it in a natural and logical order. Whenever possible, the system should be capable of sensing its environment and adapting the presentation of information accordingly.
H3: Consistency and mapping.	The user's conceptual model of the possible function/interaction with the mobile device or system should be consistent with the context. It is especially crucial that there is a consistent mapping between user actions/interactions (on the device buttons and controls) and the corresponding real tasks (e.g., navigation in the real world).
H4: Good ergonomics and minimalist design.	Mobile devices should be easy and comfortable to hold/carry, as well as robust enough to withstand damage (from environmental agents). In addition, since screen real estate is a scarce resource, it needs to be used parsimoniously. Dialogues should not contain information that is irrelevant or rarely needed.
H5: Ease of input, screen readability and glanceability	Mobile systems should provide easy ways to input data, possibly reducing or avoiding the need for the user to use both hands. Screen content should be easy to read and navigate through, notwithstanding different lighting conditions. Ideally, the mobile user should be able to obtain crucial information rapidly from the system simply by glancing at it.
H6: Flexibility, efficiency of use, and personalization.	These should allow mobile users to tailor/personalize frequent actions, as well as to configure the system dynamically in accordance with contextual needs. Whenever possible, the system should support and suggest system-based customization when crucial or beneficial.
H7: Esthetic, privacy and social conventions	These take esthetic and emotional aspects of the mobile device and system use into account, and make sure that the user's data are kept private and safe. Mobile interaction with the system should be comfortable as well as respectful of social conventions.
H8: Realistic error management.	These shield mobile users from errors: when an error occurs, the system helps users to recognize, diagnose, and, if possible, recover from the error. Mobile computing error messages should be plain and precise. They should constructively suggest a solution (which may also include hints, appropriate FAQs, etc.). If there is no solution to the error, or if the error would have a negligible effect, the system enables the user to cope gracefully with the error.

Table 3 above shows the list of heuristics. Looking at it, it seems that the authors were concerned also about the usability and accessibility of the whole mobile device, not just the applications written for it, as is apparent from the heuristic "Good ergonomics and minimalist design", which states that the device should be easy to carry. While this could be interesting for some evaluators, it's not likely to be useful for application developers who can't consider the numerous different types of devices where their applications might run on, and naturally are unable to affect the durability of the device. This issue with the heuristics is also further discussed in chapter 4. For shortcomings of the heuristics, the authors mention that binding together usability and accessibility has its' difficulties, as they are not the same thing and could be considered separately. Another issue they mention is that they limited the accessibility testing to only people with problems with sight, not with other disabilities.

The authors state that to use this methodology, the evaluators of the device / application should possess experience in using mobile applications, a good knowledge of the methodology and background in standard heuristic evaluation. (Billi et al., 2010)

3.3 AM+A heuristics

Marcus, Abromowitz & Abulkhair (2013) wrote about usability evaluation of an application, iCalamityGuide, used during times of disaster. They presented in their study a list of usability heuristics called AM+A. The authors do not describe how the list of heuristics was developed, only sources which influenced the list of heuristics. One of the authors reports working for a company called AM+A, so it is likely that this is a list of heuristics developed by them. The authors mention that when they used this list also the evaluator's expertise was depended upon, but they do not mention specifically what kind of background the evaluator should have. (Marcus, Abromowitz, & Abulkhair, 2013)

The papers used as basis for this list include the Nielsen's original list of heuristics, iOS specific usability guidelines and a number of other articles, mainly from the 90's. The authors report several issues that were found by using their list on the iOS application they evaluated. (Marcus et al., 2013)

It is noteworthy that the authors found several issues about the application even though they used a list of heuristics that seems somewhat generic and it could be utilized also for non-mobile applications. The authors do not mention who should use the list of heuristics.

Table 4 AM+A heuristics, Marcus, Abromowitz & Abulkhair (2013)

Heuristic name	**Description**
H1: Aesthetic integrity and minimalist design	Dialogs should not contain irrelevant or rarely needed information. Every extra unit of information in a dialog competes with the relevant units of information and diminishes their relative visibility. Information should be well organized and consistent with principles of visual design. Avoid information overload.
H2: Consistency and standards	Users shouldn't ask if different words/situations/actions mean the same thing. Follow platform conventions
H3: Direct manipulation/See and point	Users should be able to see on the screen what they're doing and should be able to point at what they see. This forms a paradigm of noun (object) then verb (action). When the user performs operations on the object, the impact of those operations on the object is immediately visible

H4: Error prevention	Even better than good error messages: careful design that prevents problems from occurring in the first place
H5: Feedback / Visible system status	The system should always keep users informed about what is going on, through appropriate feedback within reasonable time. Provide confirmations when the outcome of an action is not visibly apparent
H6: Fitt's Law	The time to acquire a target is a function of the distance to and size of the target
H7: Flexibility and efficiency of use	Accelerators—unseen by the novice user—may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions
H8: Help and documentation	Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, be focused on the user's task, list concrete steps to be carried out, and be concise
H9: Help users recognize, diagnose, and recover from errors	Error messages should use plain language, indicate the problem, and constructively suggest a solution
H10: Information legibility and density	Maximize the amount of data to the amount of ink or pixels used. Eliminate any decorations on charts and graphs that do not actually convey information, such as 3-dimensional embellishments. Less is More is the rule in information design as every pixel used that does not contribute to information, dilutes it
H11: Match between system and real world	The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than systemoriented terms. Follow real-world conventions, making information appear in a natural and logical order. Accommodate the ways in which users are accustomed to working
H12: Modelessness	Try to create modeless features that allow people to do whatever they want whenever they want to. Avoid using modes because they typically restrict the operations that users can perform. Modelessness gives users more control over what he or she can do and allow the user to maintain context of the work
H13: Perceived stability	In order to cope with computer-based complexity, people need stable reference points. To give users a conceptual stability, the user interface should provide a clear, finite set of objects and actions
H14: Recognition rather than recall	Make objects, actions, and options visible. Users shouldn't need to remember information from one part of the dialog to another. Instructions system use should be visible or easily retrievable whenever appropriate
H15: User control and freedom	Allow the user, not the computer to initiate and control actions. Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialog. Support undo and redo
H16: Visible interfaces/WYSIWYG	Don't hide features in applications by using abstract commands. People should be able to see what they need when needed. Most users can't and won't build elaborate mental maps and will become lost or tired if expected to do so. Clearly convey key information without making users dig or click to find it.

Table 4 above shows the list of heuristics. When comparing the list to Nielsen's, many items appear although somewhat worded differently or in different order, such as "Error prevention" in here vs the similarly named heuristic in Nielsen's list and "Feedback /

Visible system status" vs the "Provide Feedback" in Nielsen's list. However, there are some additions to the end of the list, such as heuristics "Modelessness" and "Visible interfaces/WYSIWYG". The authors also mention this themselves, listing the original list by Nielsen as one of the background materials, along with other sources. The authors do not give a lot of explanation for their list of heuristics beyond listing them, and they do not discuss of the possible shortcomings of the list.

3.4 Set of heuristics by Yanez Gomez et al.

Yanez Gomez et al., (2014) created a usability heuristics list for mobile devices. They created the list by first figuring out what problems can be faced when using mobile devices, then looking at existing literature for heuristics that have already been developed for non-mobile environments and re-arranging them to high-level goals of good usability, and in finally they developed sub-heuristics under these high-level goals. The researchers derived these sub-heuristics from high-level usability goals, and after this they added mobile-specific sub-heuristics to account for problems specific to these environments. (Yanez Gomez et al., 2014)

Table 5 Usability Heuristics by Yanez Gomez, Caballero & Sevillano (2014)

High-level heuristic name	Sub-heuristic
H1: Visibility of system status	System status feedback, Presentation adaptation, Selection/input of data, Response time, Location information
H2: Match between system and the real world (mental model accuracy)	Output of numeric information, Simplicity, Menus, Navigational structure, Metaphors/mental models
H3: User control	Menus control, Undo/cancelation, Process confirmation, Some level of personalization, Explorable interfaces
H4: Consistency	Orientation, System response consistency, Functional goals consistency, Menus/task consistency, Naming convention consistency, Design consistency
H5: Error prevention	Fat-finger syndrome
H6: Recognition rather than recall	Navigation, Menus, Input/output data, General visual cues, Memory load reduction
H7: Flexibility and efficiency of use	Search, Navigation
H8: Aesthetic and minimalist design	Navigation, Orientation, Menus, Icons, Multimedia content
H9: Help users recognize, diagnose and recover from errors	
H10: Help and documentation	
H11: Skills	
H12: Pleasurable and respectful interaction	Banking, Shopping, Input data
H13: Privacy	

Table 5 above depicts the list of heuristics by Yanez Gomez, Caballero & Sevillano. It seems that this list of heuristics has many similarities with the Nielsen's list, most of the items in Nielsen's list are covered in some form in this list of heuristics. The authors also mention this themselves. Again, the heuristic "Provide Clearly Marked Exits" is missing from this list. This list of heuristics too has some items that aren't clearly mappable to Nielsen's list, for example "Help and documentation", "Skills", "Pleasurable and respectful interaction" and "Privacy" have no clear pair in the Nielsen's list.

However, as this list has been presented a bit differently than the Nielsen's list is, it can be challenging to interpret how the heuristics map to Nielsen's ones. The authors mention that their list of heuristics is decidedly kept platform- and execution-environment agnostic, meaning that it could be useful for any mobile platforms for either native or web applications used on the device (Yanez Gomez et al., 2014). The authors also mention that their list of heuristics was primarily aimed at 'touch phones', which is synonymous with smart-phones used elsewhere in this report.

The authors had two software engineering students who had no previous usability experience use the list, and the authors state that the list helped the students identify and fix several issues in an application they were developing (Yanez Gomez et al., 2014). This list would be usable at least for non-experienced usability evaluators.

3.5 Thirteen heuristics by Joyce & Lilley (SMART)

In their paper Joyce & Lilley stated that research into heuristics for native smart-phone applications was under-represented. They cited two papers related to usability evaluation of smart-phone applications, both with their shortcomings. The two papers are **1)** study by Bertini et al., presented in chapter 3.1 and **2)** a paper by many of the same authors as the list presented in chapter 3.6, called "Usability Heuristics for Touchscreen-based Mobile Devices". This lack of suitable heuristics led to the authors developing their own set of heuristics that specifically targets these kinds of applications. The authors approached this research by developing a set of 13 heuristics using Nielsen's heuristics and the shortcomings of the papers they cited as a point of reference, gathered feedback about the heuristics from HCI experts and updated the heuristics based on this feedback. (Joyce & Lilley, 2014)

Table 6. Usability heuristics by Joyce & Lilley (2014)

Heuristic name	Refined Heuristics	Description
SMART 1	Provide immediate notification of application status	Ensure the mobile application user is informed of the application status immediately and as long as is necessary. Where appropriate do this non-intrusively, such as displaying notifications within the status bar.
SMART 2	Use a theme and consistent terms, as well as	Use a theme for the mobile application to ensure different screens look alike. Also create a style guide from which

	conventions and standards familiar to the user	words, phrases and concepts familiar to the user will be applied consistently throughout the interface, using a natural and logical order. Use platform conventions and standards that users have come to expect in a mobile application such as the same effects when gestures are used.
SMART 3	Prevent errors where possible; Assist users should an error occur	Ensure the mobile application is error-proofed as much as is possible. Should an error occur, let the user know what the error is in a way they will understand, and offer advice in how they might fix the error or otherwise proceed.
SMART 4	Display an overlay pointing out the main features when appropriate or requested	An overlay pointing out the main features and how to interact with the application allows first-time users to get up-and-running quickly, after which they can explore the mobile application at their leisure. This overlay or a form of help system should also be displayed when requested.
SMART 5	Each interface should focus on one task	Being focusing on one task ensures that mobile interfaces are less cluttered and simple to the point of only having the absolute necessary elements onscreen to complete that task. This also allows the interface to be glanceable to users that are interrupted frequently.
SMART 6	Design a visually pleasing interface	Mobile interfaces that are attractive are far more memorable and are therefore used more often. Users are also more forgiving of attractive interfaces.
SMART 7	Intuitive interfaces make for easier user journeys	Mobile interfaces should be easy-to-learn whereby next steps are obvious. This allows users to more easily complete their tasks
SMART 8	Design a clear navigable path to task completion	Users should be able to see right away how they can interact with the application and navigate their way to task completion
SMART 9	Allow configuration options and shortcuts	Depending on the target user, the mobile application might allow configuration options and shortcuts to the most important information and frequent tasks, including the ability to configure according to contextual needs.
SMART 10	Cater for diverse mobile environments	Diverse environments consist of different types of context of use such as poor lighting conditions and high ambient noise are common ailments mobile users have to face every day. While the operating system should allow the user to change the interface brightness and sound settings, developers can assist users even more for example by allowing them to display larger buttons and allowing multimodal input and output options
SMART 11	Facilitate easier input	Mobile devices are difficult to use from a content input perspective. Ensure users can input content more easily and accurately by, for instance displaying keyboard buttons that are as large as possible, as well as allowing multimodal input and by keeping form fields to a minimum.
SMART 12	Use the camera, microphone and sensors when appropriate to lessen the users' workload	Consider the use of the camera, microphone and sensors to lessen the users' workload. For instance, by using GPS so the user knows where they are and how to get there they need to go, or by using OCR and the camera to digitally capture the information the user needs to input, by allowing use of the microphone to input content which would save the

		user from having to type on the small keyboard
SMART 13	Create an aesthetic and identifiable icon	An icon for a mobile application should be aesthetic and identifiable as this is what a user sees when searching the device interface for the application they wish to launch and when scanning through app stores it will be the first item they see before the application title, description and screenshots.

In Table 6 the heuristics devised by Joyce & Lilley are visible. When comparing this list of heuristics to that of Nielsen's, almost all things pointed out in Nielsen's heuristics are covered, with the exception that Nielsen's heuristic "Provide Clearly Marked Exits" doesn't seem to have a direct pair in this list. However, it seems that rules SMART 6, 7, 8, 10, 11, 12 and 13 do not have a direct pair in Nielsen's list either. It would make sense that for example SMART 12 doesn't have a pairing in there, as Nielsen's heuristics weren't developed for a device that has easily accessible camera for data input. The authors do not mention who should use this list, however they developed it by taking feedback from experts of heuristic evaluation so one could imagine it is meant for professional evaluators.

In another article by some of the same authors, this set of heuristics was compared to two other sets of heuristics, namely traditional Nielsen's heuristics and Bertini's heuristics, which is discussed in chapter 2.3. The authors found out, by interviewing HCI experts that used their heuristics and two other lists of heuristics, that the set of heuristics by Joyce & Lilley scored higher than the two other sets. According to some of the HCI experts they interviewed, traditional Nielsen's heuristics are not focused enough on problems specific to mobile applications. (Joyce et al., 2016)

3.6 SMASH by Inostroza et al.

Inostroza, Rusu, Roncagliolo, Rusu, & Collazos (2016) present a list of usability heuristics they developed for smartphone mobile applications. Their list includes 12 heuristics, and they are presented in Table 7. They developed the list in five iterations: first by conducting a literature review and doing a guided inspection to identify usability issues, ending with 11 heuristics called TMD (usability heuristics for Touchscreen based Mobile Devices). In second iteration, the heuristics were used in practice, and the list was refined to 12 different heuristics based on data from the practical usage and expert opinions. In the third phase, the refined list of heuristics was used again in practice against Nielsen & Molich's heuristics. In fourth phase, the list of 12 heuristics was reviewed by several researchers for issues related to understandability, and were revised based on this feedback. In the final phase, 27 undergraduate students participated in evaluation of a mobile application by using the heuristics, the answers provided by the students were analysed and based on this the list was refined to its' most recent form. 16 of the evaluators had some previous heuristic evaluation experience, while 11 did not. (Inostroza, Rusu, Roncagliolo, Rusu, & Collazos, 2016)

Table 7 SMASH heuristics, Inostroza, Rusu, Roncagliolo, Rusu, & Collazos (2016)

Heuristic name	**Definition**
SMASH1: visibility of system status	The device should keep the user informed about all the processes and state changes through feedback and in a reasonable time.
SMASH2: match between system and the real world	The device should speak the users' language instead of system-oriented concepts and technicalities. The device should follow the real-world conventions and display the information in a logical and natural order.
SMASH3: User control and freedom	The device should allow the user to undo and redo his/her actions, and provide clearly pointed “emergency exits” to leave unwanted states. These options should be available preferably through a physical button or equivalent.
SMASH4: consistency and standards	The device should follow the established conventions, allowing the user to do things in a familiar, standard and consistent way.
SMASH5: error prevention	The device should hide or deactivate unavailable functionalities, warn users about critical actions and provide access to additional information.
SMASH6: minimize the user's memory load	The device should offer visible objects, actions and options in order to prevent users from having to memorize information from one part of the dialog to another.
SMASH7: customization and shortcuts	The device should provide basic and advanced configuration options, allow definition and customization of shortcuts to frequent actions.
SMASH8: efficiency of use and performance	The device should be able to load and display the required information in a reasonable time and minimize the required steps to perform a task. Animations and transitions should be displayed smoothly.
SMASH9: esthetic and minimalist design	The device should avoid displaying unwanted information overloading the screen.
SMASH10: help users recognize, diagnose, and recover from errors	The device should display error messages in a language familiar to the user, indicating the issue in a precise way and suggesting a constructive solution
SMASH11: help and documentation	The device should provide easy-to-find documentation and help, centered on the user's current task and indicating concrete steps to follow
SMASH12: physical interaction and ergonomics	The device should provide physical buttons or the equivalent for main functionalities, located in positions recognizable by the user, which should fit the natural posture (and reach) of the user's dominant hand.

A list of heuristics was used by the authors for evaluating both applications on the devices and the menu structures of the devices. When looking at the list of heuristics presented in Table 7, word ‘device’ is used in every heuristic, while some of the heuristics include things that the actual application running on the device controls. For example, SMASH9 states that unwanted information should not be displayed on the screen, but it is usually the applications running on the device that controls what is shown on the screen. In contrast heuristics SMASH3 and SMASH12 state that the device should provide physical buttons, something that the application running on the device has no control over. It is then unclear whether the list is meant to be used by application developers or the manufacturers of the device. Since the list was evaluated by some people having experience in heuristic evaluation and some who had not, it would seem this list could be utilized by novices and professionals alike.

4. Discussion

The lists of heuristics are all somewhat similar, and it seems they are all based on Nielsen & Molich's traditional list. When looking at the six lists discussed before, not all of them are aimed directly at application developers, which was the main goal when searching for these lists. It seems that some of the heuristic lists include things that are meant for the phone developers, while the same lists include things that are aimed more at application developers. It is unclear why these two things would be mixed up in the same list of heuristics. While it is indeed possible that the original equipment manufacturer (OEM), such as Apple or Samsung, develop applications that are pre-installed on the devices they sell, there are also third-party applications that can be installed on the devices. When Apple for example is developing a phone with a holistic view from hardware to software, it is possible that they could consider both ergonomics of the device itself and the data an application displays on the screen, but an individual application developer who is just making a mobile application will have no control over the device the application is run on, and thus the list of heuristics does not make much sense for them. It would also seem that there would be a lot more application developers than there are OEMs, so it is not clear why the lists would include things that the application developers have no control over. Examples of this issue are the lists by Bertini, Gabrielli & Kimani (2006), Inostroza, Rusu, Roncagliolo, Rusu, & Collazos (2016) and Billi et al. (2010). The issue is understandable for a paper published in 2006 as feature phones were still normal and installing applications from an app market place was not yet common, as the Apple's app store for example launched in 2008 (Ricker, 2008). However, as it is still an issue in the newer papers, it will make it more difficult for an application developer to find a good list of heuristics to use. In Table 8 Heuristic lists and the the apparent use of these heuristic lists has been combined to make it easier for a user of these heuristic lists to find a suitable one.

Table 8 below lists the heuristic lists discovered, along with how the authors of the publication evaluated the list they published and how the authors describe their list of heuristics. Both columns were inferred from the primary publication where the list of heuristics was presented. For example, the list by Joyce & Lilley was further discussed in another publication by the same authors, but this was not considered in this table. For the third column, it was attempted to take a direct quote (denoted by quotes around the text) from the publication, but for all publications a suitable one describing the list was not found and the goal of the list of heuristics was read from the paper.

Table 8 Heuristic lists and their perceived aim

List of heuristics	Evaluation of the heuristics	What is the list for?
Bertini, Gabrielli & Kimani	None	"A set of usability heuristics that is relevant to mobile computing"
Billi et al.	4 users, 2 researchers. Evaluation of automatically generated UIs produced by unrelated application.	"A unified methodology for evaluating mobile services/applications that is based on two fundamental principles: early assessment and the usage of ad-hoc, mobile-oriented, methods."
AM+A	None, the list of heuristics was already in use and was not the focus of the publication	A generic list of heuristics seems not to be smart-phone application specific
Yanez Gomez et al.	2 engineering students on a tablet app under development	Rearranging existing Nielsen's heuristics into new order, adding in mobile application specific heuristics to create a list for evaluating OS (operating system) level issues, along with OS menu structures, system feedback and logic of navigation
Joyce & Lilley	None, another publication evaluated later	"Thirteen heuristics tailored to the inspection of native smartphone mobile applications."
SMASH	Evaluation of Dropbox application on Windows and Android phones by 27 evaluators, half with experience in heuristic evaluation, another half with no experience	A list of heuristics for evaluating either smart-phones or their applications, unclear for which

As can be seen from the table 8 above, evaluation of the developed lists seems a bit lacking, apart from the SMASH heuristics, where the researchers used a lot of evaluators to test out their list. Half of the publications did not evaluate their list at all. It would raise more confidence in the list of heuristics when it has been thoroughly evaluated in the paper that presented it. Not doing this kind of self-evaluation does make some sense in the world of scientific publications, since it could be assumed that further research would evaluate the lists published, as is the case with Joyce & Lilley. Only the list by Joyce & Lilley mentions directly that it is meant for evaluation of smart-phone applications. The other lists discuss of operating system level usability issues, or as in the case of the list by Billi et al. and Bertini et al., discuss on a more general level of mobile computing.

In table below are the challenges presented in chapter 2.3, as discussed by (Yanez Gomez et al., 2014). As a refresher, the challenges are **C1:** limited input / output capability, **C2:** varying contexts of use and varying time of use, **C3:** different tasks when compared to desktop usage, **C4:** web-site use planned for mobile usage, **C5:** limited device resources like battery and processing power and **C6:** wide variety of devices and different user groups. The heuristics in the table Table 9 are identified only by numbers, and those numbers are the numbers of the heuristics presented in subchapters of chapter 3. The heuristic identifiers in parentheses only partially address the challenge, and this explained further below the table.

Table 9 Heuristic lists and smart-phone application issues

	C1	**C2**	**C3**	**C4**	**C5**	**C6**
Bertini, Gabrielli & Kimani	H5	H1		(H4)		H6
Billi et al.	H5	H1		(H4)		H6
AM+A	H1, H3, H10					H2, H7
Yanez Gomez et al.	H1, H7, H5, H12	H1		H7, H8		
Joyce & Lilley	H9, H12	H5, H7, H9, H10	(H5)	H5		H4, H9
Inostroza et al.	H9		(H12)	H9		H7, H12

In the Table 9 above the identifiers of the heuristics addressing the challenge are listed. The list of challenges was compared to the list of heuristics presented by the heuristic authors, and the heuristic discussing of the challenge was selected by the author of this study, who attempted to be as objective as possible when reading comparing the lists. From looking at the table, the heuristics developed by Joyce & Lilley (2014) are the ones that address most of the challenges outlined by Yanez Gomez et al. (2014).

As expected, the lists by Bertini, Gabrielli & Kimani and that of Billi et al. have same results as the heuristic lists are very similar. Both of those lists have the same main heuristics and very similar descriptions. Both consider some of the challenges presented, and for both the challenge 4 is partly covered by the heuristic 4 which states that dialogues should only have relevant information, which can be important on a mobile device with limited screen space.

AM+A has a lot of gaps in the table. The list seems to be more concerned of data displaying side of the application, while not considering data input. It also does not address the limitations of the processing power of the device nor does it discuss how the tasks done in the application should be considered differently than they would be in the desktop use-case. They also do not mention different contexts of use, while they do mention addressing users of different skill levels.

With the list of Yanez Gomez et al., it was more difficult to see if a heuristic addresses a challenge, as there is a lot of information in the article of the all sub heuristics in general, but it is difficult to find from the article a further description on some of them. As the list is very comprehensive it does consider many challenges quite well. List did not seem to address differences between desktop and mobile application tasks and the limited processing capabilities.

For the list by Joyce & Lilley, many of the heuristics discuss about aspects related to different contexts of use and different user groups. Their list was also the only one to mention usage of smart-device specific ways (such as camera or microphone) to input data into the application to overcome that challenge. The list is only one to partially answer the challenge of 'different tasks when compared to desktop environment'. Heuristic 5 in the list mentions that UI of a smart-phone application should be glanceable for users who are interrupted frequently, a challenge that might be more prominent when using a smart-phone than when using a desktop computer since smart-phones are often used in an environment with other people around. However, it can't be stated that this heuristic directly addresses this challenge.

List by Inostroza et al. does not seem to address many of the challenges directly. There is no mention of the limited input of data for mobile devices in the list of heuristics, and it does not really address the varying contexts the applications are used in. Again, as with Joyce & Lilley, the list glances at addressing C3 (different tasks between desktop usage and mobile usage). The list of heuristics mentions that the device itself should fit naturally into user's hand, which could include thinking that the tasks done on the device will be different than those done on a desktop computer with mouse and keyboard, but again it doesn't really directly address the challenge. Overall, this list of heuristics seems to least consider the issues faced by smart-phone applications, perhaps since this list also considers device-specific issues along with the application-specific issues.

It seems that none of the heuristic lists consider the challenge of low resources in a smart-phone application's execution environment. However, especially processing capability and the amount of memory has increased on smart-phones over time significantly, and it is questionable if this is as much of an issue any more, although there are certainly some areas, such as graphics-intensive calculation, where smart-phones are not nearly as powerful as desktop computers might be. Also, limited battery capability will likely be an issue long into future, as devices still need to be recharged regularly.

Another challenge that was not well-addressed by the heuristics is the different tasks when compared to desktop usage. If this challenge is to be addressed by a smart-phone application developer, it should be considered if certain tasks should be doable on a smart-phone at all, and the other way around. This challenge is related to the issue that it is difficult to show a lot of data on a screen of a smart-phone, but one should consider this further. In the end, it can be difficult to codify this challenge into a heuristic that should be addressed, but it would seem that this could be done and is a gap in all of the heuristic lists.

5. Conclusions

In this research I have outlined some of the available usability heuristics for smart-phone applications that can be found in research. There seems to be quite a few of them available, although many of them are like each other, which is not surprising since many of them are based on the original heuristics by Nielsen & Molich. It also seems that there are some gaps in the heuristic lists regarding a few challenges that mobile devices and their applications face, namely dissimilarity of tasks when compared to desktop usage and the limited processing and battery capabilities of mobile devices. However, the table 7 present in chapter 4 can be used as a starting point when considering which list of heuristics to take into use when starting on path of heuristic evaluation of a smart-phone application.

There might be heuristic lists that were not found by the author of this research, especially ones that have been published in the recent years. These newer lists might bring new insight on how modern application development should consider the smart-phone application domain area, and newer publications could be more interesting than heuristic lists that were developed before the emergence of smart-phones.

It would be interesting to see a more thorough investigation of these heuristics accompanied with an actual heuristic evaluation of a certain smart-phone application with the lists presented in here to see how the different lists perform. Also seeing the development path of an application where the developers utilized some of the heuristic lists and comparing the outcome of these paths could be of interest to verify how some of the heuristic lists perform. Finally, expanding the existing list of heuristics with the challenges outlined by Yanez Gomez et al. (2014) could be of use for the usability of future smart-phone applications. Another interesting avenue of research might be to utilize design science research to develop a list of heuristics and to evaluate it in an industry setting. A possible author for such a research might be a student finishing his or her studies who is already working in a company developing smart phone applications. Finally, as it was pointed out that gaming industry underutilizes heuristic evaluation, it would make sense to perform further research in the utilization of the usability heuristics: are they in use in smart-phone application industry and what kinds of heuristics are used.

Heuristic lists should be very usable for mobile application developers, since those heuristic lists are shown to be effective even in the hands of a person who does not have a good usability training. Thus, having an effective list of heuristics that considers the major challenges that smart-phone applications can face is a great boon to the general usability of smart-phone applications.

References

Billi, M., Burzagli, L., Catarci, T., Santucci, G., Bertini, E., Gabbanini, F., & Palchetti, E. (2010). A unified methodology for the evaluation of accessibility and usability of mobile applications. *Universal Access in the Information Society, 9*(4), 337-356.

de Lima Salgado, A., & Freire, A. P. (2014). (2014). Heuristic evaluation of mobile usability: A mapping study. Paper presented at the *International Conference on Human-Computer Interaction,* 178-188.

Heo, J., Ham, D., Park, S., Song, C., & Yoon, W. C. (2009). A framework for evaluating the usability of mobile phones based on multi-level, hierarchical model of usability factors. *Interacting with Computers, 21*(4), 263-275.

Hertzum, M., & Jacobsen, N. E. (2001). The evaluator effect: A chilling fact about usability evaluation methods. *International Journal of Human-Computer Interaction, 13*(4), 421-443.

IEEE Std 610.12-1990. IEEE standard glossary of software engineering terminology. Institute of Electrical and Electronics Engineers, Inc., 1990.

Inostroza, R., Rusu, C., Roncagliolo, S., & Rusu, V. (2013). (2013). Usability heuristics for touchscreen-based mobile devices: Update. Paper presented at the *Proceedings of the 2013 Chilean Conference on Human-Computer Interaction,* 24-29.

Inostroza, R., Rusu, C., Roncagliolo, S., Rusu, V., & Collazos, C. A. (2016). Developing SMASH: A set of SMArtphone's uSability heuristics. *Computer Standards & Interfaces, 43*, 40-52.

International Organization for Standardization [ISO]. (1998). Guidance on usability. ergonomic requirements for office work with visual display terminals (VDT). (ISO 9241-11)

International Organization for Standardization [ISO]. (2000). Information technology - software product quality - Part 1: Quality model. *Usability* (ISO/IEC 9126-1)

Joyce, G., & Lilley, M. (2014). (2014). Towards the development of usability heuristics for native smartphone mobile applications. Paper presented at the *International Conference of Design, User Experience, and Usability,* 465-474.

Joyce, G., Lilley, M., Barker, T., & Jefferies, A. (2016). Mobile application usability: Heuristic evaluation and evaluation of heuristics. *Advances in human factors, software, and systems engineering* (pp. 77-86) Springer.

Kjeldskov, J., Graham, C., Pedell, S., Vetere, F., Howard, S., Balbo, S., & Davies, J. (2005). Evaluating the usability of a mobile guide: The influence of location, participants and resources. *Behaviour & Information Technology, 24*(1), 51-65.

Marcus, A., Abromowitz, S., & Abulkhair, M. F. (2013). (2013). Heuristic evaluation of iCalamityGuide application. Paper presented at the *International Conference of Design, User Experience, and Usability,* 130-139.

Molich, R., & Nielsen, J. (1990). Improving a human-computer dialogue. *Communications of the ACM, 33*(3), 338-348.

Nielsen, J., & Molich, R. (1990). (1990). Heuristic evaluation of user interfaces. Paper presented at the *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 249-256.

Othman, M. K., Sulaiman, M. N. S., & Aman, S. (2018). Heuristic evaluation: Comparing generic and specific usability heuristics for identification of usability problems in a living museum mobile guide app. *Advances in Human-Computer Interaction, 2018*

Po, S., Howard, S., Vetere, F., & Skov, M. B. (2004). (2004). Heuristic evaluation and mobile usability: Bridging the realism gap. Paper presented at the *International Conference on Mobile Human-Computer Interaction,* 49-60.

Rajanen, M., & Rajanen, D. (2018). (2018). Heuristic evaluation in game and gamification development. Paper presented at the *Proceedings of the 2nd GamiFin Conference. Pori, Finland,*

Ricker, T. (2008). Jobs: App store launching with 500 iPhone applications, 25% free. Retrieved from https://www.engadget.com/2008/07/10/jobs-app-store-launching-with-500-iphone-applications-25-free/

Rusu, C., Rusu, V., Quiñones, D., Roncagliolo, S., & Rusu, V. Z. (2018). (2018). Evaluating online travel agencies' usability: What heuristics should we use? Paper presented at the *International Conference on Social Computing and Social Media,* 121-130.

Weiss, S. (2003). *Handheld usability* John Wiley & Sons.

Yanez Gomez, R., Cascado Caballero, D., & Sevillano, J. L. (2014). Heuristic evaluation on mobile interfaces: A new checklist. *TheScientificWorldJournal, 2014*, 434326. doi:10.1155/2014/434326 [doi]