Gerald Musser
(musser@student.tugraz.at)

Mobile Software for Young Kids

Master's Thesis

Graz University of Technology

Institute for Software Technology
Head: Univ.-Prof. Dipl-Ing. Dr.techn. Wolfgang Slany

Supervisor: Dr. Wolfgang Slany

Graz, May 3rd, 2013

Gerald Musser
(musser@student.tugraz.at)

Mobile Software für junge Kinder

(Diese Arbeit ist in englischer Sprache verfasst)

Masterarbeit

Technische Universität Graz

Institut für Softwaretechnologie
Leiter: Univ.-Prof. Dipl-Ing. Dr.techn. Wolfgang Slany

Betreuer: Dr. Wolfgang Slany

Graz, 3. Mai, 2013

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am …………………………… ………………………………………………..
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

…………………………… ………………………………………………..
date (signature)

Abstract

In the last years, the sales of mobile phones has not decreased, however, there was a shift from mobile phones to smart phones. These smart phones have made it very easy for developers to provide applications for users all around the world. One very specific user group that can be addressed are children. Children are a more and more growing user group that is increasingly becoming younger.

If children are provided with software it is not sufficient to assume that children are young adults. It should also not be supposed that the already available knowledge of design and usability concerning adults can be used without adaptions for children as well. Neither should such an assumption be made for the existing usability testing methods or the test setup.

Even if a User Centered Design approach is chosen, a suitable agile development environment should be chosen because the world of children might change quickly and the software should be able to react fast to the new expectations and requirements.

Therefore, this thesis describes already existing methodologies that fit well to a development process where only adults are involved and tries to identify the changes that have to be done to involve children in the development cycle. Adults should not only develop for children but also with children to benefit their requirements on mobile applications.

A special attention is paid to young children at the age of around three to seven years because they, more than other children, have specific needs not only concerning software applications but also when they are directly involved in a usability test or as design partners, therefore insight has been gained whether, on the one hand, Usability Log Files can be an enrichment for usability testing and on the other hand whether this passive logging could make it more comfortable and less exhausting for young children to participate in a usability test.

Keywords: children, usability engineering, software engineering, usability log, mobile computing

Zusammenfassung

In den letzten Jahren sind die Verkaufszahlen von Smartphones immer mehr angestiegen. Smartphones bieten eine einfache Möglichkeit, für Softwareentwickler, einen Markt auf der ganzen Welt zu erreichen. Einen gewissen Marktanteil bilden dabei Kinder, welche als eine sehr spezielle Nutzergruppe angesehen werden können. Die Nutzergruppe der Kinder stieg nicht nur in den letzten Jahren, sondern es ist auch ein Trend zu erkennen, dass immer jüngere Kindern als Smartphone-Besitzer aktiv werden.

Kinder so zu betrachten, als wären sie junge Erwachsene, ist nicht ausreichend, da Kinder andere Bedürfnisse und andere Erwartungen an das Design und die Funktionalität von Software haben. Des Weiteren sollte auch nicht angenommen werden, dass alle vorhandenen, auf Erwachsene ausgerichteten Usability-Laboratorien und Testmethoden ohne Weiteres im Zusammenhang mit Kindern angewendet werden können.

Der "benutzerzentrierte Design" Ansatz in Verbindung mit einem agilen Sofwareentwicklungsprozess scheint eine gute Vorgehensweise zu sein, um für die sich schnell ändernden Wünsche der Kindern angemessene Designvorschläge zu erarbeiten und diese dann auch zeitnah zu realisieren.

Diese Masterarbeit untersucht vorhandene Methoden in der Softwareentwicklung und zeigt betreffende Änderungen auf, die in Bezug auf Kinder nützlich sein können.

Besonderes Augenmerk liegt dabei auf Kindern im Alter von drei bis sieben Jahre, da diese besondere Bedürfnisse an die Software, aber auch an ihre Umwelt haben. Zusätzlich wurde versucht zu eruieren, ob Usability-Log Analysen, speziell für jüngere Kinder, die Teilnahme an Usability-Tests vereinfachen könnten.

Schlüsselwörter: Kinder, Usability Engineering, Software Engineering, Usability Log, mobile Geräte

Contents

1. Introduction

In the last years the use of mobile devices with a more advanced computing capability and connectivity or so-called smart phones has increased rapidly and will not cease over the next years. The Statista[1] website says that in the year 2010, worldwide about 300 million smart phone devices were sold. In the following year 2011, there were more than 490 million sold and in the first three quarters of 2012, it amounted to about 480 million pieces. A forecast that can be found on the same website[2] says that in the year 2014, more than one thousand million smart phones could be sold per year.

This market does not only affect adults. A research study from Statista[3] shows that in the year 2011, about 92% of 12- to 13-year-old children from Germany had access to a mobile phone and used it. For children aged 6-7 years, about 23% used mobile phones. These numbers show that children are interested in mobile phone technology and have access to it.

[1] http://de.statista.com/themen/581/smartphones/, in German, last visit on March 28th, 2013

[2] http://de.statista.com/statistik/daten/studie/12856/umfrage/absatz-von-smartphones-weltweit-seit-2007/, in German, last visit on March 28th, 2013

[3] http://de.statista.com/statistik/daten/studie/1104/umfrage/handynutzung-durch-kinder-und-jugendliche-nach-altersgruppen/, in German, last visit on March 28th, 2013

For usability experts and software developers, the outcome of this study might raise the question, "Do children have different needs and expectations concerning mobile phone applications and if so, how can we please them and develop gadgets in order to meet for their expectations?"

One open-source project that aims developing valuable applications for children is the Catrobat[4] project. This project currently tries to focus on male teenagers but in the future will try to cover all age groups.

There already been some research studies that are related to the Catriod project. One research study from Knapitsch-Scarpatetti-Unterwegen [Kna12] is about "Usability Testing of Mobile Applications for Children". The research paper from Knapitsch-Scarpatetti-Unterwegen includes a section that focused on usability tests, that covered the Catroid and Paintroid[5] projects and gave an overview of what is practical when doing usability tests with children.

This master's thesis shall extend the question about usability for children and tries to take a theoretical look at the questions: Are there other parts in the software development process in which children may be involved and can be useful partners? And if so, what are the differences and adaptions that have to be done compared to the usual process?

Furthermore, this work tries to take a closer look at the group of children aged around 3 to 7. Since a larger part of that age group will probably not be able to read and their attention span is rather limited

[4] `http://catrobat.org`, last visit on April 17th, 2013
[5] `https://github.com/Catrobat`, last visit April 17th, 2013

compared to older children, this group seems to be especially challenging for designers and developers.

Therefore, this work may be seen as a comparison between current available techniques and whether they are suitable when they are used with young children.

In the opinion of the author, it is not the question whether (young) children should use mobile applications and it is a fact that children do explore their environment. The already mentioned numbers show that the mobile phone share will rise in the following years as well as even younger children have more and more access to mobile phones or own smart phones themselves. So, it seems to be only a matter of time before children get in touch with mobile applications. If that happens, then there should be child-friendly applications available to be explored by a child. Children should not become frustrated or even be harmed in any way when using software because they try to interact with a program that was designed for adults only.

2. Children from the Psychological Point of View

This chapter provides a short overview of the evaluation states of children from the age of around two years to the age of twelve years. Chapter 2.3 will mainly deal with children at the age of three to seven years, where the focus of this work primarily lies.

Every child has its own progress speed in development, so the age groups and the corresponding developmental steps should not be seen as fixed. Therefore, the close-by developmental stages are described briefly to provide a better knowledge of the development stages of children.

Knowing the development stage and the corresponding behavior of that stage at a certain age can help to understand why some methods described in Chapter 5 about *Usability Testing With Children* may not work as well as with adults. The stages of the child development described in Table 2.1 are based on the research of Jean Piaget[1] and shall provide a starting base to identify different groups of children and their possibly different needs.

[1]Jean Piaget 9. August 1896, Neuchâtel; † 16. September 1980, Genève, was a Swiss developmental psychologist and epistemologist: [enc12b; enc12c; Tex]

Age	Stage	Sub-Stage	Main Characteristics
birth		Stage 1: training-inherent reflexes	
1 m. 2 m. 3 m.		Sub-stage 2: primary circular reactions (positive actions will more likely be repeated)	
4 m. 5 m. 6 m. 7 m.		Sub-stage 3: secondary circular reactions (discovery that actions will lead to specific results)	
8 m. 9 m. 10 m. 11 m.		Sub-stage 4: usage of known schemata in new situations	
12 m. 13 m. 14 m. 15 m. 16 m. 17 m.	Sensorimotor stage	Sub-stage 5: tertiary circular reactions (active experimentation to discover new principles of action)	Infants develop and coordinate their physical and motoric actions. This stage ends with the beginning of the development of symbolic logic.
18 m. 19 m. 20 m. 21 m. 22 m. 23 m. 24 m.		Sub-stage 6: internalization of schemes (imagination of anticipation of actions)	
3 y. 4 y.	Preoperational stage	Symbolic function sub-stage: thinking in images and symbols	At this stage, the child learns to speak. Concrete logic is developing but not yet fully understandable. The child's own perspective is still dominant over the ability to view things from other people's perspectives. An egocentric view is still dominant and playing next to each other develops slowly into cooperative play.
5 y. 6 y. 7 y.		Intuitive thought sub-stage: beginning the use of the primitive reasoning	
8 y. 9 y. 10 y. 11 y.	Concrete operational stage		The use of logic is used appropriately and the thought process of the child becomes more like that of an adult. Children begin to recognize that their thoughts may be different to those of others.
12 y. 13 y. 14 y. 15 y. 16 y. adulthood	Formal operational stage		Abstract thinking and logic improves. Identity becomes more important and social activities with friends become more interesting.

Table 2.1.: Piaget's stages of child development; m...month(s), y...years [compare enc13; enc12a; Ros99]

2.1. Ethical Responsibility

Below is an excerpt of the ethical standards for research with children from the SRCD [SRC07]. These standards shall provide the basis for the work with children. The complete standards can be found in the Appendix Section A and should be read and obeyed.

NON-HARMFUL PROCEDURES
: That means that the investigator has to make sure that the child is neither harmed physically nor psychologically.

INFORMED CONSENT
: The child should be informed in an understandable way before the research starts about anything that may affect the child. In particular, the child should be informed about anything that could affect the willingness to participate and moreover, all questions should be answered.

PARENTAL CONSENT
: The parent's agreement should be obtained preferably in written form.

DECEPTION
: Sometimes, it might be necessary not to tell the whole truth because it is in the interest of the study. This deception should be clarified later and the possibility to withdraw from the study should be given.

ANONYMITY
: The information and media may only be used where the investigator has the required permission.

UNFORESEEN CONSEQUENCES
: Any unforeseen, undesirable consequences for the participant

have to be put in order by the investigator with appropriate measures.

CONFIDENTIALITY
"The investigator should keep in confidence all information obtained about research participants.[...]" [SRC07]

INFORMING PARTICIPANTS
If any misconceptions arise at any time, they should be clarified as soon as possible.

REPORTING RESULTS
The evaluation results may be misinterpreted by parents or children, therefore, the results should be reported to them with caution.

2.2. Evolution of Children 1-2

Between one and two years of age, children go through different stages: Children of this age are also called "toddlers" [Para. 3.4 enc12a, cf.]. Especially in the second half of the second year, children acquire a skill so that they "anticipate an action with their mental image, practically attempts are not necessary any more" [Ros99, p. 77]. And the "suddenly resulting understanding is reflected in the facial expression of a child" [Ros99, p. 77]. Furthermore, "children at the end of the second year have a knowledge of about 200 words" [Ros99, p. 81].

An assumption from the point of view of a software developer, designer and usability tester is that the limited treasury of words will make it challenging or even impossible to integrate children under the age of two (1/2) into a usual software development process and find valuable usability conclusions. For this reason and because no research paper could be found that covered this topic, children under the age of two will not be considered for this work.

2.3. Evolution of Children 2-7

Children between two and seven are in the so-called *preschool period*, but that does not mean that they do (have to) attend preschool. According to Rossmann [Ros99], a main characteristic of that period is that when children do not sleep or watch television, they are almost always in motion. This is important because this improves their motor development rapidly. Rossmann also mentions that motor abilities rely on these factors to gain deliberate control over the movements of the body parts, to gain an accurate image of their own body and the ability of bilateral coordination, and the ability to coordinate both sides of the body (alternately or simultaneously). These motor skills are mainly demonstrated in running, jumping, or climbing.

Baumgarten [Bau03] concluded that within this age group, the hemispheric specialization of the brain takes place, which is externally noticeable through a hand dominance. The evolution of the brain up to the age of six has an effect on the vision of children as well [Bau03, p. 2]: "Up to this age, many children are slightly farsighted."

The attention span over this period rises and the capability of the memory increases, which leads to an understanding of symbols. The time, of obtaining the knowledge of grammar rules, will bee sooner or later in this phase and is depending on the cultural environment.
Baumgarten [Bau03, p. 2] quotes Piaget, who found some problems of children at that age:

- "inability to see more than one aspect of an object, known as centration" [Bau03]
- "difficulty in understanding another's perspective: egocentrism" [Bau03]
- "ascribing personality to inanimate objects: animism" [Bau03]
- "belief that fantasy is the same as reality" [Bau03]

Below are listed some distinctions that could be figured out more age-specifically:

Age 2: Referring to the encyclopedia [enc12a, Para 3.5.5], kids at the age of two years can answer the questions "What are you doing?", "What is this?", and "Where?" as well as questions dealing with familiar objects and events and their motor development still advances ("Grasps large crayon with fist; scribbles." encyclopedia [enc12a, Para 3.5.2]).

Age 4 According to the encyclopedia [enc12a], children at the age of four are able to paint and draw with a certain purpose. They may have an idea in mind but often have problems implementing it, which is why they create something else and become accurate at hitting nails and pegs with a hammer. Furthermore, their articulation is almost entirely understandable.

Age 5 At the age of five, children are eager to learn new things. "Many children know the alphabet and names of upper- and lowercase letters" [enc12a], but they still cannot read although they are interested in learning it. The cognitive and language development is that far advanced that children are able to tell a familiar story *while* looking at pictures in a book, and the pronunciation is almost entirely grammatically correct [enc12a]. Meggitt [Meg06, pp. 90, 91, 92] notices, that children at that age understand as well as enjoy jokes and riddles. Their fine motor skills are highly developed and children are now able to use pencils and paint brushes quite well. Children may be able to thread a large-eyed needle and sew with large stitches. The play behavior changes from playing alone beside other children to playing with others, including younger children.

Age 6 In the following age period of six to seven, the motor development of children becomes more precise and furthermore, movements become even more deliberate. Children at that age talk a lot and often talk to themselves if they encounter a problem-solving situation. Even though children talk a lot, it might not always make sense to adults and it might be hard to follow and interpret the child's monologue. Children now follow their own logic [compare enc12a].

2.4. Evolution of Children 7-12

From around the age of six, children from all over the world generally have to attend school. By starting this period of life, new roles are assigned to children. This normally results in a raise of responsibility as well. The psychological development is often strongly connected to the attendance of school. For that reason, developmental processes are difficult to describe without the knowledge of the context in which the child grows up.

A significant characteristic of this period is on the one hand that the growth rate of children decreases steadily and on the other hand that the motor development improves at the same time that children become stronger. One significant difference between boys and girls is that the athletic performance of boys is on average better whereas girls have on average a better hand-eye-coordination.

At the cognitive level, children around the age of seven gain the ability to distinguish between alive and not alive together with the ability to estimate time duration. The main cultural achievement is the ability to read [Ros99].

During this age, a first sense of logic, reasoning and simple abstractions emerges. The main influence by parents shifts to a larger influence by friends [MB03].

2.5. Evolution of Children 12 up to Early Adolescence

In this period, children become even more independent from parents and peers as in the previous phase. Nevertheless, the source of information and the source of social standards shifts from their parents to the peers. They develop abstract thinking and their logical skills are improved. Identity and social activities with friends becomes more important. The preoccupation with their own identity results in a more realistic and critical understanding of their own future, and their moral reasoning is growing. Young adolescents have a stronger opinion on their own and want to be involved more in the decision-making process [MB03; Bau03].

2.6. Summary

Research has shown that from birth to adolescence, the development of children advances very fast in different categories. From the software engineering point of view the evolutionary steps in the cognitive sector and the motor abilities might seem to be the most interesting ones. Cognitive abilities include amongst others language skills as well as logic thinking and the ability to abstract for example symbols or recognize and interpret metaphors. All these skills may affect the complexity and the hierarchical depth of software as well as the possible representation of user interfaces. For example, a textual representation is not suitable until children are able to read and in addition, the treasury of words has to be considered to find a proper user interface representation.

Motor abilities have a direct impact on the input devices and the precision of how they are used and, therefore, possibly the stage of the motor abilities of the desired child-user-group might be of special interest for designers to develop a proper usable design that is error-defusing.

In addition, the social situation cannot be left out, otherwise, for example, a software product that is intended to be used cooperatively like adults would do may fail because children at specific stages in their lives have a rather egocentric view and tend to play beside rather than with others.

For these and other reasons, children can be seen as a very specific user group especially within the adult's point of view. And even the scope of a definition like "children" does not seem to be sufficient, because if one considers especially young children aged around two to seven, they have a rapid development process and their social circumstance probably change from staying at home to attending kindergarten and afterwards, they have to attend school. Each of these stations provides different impressions and different challenges for children.

For all the above reasons and the different stages of child evolution provided in the previous chapters, children should not and cannot be treated like adults in a software engineering process. Moreover, their special interests and different abilities have to be explored and should be considered and integrated into the design process.

3. Usability Lifecycle in an Agile Environment

3.1. Overview of Usability Lifecycle

The point of time when programming a system starts is not necessarily the same point of time to start with methods that may contribute to a better handling of the system. Those methods can have their start time earlier. Nielsen [compare Table 7 Nie93, p. 72] describes the usability lifecycle with the following eleven stages:

1. *Know the user*
2. *Competitive analysis*
3. *Setting usability goals*
4. *Parallel design*
5. *Participatory design*
6. *Coordinated design of the total interface*
7. *Apply guidelines and heuristic analysis*
8. *Prototyping*
9. *Empirical testing*
10. *Iterative design*
11. *Collect feedback from field use*

Nielsen [Nie93] provides a good general overview of the stages usability engineers may come into when involved in a usability lifecycle. For this reason, those stages shall provide a basis and will be described in more detail. Jones and Marsden [JM06], Tullis and Albert [TA08], Liebal and Exner [LE11] and Markopoulos [Mar08] provide similar approaches to the important substations in a usability lifecycle even though they may not always have the same title.

Knowing the user has already started in Chapter 2 *Children from the Psychological Point of View* and will further be investigated in Chapter 3.3 *User Centered Design* that shall investigate how to determine user groups and how to put more attention to the user during the development and design lifecycle.

Competitive analysis tries to compare already existing products. Such analyses tries to find the strengths and weaknesses of a product and they shall provide useful input to the interface design which failures should be avoided and which strengths the competing products feature and, therefore, should be taken up and further improved upon.

Usability goals should be set as there is often not enough time to do all the usability research at once and in different stages of the product lifecycle, there might be different needs to investigate diverse usability characteristics. Also, it is possible that certain usability characteristics are not that important and therefore, usability research in those areas could be reduced.

Parallel design may be used to investigate different design approaches simultaneously. The best outcomes of each design can be merged in an even new advanced design. Parallel design is not mandatory in a usability lifecycle and often is resource (time) dependent.

Participatory design is the approach to introduce users as design partners in the design process. The benefit of involving real users is that questions and ideas can arise which programmers or designer never would have thought of (because in the end, designers and programmers are not the end users or at least are biased because they constantly do work on the product).

Coordination of the total interface is important because a consistent design should be anticipated. Consistency in design can contribute to a better learn-ability and memorability of the product because the user can identify reoccurring user-interface elements which do not necessarily have to again be explored concerning their functionality. To achieve a consistent design, prototyping is a possibility to visualize design ideas and may be better than formal documentation, especially when engineers are situated in a very fast evolving agile development environment. From the programming side, code-sharing and the reuse of code can contribute to a better common user-interface.

Guidelines as well as (in-house) standards can help to follow the big picture. Guidelines shall help to improve a common behavior of the system, for example. "Buttons should give the following feedback when clicked: ... "

Prototyping can be used in many ways and different design stages and is therefore described in detail in Chapter 4 *Prototyping*.

Empirical testing may be done with *Usability Tests*. Which test methods are usable for children is described in more detail in Chapter 5 *Usability Testing With Children*.

Iterative design is based on the outcomes of the usability problems. Since a redesign or improvements of an existing user-interface can lead to new, formerly not existing usability issues, it is important to iteratively test and improve the existing design to avoid the possible introduction of new user errors.

Collecting feedback from field use is oriented towards future releases or even new products. For a discussion about using log files for analysis of usability-related issues see Chapter 6 *Usability Log Analysis*.

3.2. Usability and User Experience

There are two major key words for the field of usability engineering that gather information directly through users (not necessarily end users) and tries to contribute this information directly to the engineering process. Those words are *Usability* and *User Experience (UX)*. Subsequently, there are definitions of both terms.

Usability is defined by

- International Organization for Standardization [Int98] as "*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.*"

- the Nielsen Norman Group [Gr012] in short as "*a quality attribute that assesses how easy user interfaces are to use. The word 'usability' also refers to methods for improving ease-of-use during the design process.*"

User experience is defined by

- International Organization for Standardization [Int10] as "*person's perceptions and responses resulting from the use and/or anticipated use of a product, system or service*

Note 1 to entry: User experience includes all the users' emotions, beliefs, preferences, perceptions, physical and psychological responses, behaviours and accomplishments that occur before, during and after use.

Note 2 to entry: User experience is a consequence of brand image, presentation, functionality, system performance, interactive behaviour and assistive capabilities of the interactive system, the user's internal and physical state resulting from prior experiences, attitudes, skills and personality, and the context of use.

Note 3 to entry: Usability, when interpreted from the perspective of the users' personal goals, can include the kind of perceptual and emotional aspects typically associated with user experience. Usability criteria can be used to assess aspects of user experience."

- the Nielsen Norman Group [Gro] as "*'User experience' encompasses all aspects of the end-user's interaction with the company, its services, and its products. The first requirement for an exemplary user experience is to meet the exact needs of the customer, without fuss or bother. Next comes simplicity and elegance that produce products that are a joy to own, a joy to use. True user experience goes far beyond giving customers what they say they want, or providing checklist features. In order to achieve high-quality user experience in a company's offerings there must be a seamless merging of the services of multiple disciplines, including engineering, marketing, graphical and industrial design, and interface design."*

The *Usability* definitions have in common that they want to gather measurable information about a system provided by users and that information shall be used to improve the underlying system to satisfy the end users.

Usability is one element that contributes to the system acceptability as Nielsen [Nie93, p. 24] remarks. Nielsen [Nie93, p. 26] also claimed the following five usability characteristics [compare Gro12]:

Learnability: How quick is the user able to do some work with the

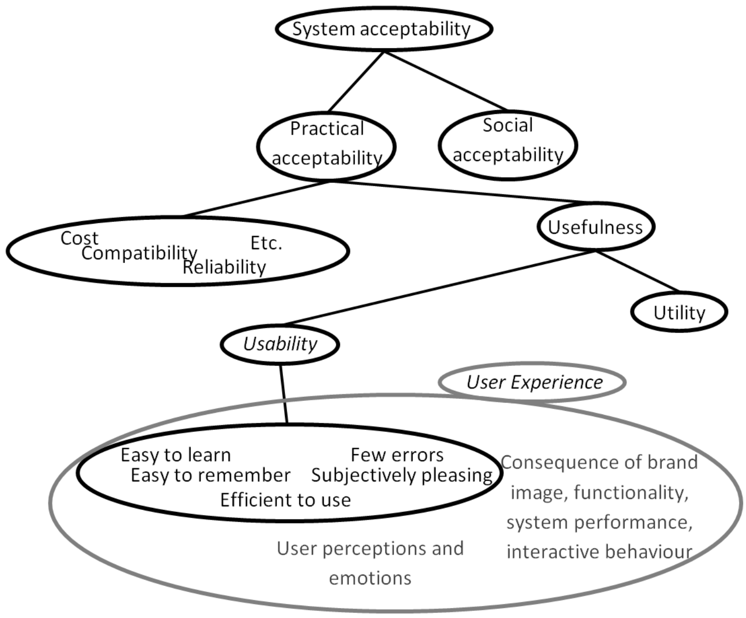


Figure 3.1.: Contributing fields to system acceptability including Usability and User Experience [based on Figure 1 from Nie93, p. 25]

system?

Efficiency: How efficient is the usage when the user already has some knowledge about the system?

Memorability: How easy is it for the user to take up work again after not having used the system for some time?

Errors: How many errors appear during the use of the system and how severe are they (is the user able to recover from an error)?

Satisfaction: How satisfying is the utilization of the system?

Other attributes that contribute to system acceptability according to Nielsen [Nie93, pp. 24, 25] are:

Social acceptability: How acceptable is the use of the system in the user's current ambiance?

Practical acceptability: How well does the system compete in the scope of cost, compatibility, reliability and other "traditional" categories?

Utility: How well does the system support the desired functionality?

Usability: Compared to utility, how well can the user deal with the supported functionality?

The *User Experience (UX)* definitions seem to have an extended perception of the system compared to the *Usability* definitions. User experience tries to include not only the directly affected measurable interactions with a system like severity and occurrences of errors. It integrates an emotional research field that tries to figure out how users react to the whole brand image and the actual computer program. Research in that direction can be especially important when software is developed for children because there might be a useful shift from efficiency to fun in a product. Although the usability definitions include a *pleasant use* factor, it could possibly not cover all the aspects that are important for a successful release of a product in an environment where children are included. For example, if parents take part in the decision-making process which products their children are allowed to use their decisions might be more affected by their user experience related to the brand image and overall perceptions and emotions without even really using the system itself. End users have such permotions (perceptions and emotions) too, and, therefore, it might be worthwhile to investigate those attributes as well as the underlying "hard facts" of the real program [compare TA08; JM06, pp. 4; 55].

Figure 3.1 shall provide an overview of the overall system acceptability in the meaning of Nielsen [Nie93] and shall indicate the *User Experience* part. The User Experience circle from Figure 3.1 includes the Usability area, which shall indicate that measuring UX can be done by adjusting usability testing techniques or even only by adjusting the goals that should be measured by a usability test.

3.3. User (and Child) Centered Design

3.3.1. Child Centered Design (CCD)

The *User-Centered Design (UCD)* tries to see the program from the end user's point of view and attempts to involve users as much as possible in every step of design and development lifecycle. Since it is not possible to have real users always at hand the *Persona* technique (Chapter 3.3.2) will be described to keep the end user in the engineers' minds.

Druin [Dru02] identified four roles so that children can participate in the *Child Centered Design (CCD)* process [compare MB03]:

User: In this role, children actually use a product. The children are observed and adults try to interpret the children's behavior with the product. An advantage of this role is that adults can gain better insight into and understanding of the interaction with products. These insight's can be used to prove the current system concept or to determine which further steps shall be initiated.
The limitation of the user role is that children only take part in the evaluation of already developed concepts and not in the actual design phases.
Methods to obtain information from children and including children as users are described in Chapter 5 *Usability Testing With Children*. Even though those usability test methods are intended

to be used during the design and development process, they can as well be used to test a released product since a release is just one step in the product and usability lifecycle.

Tester: This role directly involves children in the development process. The usability testing methods from Chapter 5 in combination with prototyping as described in Chapter 4 can be used to gain direct feedback during the development process. The feedback obtained by such test sessions has to be evaluated and subsequently may be integrated into the design and development process. To put children in the role of a tester, at least a prototype is needed.

The limitation of the tester role is that a prototype is needed for testing. The children are not involved in the initial technology creation process and, therefore, suggestions will only have an affect on ideas that have already been generated by adults.

Informant: The child informant role mostly takes place before any technology is developed. Children are observed while they interact with other similar technologies or children may be asked for their input to the relating technologies. Children as informants should always be considered when it is unclear in which direction the development should be headed.

Design Partner: Children in the design partner role shall be given the chance to contribute to the program in any way they can and throughout the whole development process. Within this role compared to the informant role, children shall be embedded more strongly in the project in whatever position possible - as real project members would be allowed to contribute in those fields where their expertise lies.

The decision to introduce a role to the project can be very resource-dependent. Particularly when the role of children as informants or design partners is established, those roles can be very time-consuming. Another limitation can emerge when children are not accepted as equal partners and, therefore, their input is not fully valued. Most likely, the relationships between adults and children is of the kind teacher-student as Figure 3.2 indicates and as the second axiom of communication from Watzlawick [Wat] declares: *Every communication has a content and relationship aspect in the way that the latter aspect determines the former aspect*[1], there may well be possible communicational misunderstandings between children and adult developers.

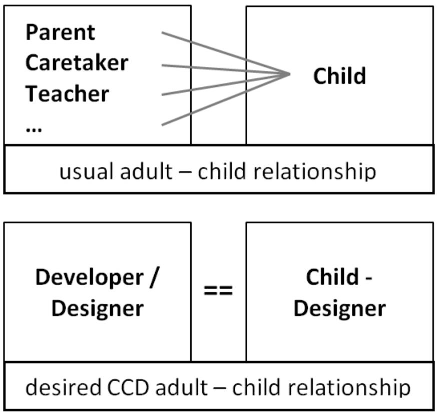


Figure 3.2.: Usual relationship between children and adults and the desired equated relationship in *Child-Centered Design (CCD)* especially in the role of informants and design partners

[1]Original excerpt in German: "*Jede Kommunikation hat einen Inhalts- und einen Beziehungsaspekt, wobei letzterer den ersten bestimmt.*"[Wat]

3.3.2. Persona - a Tool to Know the User

Persona[2] try to generate one or more prototypes of fictive persons who may get in touch with the product in the product life cycle. Each of these persons is a direct user or has some kind of relationship to the product. Such a person could be a parent who not actually uses the product but is concerned about the content their child is interacting with.

If a persona description is read, one should immediately have an image of a real person in mind who has real demands regarding the product. A persona description shall help to keep the user in mind in every development stage and tries to complement other techniques that provide information about the users. The following stages are based on the information provided by Nielsen [Nie13], Campos and Paiva [CP11], Pruitt and Grudin [PG03], Faily and Flechais [FF11] or Seffah, Kolski, and Idoughi [SKI09] and shall describe how to evolve a useful persona that is grounded by a research basis and not based only based on a fictional image.

User Research

The *user research* is mainly about how to collect information about users and how to treat that information [Nie13]. Nielsen [Nie13, p. 26] suggests starting to explore the area that is examined by doing a research about information that already exists. Such already existing data can be found in market researches, research reports or may already be internally available in the company [see also PG03, pp. 4, 5]. This information shall build the basis for choosing participants that

[2] Latin for mask or person

are further interviewed or observed. Moreover this information shall help formulate questions that shall be answered about a specific focus group and to determine why or how that group differs from another group.
Concerning user research, qualitative and quantitative methods can be useful to obtain information. Nielsen [Nie08] describes the difference between the two data collection methods as follows: "*The basic distinction here is that, in qualitative studies, the data is usually being gathered directly, whereas in quantitative studies, the data is gathered indirectly, through an instrument, such as a survey or a web server log. In field studies and usability studies, for example, the researcher directly observes how people use technology (or not) to meet their needs.*"

Analyze Data

The step of analyzing the existing data shall filter out how many different kinds of user groups exist and then help to assign those user groups to different persona personalities. Nielsen [Nie13, p. 38] suggests segmentation to differentiate the data obtained from the user research. The challenge of analyzing the data might be to filtering out the most common similarities and categorizing them and, in the next step, deciding how many personae are created and which categories are assigned to the fictive persona.

Limit Persona to a Reasonable Amount

Sometimes, it is not useful to further evolve all found persona. Reasons for decreasing the amount of personae are that a Persona should be easily remembered and be present most of the product lifecycle. If there is a large amount of fictive users, it might be hard to remember which persona fits a given story best or even what the persona

Persona/Weight	1.Persona/75	2.Persona/15	3.Persona/10	...	∑
Feature 1	1	1	0	...	90
Feature 2	1	0	2	...	95
Feature 3	-1	1	0	...	-60
...					

Table 3.1.: A prioritized feature implementation table with different personae [based on Figure 4 PG03, p. 8]

was about. Another reason might be that not all personae have a high impact on the product. For example, Pruitt and Grudin [PG03, p. 7] introduced a "*feature-Persona weighted priority matrix*" with prioritized persona. Those prioritized persons could also be used in the development cycle when it is important to make a decision of which features have to be implemented first or which features could even have an negative impact for other users as Table 3.1 indicates [compare PG03, pp. 7, 8].

Most Common Attributes of an Adult Persona

Adult persona are most likely task- and goal-oriented and provide some of the information shown in Table 3.2.

These attributes shall not be seen as a persona template because the actual outcome of the persona poster is very much dependent on the focus area and the underlying research basis. At this point, it should be mentioned again that a persona should not represent a stereotype or be seen as a job description. Often, persona definitions are represented only in a textual way, but sometimes, a slide chart with a scaling from low to high could be as well the choice of representation, for example to indicate skill-, need- or fear- levels.

Persona Attributes	
Identity	Name
	Age
	Origin
	Picture
Status	Weight of the user for the project
Goals	Goals according to the program
	Personal goals
Knowledge	General skills
	Special knowledge of the program
	General knowledge
Tasks	Most important tasks the user does with the program
Relationship	Friends/family status (if important or to make the persona more realistic)
	Relationship to the focus area
	Other users/groups of interest that might get in touch with the person and the program
Attitude / Motivation	Media usage in relation to the focus area
	Skills with the used technology
	General knowledge/usage of other technology
	Level of Motivation to use the technology of the program
Expectations	Expectations of how the program will work (related to the task)
Hobbies, daily work/life ...	

Table 3.2.: Possible attributes of an *Adult Persona* based on information from [Table 1 SKI09, p. 334], [Figure 8 Nie13, p. 78].

Child Persona

According to Antle [Ant06, pp. 23, 24], a child persona shall focus more on needs, expectations and desires of children instead of tasks or goals for adults. Moreover, she claimed that gathering information about children might not be as simple as finding information about a specific group of adults. One reason for that is that children may have problems to express their needs in interviews especially about abstract products (compare Chapter 2.3, 2.4 about the abilities of children in certain age groups). The suggestion to obtain well-grounded information from Antle [Ant06, p. 24] is "*to use a framework which guides information gathering. The framework must provide insight into what to look for, how to organize information and help with the interpretation of observations and interviews in the context of the design problem at hand*".

The Table 3.3 shall provide a quick overview of differences that could be of importance when adult and child personae are developed.

Validation and Revision of the Product Lifecycle

In the opinion of Nielsen [compare Chapter 7, from Nie13] one way to validate whether the persona meets the criteria of the expected users is, to review the final personae with the people who initially had a certain image of their users in mind. In those reviews questions about certain persona details may arise. Those questions shall thereupon be answered and justified on the basis of the underlying research data and the development process of how the persona was created. A transparent presentation of the outcomes and an open communication are helpful during this verification process.

Adult Persona	Child Persona
Name	
Age	**Age group**
Origin	
Picture	
Person goals	**Child needs**
Person background	**Child environment**
Daily work/life	**Daily routines**
Relationship to the focus area	
Media usage in relation to the focus area	
Skills with the used technology	
General knowledge/ usage of other technology	
Family status	
Hobbies	
Friends	
Weight of the persona for the project	
Tasks	**Focus on expectations and needs**

Table 3.3.: Possible differences and similarities of an adult persona chart and a child persona chart, based on information provided by Antle [Ant06, p. 24]

Pruitt and Grudin [PG03, p. 8] collected information about how often a persona was used not only to measure the persona usage but to also gain an insight into which direction the project is headed. By knowing how many times a specific persona was used and how big their influence on the product was they tried to *"roughly gauge the direction of a product as it is developed"* [PG03, p. 8].

If new significant information about the target persona arises, which might especially happen if new markets are entered, the already existing personae shall be revised. For that reason, a persona is not finished when research and presentation are completed as they are rather continuously improved.

Usage of Persona (Beyond Scenarios)

The next step in the persona development is the actual use of the created persona. Most often, they are used in scenarios. A scenario is used to describe the persona computer interaction and, therefore, shall describe a real human-computer interaction goal. A scenario is based on the user assumptions and describes a workflow how a certain goal might be achieved by a specific user. It can include assumptions about possible appearances of problems and needs of the user, as well [compare Chapter 9 from Nie13].

Besides the design and development team, a persona can be used everywhere in the company where communication about the user or an image of the user is needed. Antle [Ant06, p. 28] used their developed persona models as the basis for *"cognitive walkthroughs and to educate the usability team"* as well as to determine the priority of the bugs found by their usability tests. Pruitt and Grudin [PG03] had a similar approach with their feature-weighted matrix (see Table 3.1) where each persona represents a market share. In combination with the feature impact for the user a prioritization for new features can be based on more user relevant data. They also created flyer's, promotional items and web sites that contain persona-related material (research reports, scenarios, customer data...) to provide access for those people who are more interested in the targeted user group or to better communicate to other departments what kind of people are assumed to use the product [PG03, pp. 6, 7, 8].

Sometimes, it could be useful to create an "anti-persona" to determine for what kind of users the product will not be developed and to be able to focus more on features for users that are more important [compare PG03, p. 5].

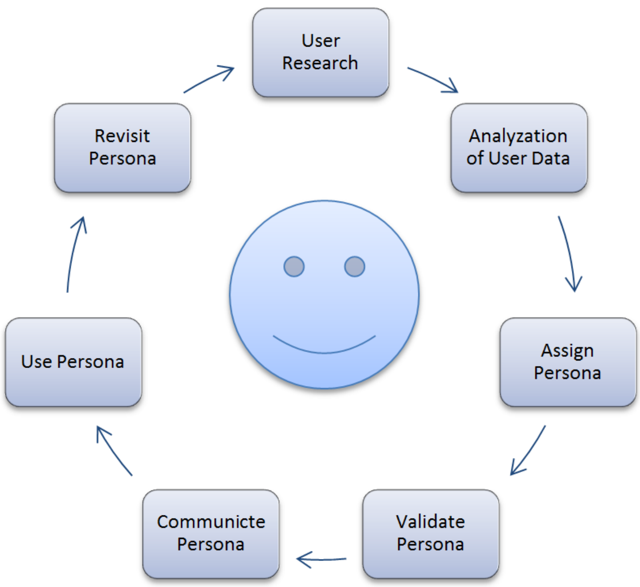


Figure 3.3.: Steps of a persona-generation cycle

3.4. Agile Usability and Software Engineering

This chapter shall provide an overview of how *Usability Engineering* (UE) methods may be combined with *Agile Software Engineering* (ASE) methods. Below is a description of the benefits of an ASE process followed by a description of the weaknesses in combination with UE that were found by Düchting, Zimmermann, and Nebe [DZN07] and suggestions of how to include ASE and UE. The two most referenced representatives of ASE methods are XP and Scrum.

Strengths of Agile Software Engineering Methods

Agile software engineering methods try to keep their iteration cycle as short as usable. Short iterations shall help to react quickly to changing requirements and a short iteration-based planning cycle tries to generate usable prototypes very early in the development phase and even tries to achieve a stable releasable product as soon as possible. This approach tries to ensure that efforts are visible to the customers and the project progress is assessable immediately during the development process. To ensure that the project is headed in the right direction during the development, there is always an onsite customer present who can be interviewed if any question arise. Software quality is ensured by collective code ownership and ongoing refactoring. To provide a project code basis that is qualitatively high, the principle of continuous integration shall help to ensure that the project code is up to date. The principle of continuous testing shall ensure that there is at any time a stable application build available that can be used immediately. The collective code ownership and the continuous integration and testing shall also help to minimize the required amount of documentation since the tests shall provide knowledge of what the code shall do and everyone from the team should be up to date with the code since everyone is allowed to modify the code if necessary. Product requirements are provided directly from the onsite customer (or his representatives). Those requirements are split up into as many small programmable tasks (or user stories) as possible and afterwards are prioritized and a time estimation is done. This iteration planning provides the basis for the work for the upcoming development cycle. The whole development process tends to rather have a direct face-to-face communication instead of reading possibly outdated documentations or specifications. For a more detailed introduction to AES, read also Chapter 1 from Brown. Figure 3.4 shows the main components of an agile software engineering method.

Weaknesses and Opportunities of Agile Software Engineering in Combination with User-Centered Design

Düchting, Zimmermann, and Nebe [DZN07, pp. 64, 65, 66] found some weaknesses and gave recommendations for ASE (in particular for XP and Scrum) to introduce user-centered requirements in an agile development process. Table 3.4 provides an overview of the outcome.

Another decision has to be made when the usability and design work have to be done and when the software engineering part is finished. There are two possibilities. The first one is that UE is done one iteration before the SE and the other is that UE and SE are carried out in parallel, where possible.

Brown [Bro13, pp. 45, 46] pointed out that the benefit of working one iteration ahead of the SE team is that the whole work of the UE team

Weakness	Recommendation
Short exploration phase especially at the beginning but within the on-going development as well .	• Use the short exploration phase to inform developers about the user's natural working environment. • Reduce documentation to a minimum and share knowledge with the whole team. • Include UE team member in the development team.
No real end users are considered. Onsite customers can be stakeholders and, therefore, might not have the same interests, concerning the software, as real users.	• Validate design mockups and prototypes with real users. • Gather information about the context of use and workflow.
Probable loss of view for the whole system and the relation between system features because the work is split into smallest possible tickets.	• Describe the workflow with use-cases (or scenario-based).
No verification that the system satisfies real user requirements.	• Review the system with UE experts and real users.
No usability system tests.	• Do usability-tests in system stages where useful workflows can be performed. Tests do not necessarily have to be performed by each iteration.

Table 3.4.: Weaknesses and recommendations for *Agile Software Engineering* that include *User-Centered Design* [see DZN07, pp. 64, 65, 66]

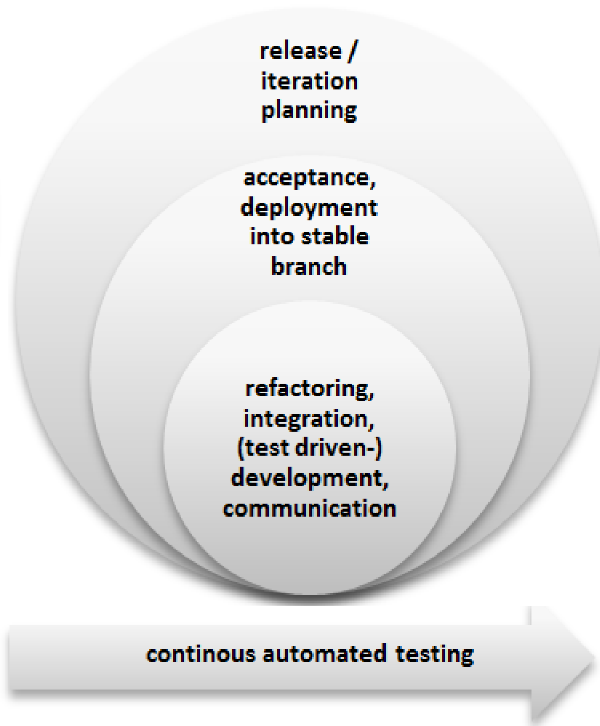


Figure 3.4.: Main components of Agile Software Engineering.

(design, customer feedback...) could be presented at once to the entire SE team and could immediately be integrated into the iteration planning of the SE team. She also found that possible weaknesses are that this might be a *mini-waterfall* approach and that the agile desire for direct and continuous communication might be corrupted and, therefore, the benefit of spreading the knowledge across the whole team might be diminished.

Another risk might lie in really fast changing agile environments where the requirements change very quickly. It could be possible that the UE work, that was done one iteration ahead, might not meet the requirements for the SE iteration any more. Also it could be possible that the deliverables from the UE team cannot be finished within one short iteration and, therefore, these deliverables cannot be provided for the iteration of the SE team [compare Chapter 2 Bro13].

The benefit of working in parallel iterations is the better visibility of the UE team and the closer collaboration of the teams and, thus, a possibly better understanding of the two teams. The deliverables cannot be outdated since they are concurrently developed and if the design effort can be cut down to the necessary work, it is possible that working in parallel may mean for the UE only doing the things needed [compare Chapter 2 Bro13].

Which iteration cycle is chosen will strongly depend on the members of the team(s) and how they individually accept each model and the empirical knowledge of what is the best practice for the outcome of the product. Nevertheless, it should be mentioned that evaluating one's own methods of operation and embracing changes is a part of an agile engineering model.

3.5. Summary

The usability lifecycle aims at improving the user interface design iteratively within different process stages regarding different usability characteristics like learnability, efficiency, memorability, errors or satisfaction.

One of those stages says that you have to know the user. One tool to get to know the user is the Persona method that reveals its strengths when the underlying research about the users has been done properly and the created Persona is in everyone's mind, not only in the designers' minds but also in the minds of those who participate in the development process. A Persona might not only be used to aid design decisions but could also be used in the iteration planning process of an agile software engineering process to determine which features have priority or may have a negative impact on the overall user experience.

The agile software process may not necessarily inhibit the design process. To the contrary, it could be an advantage. Especially when children are involved, there is a higher possibility that designers may not always exactly meet the needs and expectations of children. Therefore, an agile design process that continuously evaluates its user interaction behavior might even be a requirement to fulfill a positive user experience for children. Furthermore, it makes it possible to more often integrate children in their different roles as informants, testers, users or design partners when they are needed.

4. Prototyping

With *Prototyping*, it is meant to develop only a subset of the total features compared to the full system or reduce the level of functionality. The aim of a prototype is to test with real users in an early stage of development with a more or less realistic look and behavior of the system. Describing users, only with a written document, how the user-interface will look like or how certain functions will behave, is not always possible. For example the level of abstraction could be to high, because, in the written document there are described too many user-interaction points that cannot be visualized all at the same time by the user. Prototyping tries to reduce that abstraction level problem by providing design and functionality previews to the user in an early development stage.

4.1. Horizontal and Vertical Prototyping

When only a subset of the features is implemented, it is called vertical prototyping. The advantage of vertical prototyping is that the existing features can be tested in depth and under realistic conditions.

The reduction of functionality is also called horizontal prototyping. Since there is only little or even no underlying functionality, horizontal prototyping may be used to test or simulate interfaces of a fully featured system where no real work can be done [compare Nie93, pp. 94, 95] (see also Chapter 4.3 *Wizard of Oz Method*, for testing with prototypes).

Vertically and horizontally prototyping can be combined stepwise as Figure 4.1 indicates. In Figure 4.1 there is shown a "scenario for testing". Such a scenario is a minimalistic prototype which has only one feature and the functionality is that much reduced that no interaction with real data is possible. Since scenarios are that much limited, Nielsen [Nie93, p. 100] suggests to use them

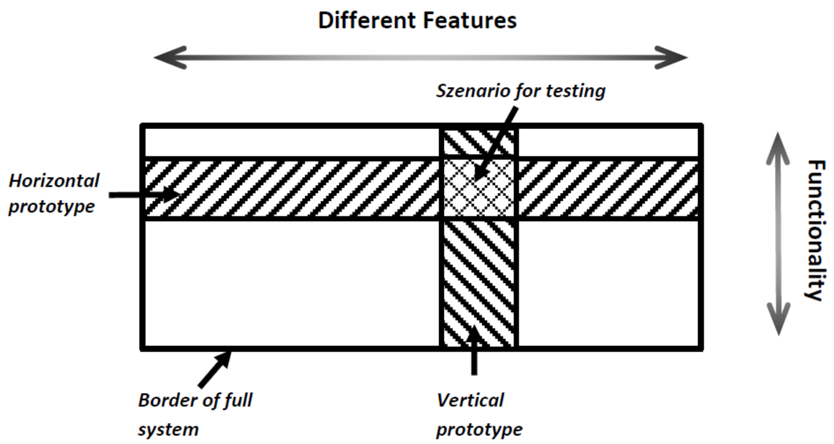


Figure 4.1.: Differences between *vertical* and *horizontal prototyping* and the limitation of features and functionality [based on Figure 9 Nie93, p. 94]

- during the design phase to understand how users will eventually interact with the system.
- during early evaluations of a design to receive a fast user feedback at low costs because there is no need to implement a working prototype.

Liebal and Exner [LE11, pp. 191, 192] see the use of prototyping with children within

Participatory Design where children are design partners and are an active part in the development process. Prototypes are realized as *Low-Fidelity Prototypes* only, which means there is no technical implementation of a system. Instead, children are able to build their own prototype with suitable resources (paper and pencils, crayons, paper-mockups...).

Expert Design is the realization of a prototype by developers and designers. Children only test the prototype but do not actively take part in the development process of the prototype. These prototypes are made by experts and thus they can be realized as *Low- and High-Fidelity Prototypes* (realized in a technical environment).

The possibilities of a prototyping process in the manner of Liebal and Exner [LE11, p. 191] is indicated in Figure 4.2 where horizontal and vertical prototyping has the same meaning as described earlier.

4.2. Low- and High-Fidelity-Prototyping

Low-Fidelity-Prototypes have the benefit that they are less costly compared to *High-Fidelity-Prototypes* and quick to realize. They can be made with rather ordinary resources like paper and pencils. This kind of prototypes is often called paper mockups. They are rather inaccurate and can be put into practice with freehand drawings that do not necessarily have to be very aesthetic. Another advantage of hand drawings at an early stage is the low threshold to change or even throw away such designs. A user-interfaces test can be done in such

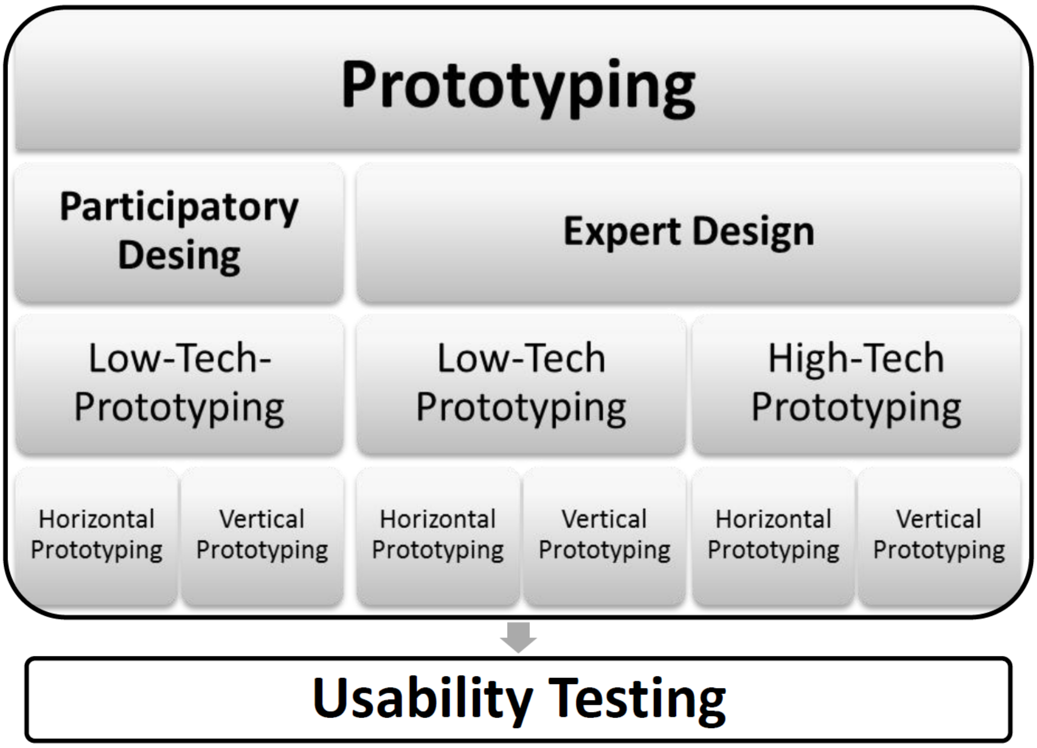


Figure 4.2.: Prototyping variants based on Liebal and Exner [LE11, p. 191] within the context of the usability engineering lifecycle; Liebal and Exner [LE11, p. 191] use the terminology *Low-Tech-* and *High-Tech* Prototyping which is the same as *Low-Fidelity-* and *High-Fidelity* Prototyping

a way that if the user pushes a paper icon/element, an administrator has to change all the paper elements that would have be affected in the planned system by the performed user interaction [compare LE11; And12, pp. 194; 78, 79].

High-Fidelity-Prototypes characteristics are that they have a longer development time and are more costly. The benefit of a *High-Fidelity-Prototype* is that the implementation either horizontally or vertically is close to the desired final system and, therefore, testing such a prototype allows a more precise feedback as well as testing in more detail [LE11; And12, pp. 193; 82].

From Liebal's and Exner's [LE11, p. 194] point of view, *High-Tech-Prototypes* can have a negative effect when a child has to evaluate such an interface because they might tend to not criticize the underlying system also they might show less understanding for the limited possibility to interact because of the restricted functionality.

4.3. The Wizard of Oz Method

The Wizard of Oz can be seen as a prototyping method. Therefore, a High- or Low-Fidelity Prototype is used where some or all of the underlying functionality is left out. The missing functionality is simulated by a human according to the user interaction. A drawback is that the human administrator, *the Wizard*, has to interpret what user interaction was performed by the testing subject and afterwards, the intended functionality of the program has to be activated or simulated remotely.

To be able to simulate the functionality, the Wizard must have the knowledge of the study domain. Therefore, the Wizard could possibly be a developer. For challenging or multiple actions at the same time, it is possible to have more than one Wizard.

From the usability testers' point of view Wizard of Oz studies might not be suitable if the environment changes fast and the Wizard has problems to follow the user interaction or if the functionality that should be provided needs a longer duration to be processed by the Wizard. Another reason for not using this method could be that the user input might not be easy to interpret and, thus, a proper simulation how the program would react is tricky (for example, if a child's handwriting should be corrected automatically, it will depend on the underlying algorithm how the handwritten letter is interpreted) [compare Chapter 12, from Mar+08].

5. Usability Testing With Children

Usability testing is understood as one of the most valuable techniques to gather direct user feedback and the most common representative is the *Think-Aloud* method (see Chapter 5.2.1). Nevertheless, as already described in Chapter 2, children go through different evaluational stages and their cognitive abilities are just evolving. Because of that, this chapter shall identify what the main differences of usability testing with adults and children are. The most common usability test methods will be described and what adaptions have to be made for children.

Finally, the Robotic Intervention method (Chapter 5.2.7) and the Problem Identification Picture Cards method (Chapter 5.3.1) will be described. These methods are especially designed for young children.

5.1. Decisions Before The Test

5.1.1. Usability Laboratory vs. Familiar Places

In general, a child-friendly location should be chosen for usability tests. Liebal and Exner [LE11] distinguish between two different location categories:

Familiar places: These places are known and frequently visited by children like kindergarten, school. Sometimes the testing environment can even be at home.
The main advantage of those places are that children feel more comfortable and relaxed even in the new situation. Hence, better findings can be expected. Other advantages are missing arrival and departure ways and that children may not completely be pulled out of their daily runs.
Disadvantages are possible disturbances. As mentioned in Chapter 2, a child's focus can switch easily. Especially this can become a problem in kindergarten (or schools), where other playing kids, teachers or parents may distract the child who is currently running the test.
Also negative factors are:

- The need to transport the equipment.
- The setup may not always be that easy to hide.
- A limited time to arrange the equipment.
- Doing a dry run might not always be possible.

Usability laboratory: *The benefit* of a usability lab compared to familiar places is a stable environment. Exterior effects can be minimized and the comparability of the results is higher. The test setup is easier to manage and if necessary, the chance to separate caretakers from the involved child that shall do the testing might take less effort. Reasons for doing this might be an undesirable influence on the child or too much intervention.

Negative aspects of a usability lab might be that children do not feel comfortable in an unknown place combined with the new situation. As a consequence, they do not provide their best support to the usability test or might even abandon the test session. To make children more comfortable, usability labs should be equipped in a child-friendly manner, for example with posters. But it should be kept in mind that accessories can also lead to distraction. The worst case could be that children are more interested in getting in touch with their environment than the actual product being tested.

5.1.2. Choosing Child Participants

When a usability test is performed in kindergarten or school, then there is no decision whom to choose for the test. Whether all children are suitable for the test session or not, because they have for example individual disabilities, all willing children should be allowed to take part. Otherwise, the children who were not allowed to take part might feel left out from a group. Afterwards they might have problems when the child group discusses their experiences [compare Mar08, p. 96].

If resources are limited, most often the time factor is one of the most critical ones, and therefore, a preselection might be reasonable. According to Nielsen [Nie94, p. 393], three to five testers are a good number of adult test subjects to find about 75% to 80% of usability problems in an interface with the *Think-Aloud* method (see Chapter 5.2.1). To find all possible usability issues Nielsen [Nie94, pp. 389, 393] found out that 15 test subjects could be enough. Nielsen, therefore, recommended to iteratively do usability testing with five people within each session. If the tested application is complex or has different groups of users (e.g., children and adults use the application), a greater number of test participants can be useful [compare Nie00; Vir92].

An estimation for an expectation of uncovered problems may be the formula $1 - (1 - p)^n$ where p is the average coverage of a single person and n is the number of test participants. A possible diagram with values for p between 0.25 to 0.40 is shown in Figure 5.1.

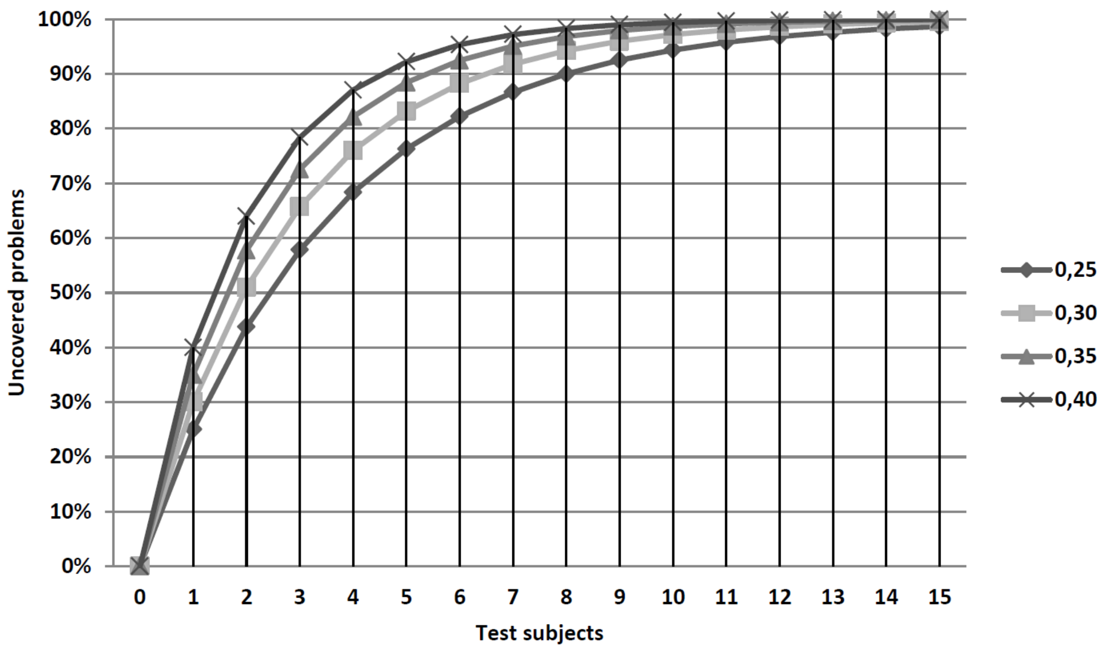


Figure 5.1.: Possible test coverage for adult testers, for the formula $1 - (1 - p)^n$ with p...single test coverage 0.25, 0.3, 0.35 and 0.4, n...number of test subjects

Another approach to estimate the success factor of usability tests could be the calculation of confidence intervals for binary success. Confidence intervals shall describe how confident one can be that a larger group of people will experience the same (problems), as an equivalent smaller group experienced in a usability laboratory. The estimation of that success factor for small groups might possibly be a problem for the "Wald formula" or the "Exact method" because, in the opinion of Tullis and Albert [TA08, p. 97], who quote Sauro and Lewis, these formulas might be too conservative or liberal.

Sauro and Lewis [Chapter 3 SL12, pp. 23, 24] found the Adjusted Wald formula that in their opinion fits best small sample sizes:

$P_{adjusted} \pm = \frac{x+\frac{2}{2}}{n+z^2}$
x ... number of persons who successfully completed a task
n ... number of persons who tried the task
and the adjustment is inserted in the standard Wald formula:

$P_{adjusted} \pm z_{(1-\frac{\alpha}{2})} \sqrt{\frac{P_{adjusted}(1-P_{adjusted})}{n_{adjusted}}}$
z ... critical value from the normal distribution for the level of confidence (e.g., 1.64 for 90%; 1.96 for 95%; 2.57 for 99%)

The confidence intervals for the adjusted Wald formula and n=5 participants with a success rate x from 0 to 5 for a confidence of 95% are summarized in Table 5.1. For a quick computation of confidence intervals for different formulas, the calculator from `http://www.measuringusability.com/wald.htm`[1] may be useful.

However, when it comes to usability testing with children, these suggestions have to be reconsidered. Especially when testing with young

[1]last visited on April 18th, 2013

	Low	High
succeeded=0		
Adjusted Wald	00%	40%
Exact	00%	45%
Wald	00%	00%
succeeded=1		
Adjusted Wald	02%	64%
Exact	00%	72%
Wald	<00%	55%
succeeded=2		
Adjusted Wald	12%	77%
Exact	05%	85%
Wald	<00%	83%
succeeded=3		
Adjusted Wald	23%	88%
Exact	15%	95%
Wald	17%	>100%
succeeded=4		
Adjusted Wald	36%	98%
Exact	28%	99%
Wald	45%	>100%
succeeded=5		
Adjusted Wald	60%	100%
Exact	55%	100%
Wald	100%	100%

Table 5.1.: Confidence interval of 95% for n=5 participants and a success rate x from 0 to 5

children at the age from three to five, it is more likely that test sessions might be discontinued and sometimes it might be hard to evaluate the children's behavior [HRA97; LE11, pp. 10, 202]. For those reasons, Liebal and Exner [LE11, p. 202] suggest the number of child testers should be at least eight. The experience from Barendregt and Bekker [BB, p. 1] is, that eleven to thirteen children uncover 80% of problems but this will depend on the complexity of the application, as session times cannot be too long for children:

Another interesting research study by Barendregt et al. [Bar+07] tried

Age (years)	3-5	5-7	7-11	11+
Max. duration (min)	10-15	15-30	30-45	45

Table 5.2.: Suggested duration of a usability test session [compare Kes+03; LE11]

to investigate if there possibly is a coherence between a child's personality and the number of found usability problems either reported verbally or indicated otherwise. She also compared if there is a significant difference in the results of the 11 best- and least-promising children. Some of the results are shown in Table 5.3. To determine who are the

	Problems found		**Frequency severity**			**Impact severity**		
	found problems	verbalised problems	high (#11)	average (#20)	low (#78)	high (#6)	average (#18)	low (#85)
most promising children	82	43	11 (=100%)	20 (=100%)	51 (=63%)	5 (=83%)	16 (=88%)	61 (=72%)
least promising children	76	28	11 (=100%)	20 (=100%)	45 (=58%)	5 (=83%)	11 (=61%)	60 (=71%)

Table 5.3.: Comparison of 11 most- and least-promising children according to the children's characteristics [Bar+07, pp. 144, 145]

most promising children before the actual usability test, the parents of the children had to fill out a questionnaire about the personality of their children[2]. Barendregt et al. [Bar+07] suggest, "*for evaluation practitioners would be to make a selection of children based on the personality characteristics, Extraversion, Friendliness and Curiosity*" because "*children would find both a large number of problems due to Curiosity, and provide self-initiated spoken output for a large proportion of these problems due to high Extraversion and low Friendliness*"[2].

An overview of the five main category groups and the according subcategories of the Blikvanger test [compare Bar+07, p. 135]:

Extraversion: Approach Seeking, Positive Emotionality, Sociability

[2]Characteristics according to the *Blikvanger* 5-13 test

Friendliness: Dominance, Agreeableness, Altruism, Affection

Conscientiousness: Conscientiousness, Impulsivity

Emotional Stability: Emotional Stability, Self-Confidence, Manageability

Intelligence: Curiosity, School Attitude, Creativity, Autonomy

5.1.3. Further Discussion About the Usability Test Setup for Children

When preparing a usability test for children, there are some things that are of greater importance than when testing with adults. Decisions that have to be made before the test when testing with children are:

How many administrators shall be present: Markopoulos et al. [Mar+08, p. 97] suggest that there are at least two adults present, one administrator for the organization and the other one should be the facilitator for the test who takes the notes. Markopoulos et al. [Mar+08, p. 97] say that "a second adult makes the event safer for both the children and the adults."

Task-based testing: Tasks may be especially challenging for young children (up to the age of six years) because the tasks have to be designed in the language of the children. Otherwise, the cognitive abilities might be overcharged as well as the short-term memory of young children is claimed to remember the task description. Task-based testing might require reading the task to the children. This can lead to the possibility that children start to ask questions about the task for clarification. Test tasks might be too challenging or too boring, which can lead to the distraction of the child or even the abandonment of the test session.

Finally, a task might be interpreted, by children, as an examination question, thus it might be necessary to explicitly note that the software is tested, not the child [compare Mar+08, p. 99].

Testing without tasks: Markopoulos et al. [Mar+08, p. 97] find testing without tasks "suitable when testing products with small children or even with toddlers" because the demand on the cognitive abilities are not so high as with tasks.

Provide time for a trial: To have a more realistic test situation, children should be able to freely practice some time with the software [LE11, p. 216].

Explain the test situation: Take enough time to explain the test situation to the child. It should be made clear that it is not the child that is tested. Also provide enough time for the child to investigate the test laboratory and answer all questions [LE11, p. 216].

One particular decision has to be made for usability testing with mobile devices and children, whether the test session shall be video recorded as well. For usability tests with a desktop computer, normally a static camera setup that has a focus on the screen or the user, should work well. For a screen recording of the test device a software could be useful as well.

Testing with mobile devices is different because, either there is no proper software to record the screen activity or the link to the recording software on a remote computer might be too weak to transmit fluent video data. Furthermore, it can be problematic to record the user properly, because something is in between the field of vision of the camera and the user.

In the last years, there have been some attempts with so-called *Usability Sleds*[3] like the one in Figure 5.2 that do not necessarily need a static camera setup but may record the finger movement very well [compare CB12]. Possible problems with children and sleds could be:

- The sleds are unnatural and can distract children.
- The sleds could be heavy, especially for young children.
- Children could possibly use the device in an unnatural way.

Figure 5.2.: Mobile Sled for usability test [image source CB12].

[3] `http://goo.gl/93nVI`

5.2. Verbalization Methods

The aim of verbalization methods is to encourage the participants to communicate what they are doing and what their thoughts are about the current work flow. Verbalizations can be elicited or spontaneous. Depending on the test method and the test environment, participants can be reminded to verbalize their feelings and thoughts or the test method itself has a technique that should keep the participants talking, like the *Co-Discovery* method (Chapter 5.2.6) or the *Peer Tutoring* method (Chapter 5.2.5) [compare Mar08; LE11].

5.2.1. Think-Aloud Method (Think Out Loud)

Method Description

The Think-Aloud (or Thinking Out Loud) method is the most noted method in literature and has become the standard practice in the usability testing of products for adults.

During the test session, the participant has to verbalize his current working steps, feelings and thoughts. Verbalization should never stop and, therefore, the facilitator should consistently remind the person of speaking out loud if he or she forgets. That gentle reminder should be done in a very neutral way because the facilitator should always stay in the background during the test and should avoid starting a dialog or helping the participant [Mar08, p. 188].

Adaptations for Children

- The product by itself can have high cognitive demands and as a consequence, there are no cognitive resources left for thinking out loud as well as it could be the opposite way, verbalization demands too much capabilities that leads to less interaction with the product. Because of that, there is a possibility that either verbalization or interaction with the product is neglected [Kes+03, p. 42].
- Another problem that might occur may be that children find it unnatural to talk when the person who is present is not responding [LE11, p. 210].
- When young children forget to verbalize, they need to be continuously reminded to keep talking. This can make children feel obliged to mention problems to please the experimenter. This could lead to non-problems being reported [DR04, p. 43].
- The not fully developed language skills of children may make it hard to clearly express their thinking [Mar08, p. 189].
- Children may be shy toward unknown adults. Those children can find it difficult to verbalize their thoughts. The *Think-Aloud* method can inhibit children from uttering their thoughts even more when an unresponsive adult requires the child to keep talking [Mar08, p. 189].
- *Think-Aloud* is a special social situation in the life of a child who may be accustomed to situations where the adult asks questions. Normally, a child is in a situation where the adult has an implicit understanding that there is a right and wrong answer or where the adult guides the child with advise and instructions [Mar08, p. 189].

- Adults seem to have it easier to slip in and out of the role of a tester. This can make it easier to understand that the product and not the performance of the tester is tested. Children tend more to think their accomplishment is under test. Children as well as adults do not want to make mistakes in the presence of others and this can lead to a different behavior, with the tested product [Mar08, p. 189].

5.2.2. Retrospective Think-Aloud Method (Retrospection)

Method Description

While the *Think-Aloud* method (Chapter 5.2.1) tries to gather verbalized information from the test user during the test session, the *Retrospective Think-Aloud* method (or Retrospection) analyzes the test session in cooperation with the test user after the actual test session. Therefore, the whole test session has to be recorded and when the video of the test session is reviewed, the facilitator asks the child questions about the interaction with the product [LE11; Mar08, pp. 211, 195].

Markopoulos [Mar08, p. 195] lists the following advantages:

- Because the verbalization happens after the test session, children have more capability for testing the underlying product and the test situation is actually closer to reality to how a child might interact with the product in practice. For this reason, the possible outcomes of the test session might be of more value.

- Another advantage is the possibility to pause or replay the video if there is an interesting incident.

On the other hand, disadvantages are [compare LE11, p. 211]:

- Children have problems to assign their thoughts to an earlier incident up to the late concrete-operational-phase[4].
- Moreover, the retrospective video analysis consumes an at least an equally amount of time as the test session. This might be a problem for children because they can provide just a limited time of attentiveness, which might lead to a limitation of the capacity for remembering.
- In addition, it is possible that a long test session leads to discomfort of the child and, thus, the possibility of an abandonment of the test session may rise.

5.2.3. Post-Task Interview Method

Method Description

In the *Post-Task Interview*, the test session is divided into small user activities. After every activity, the facilitator performs a short questioning about the previously performed task. Such an interview should only be performed after short activities while the experience is fresh in mind [Mar08, p. 203]. Markopoulos [Mar08, p. 203] also refers to two studies which showed that there were fewer reportings of problems in their answers to the questions given by the facilitator with the *Post-Task Interview* method compared to the *Think-Aloud* method (Chapter Think-Aloud Method 5.2.1).

[4]Look at Table 2.1 for an overview of Piaget's stages.

5.2.4. Active Intervention Method

Method Description

For the *Active Intervention* method, more interaction between the test user and the facilitator is required. The facilitator asks questions while a task performed by the test user to directly obtain current interpretations and opinions with various elements. The facilitator is even allowed to suggest directions for interacting with the product [Mar08, p. 202].

The questions should task-specifically be prepared in advance and children should be prepared before the test that they will be questioned during the test [LE11, p. 211].

Benefits of the *Active Intervention* method with children is the more natural way of children interacting with the facilitator. This may result in more comfort for the child tester, leading to a more qualitative analysis of the product. Another advantage of this method is that it helps the tester to continuously stay vocal, and the test session can focus on more interesting aspects of the product [Mar08, p. 202].

The *drawbacks* of the method are that the answers of the children depend strongly on the amount of questions and on the facilitator him- or herself [LE11, p. 211].

Also the questioning of the facilitator can accidentally lead the test session in an unintended direction. As a result of the questioning the product could be used in a different way as in reality [Mar08, pp. 202,203].

5.2.5. Co-Discovery Method (Constructive Interaction)

Method Description

In the *Co-Discovery* method, two test participants are able to explore an interface together and there is a natural interaction and communication between the two test users. A benefit of this method is that the child test users do not have to be trained to think aloud.

Drawbacks of this method are the higher amount of test users required and that usually the software product is not discovered in pairs, in reality. [And12; LE11; Mar08, pp. 126, 212, 207].

Adaptations for Children

- To make this technique work, it must be clearly pointed out to the children that they should cooperate during the test session and solutions should be discovered together. To encourage the cooperative work, each child should be given a separate assignment but only when the collaboration is done, the finishing of their assignment should be possible [LE11, p. 212].
- A major disadvantage of the *Co-Discovery* method is that especially young children do not play with other children as discussed in Chapter 2. Liebal and Exner [LE11, p. 212] mention that cooperative play starts at the age of six or seven and according to a study by van Kesteren[5], even that is not always true and children start to work on their assignments independently. Markopoulos [Mar08, p. 212] remarks that few authors have reported how well this technique served their purposes

[5]Kes+03.

when working with children. Moreover, Markopoulos claims problems like one child trying to dominate or trying to do everything alone.

5.2.6. Peer Tutoring Method

Method Description

Markopoulos [Mar08, p. 208], Liebal and Exner [LE11, p. 212] explain that a *Peer Tutoring* session is divided into four parts and three participants, the tutor, the tutee and a test supervisor who guides the collaboration. Tutor and tutee are children, the test supervisor is an adult.

The peer tutoring order of events is as follows:

1. Introduction of the test setup and the tutor-tutee roles.
2. Training of the tutor. The tutor is allowed to become familiar with the tested product and collect some experiences.
3. In the third part of the session, the tutor has to teach the tutee how to use the software being evaluated.
4. A concluding interview.

With this method, it is possible to observe how children use and how they talk about the product and whether children are able and willing to teach each other how to use the product [Mar08, p. 208].
An enhancement of the *Peer Tutoring* method compared to the *Co-Discovery* method described before is that children do not have to collaborate.

5.2.7. Robotic Intervention Method

Method Description

The *Robotic Intervention* method is a variation of the *Active Intervention* method described in Chapter 5.2.4 with the difference that the facilitator who is located somewhere next to the child tester is replaced by a *Social Robot*. Such a robot is able to interact and communicate with the child tester. This robot may be remote-controlled by an administrator in another room to avoid the presence of the administrator in the test room. The procedure for the *Robotic Intervention* is the same as for the *Active Intervention* except that the robot is introduced to the child tester and that it is made clear to the child that interaction is provided through the robot. The administrator of the robot follows the predefined protocol as if it would be an *Active Intervention* protocol except that human-like emotions and directions are provided through the social robot [compare Mar+08].

5.3. Non-Verbalization Method

5.3.1. Problem Identification Picture Cards Method (PIPC)

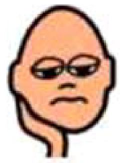
Method Description

The *Problem Identification Picture Cards* method (PIPC) is a supportive technique for usability tests. When children have problems to verbalize their feelings and thoughts during a usability test, this method may help.

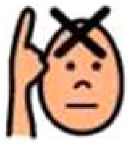
If children are not able to express their feelings in a usual manner, they have the possibility to pick up a card. Each of the cards shows a problem-symbolizing picture. The picture card has to be picked up and placed in a box. Through the video recording, the indication of a problem can be analyzed afterward. If facilitators do not understand why a particular card is used, they can actively intervene and ask the child for an explanation [compare LE11; Mar08, pp. 214, 197].

Barendregt, Bekker, and Baauw [BBB08] used the pictures from Table 5.4.

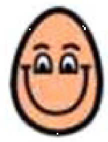
They decided to limit the picture cards to only eight pictures so that children are not overwhelmed with remembering the concepts of the cards. Another reason for limiting the number of available cards is the possibility to use the same card for different kinds of problems. Barendregt, Bekker, and Baauw [BBB08, p. 97] give the example that children



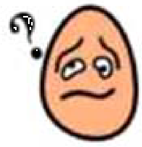
(a) boring



(b) don’t know/ understand

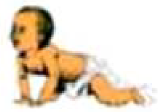


(c) fun



(d) difficult

(e) take too long



(f) childish

(g) silly/strange

(h) scary

Table 5.4.: Pictures used for a PIPC usability test Barendregt, Bekker, and Baauw [image source BBB08, p. 94]

could say that it is difficult when something is hard to see or hear as well as they could say that something is difficult to click because it is very small. For both examples, the picture card d. for difficult from Table 5.4 might be used. To determine the indicated problem, the context where the issue occurred has to be investigated together with the picture card.

Furthermore, this method could be used to distinguish on the one hand between usability problems related to perception, cognition or action and on the other hand fun problems [compare BBB08, pp. 97, 98]:

Usability problems related to

Perception is the ability of a child to recognize different influences (effects) from the game and assign them to specific events (actions) in the game. A perception problem may occur if something is *difficult* to hear or see (picture 5.4 d.).

Cognition usability problems appear if children do *not know what to do or how to do or do not understand* what to do (picture 5.4 b.).

Action usability problems are related to performing physical actions that are *difficult* to perform (picture 5.4 d.).

Fun problems in relation to the pictures

Boring (picture 5.4 a.) might be a game if the challenge level is too low.

Boring (picture 5.4 a.) picture cards may be related to curiosity problems as well.

Difficult (picture 5.4 d.) fun problems may be experienced by children when the challenge level is too high.

Takes too long (picture 5.4 e.) picture cards can be used when children have control problems and they think that something takes too long.

Childish (picture 5.4 f.) problems may be encountered by children because of fantasy problems. The game might be aimed at younger children.

Silly/Strange (picture 5.4 g.) problems may be encountered by children when the story is in-congruent with the experience of the child.

Scary (picture 5.4 h.) problems may be encountered by children because of fantasy problems. The game might be aimed at older children.

The fun picture card (picture 5.4 c.) was not used to indicate problems. This card should indicate that the evaluation is about fun as well.

PIPC combined with the Think-Aloud Method

The research results from Barendregt, Bekker, and Baauw [BBB08, pp. 101, 102] show that there is a significant positive difference between the number of problems expressed with the *PIPC* method and the *Think-Aloud* method (Chapter 5.2.1) as there is no significant difference of the number of verbalized problems between these two methods, which means that the PIPC method seems to have no negative effect on the average number of verbalized problems.

Improvements

Improvements to the original *PIPC* method according to Barendregt, Bekker, and Baauw [BBB08, pp. 102, 103] could be made in the following sections:

- When children put the cards into the box, they have to shift their attention from the game to the box and this requires supplementary effort. Placing the picture cards within closer proximity to the screen may reduce the extra effort.
- Pointing at the picture card instead of picking it up and placing it in a box could be sufficient.
- There could be a better selection of pictures than the chosen ones from Table 5.4.
- Results for the *PIPC* method have only been published for adventure games. It is unclear how this method complies with other types of games. Especially with software that changes its environment swiftly, picking up picture cards and placing them in a box could be too challenging.

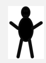
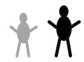
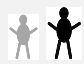
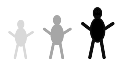
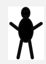
5.4. Summary

To perform a usability test with children, the current methods are more or less suitable with restrictions and adaptations. The younger the children, the more constraints and a greater effort in setting up the test environment, maintaining the test session and evaluating the results have to be made. Compared to adolescents, children have special or other needs not only concerning the test environment and the usability method. They also have other expectations, for example, how a software product has to be designed or which functionality has to be implemented.

Particularly young children have a naturally lower cognitive level than adults and, therefore, some usability techniques like *Think-Aloud* where concentration has to be divided between an effort in verbalizing current feelings or thoughts and using and testing the product (at the same time) may overcharge children in certain situations. Another challenge is the limited time span during which young children are able to concentrate solely on a single exercise, as shown in Table 5.2.

Another aspect that should be considered is where the usability test takes place. As already mentioned, especially the usability laboratory should be prepared in a child-friendly way but not too pleasant so that the child's focus shifts from the product under test to its environment. And when the test session takes place in a natural environment of a child (at school...), it should be considered that the usual daily routines may influence the usability test just like an uneasy environment might lead to no usable data.

Either way, when it comes to usability testing with children and especially young children, the expectation should not be that everything works as planned. Rather, one should be prepared to quickly switch the test methodology (for example from task-based testing to no tasks because the tasks are too complex...) where possible, to gain at least some insights into the child interaction with the product.

Method Name	Description	Required Skills	Additional Skills	Age Group
Think-Aloud	Instruct to think out loud during the test (and remind to verbalize)	thinking aloud (without having a counterpart)		
Retrospective Think-Aloud	Ask questions during a video review	remember and answer questions about previously done test session	patience to watch video about previous test session	
Post-Task Interview	Ask questions after right after a completed task			
Active Intervention	Ask questions during the usability test session	answer questions about current plans and actions		
Robotic Intervention	A variation of active intervention. The facilitator is replaced by a remote-controlled human like robot	same as active intervention		
Co-Discovery	Two testers may explore the test object together	communicate and express to other children		
Peer Tutoring	One tester teaches the other a previously learned task	teach another child	teach only by explaining (not by taking control of task)	
PIPC	Possibility to express feelings by selecting picture cards instead of verbalization			

 . . . age 2-7,  . . . age 7-11,  . . . age 11+

Table 5.5.: Usability test methods and their possible age group

6. Usability Log Analysis

This chapter shall investigate which usability metrics are possible and feasible to collect in the context of mobile applications and in which way these usability logs are analyzable. Moreover, the question is whether it is possible to build an association between the collected data and the user behavior. Furthermore, it should be possible to derive information about what improvements should be done to a user interface by analyzing the logs. At least it shall be given an outlook whether existing approaches can be used with children as well or if further research or adjustments have to be done. The stages of a log analysis are the application instrumentation, the logging itself and the log analysis as shown in Figure 6.1.

Hilbert and Redmiles [HR01, p. 572] and Hartman and Bass [HB05, p. 825] determined the overall problems that have to be considered when usability logs are introduced in the usability engineering process as follows:

Selection problem: This problem is related to the question, "What should be logged in which situation?"

Context problem: If the log data is not sufficient to understand or analyze a (usability) problem, "How can more information be made available from other sources?"

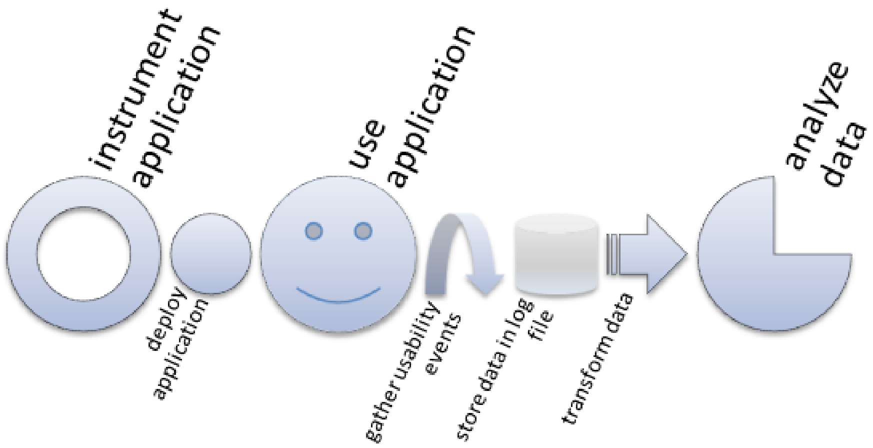


Figure 6.1.: The stages of the log process [based on Figure 1 Bat+09, p. 46]

Reduction problem: How can the data be reduced to get quicker access to identify the important issues?

Abstraction problem: Logs are most likely collected on a low-level code basis. This information has to abstracted to a high-level user interface information.

6.1. Strengths and Weaknesses of (Automatic) Log Analysis in a Mobile Context

The following strengths and weaknesses of *Usability Log Analysis* are based on the research articles from Balagtas-Fernandez and Hussmann [BH09], Lettner and Holzmann [LH12], Kaikkonen et al. [Kai+05] and Mayz, Curtino, and De la Rosa [MCD12].

Strengths of Log Analysis in a Mobile Context

- Often, cameras are used to record usability studies but sometimes, the line-of-sight between the camera and the test device might be concealed by the testing person. Another problem could be the small display size of the device that might be poorly visible for the camera. Logs can provide clear statistical data about the occurring events even in a fast-changing environment.
- If the user does not indicate usability problems or they are not perceived as such by the facilitators, there is still a chance that the usability issue can be detected by analyzing the log.
- Real world conditions, influences from the environment and different contexts of use can be tested that might not easily be emulated in a laboratory.
- The usability test session may not necessarily be bound to a fixed time slot and place.
- There is a chance to gain access to a greater range of users with different backgrounds (e.g., different languages, cultures, knowledge of the technology...).
- Non task-based usability studies and the absence of an observer may provide more realistic information about the tested system.
- If the logging behavior can be triggered without the modification of the underlying code, no additional effort has to be made by developers.
- Logs are objective.
- Logging does not require an observer, they are processed unsupervised.
- Logs are obtained automatically.
- Logs can be analyzed and visualized automatically.

- Easier usability error or bug recovery with logs.
- There is a possibility to verify a positive (or negative) effect of an interface change.
- It is easier to get an insight into how the whole application is used ("the big picture").

Weaknesses of Log Analysis in a Mobile Context

- Analyzing logs can only be done with a high-fidelity prototype.
- Logging events may afford arbitrary access to the program code to instrument the logging behavior.
- There are currently only few frameworks for mobile software that provide an insight into user behavior and those available more often focus on business-related metrics instead of usability metrics.
- If the log analysis is carried out within a field test that is observed, there could be a possible time overhead for the field test preparation.
- If the logging behavior cannot be triggered automatically, additional effort from the developers has to be done to implement log-function-calls.
- Logging events might possibly produce an amount of data that is too large and impracticable for the user to upload on mobile software.
- Logs do not provide higher-level context information (such as video recordings can provide).

Method Name and Parameters	Description
startNewLogFile(platformName)	Tells the EvaHelper framework to create a new log file for a specific platform
setBasicLogInfo(screenName)	Set the screen name for each section of the application
log(componentName, componentType, action)	Log information
log(screenName, componentName, componentType, action)	Log information
logComment(anyLogMessage)	Log information

Table 6.1.: EvaHelper framework methods for logging [from Table 1 BH09, p. 522]

6.2. Frameworks to Collect Logs for Usability Metrics

6.2.1. Manual Instrumentation

By *Manual Instrumentation* approach, it is meant that a basic framework is added that provides the possibility to log events by simply calling a predefined function. Balagtas-Fernandez and Hussmann [BH09] tried to realize the logging functionality for the Android platform in a framework called the Evaluation Helper (EvaHelper) with the five methods from the EvaLogger class that are listed in Table 6.1.

The strength of manual logging is that the complexity of how the logging is achieved is rather low and log information can be obtained where it is intended and needed and, therefore, the amount of gathered log information might be reduced.

Even though this approach is kept rather simple and understandable, it's weakness lies within it's strengths. Due to the necessity to trigger the log information explicitly within the code, there is a chance to miss necessary information because either the call to the log functions is missing or placed at the wrong position or the wrong information might have been collected. These problems might increasingly appear, if the project code is quite large, or in an agile software engineering environment where the project code is intended to be changed rather fast [compare Bat+09; BH09, pp. 46; 523].

6.2.2. Aspect-Oriented Instrumentation

To avoid missing possible log information because the direct call to the corresponding log method in the code was not implemented either at the right position or left out for any reason, Balagtas-Fernandez and Hussmann [BH09, p. 523] and Lettner and Holzmann [LH12] suggest to use an aspect-oriented programming[1] approach for the logging framework.

In short, this approach automatically logs information if certain predefined patterns of methods or classes are called during the program execution. This may work very well if the underlying code deals with interfaces that have (hopefully more and not less) fixed signature. Almost every programming language has such reference points. Some possible points of interest for Android[2] underlying Java-based programming language are listed in Table 6.2.

[1] Aspect-oriented programming: `http://en.wikipedia.org/wiki/Aspect-oriented_programming`, last visit on April 25th, 2013

[2] Android developer API: `http://developer.android.com/reference/packages.html`, last visit on April 25th, 2013

Method Name	**Description**
android.app.Activity.**on*** These Android Activities are one of the main points where user interfaces might be placed in Android. They might be suitable for logging transitions within an Activity (pause, resume) or between Activities. For a better understanding of an Activity's lifecycle, see Figure 6.2	
onCreate()	*"Called when the activity is first created."*[And]
onStart()	*"Called when the activity is becoming visible to the user."*[And]
onResume()	*"Called when the activity will start interacting with the user."*[And]
onPause()	*"Called when the system is about to start resuming a previous activity."*[And]
onStop()	*"Called when the activity is no longer visible to the user."*[And]
onDestroy()	*"The final call you receive before your activity is destroyed."*[And]
android.view.View.OnClickListener The OnLickListener is just one representative for a whole listener family. Listeners are an interface definition for a callback to be invoked when a view is clicked. Views can be buttons or menu entries ...	
onClick(View v)	*"Called when a view has been clicked."*[And]

Table 6.2.: Excerpt of possible aspect-oriented interface points for the Android-, Java-based programming language.

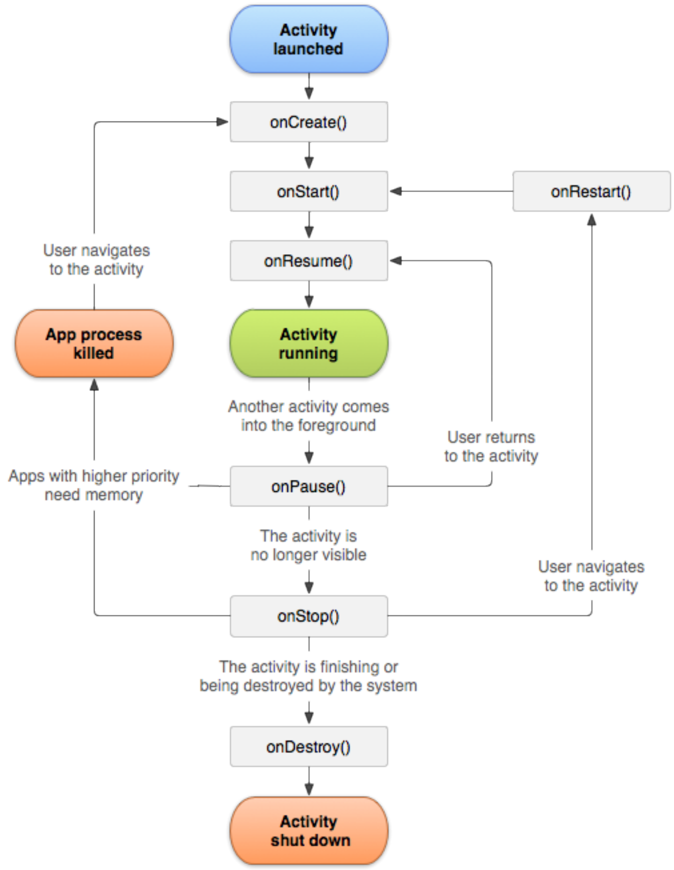


Figure 6.2.: Android-Activity lifecycle [image source And]

This programming approach for usability logs seems certainly better than the manual instrumentation but nevertheless, there could as well be criticism that the aspects have to be defined and programmed and, therefore, there is again a possibility that useful usability log information may be left out.

From the point of view of the manual instrumentation approach, the need to manually code the log methods within the programming code was pointed out as a weakness. In fact, concerning the aspect-oriented approach, this could turn out as a little strength because if there is the need to gather information within the "programmed business logic" and not on the interface basis, it might be hard to obtain the required information [compare Bat+09; BH09].

Hartman and Bass [HB05] tried to realized an aspect-oriented framework named LECAB. LECAB should avoid the high effort of manually instrumenting the code by defining aspects in Java. To apply the framework easily to different applications the aspects were written for a specific GUI toolkit [HB05, p. 823]. Hartman and Bass [HB05, p. 832] compared the success of their aspect-oriented approach to a framework that was manually instrumented and determined the following,

- "*The log messages were more verbose than in the manual instrumentation approach.*"
- "*They missed changes to the state of the application.*"

6.2.3. Interactive Usability Instrumentation

This section describes the *Interactive Usability Instrumentation (IUI)* approach and summarizes the ideas of the research article by Bateman et al. [Bat+09].

Criticism of the previously described aspect-oriented instrumentation concept is that the aspects have to be defined by the developers themselves and, therefore, the decision what information is logged is left to them. Unfortunately, it is most likely that the developers are not the ones that carry out usability tests nor analyze the findings or investigate the logs (unless they are used for bug tracking or system metrics as well). Thus, their understanding of the importance what data shall be collected might differ from the expectations of usability engineers. For those reasons, it might be better to shift the decision of what is logged to the usability engineers.

Bateman et al. [Bat+09] tried to solve the above mentioned problems in their UMARA framework. Their framework is based on the aspect-oriented programming approach, but additionally provides the ability to interactively decide which user interaction points should be logged. This approach tries to avoid the need to instrument the framework completely by a software engineer. Bateman et al. [Bat+09, p. 45] suggest that usability engineers should be able to click directly on the desired user interface elements within the application. For these elements usability engineers have the possibility to provide own names that shall appear in the log. Their UMARA framework also provides [Bat+09, p. 48],

- *"A method for grouping low-level interface actions into higher level activities that are meaningful to the usability test."*
- *"A mechanism for evaluating the previous methods through viewing when and what data will be logged, during the instrumentation process."*

The strengths of only logging the desired aspects are that the amount of collected data is reduced to the required and, therefore, the analysis of the event logs might become easier.

6.3. Analysis of Logs

Before usability logs may by analyzed, the questions is what are UI events of interest?

Hilbert and Redmiles [HR01, p. 391] describe *High Frequency Band Events* as events, that take place within a duration of 10 milliseconds to 1 second. Such events may be button clicks or mouse events. The occurrence of high frequency band events is assumed to happen in a serial order and are the most commonly focused user interaction events. The other type of events are *Low Frequency Band Events* that have a longer duration. According to Hilbert, such an event could be a project event. High- and low-frequency band events may overlap and as a result, the following characterization of events may emerge [HR01, pp. 391, 392],

Synchronous and asynchronous events: Due to the short duration of high-frequency events, they will occur normally in a synchronous way. Low-frequency events may occur asynchronously due to their longer duration. The categorization of the event has to be considered as well as the time of the occurrence of the event when the right "methods used to sample, capture, and analyze data" are chosen.

Composition of events: This means that there can be different levels of abstraction of the log events. Possibly higher-level frequency band events may be abstracted to lower-level frequency bands.

Inference between different frequency band events: How high-frequency band events can be abstracted to low frequency band events, cannot be determined through the logged events. Hilbert and Redmiles [HR01, p. 392], therefore, suggest an external model like a grammar to describe this combination of circumstances.

To investigate the sequence of events and therefore the possible workflow or usage behavior of the user, some suggestions how to represent the event behavior are graphs, state diagrams Markov-chains or a grammar.

However, Hilbert and Redmiles [HR01, pp. 392, 393] found out that if only the sequence of the events is considered, there is a possible chance of misinterpretation. For the grammar description they take as an example a print job:

The direct activation via the toolbar of the print event is described as "A" in the grammar.
An activation indicating that the print job was started from a menu could be "B".
Another possible activation where possibly a print dialog is used as "CD" and so on.

So, from a high-level event point of view, those events may appear as:

AAAA
ABCDA
CDCDAA

Without the consideration of higher level events, this grammar would not reveal that these events all belong to the print job and on the lexical level, it might be difficult to automatically detect, compare or characterize those sequences [compare HR01, pp. 392, 393].

Another aspect that has to be considered is that not every usability event might be meaningful without knowing the surrounding context or might be differently interpreted with the knowledge of some preconditions or a resulting history of other events. When those events are analyzed individually, they might simply provide too little information to be interpreted in a meaningful way, without the knowledge of the context.

Hilbert and Redmiles [HR01, p. 393] give several explanations of this contextual problem. One of those examples is "WINDOW-OPENED ErrorDialog": Without the knowledge of prior events, it might be insufficient to only know that an error dialog appeared. The knowledge of the error dialog event does not uncover in which context the error occurred and, therefore, does not lead to the real problem, what might be the intrinsic information for the usability analyst.

Another decision has to be made on which level of abstraction the user interaction logs will be analyzed. Two common abstraction levels might be on the one hand high-frequency touch events that are analyzed on a low-level time-based scale or on the other hand, a higher level of task-based abstractions. Hilbert and Redmiles [HR01, p. 394] call this *Composition of Events*.

For mobile applications, Lettner and Holzmann [LH12, pp. 123, 124] found the following usability metrics very useful to identify navigational problems. Those metrics shall be understood as one Android-Activity, but some metrics interpreted in the right way can also describe relationships between screens. For example, to describe a workflow of the user while navigating through the application. Such an Android-Activity can simply be described as one screen page where user interaction activities are intended to be grouped together thematically:

Count-based metrics are the effectively counted incidence of "button clicks, session times, screen calls etc." Those metrics may be used for example to identify frequently used user interaction points and those which are not used.

Device statistics such as operating system and hardware information etc. That information may be useful to identify whether a usability problem is related to specific hardware like different screen resolutions might have an impact on the overall contentedness of the application usage.

Combination of count based metrics and device statistics to derive usability-related metrics like hit/miss ratios and, therefore, identify elements in the application that might be misleading like localization problems.

Voluntarily provided user data to compute user satisfaction, loyalty and a customer lifetime value[3]. This data may provide information to define user groups for the application and could be used in conjunction with different filters to provide a better visualization of usability-related issues.

One of the most valuable ways to aid usability engineers to interpret the log files might be an abstraction of the low-level metrics and represent them in a graphical way. What is visualized and in which way strongly depends on the question what was logged and what question in terms of usability has to be answered. Lettner and Holzmann [LH12, p. 124] use color to indicate the different amount of visits per screen, to provide a faster and easier understanding of the data displayed. Such a kind of visualization could look like indicated in Figure 6.3.

Another strength might arise from the possibility to search logs and

[3]CLV: `http://en.wikipedia.org/wiki/Customer_lifetime_value`, last visit on April 25th, 2013

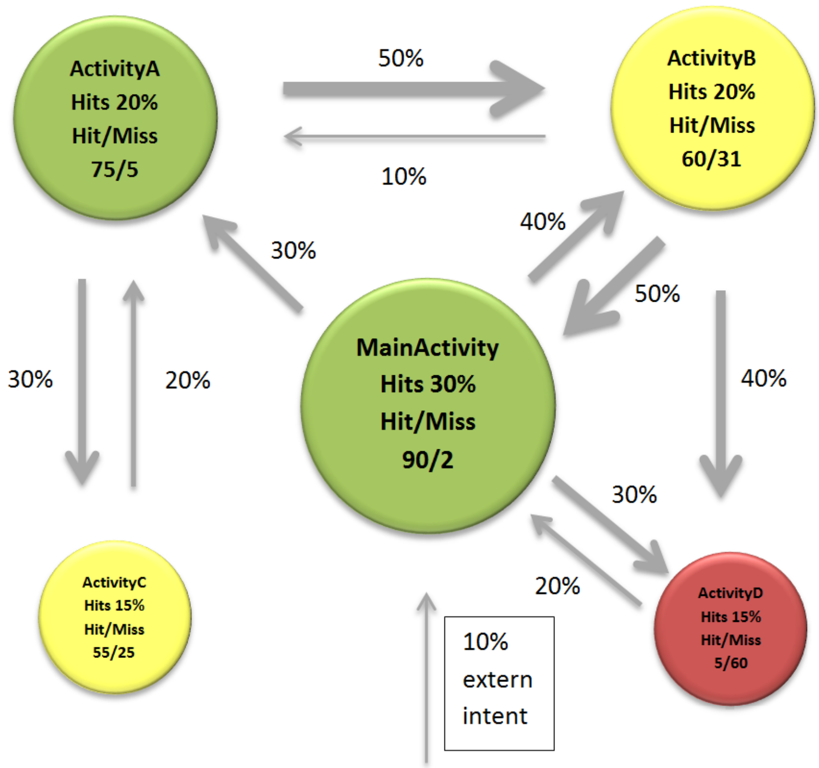


Figure 6.3.: Graphical representation of a fictive application usage. Greater nodes indicate higher usage. The colors shall indicate the hit/miss ratio. Red ... bad (ActivityD); Yellow ... average (ActivityB & C); Green ... good (MainActivity & ActivityA). The gray transition arrows could also use colors to indicate if users immediately returned from an Activity (because they were mislead....) [based on Figure 5 LH12, p. 124]

filter out different usability events or user patterns. This may especially ease the comparison between different user groups as well as the comparison of different periods of usage or whole usability test sessions. The potential to search logs for different usability events may also be powerful in combination with other resources that contribute information about the user interaction, like a video recording from a usability test session. The search and filter function from the log could be used to determine when certain usability events occurred and the log timestamps can further be used to quickly navigate to the desired scenes in the video recordings [compare Bat+09, p. 47].

The biggest drawback of a log is the lack of explicit user feedback. For example, the *Think-Aloud* usability test method (see Chapter 5.2.1) tries to encourage the testing subjects to continuously verbalize their feelings and thoughts about their current interaction with the application. Parts of this supplementary contextual information may be tried to be collected with previously defined questionnaires that might have a similarity to the *Post-Task Interview* method (see Chapter 5.2.3).

6.4. Summary

Usability logs might be interesting to achieve usability-related information with mobile devices because:

- Logs may be used to observe the users in the field.
- The small screen size of mobile devices could make it difficult to record the user interaction and, therefore, a later reproduction of the error for a detailed analysis could be difficult.
- Logs are objective.
- Logs do not influence the user during the application use.

The fact that usability logs do not influence the user might be of particular interest when children are involved in a usability study because they cannot be distracted by an administrator and vice versa, the administrator does not affect the children in their behavior. Since young children forget to verbalize their feelings and thoughts and the administrators should not continuously prompt children to verbalize them, because they could feel uncomfortable about that and abort the test session, logs may still provide information about usability events and the workflow.

How the log behavior is instrumented is of great importance because it directly affects which low level metrics can be collected and furthermore, which information can be filtered out from a log.

Nevertheless, the full strength of log analysis will be uncovered with another input from a usability study when that information can be assembled with the log information. If such synchronization can be achieved, like event timestamps from the log can be used to easily find usability issues in a video recording, then both techniques can benefit from each other.

7. Conclusion

It is obvious that children are not young adults and psychological studies have determined different primary stages children pass through. Two abilities that may be of special interest when it comes to usability testing with children might be the different stages of cognitive and motor capabilities. Those stages might have a direct impact on how to design proper software for young children and when children are intended to participate in a usability test. The cognitive abilities may have an direct impact on the possible complexity of the software and the motor abilities may have an direct impact on the possible size of buttons or other interaction points.

Other specific information about children might also be that children of different ages have different fears and needs not only regarding software products but, moreover, in a social context that may lead to different approaches how software is discovered or used.

Those differences lead to multiple different requirements of the software and the need to know at least the significant user groups or *Persona* (Chapter 3.3.2) that will use the software product early in the design and development process. Otherwise, there is a great chance that the software requirements will not meet the child's requirements.

An approach that tries to satisfy the user needs is *User-Centered Design* (UCD, see Chapter 3.3). UCD tries to involve real users as much as possible in early design and evaluation processes. The user can therefore participate in different roles like a design partner or usability tester.

It would be convenient having a permanent representative of a user present in early development stages. Especially when children are involved their input can be very valuable for the product. However, having a permanent representative is difficult. For example, development offices are not co-located and, therefore, it would be unrealistic that a onsite customer or user is always present.

The Persona method is a powerful tool to define different fictive users that can be used for various departments. Those Personae may not only be used to keep the user present in everyone's mind, but it may as well be used to do an estimation which features are most important to implement in the next iteration circle [PG03]. One factor of success for an effective use of the Persona that could be determined is the need that they have to be used in as many processes as possible. It is not sufficient to announce the outcome of a Persona research once. They have to be present in everyone's mind. If it is spoken about a Persona, everyone should immediately have a person's image in mind without further explanation of the person's attributes. To provide a realistic Persona, a well-grounded research is mandatory. Finally, for a child Persona there need to be some adaptions made, like, *goals*, should be replaced by *needs* and the *user tasks* should focus on *expectations* [Dru02].

One of the state of the art techniques to obtain user feedback is usability testing, and the most commonly used method is *Think-Aloud*.

For children and especially young children, doing a usability test and simultaneously provide a verbalized feedback about their feelings, thoughts or a description about the current progress of their desired work steps is challenging. This behavior needs to divide or totally shift the cognitive load from one task, work with the application, to another, verbalization. This can lead to problems because either the workflow is interrupted in a way that the normal use of the application is discontinued or verbalization is completely left out [Kes+03]. Even though the *Problem Identification Picture Cards* method (Chapter 5.3.1) provides an alternative and simpler way to indicate a usability problem, it still has the disadvantage that the chosen picture has to be interpreted within the context of the problem [BBB08]. This, therefore, can still lead to misinterpretation.

A shot summary of possible drawbacks of currently usability test methods are [BBB08; LE11; Mar08]:

- The already limited attention span of children may be further reduced because of the higher cognitive demands and a consequential narrower margin of concentration.
- An unintended bias could arise because of the unusual "stop and go" usage of the application.
- If children forget about the verbalization and they are continuously prompted to verbalize, this might result in disaffection of the children or false reporting of usability issues to please the facilitator.

To avoid the above mentioned problems, usability log files (Chapter 6 may provide an unobtrusive way of observing the behavior of children with an application. The largest strengths of usability logs when children do the testing are:

- Logs are objective, without any bias and there is no need to interrupt children to please a testing method (e.g., prompt them to verbalize).
- Logs do not need a facilitator that could possibly have an influence on the usability study, either that children want to please the facilitators or that they are unintendedly influenced by an unknown person.
- Logs do not have to restrict the test session to a fixed time slot nor a fixed location. Children can do the test whenever and wherever they want. Mobile applications and mobile devices are not intended to be statically bound to a fixed location and, therefore, a proper usability lab setup might be difficult to achieve as well.

The weaknesses of a usability test study can be compensated by a usability log and vice versa. Thus they seem to complement each other very well. Another benefit of the parallel use is the possibility to synchronize them and furthermore have the ability to filter and group possible events of interest for a faster investigation of the severity of the problem as well as to compare possible improvements of different design iterations [Bat+09].

Nevertheless, to obtain a proper usability log file that may be analyzed, one should keep in mind that an appropriate instrumentation has to be chosen as well as enough contextual information has to be gathered to abstract low-level metrics and identify high-level user patterns [Bat+09].

Further research could be done if the built-in front camera of mobile devices could be used for video recordings. For example, almost every

current tablet computer[1] has a built-in webcam and about 58%[2] of the current available Android[3] mobile phones have a built-in front camera. Even if this camera has a limited recording quality, it could still be used to record the children while testing the application. This could be especially valuable if the test session is not conducted in a usability laboratory where gathering contextual information is difficult.
In any case, it is certain that the participating children and their parents have to be informed explicitly about the video recording.

Another idea could be to take screenshots from the application. To provide a visual history of the user interaction screenshots may be triggered by specific user interaction events.
Besides screenshots, the logging of touch coordinates might help to reproduce the usage behavior. These touch-events could possibly also be used to do a play-back of the test session. Furthermore, touch coordinates could provide useful information whether certain elements (e.g., buttons, list-elements...) have a proper size for children. For example, if a lot of touch-events occur around a button the assumption might be that the button is too small.

The excessive logging of events will only make sense if the gathered data can be analyzed with "Log-Analyzation Tool".

Furthermore, it should be investigated whether gaze or eye-tracking could be done with the help of the built-in camera of mobile devices and whether there are solutions available that work in field-tests as well.

[1] As a current reference, the website from `http://goo.gl/Onb9Y` was used and 884 from 884 tablet PCs had a built-in webcam; last visit on April 17th, 2013

[2] As a current reference, the website from `http://goo.gl/wgctI` was used and 391 from 672 smart phones had a front camera; last visit on April 17th, 2013

[3] `http://www.android.com`, last visit on April 17th, 2013

Finally, an agile software engineering method (see Chapter 3.4) is the most practical when it comes to developing applications for children.

Agile stands for short development iterations and within these short cycles, the most valuable tickets for the user are taken into account first. Whether these tickets are really valuable for the user or not, could possibly be determined by analyzing the log statistics. At least statistics could give an insight if new features are used or the usage of improved features rises. To ensure if certain goals are met, Sauro and Lewis [SL12] suggest to compare test tasks against benchmarks. For more information about comparing usability outcome against benchmarks a good starting point could be Chapter 4, *"Did We Meet or Exceed Our Goal?"*, from Sauro and Lewis [SL12]. Nevertheless, short iterations of an agile development cycle, provide a flexibility to reconsider the already existing design and determine the right design improvements and new features for the upcoming iteration. Thus, it is possible to integrate previously unknown information really fast into the development process. The outcome of a usability test may provide the basis for the considerations of the next iteration planning. This re-evaluation process is recommended as well for the Persona and all other parts of the engineering process. These self-evaluation processes are essential, especially if new information is available, for example, a new study which reveals information that is relevant to one of the Persona.

List of Figures

List of Tables

Appendix

Appendix A.

SRCD Ethical Standards for Research with Children

Updated by the SRCD Governing Council, March 2007
The principles listed below were published in the 1990-91 Directory, except for Principles 15 and 16, first published in the Fall 1991 Newsletter.

Principle 1. NON-HARMFUL PROCEDURES: The investigator should use no research procedure that may harm the child either physically or psychologically. The investigator is also obligated at all times to use the least stressful research procedure whenever possible. Psychological harm in particular instances may be difficult to define; nevertheless, its definition and means for reducing or eliminating it remain the responsibility of the investigator. When the investigator is in doubt about the possible harmful effects of the research procedures, consultation should be sought from others. When harm seems inevitable, the investigator is obligated to find other means of obtaining the information or to abandon the research. Instances may, nevertheless, rise in which exposing the child to stressful conditions may be necessary if diagnostic or therapeutic benefits to the child are associated with the research.

In such instances careful deliberation by an Institutional Review Board should be sought.

Principle 2. INFORMED CONSENT: Before seeking consent or assent from the child, the investigator should inform the child of all features of the research that may affect his or her willingness to participate and should answer the child's questions in terms appropriate to the child's comprehension. The investigator should respect the child's freedom to choose to participate in the research or not by giving the child the opportunity to give or not give assent to participation as well as to choose to discontinue participation at any time. Assent means that the child shows some form of agreement to participate without necessarily comprehending the full significance of the research necessary to give informed consent. Investigators working with infants should take special effort to explain the research procedures to the parents and be especially sensitive to any indicators of discomfort in the infant. In spite of the paramount importance of obtaining consent, instances can arise in which consent or any kind of contact with the participant would make the research impossible to carry out. Non-intrusive field research is a common example. Conceivably, such research can be carried out ethically if it is conducted in public places, participants' anonymity is totally protected, and there are no foreseeable negative consequences to the participant. However, judgments on whether such research is ethical in particular circumstances should be made in consultation with an Institutional Review Board.

Principle 3. PARENTAL CONSENT: The informed consent of parents, legal guardians or those who act in loco parentis (e.g., teachers, superintendents of institutions) similarly should be obtained, preferably in writing. Informed consent requires that parents or other responsible adults be informed of all the features of the research that may affect their willingness to allow the child to participate. This information should include the profession and institution affiliation of the investigator. Not only should the right of the responsible adults to refuse consent be respected, but also they should be informed that they may refuse to participate without incurring any penalty to them or to the child.

Principle 4. ADDITIONAL CONSENT: The informed consent of any persons, such as schoolteachers for example, whose interaction with the child is the subject of the study should also be obtained. As with the child and parents or guardians informed consent requires that the persons interacting with the child during the study be informed of all features of the research which may affect their willingness to participate. All questions posed by such persons should be answered and the persons should be free to choose to participate or not, and to discontinue participation at any time.

Principle 5. INCENTIVES: Incentives to participate in a research project must be fair and must not unduly exceed the range of incentives that the child normally experiences. Whatever incentives are used, the investigator should always keep in mind that the greater the possible effects of the investigation on the child, the greater is the obligation to protect the child's welfare and freedom.

Principle 6. DECEPTION: Although full disclosure of information during the procedure of obtaining consent is the ethical ideal, a particular study may necessitate withholding certain information or deception. Whenever withholding information or deception is judged to be essential to the conduct of the study, the investigator should satisfy research colleagues that such judgment is correct. If withholding information or deception is practiced, and there is reason to believe that the research participants will be negatively affected by it, adequate measures should be taken after the study to ensure the participant's understanding of the reasons for the deception. Investigators whose research is dependent upon deception should make an effort to employ deception methods that have no known negative effects on the child or the child's family.

Principle 7. ANONYMITY: To gain access to institutional records, the investigator should obtain permission from responsible authorities in charge of records. Anonymity of the information should be preserved and no information used other than that for which permission was obtained. It is the investigator's responsibility to ensure that responsible authorities do, in fact, have the confidence of the participant and that they bear some degree of responsibility in giving such permission. In complying with requirements for data sharing, researchers need to carefully consider whether they have provided data which, if combined, risks violating participant anonymity.

Principle 8. MUTUAL RESPONSIBILITIES: From the beginning of each research investigation, there should be clear agreement between the investigator and the parents, guardians or those who act in loco parentis, and the child, when appropriate, that defines the responsibilities of each. The investigator has the obligation to honor all promises and commitments of the agreement.

Principle 9: JEOPARDY: When, in the course of research, information comes to the investigator's attention that may jeopardize the child's well-being, the investigator has a responsibility to discuss the information with the parents or guardians and with those expert in the field in order that they may arrange the necessary assistance for the child. Researchers need to be aware that they may obtain findings suggesting that a child's health and well-being might be in jeopardy, that these findings may include false positives, and they should be knowledgeable about current human subjects procedures and regulations for informing families of incidental findings.

Principle 10. UNFORESEEN CONSEQUENCES: When research procedures result in undesirable consequences for the participant that were previously unforeseen, the investigator should immediately employ appropriate measures to correct these consequences, and should redesign the procedures if they are to be included in subsequent studies.

Principle 11. CONFIDENTIALITY: The investigator should keep in confidence all information obtained about research participants. The participants' identity should be concealed in written and verbal reports of the results, as well as in informal discussion with students and colleagues. When a possibility exists that others may gain access to such information, this possibility, together with the plans for protecting confidentiality, should be explained to the participants as part of the procedure of obtaining informed consent.

Principle 12. INFORMING PARTICIPANTS: Immediately after the data are collected, the investigator should clarify for the research participant any misconceptions that may have arisen. The investigator also recognizes a duty to report general findings to participants in terms appropriate to their understanding. Where scientific or humane values justify withholding information, every effort should be made so that withholding the information has no damaging consequences for the participant.

Principle 13. REPORTING RESULTS: Because the investigator's words may carry unintended weight with parents and children, caution should be exercised in reporting results, making evaluative statements, or giving advice.

Principle 14. IMPLICATIONS OF FINDINGS: Investigators should be mindful of the social, political and human implications of their research and should be especially careful in the presentation of findings from the research. This principle, however, in no way denies investigators the right to pursue any area of research or the right to observe proper standards of scientific reporting.

Principle 15. SCIENTIFIC MISCONDUCT: Misconduct is defined as the fabrication or falsification of data, plagiarism, misrepresentation, or other practices that seriously deviate from those that are commonly accepted within the scientific community for proposing, conducting, analyzing, or reporting research. It does not include unintentional errors or honest differences in interpretation of data. The Society shall provide vigorous leadership in the pursuit of scientific investigation that is based on the integrity of the investigator and the honesty of research and will not tolerate the presence of scientific misconduct among its members. It shall be the responsibility of the voting members of Governing Council to reach a decision about the possible expulsion of members found guilty of scientific misconduct.

Principle 16. PERSONAL MISCONDUCT: Personal misconduct that results in a criminal conviction of a felony may be sufficient grounds for a member's expulsion from the Society. The relevance of the crime to the purposes of the Society should be considered by the Governing Council in reaching a decision about the matter. It shall be the responsibility of the voting members of Governing Council to reach a decision about the possible expulsion of members found guilty of personal misconduct.

Bibliography

[And] Google Android. *Android Activity Lifecycle*. URL: http://developer.android.com/reference/android/app/Activity.html#ActivityLifecycle (visited on 04/08/2013).

[And12] Keith Andrews. *Human Computer Interaction*. Lecture Notes. Aug. 2012.

[Ant06] Alissa Nicole Antle. "Child-personas: fact or fiction?" In: *Proceedings of the 6th conference on Designing Interactive systems*. DIS '06. University Park, PA, USA: ACM, 2006, pp. 22–30. ISBN: 1-59593-367-0. DOI: 10.1145/1142405.1142411. URL: http://doi.acm.org/10.1145/1142405.1142411.

[Bar+07] Wolmet Barendregt et al. "Predicting effectiveness of children participants in user testing based on personality characteristics." In: *Behaviour & Information Technology* 26.2 (2007), pp. 133–147. DOI: 10.1080/01449290500330372. eprint: http://www.tandfonline.com/doi/pdf/10.1080/01449290500330372. URL: http://www.tandfonline.com/doi/abs/10.1080/01449290500330372.

[Bat+09] Scott Bateman et al. "Interactive usability instrumentation." In: *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems*. EICS '09. Pittsburgh, PA, USA: ACM, 2009, pp. 45–54. ISBN: 978-1-60558-600-7. DOI: 10.1145/1570433.1570443. URL: http://doi.acm.org/10.1145/1570433.1570443.

[Bau03] Miki Baumgarten. "Kids and the internet: a developmental summary." In: *Comput. Entertain.* 1.1 (Oct. 2003). ISSN: 1544-3574. DOI: 10.1145/950566.950584. URL: http://doi.acm.org/10.1145/950566.950584.

[BB] W. Barendregt and M.M. Bekker. "Guidelines for user testing with children." URL: http://w3.id.tue.nl/fileadmin/id/objects/doc/Guidelines_for_user_testing_with_children.pdf.

[BBB08] Wolmet Barendregt, MathildeM. Bekker, and Ester Baauw. "Development and evaluation of the problem identification picture cards method." English. In: *Cognition, Technology & Work* 10 (2 2008), pp. 95–105. ISSN: 1435-5558. DOI: 10.1007/s10111-007-0066-z. URL: http://dx.doi.org/10.1007/s10111-007-0066-z.

[BH09] Florence Balagtas-Fernandez and Heinrich Hussmann. "A Methodology and Framework to Simplify Usability Analysis of Mobile Applications." In: *Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering*. ASE '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 520–524. ISBN: 978-0-7695-3891-4. DOI: 10.1109/ASE.2009.12. URL: http://dx.doi.org/10.1109/ASE.2009.12.

[Bro13] Diana DeMarco Brown. *Agile user experience design; a practitioner's guide to making it work*. 1. ed. ID: AC09586619. San Francisco, Calif.: Morgan Kaufmann, 2013, XIII, 176 S. ISBN: 978-0-12-415953-2; 0-12-415953-2. URL: http://permalink.obvsg.at/AC09586619.

[CB12] Andy Clyce and Shiloh Barnat. *Mobile usability testing: open source evolution call-to-action*. 2012. URL: http://www.lokion.com/nimble/mobile-usability-testing-open-source-evolution-call-to-action/ (visited on 04/17/2013).

[CP11] Joana Campos and Ana Paiva. "A personal approach: the persona technique in a companion's design lifecycle." In: *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part III*. INTERACT'11. Lisbon, Portugal: Springer-Verlag, 2011, pp. 73–90. ISBN: 978-3-642-23764-5. URL: http://dl.acm.org/citation.cfm?id=2042182.2042190.

[DR04] Afke Donker and Pieter Reitsma. "Usability testing with young children." In: *Proceedings of the 2004 conference on Interaction design and children: building a community*. IDC '04. Maryland: ACM, 2004, pp. 43–48. ISBN: 1-58113-791-5. DOI: 10.1145/1017833.1017839. URL: http://doi.acm.org/10.1145/1017833.1017839.

[Dru02] Allison Druin. "The Role of Children in the Design of New Technology." In: (2002). URL: http://www.cs.umd.edu/hcil/kiddesign/design_process.shtml.

[DZN07] Markus Düchting, Dirk Zimmermann, and Karsten Nebe. "Incorporating User Centered Requirement Engineering into Agile Software Development." In: *Human-Computer Interaction. Interaction Design and Usability*. Ed. by JulieA. Jacko. Vol. 4550. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, pp. 58–67. ISBN: 978-3-540-73104-7. DOI: 10.1007/978-3-540-73105-4_7. URL: http://dx.doi.org/10.1007/978-3-540-73105-4_7.

[enc12a] Wikipedia: The free encyclopedia. *Child development stages*. Oct. 2012. URL: http://en.wikipedia.org/wiki/Child_development_stages (visited on 10/24/2012).

[enc12b] Wikipedia: The free encyclopedia. *Jean Piaget*. Oct. 2012. URL: http://en.wikipedia.org/wiki/Jean_Piaget (visited on 11/08/2012).

[enc12c] Wikipedia: The free encyclopedia. *Jean Piaget*. Nov. 2012. URL: http://de.wikipedia.org/wiki/Jean_Piaget (visited on 11/08/2012).

[enc13] Wikipedia: The free encyclopedia. *Piaget's theory of cognitive development*. Jan. 2013. URL: http://en.wikipedia.org/wiki/Piaget's_theory_of_cognitive_development (visited on 01/16/2013).

[FF11] Shamal Faily and Ivan Flechais. "Persona cases: a technique for grounding personas." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. Vancouver, BC, Canada: ACM, 2011, pp. 2267–2270. ISBN: 978-1-4503-0228-9. DOI: 10.1145/1978942.1979274. URL: http://doi.acm.org/10.1145/1978942.1979274.

[Gro] Nielsen Norman Group. *User Experience - Our Definition*. URL: http://www.nngroup.com/about-user-experience-definition/ (visited on 02/22/2013).

[Gro12] Nielsen Norman Group. *Usability 101: Introduction to Usability. What — Definition of Usability*. 2012. URL: http://www.nngroup.com/articles/usability-101-introduction-to-usability/ (visited on 02/22/2013).

[HB05] Gregory S. Hartman and Len Bass. "Logging events crossing architectural boundaries." In: *Proceedings of the 2005 IFIP TC13 international conference on Human-Computer Interaction*. INTERACT'05. Rome, Italy: Springer-Verlag, 2005, pp. 823–834. ISBN: 3-540-28943-7, 978-3-540-28943-2. DOI: 10.1007/11555261_65. URL: http://dx.doi.org/10.1007/11555261_65.

[HR01] David M. Hilbert and David F. Redmiles. "Large-Scale Collection of Usage Data to Inform Design." In: *Conference on Human-Computer Interaction*. IFIP TC 13 8 (July 2001), pp. 569–576. URL: http://www.ics.uci.edu/~redmiles/publications.html.

[HRA97] Libby Hanna, Kirsten Risden, and Kirsten Alexander. "Guidelines for usability testing with children." In: *interactions* 4.5 (Sept. 1997), pp. 9–14. ISSN: 1072-5520. DOI: 10.1145/264044.264045. URL: http://doi.acm.org/10.1145/264044.264045.

[Int10] International Organization for Standardization. *Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems*. Norm. 2010.

[Int98] International Organization for Standardization. *Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability*. Norm. 1998.

[JM06] Matt Jones and Gary Marsden. *Mobile interaction design*. ID: AC04987960. Chichester: Wiley, 2006, XX, 377 S. ISBN: 0-470-09089-8; 978-0-470-09089-3. URL: http://permalink.obvsg.at/AC04987960.

[Kai+05] Anne Kaikkonen et al. "Usability Testing of Mobile Applications: A Comparison between Laboratory and Field Testing." In: (2005). URL: http://www.upassoc.org/upa_publications/jus/2005_november/mobile.html.

[Kes+03] Ilse E. H. van Kesteren et al. "Assessing usability evaluation methods on their effectiveness to elicit verbal comments from children subjects." In: *Proceedings of the 2003 conference on Interaction design and children*. IDC '03. Preston, England: ACM, 2003, pp. 41–49. ISBN: 1-58113-732-X. DOI: 10.1145/953536.953544. URL: http://doi.acm.org/10.1145/953536.953544.

[Kna12] Ferdinand R. Knapitsch-Scarpatetti-Unterwegen. *Usability testing of mobile applications for children*. ID: AC10673757; Nebent. Usability Testen von mobilen Anwendungen für Kinder; Graz, Techn. Univ., Master-Arb., 2012. 2012, XII, 113 Bl. URL: http://permalink.obvsg.at/AC10673757.

[LE11] Janine Liebal and Markus Exner. *Usability für Kids; ein Handbuch zur ergonomischen Gestaltung von Software und Websites für Kinder*. 1. Aufl. ID: AC08769712. Wiesbaden: Vieweg + Teubner, 2011, XIV, 243 S. ISBN: 978-3-8348-1624-5; 3-8348-1624-8. DOI: 10.1007/978-3-8348-8206-6. URL: http://d-nb.info/1011018209/04;%20http://permalink.obvsg.at/AC08769712.

[LH12] Florian Lettner and Clemens Holzmann. "Automated and unsupervised user interaction logging as basis for usability evaluation of mobile applications." In: *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia*. MoMM '12. Bali, Indonesia: ACM, 2012, pp. 118–127. ISBN: 978-1-4503-1307-0. DOI: 10.1145/2428955.2428983. URL: http://doi.acm.org/10.1145/2428955.2428983.

[Mar+08] Panos Markopoulos et al. "Testing interactive products with the robot intervention method." In: *Proceedings of the 7th international conference on Interaction design and children*. IDC '08. Chicago, Illinois: ACM, 2008, pp. 117–120. ISBN: 978-1-59593-994-4. DOI: 10.1145/1463689.1463735. URL: http://doi.acm.org/10.1145/1463689.1463735.

[Mar08] Panos Markopoulos. *Evaluating children's interactive products; principles and practices for interaction designers*. ID: AC08869444. Amsterdam: Morgan Kaufmann, 2008, p. 1 v. ISBN: 9780123741110; 0123741114. URL: http://www.sciencedirect.com/science/book/9780123741110;%20http://permalink.obvsg.at/AC08869444.

[MB03] Panos Markopoulos and Mathilde Bekker. "Interaction design and children." In: *Interacting with Computers* 15.2 (2003). Interaction Design and Children, pp. 141–149. ISSN: 0953-5438. DOI: 10.1016/S0953-5438(03)00004-3. URL: http://www.sciencedirect.com/science/article/pii/S0953543803000043.

[MCD12] M.A. Mayz, D.M. Curtino, and A. De la Rosa. "Avoiding laboratories to collect usability data: Two software applications." In: *Information Systems and Technologies (CISTI), 2012 7th Iberian Conference on*. 2012, pp. 1–5.

[Meg06] Carolyn Meggitt. *Child Development. An illustrated guide*. 2nd edition. Heinemann Educational Publishers, Jan. 2006. ISBN: 0435420488.

[Nie00] Nielsen Norman Group: Jakob Nielsen. *Why You Only Need to Test with 5 Users*. Mar. 2000. URL: http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/ (visited on 02/06/2013).

[Nie08] Jakob Nielsen. *When to Use Which User Experience Research Methods*. Oct. 2008. URL: http://www.nngroup.com/articles/which-ux-research-methods/ (visited on 03/16/2013).

[Nie13] Lene Nielsen. *Personas - User Focused Design*. ID: AC08985047. 2013. URL: http://dx.doi.org/10.1007/978-1-4471-4084-9;%20http://media.obvsg.at/AC08985047-4001;%20http://media.obvsg.at/AC08985047-1001;%20http://permalink.obvsg.at/AC08985047.

[Nie93] Jakob Nielsen. *Usability Engineering*. AP Professional, 1993. ISBN: 0-12-518406-9.

[Nie94] Jakob Nielsen. "Estimating the number of subjects needed for a thinking aloud test." In: *International Journal of Human-Computer Studies* 41.3 (1994), pp. 385–397. ISSN: 1071-5819. DOI: 10.1006/ijhc.1994.1065. URL: http://www.sciencedirect.com/science/article/pii/S1071581984710652.

[PG03] John Pruitt and Jonathan Grudin. "Personas: practice and theory." In: *Proceedings of the 2003 conference on Designing for user experiences*. DUX '03. San Francisco, California: ACM, 2003, pp. 1–15. ISBN: 1-58113-728-1. DOI: 10.1145/997078.997089. URL: http://doi.acm.org/10.1145/997078.997089.

[Ros99] Peter Rossmann. *Einführung in die Entwicklungspsychologie des Kindes- und Jugendalters*. 2. Nachdruck. Hans Huber, 1999. ISBN: 3456827237.

[SKI09] Ahmed Seffah, Christophe Kolski, and Djilali Idoughi. "Persona comme outil de design de services interactifs: principes et exemple en e-maintenance." In: *Proceedings of the 21st International Conference on Association Francophone d'Interaction Homme-Machine*. IHM '09. Grenoble, France: ACM, 2009, pp. 333–336. ISBN: 978-1-60558-461-4. DOI: 10.1145/1629826.1629880. URL: http://doi.acm.org/10.1145/1629826.1629880.

[SL12] Jeff Sauro and James R. Lewis. *Quantifying the user experience; practical statistics for user research*. ID: AC08938464. Amsterdam u.a.]: Elsevier, 2012, XV, 295 S. ISBN: 978-0-12-384968-7; 0-12-384968-3. URL: http://permalink.obvsg.at/AC08938464.

[SRC07] Society for Research in Child Development. *SRCD Ethical Standards for Research with Children*. Mar. 2007. URL: http://www.srcd.org/index.php?option=com_content&task=view&id=68 (visited on 10/24/2012).

[TA08] Tom Tullis and Bill Albert. *Measuring the user experience; collecting, analyzing, and presenting usability metrics*. ID: AC06753450. Amsterdam u.a.]: Elsevier, 2008, XVII, 317 S. ISBN: 0-12-373558-0; 978-0-12-373558-4. URL: http://media.obvsg.at/AC06753450-1001;%20http://permalink.obvsg.at/AC06753450.

[Tex] Martin R. Textor. *Jean Piaget*. URL: http://www.kindergartenpaedagogik.de/1226.html (visited on 11/08/2012).

[Vir92] Robert A. Virzi. "Refining the test phase of usability evaluation: how many subjects is enough?" In: *Hum. Factors* 34.4 (Aug. 1992), pp. 457–468. ISSN: 0018-7208. URL: http://dl.acm.org/citation.cfm?id=141691.141700.

[Wat] Paul Watzlawick. *Die Axiome von Paul Watzlawick*. URL: http://www.paulwatzlawick.de/axiome.html (visited on 03/06/2013).