

NFC Home Control

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering & Internet Computing

eingereicht von

Andreas Bretschneider

Matrikelnummer 0327444

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Assoc. Prof. Dipl. Ing. Dr. Hilda Tellioğlu

Wien, 03.12.2013

(Unterschrift Verfasser)

(Unterschrift Betreuung)

NFC Home Control

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering & Internet Computing

by

Andreas Bretschneider

Registration Number 0327444

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Assoc. Prof. Dipl. Ing. Dr. Hilda Telliöglu

Vienna, 03.12.2013

(Signature of Author)

(Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Andreas Bretschneider
Maria vom Siege 2/9, 1150 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)

(Unterschrift Verfasser)

Danksagung

Ich möchte hiermit meiner großen Liebe danken, die mich während der Erstellung dieser Arbeit mit Tipps zum Verfassen wissenschaftlicher Arbeiten unterstützte. Besonderer Dank gilt auch ihrem Vater. Er half mir tatkräftig und ohne jegliche Verzögerung beim Korrekturlesen dieser Arbeit. Außergewöhnlicher Dank gilt Masahiro Sakurai, dem indirekten Namensgeber unseres Hundes Kirby, der mich stets aufmunterte wenn ich müde wurde. Ebenso möchte ich mich bei dem göttlichen Zufall bedanken, der diesem Planeten Leben einhauchte und mit seinen physikalischen Gesetzen das möglich machte, was für uns alltäglich ist und dennoch an ein Wunder gleicht.

Der größte Dank gilt jedoch meiner Diplomarbeitsbetreuerin Hilda Tellioglu. Ohne sie wäre die Erstellung dieser Arbeit nicht möglich gewesen.

Abstract

Conventional interaction mechanisms for remote control of devices in the household are limited to pressing buttons on a device or using a touchscreen. The necessary actions require an increased amount of concentration and attention. The aim of this work is to design, implement and evaluate an activity-based remote control which additionally enables interaction by reading NFC tags with an Android smartphone. The interaction shall be implemented as user-friendly as possible in order to be carried out with a single movement of the hand. Through the determination of places at which specific actions are usually performed and the placement of NFC tags, the user is offered an additional opportunity to interact. The whole process has been supported through user-centered methods to clarify the question of user acceptance and the impact on usability. By applying an iterative design process it has been possible to react to problems and new research questions that arose. The developed solution has been evaluated at the end of each iteration by means of user tests. In order to provide an optimal basis for carrying out the tests, a variety of devices has been integrated into the remote control system. The focus of the evaluations has been on learnability, user satisfaction and the potential added value for the users. To clarify the questions, both unstructured observations and interviews, as well as usability metrics were utilized. Depending on the situation and a meaningful placement, the interaction with the NFC tags has proven to be extremely useful. During this work possible extensions of the software for remote control and points for improvement of the currently prevailing technical implementation of NFC readers in smartphones have been identified.

Kurzfassung

Herkömmliche Interaktionsmechanismen zur Steuerung und Fernbedienung von Geräten im Haushalt beschränken sich auf das Drücken von Tasten oder die Bedienung durch einen Touchscreen. Die notwendigen Handlungen erfordern ein erhöhtes Maß an Konzentration und Aufmerksamkeit. Das Ziel dieser Arbeit ist die Entwicklung und Evaluierung einer aktivitätsbasierten Universalfernbedienung, welche zusätzlich die Steuerung durch das Einlesen von NFC-Tags mit einem Android Smartphone ermöglicht. Die Interaktion soll dabei möglichst benutzerfreundlichen umgesetzt werden, um mit einem einzigen Handgriff durchgeführt werden zu können. Durch die Bestimmung von Orten an denen bestimmte Aktionen üblicherweise durchgeführt werden und der Platzierung von NFC-Tags, wird der BenutzerIn dadurch eine zusätzliche Möglichkeit zur Bedienung geboten. Der gesamte Prozess wurde durch benutzerzentrierte Methoden gestützt, um die Frage nach der Benutzerakzeptanz und den Auswirkungen auf die Usability zu klären. Durch die Anwendung eines iterativen Entwicklungsprozesses konnte auf angefallene Probleme und neue Forschungsfragen reagiert werden. Am Ende jeder Iteration stand jeweils die Evaluierung der entwickelten Lösung mit Hilfe von Benutzertests. Um eine optimale Grundlage zur Durchführung der Tests zu bieten, wurde eine Vielzahl von Geräten in die Fernbedienung integriert. Bei der Evaluierung standen die Fragen nach der Erlernbarkeit, der Benutzerzufriedenheit und dem möglichen Mehrwert im Vordergrund. Zur Klärung der Fragen kamen sowohl unstrukturierte Beobachtungen und Interviews als auch Usability Metriken zum Einsatz. In Abhängigkeit von der jeweiligen Situation und einer sinnvollen Platzierung erwies sich die Interaktion mit den NFC-Tags als äußerst nützlich. Während dieser Arbeit wurden mögliche Erweiterungen der Software zur Fernbedienung sowie verbesserungswürdige Punkte der momentan vorherrschenden technischen Umsetzung von NFC Lesegeräten in Smartphones identifiziert.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Fragestellung	2
1.3	Aufbau dieser Arbeit	3
2	Technologien	5
2.1	Smartphones	6
2.2	Near-Field-Communication (NFC)	9
2.3	Mikrocontroller	12
2.4	Fernbedienung im Haushalt	17
2.5	Heimautomation	19
2.6	Ubiquitous-Computing	23
2.7	Tangible User-Interfaces	25
2.8	Referenzprojekte	26
2.8.1	Umgebungssteuerung mit NFC und KNX	26
2.8.2	SiteView	28
3	Methoden	31
3.1	Iterative Entwicklung	31
3.2	User-Centered Design (UCD)	33
3.3	Personas	35
3.4	Szenarios	36
3.5	Use-Cases	37
3.6	Interviews	38
3.7	Beobachtungen	40
3.8	Diary studies	40
3.9	Usability	42
3.9.1	Affordance	43
3.9.2	Metapher	44
3.9.3	Usability Metriken	45
3.10	Technology Adoption Lifecycle	48
4	Entwicklung	51

4.1	Der Entwicklungsprozess	51
4.2	Erste Anforderungsphase	51
	4.2.1 Anforderungen an die Interaktion mit NFC-Tags	52
	4.2.2 Rollen und Ziele	52
	4.2.3 Personas	53
	4.2.4 Szenarios	56
4.3	Erste Entwicklungsphase	57
	4.3.1 Android Smartphone	57
	4.3.2 Beleuchtung	58
	4.3.3 Fenster	61
	4.3.4 Kaffeemaschine	65
	4.3.5 NFC-Tags	67
	4.3.6 Unterhaltungselektronik	68
	4.3.7 Android Universalfernbedienung reYmote	68
4.4	Erste Testphase	80
	4.4.1 Ziele	80
	4.4.2 Testpersonen	81
	4.4.3 Vorgehensweise	81
	4.4.4 Testfälle	82
4.5	Erste Evaluierungsphase	83
	4.5.1 Benutzertests	83
	4.5.2 SUS	86
	4.5.3 MUSiC	87
	4.5.4 Analyse und Zielsetzung	87
4.6	Zweite Anforderungsphase	89
	4.6.1 Gestaltung des Benutzertests	89
	4.6.2 Ausführung von Kommandos	89
	4.6.3 NFC Lesebereich	89
	4.6.4 Kontextsensitive Steuerung	89
	4.6.5 Konsistente Grafiken	90
	4.6.6 HTTP Probleme	90
	4.6.7 Logging	90
4.7	Zweite Entwicklungsphase	90
	4.7.1 Ausführung von Kommandos	90
	4.7.2 NFC Lesebereich	91
	4.7.3 Kontextsensitive Steuerung	94
	4.7.4 Konsistente Grafiken	96
	4.7.5 HTTP Probleme	97
	4.7.6 Logging	97
4.8	Zweite Testphase	98
	4.8.1 Ziele	98
	4.8.2 Testpersonen	98
	4.8.3 Vorgehensweise	98

4.9	Zweite Evaluierungsphase	98
4.9.1	Benutzertest	99
4.9.2	Analyse	101
5	Analyse	103
5.1	Entwicklung	103
5.2	Fazit	109
5.3	Weiterführende Entwicklungen	110
5.3.1	Mehrbenutzerbetrieb durch Synchronisierung	111
5.3.2	Erleichterung der Konfiguration	111
5.3.3	Gerätedatenbanken	111
5.3.4	Android NFC im Standby	111
5.3.5	Positionierung der NFC Antenne	112
5.3.6	Wearable NFC Lesegerät	112
6	Zusammenfassung	113
	Literaturverzeichnis	115
	Webseiten	123
	Abbildungsverzeichnis	127
	Tabellenverzeichnis	129

Einleitung

Dieses Kapitel gibt einen groben Überblick über die Arbeit. Die **Motivation** beschreibt die momentane Situation der Wege zur Heimsteuerung in Kürze und rechtfertigt die Erstellung und Durchführung dieser Arbeit. Außerdem werden in diesem Abschnitt auch die grundlegenden Gedanken, die zur Erstellung dieser Arbeit führten, erläutert und ein grober Überblick über die angestrebten Entwicklungen gegeben. Der Abschnitt **Fragestellung** gibt einen tiefen Einblick in die gestellten Forschungsfragen. Abschließend liefert ein weiteres Unterkapitel Informationen über den **Aufbau dieser Arbeit**.

1.1 Motivation

Eine Studie von Logitech aus dem Jahr 2010 ergab, dass der mehr als die Hälfte der Haushalte in Frankreich, Australien und dem Vereinigten Königreich vier oder mehr Geräte im Wohnzimmer beherbergen, die per Fernbedienung bedient werden müssen. Diesen Umstand haben sich Hersteller von Universalfernbedienungen, wie etwa auch Logitech selbst, zunutze gemacht und eigene Produkte zur Vereinfachung der Fernsteuerung entwickelt. Teurere Geräte sind bereits mit einem Farbbildschirm oder sogar einem Touchscreen ausgestattet [91]. Billigere Geräte sehen nahezu wie herkömmliche Fernbedienungen aus.

Universalfernbedienungen bieten im Allgemeinen kein wesentlich anderes Bedienkonzept als die original mitgelieferten Geräte. Die Harmony Reihe von Logitech bezeichnet sich selbst jedoch als aktivitätsbasierte Fernbedienung und übernimmt so die Steuerung mehrerer Geräte mit nur einem einzigen Tastendruck [1].

Dennoch leiden auch solche ausgereiften Universalfernbedienungen an vielen der Schwächen ihrer einfacheren Vorgänger. Durch die zahlreichen Neuerungen im Bereich der Heimautomation und der Verbreitung von Netzwerkschnittstellen in herkömmlichen Haushaltsgeräten unterstützen Universalfernbedienung insgesamt nur einen Bruchteil der möglichen Geräte. Selbst

wenn sie die Beleuchtung des Eigenheims steuern könnten, müssten die BenutzerInnen immer noch nach der Fernbedienung suchen. Durch die Vielzahl an möglichen Einsatzorten im Zuhause würde sich auch die Anzahl der Orte, an denen man sie verlegen kann, erhöhen. Außerdem bieten sie die selben herkömmlichen Mechanismen zur Interaktion wie ihre Vorgänger. Die Bedienung einer Fernbedienung erfordert Konzentration und Aufmerksamkeit zur Suche nach der richtigen Taste. Während Fernbedienungen die ausschließlich über Tasten bedient werden und keinen Bildschirm bieten, durch Übung auch blind bedient werden können, ist dies bei teureren Geräten mit Bildschirm oder Touchscreen sogar oft noch schwieriger.

Es soll daher ein System zur Bedienung von Haushaltsgeräten mit Hilfe von NFC-Tags und einem NFC fähigen Android Smartphone entwickelt werden. Die Interaktion soll möglichst wenig Konzentration und Aufmerksamkeit benötigen und als zusätzliche Benutzerschnittstelle zu einer Touchscreen-Fernbedienung am Smartphone dienen. Durch das Anbringen von Tags an Orten, an denen Funktionen gewünscht werden, soll die BenutzerIn mit ihrem Smartphone schnell und unkompliziert Aktionen auslösen können. Das System soll mit einer Vielzahl von Geräten kommunizieren können und mit besonderem Fokus auf benutzerzentrierte Methoden entwickelt werden. Mit Hilfe von Benutzertests sollen die im folgenden Abschnitt definierten Forschungsfragen geklärt werden.

Einige Arbeiten haben sich bereits mit verwandten Themen beschäftigt. Sie legten jedoch keinen Fokus auf benutzerzentrierte Methoden und die Evaluierung durch Tests mit BenutzerInnen. Fragen zur Benutzerakzeptanz und Usability blieben dabei größtenteils unbehandelt. Die entwickelten Systeme verfolgten andere Ziele und halfen bei der Beantwortung anderer Forschungsfragen.

1.2 Fragestellung

Im Folgenden werden die wesentlichen Fragestellungen genannt. Sie lassen sich in die drei Kategorien **Entwicklung**, **Bedienbarkeit** und **Benutzerakzeptanz** einordnen. Die Kategorien werden nun mit detaillierten Fragestellungen näher erläutert.

Entwicklung:

- Was ist notwendig, um eine Universalfernbedienung die per NFC-Tags bedient werden kann, für Android Geräte zu entwickeln?
- Wie ist dabei ein Höchstmaß an Bedienkomfort zu erreichen?
- Wie sollen Tags gestaltet werden, um zu gefallen und die BenutzerInnen bestmöglich zu unterstützen?

Usability:

- Sind die BenutzerInnen mit NFC-Tags ähnlich effizient wie mit der Bedienung durch eine ausgereifte Touchscreen-Fernbedienung alleine?

- Ist die Verwendung einer solchen Universalfernbedienung leicht zu erlernen?
- Steigert ein solches System die Benutzerzufriedenheit?

Benutzerakzeptanz:

- Sehen BenutzerInnen einen Mehrwert in dem entwickelten System, durch einen größeren, subjektiv wahrgenommenen Nutzen?
- Was ist notwendig, dass BenutzerInnen ein solches System in ihr Eigenheim integrieren wollen?

1.3 Aufbau dieser Arbeit

Diese Arbeit gliedert sich in die Kapitel **Einleitung**, **Technologien**, **Methoden**, **Entwicklung** und **Schlusswort**. Das Kapitel **Technologien** gibt einen tiefen Einblick in alle Technologien, mit denen diese Arbeit in Berührung kam. Neben einem technischen Einblick werden ebenso geschichtliche wie aktuelle Themen besprochen. Das Kapitel **Methoden** nennt Methoden, die in der Entwicklung dieser Arbeit zum Einsatz kamen oder deren Einsatz in Betracht gezogen wurde. Das Kapitel **Entwicklung** beschäftigt sich schließlich im Detail mit allen praktischen Arbeiten, die zur Erstellung des gewünschten Systems, sowie zur Beantwortung der Forschungsfragen durchgeführt wurden. In diesem Abschnitt werden ebenfalls Analysen der Benutzertests präsentiert. Die **Analyse** bietet eine zusammenfassende Behandlung aller Erkenntnisse dieser Arbeit sowie einen Ausblick auf mögliche zukünftige Entwicklungen. In der **Zusammenfassung** findet sich eine kurze Übersicht über den Verlauf dieser Arbeit sowie das persönliche Resümee des Autors.

Technologien

Dieses Kapitel gibt einen Überblick über die verschiedenen Technologien, die in dieser Arbeit verwendet wurden. Neben Erläuterungen und Definitionen werden auch verwandte Technologien vorgestellt und relevante geschichtliche Informationen zu den Themengebieten geboten. Zu Beginn wird ein genauerer Blick auf die Historie und die aktuellen Entwicklungen bei **Smartphones** geworfen. Danach wird das populäre, mobile Betriebssystem **Android** vorgestellt. Damit einhergehend wird die Technologie **NFC**, welche aktuell in von vielen Android Smartphones unterstützt wird, näher betrachtet. Anschließend werden **Mikrocontroller** anhand von Beispielen und durch den Vergleich mit Mikroprozessoren erläutert. Ein Abschnitt über die **Arduino** Plattform gibt einen Überblick über die Besonderheiten der beliebten Entwicklungsplattform. Nach einem kurzen Rückblick auf die Grundlagen der Kommunikation wird ein Überblick über aktuelle Entwicklungen im Bereich der **Fernbedienung** im Haushalt gegeben. Ein Abschnitt über **Heimautomation** gibt einen tiefen Einblick in moderne Technologien zur Steuerung des Eigenheims und die zugrundeliegenden Konzepte. Nach näherer Betrachtung der visionären Technologien **Ubiquitous Computing** und **Tangible User Interfaces** bildet die Analyse und Zusammenfassung besonderer **Referenzprojekte** den Abschluss des Kapitels.



Abbildung 2.1: IBM Simon [92]

2.1 Smartphones

Smartphones entstanden aus der Kombination von einfachen Mobiltelefonen (sogenannten *Feature Phones*) und PDAs (*Personal Digital Assistant*). Im Folgenden wird daher zuerst der Begriff und die Geschichte des PDAs erläutert und anschließend näher auf Smartphones eingegangen [2].

Als Vorgänger des PDA werden im Allgemeinen auf Papier basierende Systeme zur Verwaltung von Informationen angesehen. Die bekanntesten Vertreter davon sind das Filofax (Portmanteau aus *File Of Facts*) und das Rolodex (Portmanteau aus *Rolling And Index*). Mit ihnen wurden Kontaktdaten, Termine, Aufgaben und Notizen verwaltet. Kritiken über die ersten PDAs enthielten daher oftmals Referenzen und Vergleiche mit ihren Vorgängern [3].

Mit PDAs werden typischerweise die selben Informationen wie mit ihren Vorgängern aus Papier verwaltet. Als erster PDA gilt der Psion Organizer II aus dem Jahre 1984 [2]. Als Anzeigedienste ein zweizeiliger 16-Zeichen LC-Bildschirm. Zur Eingabe wurden 36 Multifunktions Tasten verwendet. Neben der Möglichkeit zur Verwaltung persönlicher Informationen war das Gerät bereits mit Lotus Notes 1-2-3 kompatibel und erlaubte die Bearbeitung von Tabellen. Mit der COMMS-LINK Erweiterungskarte erhielt der Psion Organizer II eine RS232 Schnittstelle zur Kommunikation mit PCs. Mit der dazugehörigen Programmiersprache OPL (Organiser Programming Language) konnten sogar eigene Programme erstellt werden [4]. Ein Unternehmen konnte so also auch damals schon die Dateneingabe mit Mobilgeräten realisieren.

Der Begriff PDA wurde erstmals von Apple CEO John Sculley verwendet, als er das Apple Newton MessagePad bei der CES 1992 in Las Vegas vorstellte. Das Gerät wurde mit einem Touchscreen und einem Eingabestift bedient [2].

Als das erste Smartphone wird das IBM Simon aus dem Jahr 1993 angesehen (siehe Abbildung 2.1) [93]. Es kostete bei seiner Einführung \$899 und galt mit seinem Gewicht von 0.51kg und seinen Maßen von 20cm x 6.4cm x 3.8cm selbst für damalige Verhältnisse als klobig und schwer [5]. Das Gerät war mit einem x86 kompatiblen 16 MHz, 16-bit Mikroprozessor und 1MB RAM und ROM ausgestattet [93]. Das Betriebssystem basierte auf DOS und enthielt bereits Programme die für PDAs typisch waren. Es existierte ebenfalls ein optional erhältliches Kit für die Verbindung mit einem PC. Zusätzliche Software konnte über die Datenschnittstelle eingespielt oder per Erweiterungskarte verfügbar gemacht werden. Die Akkulaufzeit betrug 8 Stunden im Standby und etwa eine Stunde bei Verwendung [5].

Ebenfalls wie beim PDA entstand das erste Gerät der Kategorie Smartphones auch zu einem anderen Zeitpunkt als der Begriff selbst. Der Begriff Smartphone wurde erst im Jahr 1997 von Ericsson eingeführt. Sie bezeichneten mit dem GS 88 'Penelope' als erste ein Mobiltelefon als 'Smart Phone' [6].

Den großen Durchbruch in der Entwicklung von Smartphones gelang Apple mit ihrem iPhone

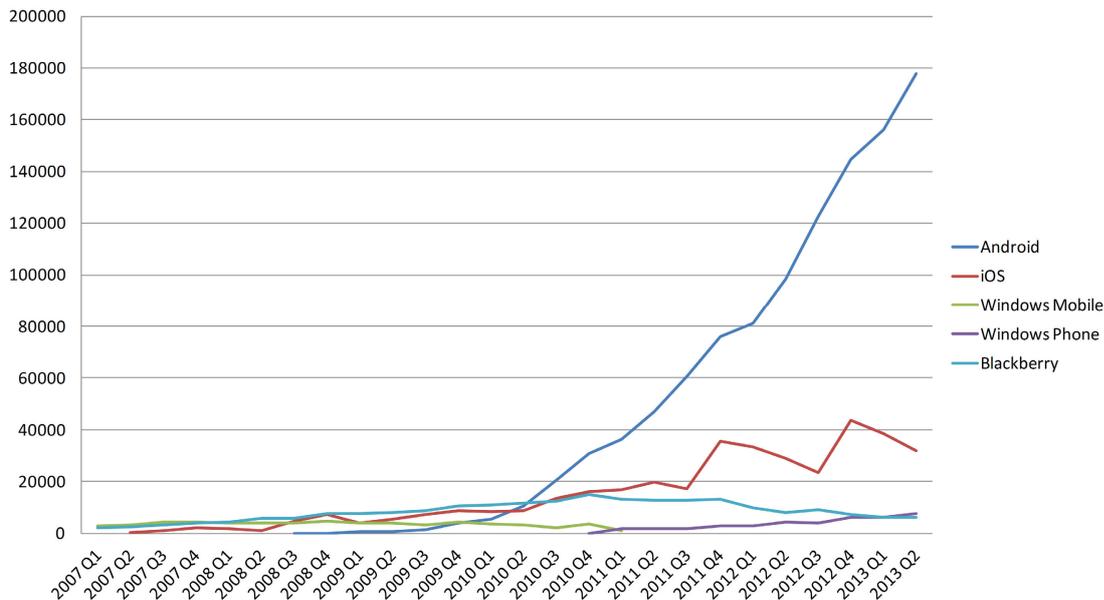


Abbildung 2.2: Mobile OS Verkäufe in Tausenden (abgeändert, reproduziert nach [94])

und dem dazugehörigen Betriebssystem iOS im Juni 2007. Im Oktober 2008 brachte HTC das erste Smartphone mit Android Betriebssystem auf den Markt [7]. Diese zwei Betriebssysteme dominieren zur Zeit den Smartphone Markt. Abbildung 2.2 zeigt ein Diagramm der historischen Zahlen an verkauften Smartphones der wichtigsten mobilen Betriebssysteme. Android hält mit ungefähr 178 Millionen verkauften Geräten im zweiten Quartal 2013 etwa 79% Marktanteil [95] [94].

Moderne Smartphones bieten weitaus mehr als ihre Vorgänger. Als Beispiel wird das Google Nexus 5 herangezogen welches am 31.10.2013 veröffentlicht wurde. Neben den bereits herkömmlichen Kommunikationsschnittstellen WiFi, Bluetooth und diverser Mobilfunkstandards bietet es unter anderem folgende Spezifikationen:

- 4.95" Full HD IPS Bildschirm
- Quadcore 2.26 GHz Prozessor
- 8MP Kamera
- 130g Gewicht
- NFC
- Bis zu 300 Stunden Standby
- 2GB RAM und 16 oder 32GB interner Speicher [96]

In weiterer Folge werden nun speziell für die Softwareentwicklung interessante Besonderheiten des Android Betriebssystems näher beschrieben, da Entwicklungen dieser Arbeit auf der Android Plattform durchgeführt wurden.

Android OS

Android ist ein auf Linux basierendes Betriebssystem für mobile Geräte mit Touchscreen wie Smartphones und Tablets. Einen Tag bevor das erste Android Smartphone käuflich zu erwerben war, wurde der Quellcode für das Betriebssystem offen gelegt. Der Quellcode steht unter der Apache Software Lizenz [7]. Als Konsequenz dessen existieren eine Vielzahl an modifizierten Varianten des Betriebssystems verschiedenster Hersteller und aus Community-Projekten [97] [98]. Ableger von Projekten, die unter der Apache Lizenz stehen, müssen jedoch selbst nicht unter der Apache Lizenz stehen [99]. Die modifizierten Versionen des Android OS sind daher nicht zwingend quelloffen.

Der Politik der Verbreitung von Android Updates und der zugrundeliegende Prozess unterscheidet sich stark von dem des direkten Konkurrenten iOS. Android Updates müssen zuerst vom Hersteller und (falls vorhanden) anschließend vom Mobilfunkbetreiber freigegeben werden. Je nach gewünschter Individualisierung und Reaktionszeit der involvierten Parteien kann dieser Prozess viel Zeit in Anspruch nehmen [100] [101]. Das Motorola Xoom Tablet ohne Datenschnittstelle erhielt sein Update auf Version 4 im Jänner 2012. Das gleiche Modell mit Mobilfunkschnittstelle erhielt das Update jedoch erst 5 Monate später. Eine Ausnahme dabei bilden die Geräte der 'Nexus' Reihe die Android Updates als Erste erhalten sollen. Doch selbst dieses Versprechen wurde gebrochen, als Mobilfunkbetreiber Updates für die Geräte zurückhielten [101]. Eine Untersuchung des Updateverhaltens von iOS BenutzerInnen zeigte, dass nach einem Monat etwa 73% der BenutzerInnen auf das neue iOS 7 gewechselt haben [102]. Im Vergleich dazu zeigt die im Juli 2013 veröffentlichte Android Version 4.3 (API-Level 18) selbst drei Monate später erst eine Verbreitung von 2.3% [103]. Das Ergebnis ist eine Vielzahl an verschiedenen Versionen und Varianten des Betriebssystems, die teilweise Fehlerquellen darstellen und während der Entwicklung speziell beachtet werden müssen.

Android Applikationen werden normalerweise in der sehr weit verbreiteten Programmiersprache Java geschrieben [104]. Obwohl Java von Beginn an als *Write Once, Run Anywhere* angepriesen wurde, bewahrheitet sich dieser Leitsatz nicht vollkommen. Gerade im Bereich von mobilen Geräten mit limitierten Implementierungen der Standard-Bibliotheken und unterschiedlichen oder gar nicht vorhandenen VMs (*Virtual Machine*) wird das ursprüngliche Versprechen gebrochen. Auch Android nutzt eine alternative Implementierung der Java API (Application Programming Interface). Android Applikationen werden in der Dalvik VM ausgeführt, welche sich besonders gut für den Einsatz auf mobilen Endgeräten mit stark limitierten Ressourcen eignet. Es ist fraglich, ob es eine bessere Wahl für Google gewesen wäre, der Fragmentierung des Java Ökosystems entgegenzuwirken und auf Standards zu bauen. Durch den Vorsprung von Apples iOS und dem niedrigen Tempo in der an Standards gearbeitet wird, dürfte dies keine denkbare Option gewesen sein [8].

Android bietet der ProgrammiererIn die vier Applikationskomponenten **Activities**, **Services**, **Broadcast Receiver** und **Content Provider** als Grundbausteine für Applikationen. Sie bieten dem Betriebssystem unterschiedliche Wege und Zugänge zu einem Programm.

Bei einer **Activity** (deutsch: Aktivität) handelt es sich um die Klasse, die ein Fenster erzeugt, in dem die Applikation ihre Benutzerschnittstelle darstellen kann. **Services** (deutsch: Dienste) laufen als Hintergrundprozess und eignen sich daher besonders für lang dauernde Operationen. **Broadcast Receiver** (deutsch: Rundsendungsempfänger) lauschen nach bestimmten systemweiten Nachrichten, um darauf gegebenenfalls zu reagieren. **Content Provider** bieten die Möglichkeit Schnittstellen zu Datenquellen zu erstellen [105] [106].

Ein besonders interessanter Teil von Android sind **Intents** (deutsch: Absicht, Vorhaben) und **Intent Filter**. **Intents** ermöglichen:

- Starten von **Activities**
- Starten von **Services**
- Aussenden von **Broadcasts** [107]

Explizite **Intents** werden an bestimmte Komponenten gerichtet. Sie finden sehr häufig Anwendung, da sie beispielsweise zum Starten einer weiteren **Activity** verwendet werden müssen, um eine andere Benutzerschnittstelle darzustellen.

Implizite **Intents** hingegen sind nicht an eine bestimmte Komponente gerichtet. Beim Starten einer **Activity** per implizitem **Intent** sucht das Betriebssystem alle in Frage kommenden Applikationen. Anschließend präsentiert Android der BenutzerIn die zur Verfügung stehenden Applikationen für die gewählte Aufgabe. Um implizite **Intents** verarbeiten zu können, muss die empfangende Applikation entsprechende **Intent Filter** definieren. Mit ihnen spezifiziert die Applikation für welche ‘Absichten’ sie geeignete Komponenten besitzt [107]. Somit ist die Modularität, einer der Grundgedanken von Linux, fester Bestandteil des Android Betriebssystems.

2.2 Near-Field-Communication (NFC)



Abbildung 2.3: NFC N-Mark [108]

NFC (*Near-Field-Communication*, deutsch: Nahfeldkommunikation) ist eine Technologie zur drahtlosen Kommunikation über sehr kurze Strecken. Damit sind zwar theoretisch Entfernun-

gen von bis zu 20 cm überbrückbar, es werden jedoch in der Praxis meist nicht mehr als vier Zentimeter erreicht. NFC ist eine Sammlung von Standards, basierend auf anerkannten und erprobten Standards der Technologie RFID (*Radio-Frequency Identification*) [9]. Daher wird im Folgenden zuerst ein Überblick über RFID gegeben und danach auf die Besonderheiten von NFC eingegangen.

RFID Systeme bestehen in der Regel aus einem aktiven Lesegerät und einem passiven Transponder. Der Transponder wird als Tag (deutsch: Etikett) bezeichnet. Der Tag wird mit dem Lesegerät ausgelesen, sobald er sich in Lesereichweite befindet. Geräte mit geringer Reichweite müssen den Tag daher nahezu berühren. Aktive Geräte senden selbst Signale und Daten und brauchen daher eine Stromquelle, um betrieben zu werden. Aktive Tags werden in der Regel mit Batterien betrieben. Passive Geräte senden keine Signale, solange sie nicht durch elektromagnetische Wellen mit der richtigen Frequenz angeregt werden [10].

Als erstes Gerät dass einem passiven RFID-Tag ähnelt, wird Leon Theremins Abhörgerät *The Great Seal Bug* angesehen. Da es sich um eine passive Wanze handelt, galt sie als besonders schwer auffindbar. Die Wanze selbst benötigte nämlich keine Energiequelle und sendete lediglich wenn sie gerade abgehört wurde [11].

In der Praxis finden sich eine Vielzahl verschiedener RFID Architekturen und Technologien für unterschiedlichste Anwendungsgebiete. Darunter finden sich auch neben ARPT (*Active-Reader Passive-Tag*) auch PRAT (*Passive-Reader Active-Tag*) und ARAT (*Active-Reader Active-Tag*) Systeme sowie BAP (*Battery Assisted Passive*) Tags. Die Reichweite von PRAT Systemen kann bis zu 600 Meter betragen [10]. Der Preis von entsprechenden Tags kann je nach Technologie wenige Cent bis zu einigen Euros betragen [12].

Die Hauptanwendungsgebiete von RFID befanden sich bisher größtenteils in den folgenden Bereichen [13]:

- Überwachung von Lieferketten
- Zutrittskontrolle
- Verfolgung von Gepäckstücken
- Elektronische, mobile Zahlungen
- Verwaltung von Viehbestand
- Verwaltung von Büchern und Hilfe beim Wiedereinordnen

2004 gründete NXP, Sony und Nokia das NFC Forum mit dem Ziel bestehende Standards zu vereinen und eine neue Technologie zu erschaffen. NFC ist daher teilweise mit RFID Technologie kompatibel [14]. Um den Wiedererkennungswert zu steigern, wurde vom NFC Forum das N-Mark Logo (siehe Abbildung 2.3) eingeführt. Es soll von den Herstellern an Geräten, in Software, auf Tags und in den Medien eingesetzt werden, um das Vorhandensein von Diensten zu

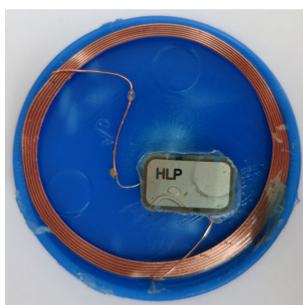


Abbildung 2.4: Geöffneter NFC-Tag

signalisieren und die Positionierung der NFC-Geräte anzuzeigen [108]. Die Technologie wird aufgrund ihrer Reichweite vom NFC-Forum als 'von Natur aus sicher' bezeichnet [109].

NFC-Geräte können in drei verschiedenen Betriebsmodi verwendet werden: *Reader/Writer*, *Peer-to-Peer* und *Card Emulation* Modus. Im *Reader/Writer* Modus können passive NFC-Tags mit Daten beschrieben und ausgelesen werden. Der *Peer-To-Peer* Modus erlaubt es zwei aktiven NFC-Geräten direkt miteinander zu kommunizieren und Daten auszutauschen. Im *Card Emulation* Modus agiert das Gerät wie ein NFC-Tag [14].

Das erste Mobiltelefon mit NFC Schnittstelle war das im Februar 2006 veröffentlichte Nokia 6131 [110]. Die NFC Antenne befindet sich in der oberen Schale des klappbaren Mobiltelefons [15]. Bis zur Veröffentlichung des Google Nexus S im Jahr 2010 waren nur sehr wenige Mobiltelefone mit NFC Schnittstelle erhältlich [9]. Mittlerweile sind sogar schon viele günstige Android Smartphones mit NFC ausgestattet. Der 2012 veröffentlichte und wieder eingestellte Mediaplayer Google Nexus Q enthält ebenfalls eine NFC Schnittstelle. Damit konnte die BenutzerIn einen Link zu der dazugehörigen Applikation aufrufen. Auch das *Pairing* (deutsch: Kopplung) mit zusätzlichen Android Geräten wurde mit NFC ermöglicht. So konnten Gäste einfach ihre eigenen Smartphones und Tablets mit dem Gerät verbinden und ihre eigenen Medien auf dem Gerät wiedergeben [111].

Die typischen Anwendungen von NFC sind ähnliche wie die von RFID. Durch die stärkere Präsenz im Privatgebrauch ergeben sich jedoch zahlreiche zusätzliche Verwendungsmöglichkeiten wie etwa [112]:

- Situationsabhängige Einstellungsprofile durch Tags an diversen, gebräuchlichen Orten
- Teilen von Hotspot-Zugangsdaten
- Pairing von Bluetooth Geräten wie Tastaturen, Lautsprechern oder Kopfhörern

RFID und NFC-Tags gibt es in unterschiedlichen Verpackungen wie Aufklebern, Visitenkarten, Schlüsselanhängern und Armbändern. Abbildung 2.4 zeigt die Spule und den Chip eines geöffneten NFC Schlüsselanhängers. Das steigende Interesse an der Technologie NFC brachte ein



Abbildung 2.5: NFC-Ring

Kickstarter (eine Internetplattform zur Projektfinanzierung per *Crowdfunding*) Projekt für die Entwicklung eines NFC Rings (siehe Abbildung 2.5) zum Erfolg. Das Projekt wurde zu 806% finanziert und brachte £241.947 anstatt der ursprünglich veranschlagten £30.000. Der NFC Ring enthält zwei NTAG203 Tags auf gegenüberliegenden Seiten des Rings. Dazwischen befindet sich eine abschirmende Schicht, um Interferenzen zu minimieren. Somit unterstützt der NFC Ring zwei verschiedene Interaktionen. Die Innenseite des Rings befindet sich in der Handfläche und soll für private Informationen verwendet werden, wie zum Beispiel zum Öffnen eines Türschlosses oder zum Entsperren des Smartphones. Die Außenseite wird mit der geschlossenen Faust verwendet und soll der BenutzerIn Platz für öffentliche Informationen wie etwa eine elektronische Visitenkarte bieten [113].

2.3 Mikrocontroller

Um die Besonderheiten eines Mikrocontrollers zu erläutern, wird zuerst auf den Begriff Mikroprozessor in Kürze näher eingegangen. Bei einem Mikroprozessor handelt es sich um eine normale CPU (Central Processing Unit) wie man sie in PCs vorfindet. Er ist über einen Bus mit einer Vielzahl von Komponenten wie Speicher, Laufwerken, Controllern und Ausgabegeräten verbunden. Typischerweise wird ein Mikroprozessor zumindest mit Speicher und einem Ausgabegerät betrieben, um sinnvoll eingesetzt werden zu können.

Mikrocontroller (oft abgekürzt durch μC , uC oder MCU) enthalten bereits alle notwendigen Komponenten, um sinnvolle Aufgaben zu übernehmen und mit der Umwelt zu interagieren. Die Komponenten sind über einen internen Bus miteinander verbunden. Zu den Komponenten in einem Mikrocontroller gehören unter anderem [16]:

- Prozessor
- Speicher
- Analoge und digitale *GPIO Pins*
- *Timer/Counter*
- *Interrupt Controller*

Die **GPIO Pins** (*General Purpose Input Output*, deutsch: universelle Eingabe-Ausgabe) des Mikrocontrollers sind seine Schnittstelle zur Außenwelt. Sie können als analoge und digitale Ein- und Ausgänge benutzt werden und somit Signale einlesen oder ausgeben. In einem Radiowecker sind beispielsweise Pins mit den Tasten verbunden, um auf Tastendrucke reagieren zu können, während andere Pins die Ausgabe auf der Anzeige übernehmen. **Timer** (deutsch: Zeitgeber) ermöglichen die regelmäßige Ausführung von Aktionen. Mit ihnen kann man beispielsweise beim Radiowecker, das blinkende Trennzeichen zwischen der Anzeige von Stunden und Minuten implementieren. **Timer** sind eigentlich Zähler (engl: **Counter**), die nebenläufig per Hardware hochgezählt werden. Anstelle den Zähler nach einer bestimmten Anzahl von Taktzyklen der CPU zu inkrementieren, kann auch ein digitales Signal an einem **GPIO Pin** als Auslöser verwendet werden. Ein **Timer** kann an bestimmten Zählpunkten einen **Interrupt** (deutsch: Unterbrecher) auslösen. Dadurch wird der normale Programmfluss unterbrochen, um auf den Zählpunkt reagieren zu können. **Interrupts** können auch durch andere Ereignisse ausgelöst werden, wie etwa bei der steigenden oder fallenden Flanke eines digitalen Eingangssignals [16] [17] [114].

Zahlreiche Hersteller wie zum Beispiel Freescale, Atmel, Texas Instruments, Microchip, Intel und NXP produzieren Mikrocontroller. Als Beispiel für einen Mikrocontroller wird nun jedoch der ATmega328 aus der 8-bit AVR Serie von Atmel herangezogen. Sie werden auf den Platinen der klassischen Arduino Uno (siehe Abschnitt 2.3) verbaut und sind sehr weit verbreitet. Der ATmega328 wird mit einer Taktfrequenz von maximal 20 MHz betrieben, wobei die meisten Instruktionen in einem einzigen Takt ausgeführt werden. Somit werden bis zu 20 MIPS (*Million Instructions Per Second*) erreicht [115]. Das entspricht in etwa der Rechenleistung eines Intel 80486 mit 25 MHz [116].

Ein Mikrocontroller enthält typischerweise folgende drei Arten von Speicher: **Flash-ROM**, **EEPROM**, und **RAM**. Das **Flash-ROM** enthält den Maschinencode des auszuführenden Programms und ist typischerweise zur Laufzeit nicht beschreibbar. Das **EEPROM** wird zum Speichern von Daten wie Messwerten und Einstellungen verwendet. Das **RAM** bietet Speicher für Variablen und den Stack. Tabelle 2.1 gibt einen Überblick über die Spezifikation und verdeutlicht dass **Flash** und **EEPROM** zwar nichtflüchtig sind, jedoch auch nicht unbegrenzte Schreibzugriffe erlauben [18].

	Flash-ROM	EEPROM	RAM
Schreibzugriffe	10.000	100.000	∞
Flüchtig	✗	✗	✓
ATmega328P	32KB	1KB	2KB

Tabelle 2.1: ATmega328P Speicher

Verglichen mit modernen PCs, die mit Prozessoren im GHz Bereich und Speicher im Gigabyte und Terabyte Bereich ausgestattet sind, wirken die Ressourcen eines Mikrocontrollers stark limitiert. Ihr Einsatzgebiet ist jedoch auch ein anderes. Milliarden von ihnen werden heutzutage jährlich produziert und sie finden sich in nahezu jedem elektronischen Gerät, von Haushaltsgerä-

ten wie Fernbedienungen und Kaffeemaschinen bis hin zur Treibstoffeinspritzung und Ansteuerung des ABS in einem Auto. Dort erfüllen sie eine spezifische Aufgabe für die ihre Ressourcen ausreichend sind. Mikrocontroller finden auch in sicherheitskritischen Systemen Anwendung, in denen eine garantierte Reaktionszeit des Systems überlebenswichtig ist [16] [17]. Im Folgenden werden die wichtigsten Vorteile eines Mikrocontrollers erläutert. Dabei handelt es sich um:

- Kosten
- Energieverbrauch
- Verlässlichkeit
- Veränderbarkeit

Mikrocontroller sind **billig**. Die treibende Kraft hinter ihrer Entwicklung war die Senkung von Kosten [16]. Die Preise variieren je nach Ausstattungsvariante und bewegen sich hauptsächlich im einstelligen Euro Bereich. Einfachste Modelle sind sogar bereits zu Preisen unter einem Euro zu bekommen. Ein ATmega328P-PU kostete zum Zeitpunkt der Erstellung dieser Arbeit ungefähr 4.78 € [117].

Mikrocontroller haben einen **niedrigen Energieverbrauch**. Die Stromaufnahme des ATmega328P beträgt bei einer Spannung von 1.8 Volt und einer Taktfrequenz von 1 MHz lediglich 0.2 mA. Das entspricht einer Leistung von 0.36 mW im aktiven Modus. Der Stromverbrauch ist jedoch stark abhängig von zahlreichen anderen Faktoren und Konfigurationen und nimmt dementsprechend zu. Dazu zählt beispielsweise die Verwendung von GPIO Pins, Zeitgebern und des Schlafmodus [115].

Mikrocontroller sind **langlebig**. Die durchschnittliche MTBF (*Mean Time Between Failure*, deutsch: mittlere Zeit zwischen Ausfällen) eines herkömmlichen PCs wird mit ungefähr 25 000 Betriebsstunden (3 Jahre) geschätzt [19]. Atmel berechnete $1.69 * 10^7$ Stunden (1929 Jahre) als MTBF bei 65 °C für ihre 8-bit AVR Mikrocontroller Serie [115].

Das Verhalten eines Mikrocontrollers ist **veränderbar**. Probleme direkt in Hardware zu lösen, resultiert in einer besseren *Performance* (deutsch: Leistung). Änderungen sind jedoch kostspielig oder gar unmöglich. Durch den Einsatz von Software kann das Systemverhalten verändert werden, ohne dafür Veränderungen an der Hardware vornehmen zu müssen [16].

Um Kosten und Energieverbrauch zu minimieren, gilt es also den am besten geeigneten Mikrocontroller auszuwählen, der nicht wesentlich mehr bietet als benötigt. Dazu müssen zuerst die Anforderungen an das System genauestens analysiert werden. Als Faustregel gilt es mit 80% des verfügbaren Speichers zu planen, um Raum für unvorhergesehene Änderungen und Erweiterungen zu bieten [16]. Etwaige kritische Algorithmen und deren gewünschte Ausführungszeit muss untersucht werden, um einen passenden Mikrocontroller auswählen zu können. Bei sicherheitskritischen Systemen kann auch die Interrupt Latenz eine entscheidende Rolle spielen, um die Ausführungszeit von Interaktionen mit dem System garantieren zu können [17].

Arduino

Die Arduino Plattform bietet verschiedene, mit einem Mikrocontroller und weiteren Komponenten bestückte Platinen sowie eine dazugehörige IDE (*Integrated Development Environment*, deutsch: Integrierte Entwicklungsumgebung). Besonderer Beliebtheit erfreut sich die Plattform bei DesignerInnen und KünstlerInnen zur Erstellung von Prototypen [20]. Der klassische Arduino Uno ist mit seinem Preis von ungefähr 24 € [118] jedoch um einiges teurer als ein einzelner ATmega328P Mikrocontroller für 4.78 € [117].

Die Arduino Hardware und Software bietet jedoch ein gewisses Maß an zusätzlichem Komfort. Dadurch soll das notwendige Vorwissen minimiert werden, um Neulingen den Einstieg in die Bereiche Elektronik und Programmierung von Mikrocontrollern zu erleichtern. Der niedrige Preis macht Fehler leistbar. Dieser Umstand senkt die Berührungängste mit der Technologie und fördert somit die Experimentierfreudigkeit zusätzlich [20]. Im Vergleich zu den 24 € für einen Arduino Uno kostet das Atmel Entwicklungskit der vorigen Generation (STK500) etwa 110 € [119]. Das aktuelle Modell (STK600) kann zwar auch über USB angebunden werden, kostet dafür aber ungefähr 275 € [120]. Durch die geringe Größe der Arduinos und ihrem vergleichsweise kleinen Preis können Arduinos auch in kommerziellen Produkten verbaut werden [121]. Abbildung 2.6 zeigt einen Arduino Ethernet und ein Atmel STK500 zum Größenvergleich.

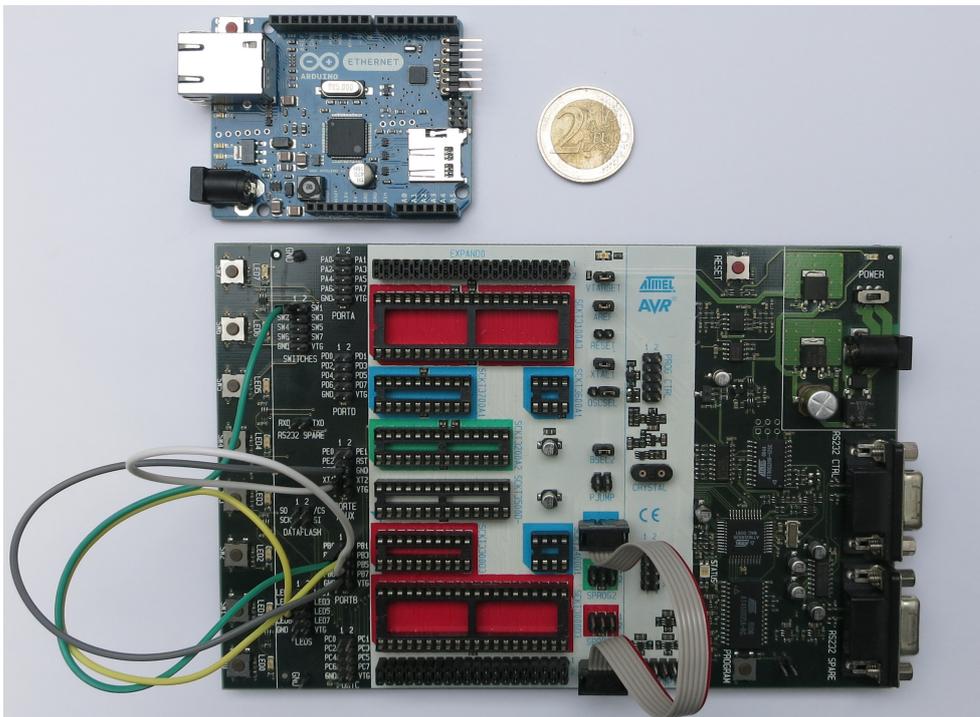


Abbildung 2.6: Arduino Ethernet und Atmel STK500

Die Arduino IDE ist für die großen Betriebssysteme Windows, Macintosh und Linux frei erhältlich. Das Einspielen von Programmcode wird über den USB Anschluss durchgeführt. Somit ist die Arduino Entwicklung praktisch von jedem modernen PC aus möglich [20]. Im Gegensatz dazu muss die weit verbreitete Atmel Entwicklungsplatine der vorigen Generation (STK500) per serieller Schnittstelle angebunden werden [122] und die Entwicklungsumgebung 'Atmel Studio' ist lediglich für Windows verfügbar [123]. Die Arduino IDE ist in Java geschrieben und basiert auf der *Open Source* (deutsch: quelloffen) IDE 'Processing'. Die Arduino Sprache ist eine Untermenge der Programmiersprache C. Zur Übersetzungszeit wird der Programmcode modifiziert, um gültigen C Code zu erhalten, der anschließend mit den herkömmlichen Programmen zur AVR Entwicklung kompiliert und auf den Mikrocontroller geladen wird [20]. Die Abstraktion von der Sprache C soll Einsteigern und Nicht-Programmierern den Einstieg weiter erleichtern [20].

Durch spezielle Platinen die auf den Arduino gesteckt werden (sogenannte *Shields*) kann er schnell um weitere Schnittstellen erweitert werden, wie beispielsweise mit einem Ethernet Interface oder einem LCD-Bildschirm. Die Erstellung eigener *Shields* ist natürlich auch möglich. So können komplette Projekte auf einer kleinen Platine gebaut und bei Bedarf ausgetauscht werden [21]. Abbildung 2.7 zeigt einen Arduino mit Ethernet- und XBee-Shield. Das XBee-Shield ermöglicht drahtlose Kommunikation über Zigbee [124] (siehe Abschnitt 2.5).

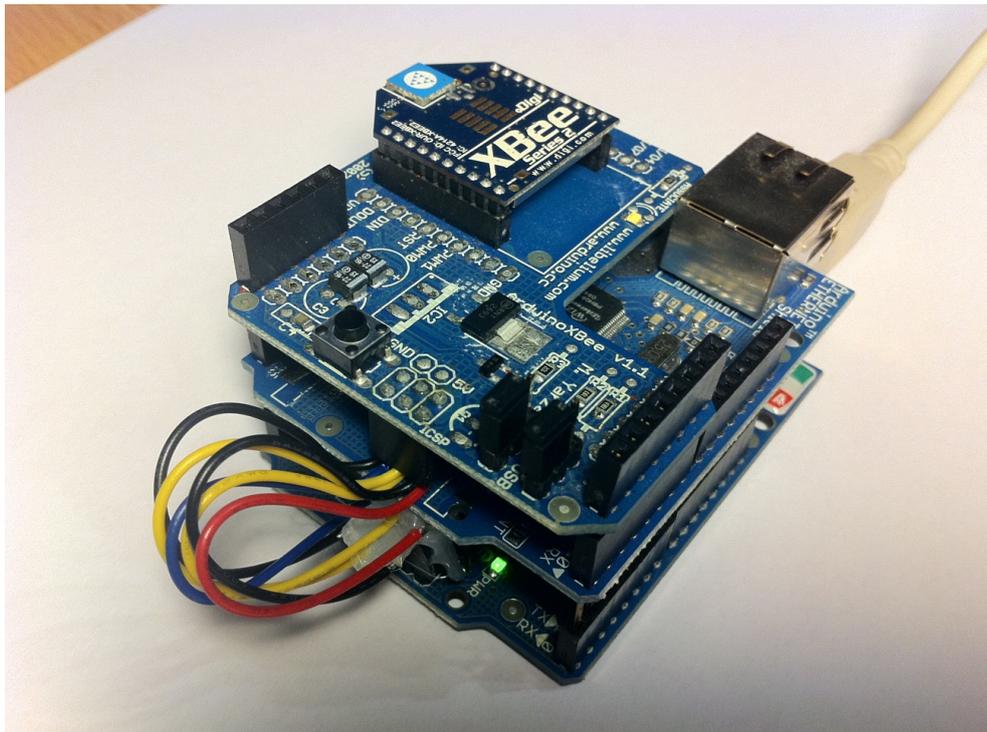


Abbildung 2.7: Arduino mit Ethernet- und XBee-Shield [125]

2.4 Fernbedienung im Haushalt

In diesem Kapitel werden Technologien zur Fernbedienung von Geräten im Haushalt näher erläutert. Um Geräte fernbedienen zu können, müssen Signale übertragen werden. Zur Veranschaulichung der Grundlagen von Kommunikation wird das Shannon-Weaver Modell (auch Sender-Empfänger Modell genannt) aus dem Jahr 1949 herangezogen (siehe Abbildung 2.8). Das Sender-Empfänger Modell besteht aus den Komponenten: **Quelle**, **Sender**, **Übertragungsmedium**, **Empfänger**, **Ziel** und einer **Störquelle**.

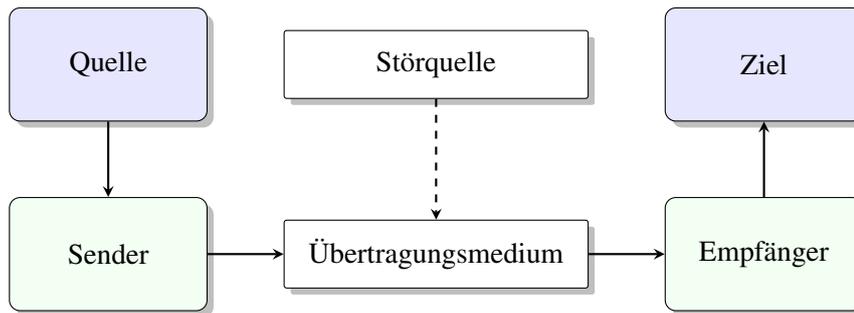


Abbildung 2.8: Shannon-Weaver Kommunikationsmodell (abgeändert, reproduziert nach [22])

Die **Quelle** erstellt eine Nachricht und möchte diese an das Ziel übermitteln. Typischerweise wählt sie dafür eine Nachricht aus einer begrenzten Menge an möglichen Nachrichten. Der **Sender** wendet eine Reihe von Operationen auf die Nachricht an, um sie als Signal übermitteln zu können. Anschließend wird das Signal vom **Sender** durch ein **Übertragungsmedium** an den **Empfänger** übermittelt. Beim **Übertragungsmedium** handelt es sich typischerweise um einen Drahtweg, Funkweg oder Lichtweg. Der **Empfänger** wendet Operationen, die invers zu denen vom **Sender** sind, auf das Signal an, um die darin enthaltene Nachricht zu erhalten. Das Ergebnis wird anschließend an das **Ziel** der Informationsübermittlung weitergegeben. Am Weg kann die Nachricht unterschiedlichen **Störquellen** ausgesetzt sein und möglicherweise durch sie verzerrt werden [23].

Bei Haushaltsgeräten finden sich Vertreter aller der eben erwähnten Übertragungsmedien wieder. Die Infrarotfernbedienung ist das Überbleibsel des Heimautomation-Booms der 70er Jahre [24]. Sie werden heutzutage mit nahezu jedem Audio- und Videogerät (Fernseher, DVD-Player, usw.) mitgeliefert. Infrarotfernbedienungen benötigen eine Sichtverbindung mit dem Gerät. Es existieren unterschiedliche Protokolle, welche die Übermittlung von Infrarotsignalen spezifizieren. Das RC-5 Protokoll von Philips benutzt beispielsweise 14 bits für das Kommando und eine Trägerfrequenz von 36 KHz [126].

Die Logitech Harmony Serie an Universalfernbedienungen für Geräte mit Infrarotschnittstelle findet sehr weite Verbreitung. Neben den typischen Funktionen einer Infrarot Universalfernbedienung wie einer Datenbank an Geräten verwendet sie das Konzept von *Activities*. Anders als bei herkömmlichen Universalfernbedienungen muss die BenutzerIn die Geräte nicht mehr einzeln bedienen, sondern die Fernbedienung übernimmt die Steuerung mehrerer Geräte [1].

'The Harmony 1100 is an Activity-based infrared (IR) universal remote control... When you select an Activity, your Harmony 1100 sends a series of commands to your entertainment system to turn on the necessary devices and configure them for the selected Activity' [1, S. 5]

Samsung stattet aktuell einige seiner höherpreisigen Android Smartphones (siehe Abschnitt 2.1) mit Infrarotsendern zur Fernsteuerung von Geräten aus [127]. Eine API zur Ansteuerung der IR-LED wurde allerdings von Samsung nicht veröffentlicht. Mittlerweile finden sich jedoch Programmbeispiele zur Verwendung im Internet. Die Implementierung von Samsung verwendet einen Service, der auf Aufforderungen zum Senden von Daten wartet. Da die Klassendefinition des Service zur Kompilierzeit jedoch nicht vorhanden ist, behilft man sich mit *Reflection*. Mit *Reflection* können Objekte zur Laufzeit untersucht werden. In diesem Beispiel wird damit die Methode 'write_irsend' des Service gesucht und anschließend aufgerufen. Der Service wird auf folgende simple Weise verwendet [128]:

```
// Daten zum Senden
String data = hex2dec("0000 006d 0022 0002 0152 00aa 0015...");

// Service holen
Object irdaService = this.getSystemService("irda");
Class c = irdaService.getClass();

try
{
    // Methode holen – hier passiert die Reflection
    Class p[] = { String.class };
    Method irWrite = c.getMethod("write_irsend", p);

    // Senden
    irWrite.invoke(irdaService, data);
}
catch (Exception e)
{
    e.printStackTrace();
    throw new RuntimeException("Failed invoking service");
}

...

protected String hex2dec(String irData)
{
    List<String> list = new ArrayList<String>(Arrays.asList(irData.split(" ")));
    list.remove(0); // dummy
    int frequency = Integer.parseInt(list.remove(0), 16); // frequency
    list.remove(0); // seq1
    list.remove(0); // seq2

    for (int i = 0; i < list.size(); i++)
    {
        list.set(i, Integer.toString(Integer.parseInt(list.get(i), 16)));
    }
}
```

```

    }

    frequency = (int) (1000000 / (frequency * 0.241246));
    list.add(0, Integer.toString(frequency));

    String irData = "";
    for (String s : list)
    {
        irData += s + ",";
    }
    return irData;
}

```

Funkübertragung wird zur Steuerung vor allem im Bereich der Heimautomation (siehe Abschnitt 2.5) angewendet. Sie findet sich jedoch auch in einer Vielzahl anderer Geräte wieder, wie beispielsweise in Klimaanlage und Garagentoren [126]. Neben den populären Systemen der Heimautomation existieren auch eine Vielzahl proprietärer, teilweise analog kommunizierender Lösungen zur Steuerung von Verbrauchern im Eigenheim [25].

Viele Fernseher sind heutzutage bereits mit einer Ethernetschnittstelle ausgestattet. Neben Funktionen wie der Wiedergabe von Medien aus dem Netzwerk [129] können solche Geräte oft auch mit einem Smartphone über das Netzwerk bedient werden [126] [130].

Ein Protokoll zur Steuerung, Überwachung und Diagnose von Kaffeemaschinen entstand als Aprilscherz-RFC im Jahre 1998. Der RFC2324 spezifiziert das Hyper Text Coffee Pot Control Protocol (HTCPCP) als Erweiterung des HTTP Protokolls mit dem URI Schema 'coffee:', den zusätzlichen Methoden BREW und WHEN und dem Fehlercode 418 *I'm a teapot* [126].

Gateways finden in allen Bereichen der Kommunikation ihre Anwendung ebenso bei der Steuerung von Geräten im Heimbereich. Gateways konvertieren ein Signal in ein anderes Übertragungsmedium [26]. Die Logitech Harmony 1100 bietet beispielsweise einen *RF Wireless Extender*, mit dem bis zu acht Geräte mit Infrarotschnittstelle außerhalb des Sichtfeldes bedient werden können [1].

2.5 Heimautomation

Das Ziel von Heim- und Gebäudeautomation ist es die Interaktion zwischen dem Menschen und Geräten sowie den Geräten untereinander zu verbessern. Die Gebäudeautomation beschäftigt sich, im Gegensatz zur Heimautomation, mit großen Installationen wie zu Beispiel in Bürokomplexen und Krankenhäusern. Die Motivation zum Einsatz der Technologien ist der zusätzliche Komfort und die niedrigeren Kosten im Betrieb. Die Betriebskosten eines Gebäudes sind über die gesamte Lebensdauer typischerweise sieben mal so hoch wie die der Errichtung. Diese Kosten werden durch den optimierten Einsatz von Heizung, Belüftung, Beleuchtung, Beschattung und Klimaanlage gesenkt. Durch die bessere Energieeffizienz wird außerdem die Umwelt geschont. Weitere Einsparungen ergeben sich durch niedrigere Personal- und Wartungskosten [27]

[28]. Es existieren sogar Modelle und Normen zur Bestimmung der thermischen Behaglichkeit, um den Komfort in Gebäuden messbar zu machen und planen zu können [29].

'Home automation will be achieved not with the household robot but with embedded computing power and memory within dozens of pieces of domestic equipment, each of which will communicate with the user and other equipments.' [26, S. 1]

Die Heimautomation wird als eng verwandter oder Teilbereich der Gebäudeautomation gesehen. Es handelt sich dabei jedoch um mehr als die bloße Anwendung von Technologien aus der Gebäudeautomation im privaten Heim [28]. Die Systeme im Heimgebrauch sind wesentlich kleiner als typische Installationen der Gebäudeautomation. Die Einbindung von Unterhaltungssystemen und Haushaltsgeräten nimmt jedoch einen hohen Stellenwert ein. Auch die Einfachheit der Installation solcher Systeme ist besonders wichtig für den Privatverbraucher. Die Möglichkeit zur drahtlosen Kommunikation oder die Nutzung bestehender Stromleitungen spielen daher eine wesentliche Rolle. Durch die weite Verbreitung von Smartphones und anderen mobilen Geräten wird auch *Pervasive Computing* bzw. *Ubiquitous Computing* (siehe Abschnitt 2.6) für die zukünftige Entwicklung eine Rolle spielen [27].

Aber auch die Anschaffungskosten haben sich als wichtiger Faktor für die Akzeptanz von Heimautomationsssystemen herausgestellt. Einer Studie von Siemens und Bosch zufolge haben zwar mehr als 70% der Deutschen Interesse an Heimautomation, weniger als 20% würden jedoch mehr als 50 € dafür investieren [28].

Im Folgenden werden die Herkunft und die Besonderheiten einiger der wichtigen und gebräuchlichen Technologien, die ihren Einsatz in der Heimautomation und teilweise auch in der Gebäudeautomation finden, näher erläutert. Ein Vergleich der technischen Spezifikationen von auf Funk basierenden Systemen ist in der Tabelle 2.2 zu finden.

Übertragungstechnologien

Durch die Vermeidung baulicher Maßnahmen bei nachträglicher Installation und Erweiterung bestehender Systeme erfreuen sich besonders Powerline- und Funk-Systeme großer Beliebtheit. Alle der im Folgenden vorgestellten Systeme unterstützen jedoch auch Funkmodule. Somit können auch batteriebetriebene Komponenten beispielsweise außerhalb des Hauses eingesetzt werden [28].

Das 2.4 GHz Band ist mit WLAN und Bluetooth das populärste weltweit. Durch die kurze Wellenlänge erlaubt es hohe Datenraten auf kurze Distanzen. Durch die weite Verbreitung ist das Frequenzband jedoch auch stark belegt. Heim- und Gebäudeautomation kommt mit niedrigeren Datenraten aus und wick daher auf das ISM Band um 900 MHz aus. Dadurch ergeben sich auch Vorteile wie bessere Funkwellenausbreitung. Leider unterscheiden sich die Frequenzbänder in Europa und den USA, wodurch die Geräte nicht miteinander kompatibel sind. Die notwendigen Änderungen an der Hardware beschränken sich jedoch auf der Anpassung eines einzigen Oszillators [30].

Eine weitere wichtige Funktionalität von funkbasierten Systemen ist die Implementierung von Routing-Mechanismen. Anders als bei Punkt-zu-Punkt Verbindungen können damit Knoten die sich außerhalb der direkten Reichweite befinden angesprochen werden [28]. Welche der Systeme Routing unterstützen ist der Tabelle 2.2 zu entnehmen. Systeme ohne Routing Unterstützung behelfen sich mit Repeater-Stationen, um die notwendige Reichweite in großen Gebäuden gewährleisten zu können [31].

X-10

X-10 ist ein auf Powerline basierendes System. Als Übertragungsmedium dienen also die in einem Gebäude vorhandenen Stromleitungen [28].

Die Technologie wurde von Pico Electronics im Jahr 1978 erstmals der amerikanischen Öffentlichkeit vorgestellt. Zur damaligen Zeit war es noch nicht selbstverständlich, dass selbst ein Fernsehgerät mit einer Fernsteuerung bedient werden konnte. Bei den Arbeiten an der Entwicklung des mittels Ultraschall fernbedienbaren Plattenwechslers namens 'Accutrac 2000' kam ihnen die Idee zur Fernsteuerung von Lampen und Haushaltsgeräten [32].

Das X-10 System findet in den USA weite Verbreitung. Der Erfolg in Europa blieb aufgrund von Regulierungsmaßnahmen im Telekommunikationsbereich jedoch lange aus. Durch die Erweiterung auf drahtlose Kommunikation sowie Lockerungen im Bereich der Regulierungen stellt es jedoch mittlerweile eine günstige Alternative dar [28].

X-10 ist ein äußerst langsames Protokoll aber zumindest ausreichend schnell für den Zielbereich. Ein Kommando braucht mit einer Netzfrequenz von 60 Hz immerhin 0.8 Sekunden zur Übermittlung.

Z-Wave

Z-Wave wurde von Zensys entwickelt, einer dänischen Firma welche 1999 von zwei Ingenieuren gegründet wurde. Die erste Generation von Z-Wave Hardware wurde jedoch erst 2003 verkauft und baute damals noch auf einem einfachen Atmel Mikrocontroller auf. Im Jahr 2005 wurde die Z-Wave Allianz gegründet. Sie umfasst mit mehr als 200 Unternehmen alle Hersteller kompatibler Geräte. Ziel der Allianz ist die Kontrolle und Zertifizierung der Interoperabilität sowie die Weiterentwicklung und Marktverbreitung des Standards.

Z-Wave Geräte basieren immer noch auf Hardware und Firmware von Zensys, wodurch die Funk- und Netzwerk-Ebene vorgegeben ist. Die Hersteller verbauen die Chips und implementieren die für die gewählte Geräteklasse notwendigen Kommandos und grundlegende Protokollfunktionen des Knotentyps wie beispielsweise **Assoziation** [25]. Eine Analyse des Stromverbrauchs des Z-Wave Chips der vorigen Generation (ZW0201) ergab bei einer Stromversorgung zwischen 2.2V und 3.6V einen Verbrauch von 23 mA beim Senden, 21 mA beim Empfangen und 3 μ A im Schlafmodus [33]. Bei Bestellungen ab 100.000 Stück kosten sie ungefähr 3\$ [131].

Das Protokoll unterscheidet zwischen *Controller* und *Slave* Knotentypen. Der primäre *Controller* verwaltet das Netzwerk, bindet neue Knoten ein und stellt Routing-Tabellen für alle anderen *Controller* bereit. *Slaves* können als Eingabe und Ausgabe dienen und werden normalerweise vom Controller angesteuert [33]. Mittels **Assoziation** können *Slaves* auch direkt verknüpft werden. So kann beispielsweise ein Bewegungsmelder direkt ein Licht schalten oder ein verkabelter Z-Wave Lichtschalter mit einem batteriebetriebenen, kabellosen Lichtschalter geschaltet werden [25].

ZigBee

Die ZigBee Allianz wurde 2002 gegründet und besteht aus einem Zusammenschluss von mehr als 230 Unternehmen [25]. Die untersten Protokoll und Funkschichten von ZigBee basieren auf dem offenen Standard IEEE 802.14.4. ZigBee entwickelte darauf die Netzwerk- und Routing-Schicht und eine Schicht zur Unterstützung von Applikationen. Es sind jedoch nur wenige ZigBee zertifizierte Produkte erhältlich [31]. Den Herstellern wird bei der Anwendungsimpementierung sehr viel Freiheit gelassen, wodurch ZigBee Geräte unterschiedlicher Hersteller meist nicht miteinander kompatibel sind [25]. ZigBee arbeitet primär auf dem 2.4 GHz Band, bietet jedoch die Möglichkeit wahlweise im 868 MHz Band zu operieren [31] [25].

io-homecontrol

io-homecontrol wurde 2002 von den Firmen Velux und Somfy gegründet. Velux ist spezialisiert auf die Entwicklung und Herstellung von Dachflächenfenstern und Sonnenschutzlösungen. Somfy produziert Antriebe und Steuerungen für Rollläden, Markisen, Hof- und Garagentore. Mittlerweile besteht das io-homecontrol Partnernetzwerk aus insgesamt 10 namhaften Herstellern und einem gewachsenen Produktumfang [132].

io-homecontrol ist kein offenes Protokoll. Ein Hardware Hersteller muss sich um eine Partnerschaft bewerben, um io-homecontrol kompatible Geräte vertreiben zu dürfen. Mit Hilfe der genauen Kontrolle des Partnernetzwerks soll der Verbund ausschließlich aus aktiven, namhaften Mitgliedern bestehen und sich durch absolute Interoperabilität auszeichnen [34].

KNX

Die KNX-Association wurde im Jahr 1999 mit dem Ziel gegründet einen offenen Standard für Heim- und Gebäudeautomation zu erschaffen. KNX entstand aus den unabhängigen Standards EIB, BCI und BatiBus und übernahm die Übertragungsmedien Twisted-Pair und Powerline. Zusätzlich dazu wurde der Funk-Übertragungsstandard KNX RF entwickelt. EIB wird jedoch weiterhin vermarktet.

KNX Module können mit einer Gruppenadresse versehen und zusammengefasst werden. So können mehrere Geräte gleichzeitig gesteuert werden und verhalten sich dabei wie ein Gerät. Dadurch wird außerdem der im Betrieb erzeugte Datenverkehr verringert.

EnOcean

EnOcean ist ein im Jahre 2001 gegründetes Spin-Off des Siemens-Konzerns. Mittlerweile sind der daraus entstandenen Allianz mehr als 100 Unternehmen beigetreten [25]. EnOcean zeichnet sich besonders durch die Philosophie des batterie- und kabellosen Betriebs aus. Daraus ergeben sich viele Vorteile wie etwa Ausfallsicherheit, geringerer Wartungsaufwand und niedrigere Umweltbelastung.

Die zum Betrieb benötigte Energie wird durch *Energy Harvesting/Scavenging* erzeugt. Neben Solarzellen kann beispielsweise auch mit Hilfe piezoelektrischer Elemente, die mechanische Energie beim Betätigen einer Taste dazu verwendet werden, um ein Signal zu senden.

Ein weitere Besonderheit an EnOcean ist die sehr hohe Datenrate von 120 kbit/s, sowie eine sehr geringe Datenmenge von nur 6 byte. Die Pakete sind somit ein Zehntel bis ein Hundertstel so groß wie die konkurrierender Technologien und die Übermittlung dauert weniger als eine Millisekunde. Dadurch wird das Kollisionsrisiko und der Energieverbrauch minimiert. Diese Tatsache begünstigt solarbetriebene Geräte zusätzlich, die durch die geringen Aufwendungen zur Kommunikation oft tagelang ohne Sonnenlicht auskommen [35] [30].

Aufgrund der vergleichsweise hohen Kosten konnte sich EnOcean bislang außerhalb von Europa noch nicht durchsetzen [25].

Protokoll	Datenrate	Sicherheit	Routing	Knoten
Z-Wave	9.6 kbit/s	●	✓	232 pro Controller
ZigBee	250 kbit/s	✓	✓	65536
io-homecontrol	38.4 kbit/s	✓	✗	?
KNX (RF)	16.4 kbit/s	✗	✗	57600 pro Domäne
EnOcean	120 kbit/s	✗	✗	2 ³²

Tabelle 2.2: Vergleich RF-basierender Systeme der Heimautomation

2.6 Ubiquitous-Computing

'A few places in the world have begun work on a possible next generation computing environment in which each person is continually interacting with hundreds of nearby wirelessly interconnected computers.' [36, S. 1]

'The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.' [37, S. 1]

Der Begriff *Ubiquitous-Computing* (*ubiquitous*, deutsch: allgegenwärtig) wurde von Mark Weiser eingeführt. Er hatte die Vision einer Umgebung aus hunderten drahtlosen, miteinander ver-

bundenen Computern, die auf die Bedürfnisse und Aktionen des Menschen reagieren. Die Computerisierung, die Geräte und deren Vernetzung soll für die BenutzerIn transparent und unsichtbar sein [36]. Dadurch werden die Geräte einfach unbewusst vom Menschen bei der Verrichtung alltäglichen Aufgaben genützt und von ihnen unterstützt [37].

Eine Unterscheidung zwischen *Ubiquitous-Computing* und *Pervasive-Computing* (*pervasive*, deutsch: durchdringend) ist schwer möglich. Mühlhäuser et al. [38] bezeichnet sie als Synonyme, wobei der Begriff *Ubiquitous-Computing* eher in der Industrie und *Pervasive-Computing* eher in der Wissenschaft Verwendung findet [38].

In Weisers ursprünglicher Vision aus dem Jahr 1991 gab es ubiquitäre Computer in drei verschiedenen Kategorien. Die *Active Badge* ist ein System zur Lokalisierung von Personen in Gebäuden. Das Gerät ist nicht größer als ein Mitarbeiterausweis und fällt in die Kategorie *Tab*. *Pads* sind portable Computer, die jedoch nicht wie sonst üblich einer einzelnen BenutzerIn gehören. Man transportiert sie auch nicht mit sich. Sie sind an vielen Orten im Haushalt zu finden und jeder kann sie benutzen. *Boards* sind große Bildschirme die beispielsweise in Pinnwände oder Tafeln integriert sind. Sie könnten automatisch die gewünschten Informationen einer BenutzerIn anzeigen. Ein *Board* kann auch als digitales Bücherregal dienen, mit dem die BenutzerIn Bücher und Zeitschriften auf ihre *Tabs* und *Pads* transferieren kann [37].

Folgende Elemente werden aus heutiger Sicht als grundlegende Eigenschaften von Ubiquitous-Computing angesehen:

- Ubiquitärer Zugang
- *Context Awareness* (deutsch: Kontextbewusstsein)
- Intelligenz
- Natürliche Interaktion

Ubiquitärer Zugang beschreibt die grundlegende Vision von Weiser. Allgegenwärtige Sensoren dienen aufgrund von Ereignissen und zeitlicher Parameter als Eingabe in die virtuelle Welt wobei Aktuatoren proaktiv oder reaktiv eine Ausgabe liefern [39]. **Context Awareness** beschreibt die Fähigkeit eines Systems, Objekte und Personen erkennen und lokalisieren zu können. Jegliche relevante Information wie die Präsenz anderer Entitäten, deren Emotionen und Intentionen soll durch das System verarbeitet werden. **Intelligenz** bezieht sich auf die Möglichkeit des Systems, sich an die BenutzerIn anzupassen und ihr Verhalten und ihre Gewohnheiten zu erkennen. Das Darstellen von Emotionen fällt auch unter diese Eigenschaft. **Natürliche Interaktion** schreibt eine natürlichere Art der Verwendung eines Systems vor als heutzutage üblich. Dazu gehört beispielsweise die Steuerung mittels Sprache und Gesten.

2.7 Tangible User-Interfaces

'In the last two decades, Tangible User Interfaces (TUIs) have emerged as a new interface type that interlinks the digital and physical worlds. Drawing upon users' knowledge and skills of interaction with the real non-digital world, TUIs show a potential to enhance the way in which people interact with and leverage digital information.' [40, S. 1]



Abbildung 2.9: Nachbau der Marble Answering-Machine [133]

Mit dem Begriff *Tangible User Interfaces* (*tangible* deutsch: greifbar) sind Benutzerschnittstellen gemeint, die die reale mit der digitalen Welt verknüpfen. Solche Schnittstellen bieten buchstäblich greifbare Informationen oder Steuerungen und können als Eingabe- oder Ausgabemöglichkeit dienen. Durch haptisches, visuelles oder auditorisches Feedback kann die BenutzerIn direkte Rückmeldung über ihre Aktionen erhalten [40].

Als möglicherweise das erste *Tangible User Interface* wird die *Slot Machine* vor Radia Perlman aus dem Jahre 1976 angesehen. Sie bot eine neuartige Benutzerschnittstelle für die *Logo turtle*. Dabei handelt es sich um einen Roboter, der mit der Programmiersprache Logo programmiert wird. Die Sprache und der Roboter entstanden aus Forschungsarbeiten zu Methoden, die Kindern Informatik näher bringen soll. Radia Perlman fand heraus dass die Benutzerschnittstelle zur Programmierung die größte Hürde für die Jugendlichen darstellte. Daraus entstand unter anderem die Slot-Machine. Die ProgrammiererInnen geben dabei die sequentiellen Aktionen der Schildkröte in Form von Karten in einen Schacht [41].

Die *Marble Answering Machine* von Durrell Bishop aus dem Jahr 1992 wird oft als sehr einflussreiches und inspirierendes Konzept genannt. Abbildung 2.9 zeigt einen prototypischen Nachbau des Geräts. Die *Marble Answering Machine* verwendet farbige Murmeln als physikalische Repräsentation von Sprachnachrichten. Erhält man eine Nachricht, rollt eine Murmel aus dem Ge-

rät in eine dafür vorgesehene Schale. Die Murmeln können anschließend in eine weitere Schale gelegt um die dazugehörige Nachricht abzuspielen [40].

Die visionäre Publikation *Tangible Bits* von Ishii und Ullmer [42] erwähnte erstmals den Begriff *Tangible User Interface*. In ihrer Vision wird die gesamte Welt zur Benutzerschnittstelle, durch die Integration alltäglicher Gegenstände und deren Verbindung mit digitaler Information [40] [43] [44]. Der Bereich des *Ubiquitous Computing* hat viele Gemeinsamkeiten mit ihrer Vision von *Tangible User Interfaces* und hatte nach eigenen Angaben großen Einfluss auf ihre Forschung [45].

2.8 Referenzprojekte

Im Folgenden werden Referenzprojekte näher betrachtet, die für die Zielsetzung dieser Arbeit als Inspiration und Informationsquelle dienen.

2.8.1 Umgebungssteuerung mit NFC und KNX

Llorente-Carmona et al. [46] führten ein Forschungsprojekt zur Heimsteuerung mittels eines NFC fähigen Mobiltelefons durch. Die Steuerung der Geräte wurde mittels KNX (kabelgebunden, siehe Abschnitt 2.5) gelöst. Dazu sendete das Mobiltelefon die Steuerungsaufgaben via Bluetooth an einen PC der als KNX Gateway fungierte. Das Ziel ihrer Arbeit war es älteren Menschen neue Interaktionsmöglichkeiten zur Steuerung von Geräten zu bieten. Sie setzten für das System folgende Detailziele:

- *Easy-To-Use*: Einfache Bedienung, niedriger Lernaufwand
- *Accessible*: Zugänglich für ältere Menschen
- *Useful*: Lebensqualität erhöhen, Abhängigkeit von anderen reduzieren
- *Attractive*: Der BenutzerIn Vorteile in der Bedienung bieten
- *Feasible and Scalable*: Leicht in bestehende KNX Installationen zu integrieren

In ihrer Arbeit erwähnen sie mehrmals das Problem des nicht Vorhandenseins von Schaltern an Stellen, an denen die BenutzerIn alltägliche Tätigkeiten durchführt. Außerdem betonen sie die Wichtigkeit möglichst viele Geräte in das Zielsystem zu integrieren. Sie definierten drei Anwendungsgebiete für die Interaktion via NFC:

- Steuerung von Geräten mit Infrarotschnittstelle
- Szenen Steuerung
- Positionskontext-sensitive Steuerung

Um die Interaktion mit Geräten, die über eine Infrarotschnittstelle gesteuert werden, zu demonstrieren, wurde ein NFC-Tag auf der Armlehne eines Sofas angebracht. Liest man den Tag ein, erscheint der Text 'ON' mit einer Filmklappe darunter auf dem Bildschirm. Tippt man auf den Bildschirm, wechselt der Text zwischen 'ON' und 'OFF'. Um die gewünschte Aktion durchzuführen, muss man das Gerät vom NFC-Tag entfernen. Ein vibrieren des Mobiltelefons bestätigt die erfolgreiche Ausführung.

Sie nutzten die NFC-Tags auch für das Starten von Szenen und definierten die im Folgenden zusammengefassten Szenarien. Die BenutzerIn kommt Zuhause an und legt dabei ihren Schlüsselbund in einen Korb nahe dem Eingang. Am Schlüsselbund befindet sich ein NFC-Tag und im Korb befindet sich ein NFC-Lesegerät. Durch das Einlesen des Tags wird automatisch das Licht eingeschaltet, das Radio spielt eine angenehme Musik und eine SMS wird an eine Pflegeperson geschickt, um zu signalisieren, dass die Person zuhause angekommen ist. Die Szene zum Verlassen des Hauses schaltet alle Geräte und Lichter aus und sendet ebenfalls eine SMS. Dazu muss die BenutzerIn einen NFC-Tag im Eingangsbereich einlesen. Am Nachttisch befindet sich ein NFC-Tag, auf dem die BenutzerIn ihr Mobiltelefon ablegt. Dadurch werden ebenfalls alle Geräte und Lichter ausgeschaltet. Durch das Einlesen von NFC-Tags auf der Rückseite von Büchern wird eine passende Lichtszene aktiviert.

Als weitere Möglichkeit nennen sie die Verwendung eines kommerziellen Systems zur Positionsbestimmung von Personen im Haus durch ein spezielles Armband. Dabei könnte beispielsweise ein NFC-Tag in der Armlehne eines Rollstuhls dazu dienen, Geräte in der Umgebung zu steuern.

Sie führten Benutzertests in einem Altersheim mit 14 Personen im Alter von 60 bis 74 Jahren durch. Dabei testeten sie jedoch nicht die vorher definierten Szenarien sondern beschränkten sich ausschließlich auf die Grundlegende Interaktion mit NFC. Die meisten BenutzerInnen waren beeindruckt von der Technologie und davon, dass sie die Prinzipien der Interaktion nach einer kurzen Einführung verstanden haben.

Als Vorteile der Interaktion nannten sie den niedrigen Lernaufwand und die intuitive Steuerung. Ältere Leute tragen jedoch nicht gerne immer ein Mobiltelefon bei sich, wodurch andere Lösungen wie etwa ein Armband mit NFC Lesegerät besser angenommen werden könnten.

Als Vorteile der Hardwarelösung sehen sie die große Anzahl an verfügbaren Geräten und den geringen Kosten. Die geringe Verbreitung von NFC in Mobiltelefonen und die Notwendigkeit einen PC als Gateway zu verwenden, würden sich jedoch negativ auf die Benutzerakzeptanz auswirken [46].

Die in der Arbeit von Llorente-Carmona et al. [46] erwähnte Problematik der nicht vorhandenen Schalter ist nicht vollkommen schlüssig. Das Problem ließe sich durch die Verwendung drahtloser Geräte lösen. Somit könnten zusätzliche Schalter auch auf rollstuhlgerechter Höhe angebracht werden. Die niedrigen Kosten für Hardware aus dem Bereich Heimautomation, sind

aus gegenwärtiger Sicht auch nicht nachvollziehbar und stehen in Konflikt mit der Schalter-Problematik. Einer der Vorteile des Systems das in dieser Arbeit entwickelt wird, sind die niedrigen Kosten von NFC-Tags im Vergleich zu den Kosten eines zusätzlichen Schalters.

Ein weiterer Kritikpunkt ist die geringe Involvierung der BenutzerInnen in den Designprozess und die anschließende Evaluierung. Daher steht die Analyse der Benutzerakzeptanz und die Entwicklung neuer Interaktionen im Vordergrund dieser Arbeit. Mit Hilfe qualitativer Methoden soll die Interaktion der BenutzerInnen mit der Umgebung genauer untersucht werden. Verschiedenste Szenarien sollen dabei durchgespielt werden, um mögliche Probleme aufzudecken.

Die vorgestellten Szenarien bezüglich Steuerung von Haushaltsgeräten ist nicht ausreichend für eine sinnvolle Verwendung. Die Interaktion mit dem Fernseher wirkt starr und langwierig, wenn man bedenkt, dass sich die NFC-Antenne des verwendeten klappbaren Mobiltelefons in der oberen Schale des Geräts befindet [15]. Darüber hinaus benötigt die BenutzerIn danach ohnehin eine Fernbedienung, um den Fernseher im Detail zu bedienen. Die dazugehörigen Szenarien sollen daher in dieser Arbeit noch wesentlich realistischer gestaltet werden und der BenutzerIn soll eine Möglichkeit zur weiteren Detailsteuerung gegeben werden.

Die Empfehlung, möglichst viele Geräte in die Steuerung zu integrieren, wird von dieser Arbeit angenommen.

2.8.2 SiteView

SiteView [47] ist ein *Tangible User Interface* zur Steuerung und Programmierung von Heimautomatisierungsanlagen. Es adressiert das Problem des nötigen technischen Hintergrundwissens und die Ungewissheit über die Folgen der Interaktionen.

Das notwendige technische Hintergrundwissen soll mit dem *Tangible User Interface* minimiert werden. Es besteht aus einem Raumplan (*World-In-Miniature*), drei RFID-Sensoren (*Condition Composer*) und zahlreichen *Tangibles* für Bedingungen und Aktionen (*Interactors*). Feedback für ihre Handlungen erhält die BenutzerIn über die Regelanzeige (*Rules Display*) und die Umgebungsanzeige (*Environment Display*).

Die Regelanzeige zeigt der BenutzerIn die aktuell angewendete Regel sowie eine Liste aller im System gespeicherten Regeln. Beim Durchsuchen von Regeln können mit Hilfe der *Condition Composer* Bedingungen für die Filterung der Liste definiert werden. Die Regelanzeige dient als Feedback bei der Erstellung von Regeln sowie als Fehleranzeige. Regeln werden als Sätze in natürlicher Sprache dargestellt, um leichter verstanden werden zu können.

Mit Hilfe von 192 Fotos des Raums, in dem die Applikation integriert wurde, stellt die Umgebungsanzeige die Auswirkungen von Regeln und Aktionen dar. Die Anzeige wird auf die Rückseite einer 60 cm x 90 cm großen und transluzenten Oberfläche projiziert. Durch die Verwendung der selben Maße für den Raumplan wird der Realismus der *World-In-Miniature* gesteigert [47].

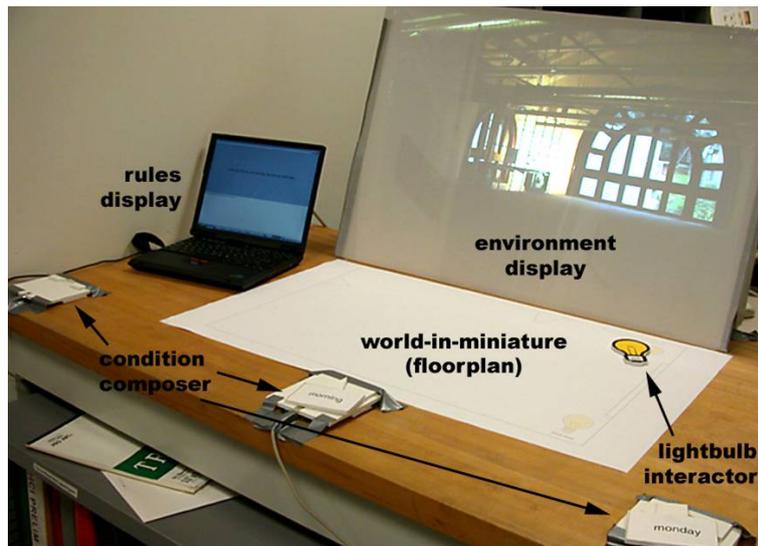


Abbildung 2.10: SiteView [47]

Bei den *Tangibles* für Aktionen handelt es sich um keine RFID Tags. Statt dessen wird ihre Positionierung durch optische Methoden festgestellt. Dabei handelt es sich um die Aktionen ‘Licht an’, ‘Licht aus’ und ‘Temperatur stellen’. Das Einstellen der Temperatur wurde mit einem linearen Schiebepotentiometer von Phidgets gelöst, mit dem die BenutzerIn eine Temperatur zwischen 13 °C und 30 °C für den Thermostat setzen kann.

Die *Tangibles* für Bedingungen wurden mit RFID Tags realisiert. Sie haben an der oberen Kante einen zur Kategorie (Wetter, Wochentag, Tageszeit) passende geometrische Form (Halbkreis, Quadrat, Dreieck). Die selben geometrischen Formen fanden sich als Schaumstoffaussparung auf den RFID Lesegeräten wieder. Um Interferenzen zu vermeiden, wurden die *Condition Composer* 75 cm voneinander entfernt angebracht.

In dieser Arbeit wird die Unterscheidung der Tags verschiedener Kategorien von Aktionen mittels Grafik, Farbe und Form aufgegriffen.

Methoden

In diesem Kapitel werden jene Methoden näher betrachtet, die für den Entwicklungsprozess in Erwägung gezogen wurden. Die Abschnitte zu **iterativen Entwicklungsprozessen** und **User-Centered Design (UCD)** geben einen tiefen Einblick in Methoden zur Entwicklung besserer Systeme. Im folgenden Abschnitt wird der Begriff **Usability** am Beispiel verschiedener Definitionen näher erläutert und ein genauerer Blick auf ausgewählte Teilbereiche geworfen. Abschließend wird mit dem **Technology Adoption Lifecycle** die Integration von Innovationen in das tägliche Leben der Menschen betrachtet.

3.1 Iterative Entwicklung

Softwareentwicklungsprozesse unterteilen den Projektablauf typischerweise in abgegrenzte Phasen wie beispielsweise:

- Planung
- Spezifikation
- Design
- Implementierung
- Integration
- Testen
- Betrieb
- Wartung

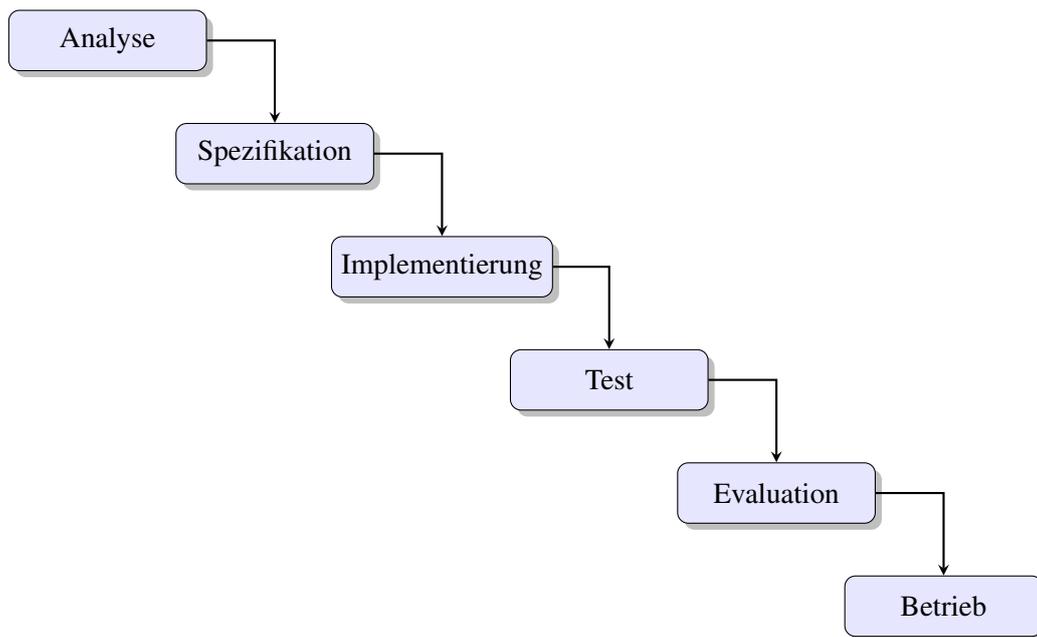


Abbildung 3.1: Klassisches Wasserfallmodell (reproduziert nach [49])

Die Anzahl der Phasen und deren Ziele können je nach Prozess und Einsatzgebiet variieren. Um die Entstehung und die Besonderheit iterativer Entwicklungsprozesse besser verstehen zu können, wird zuerst das klassische Wasserfallmodell näher erläutert. Beim Wasserfallmodell handelt es sich nicht um einen iterativen Entwicklungsprozess. Die einzelnen Phasen werden einmalig nacheinander durchlaufen. Änderungen der Artefakte aus früheren Phasen sind hierbei nicht vorgesehen. Abbildung 3.1 veranschaulicht den Prozess. Eine gute und ausgiebige Analyse und Spezifikation im Vorhinein soll hierbei zielführend wirken und teure Änderungen in späten Projektabschnitten verhindern [48]. Eine Studie von Cusumano und Selby [50] aus dem Jahr 1997 zeigte jedoch, dass 30% der Anforderungen in Projekten von Microsoft nicht im Vorhinein erhoben wurden, sondern erst während der Entwicklung aufkamen. Es ist also oft nicht möglich, sie in einer sehr frühen Projektphase vollständig zu erheben. Ebenso falsch ist es zu versuchen, einmal erstellte Spezifikationen komplett unangetastet zu lassen. Projekte können daran scheitern oder ein unbrauchbares Produkt hervorbringen. Dieser Umstand ist die größte und am meisten kritisierte Schwäche des Wasserfallmodells. Das Modell verlor schließlich an Bedeutung, da Entwicklungsprozesse die besser auf Feedback reagieren konnten, in der Regel bessere Produkte hervorbrachten [50].

The Chaos Report der Standish Group [51] aus dem Jahr 1995 listet ‘unvollständige Anforderungen’ und ‘sich ändernde Anforderungen’ auf Platz 2 und 3 der größten Risiken für *Challenged Projects*. Dabei handelt es sich um abgeschlossene Projekte, die ihr zeitliches und finanzielles Budget überschritten haben und mit weniger als der ursprünglich spezifizierten Funktionalitäten ausgeliefert wurden. ‘Unvollständige Anforderungen’ belegt sogar Platz 1 der Risiken für fehlgeschlagene Projekte [51].

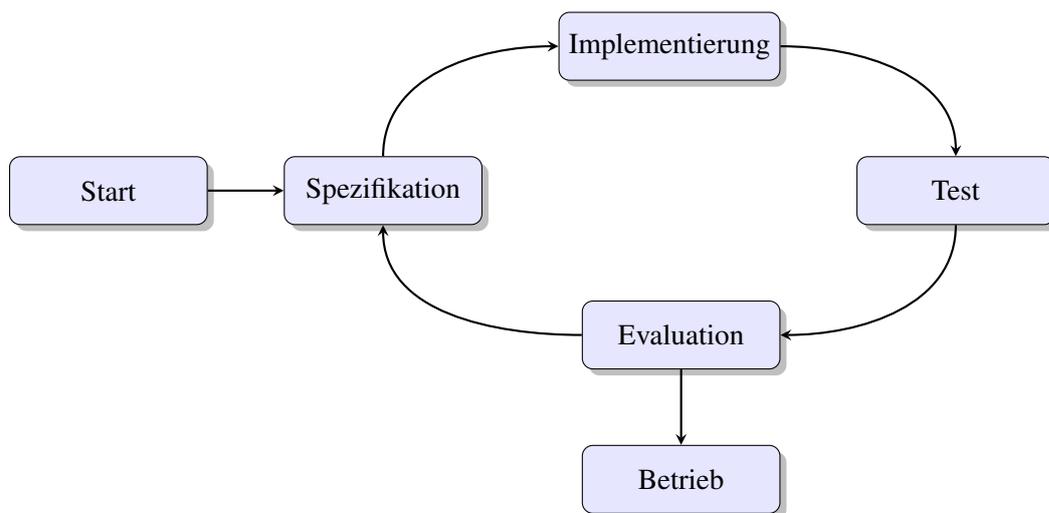


Abbildung 3.2: Iterativer Prozess (abgeändert, reproduziert nach [52])

Iterative Prozesse wirken diesem Problem entgegen, indem sie das wiederholte sequentielle Durchlaufen der einzelnen Phasen vorschreiben. Abbildung 3.2 veranschaulicht das Prinzip. In jeder Iteration werden Anforderungen erhoben, Funktionalitäten spezifiziert und implementiert, Tests durchgeführt, deren Ergebnisse ausgewertet und als Grundlage für die nächste Iteration verwendet. Durch die regelmäßige Evaluierung und Planung kann besser auf Änderungen reagiert werden. Die Anzahl der Iterationen ist hierbei variabel und wird entsprechend der Gegebenheiten gewählt. Doch Anforderungen ständig zu ändern ist ebenso nutzlos wie sie über den kompletten Projektverlauf beibehalten zu wollen. Daher ist es empfehlenswert, sie zumindest innerhalb einer Iteration unangetastet zu lassen [48].

3.2 User-Centered Design (UCD)

Der Begriff *User-Centered Design* erlangte durch eine Publikation von Donald A. Norman [53] im Jahr 1986 Bekanntheit. Er betont die Wichtigkeit, ein gutes Verständnis für die BenutzerIn zu haben, um bessere Systeme zu erschaffen. Die Miteinbeziehung der BenutzerIn wurde damals jedoch nicht explizit erwähnt [54].

‘The needs of the users should dominate the design of the interface, and the needs of the interface should dominate the design of the rest of the system.’ [54, S. 1]

Constantine und Lockwood bezeichnen *User-Centered Design* in [55] als eine lose Ansammlung von Methoden zu einer Philosophie des Verständnis und der Miteinbeziehung von BenutzerInnen in den Designprozess eines Systems. Gleichzeitig erinnern sie die LeserIn daran, dass die Anwendung von UCD Methoden gute Designarbeit nicht ersetzen kann [55].

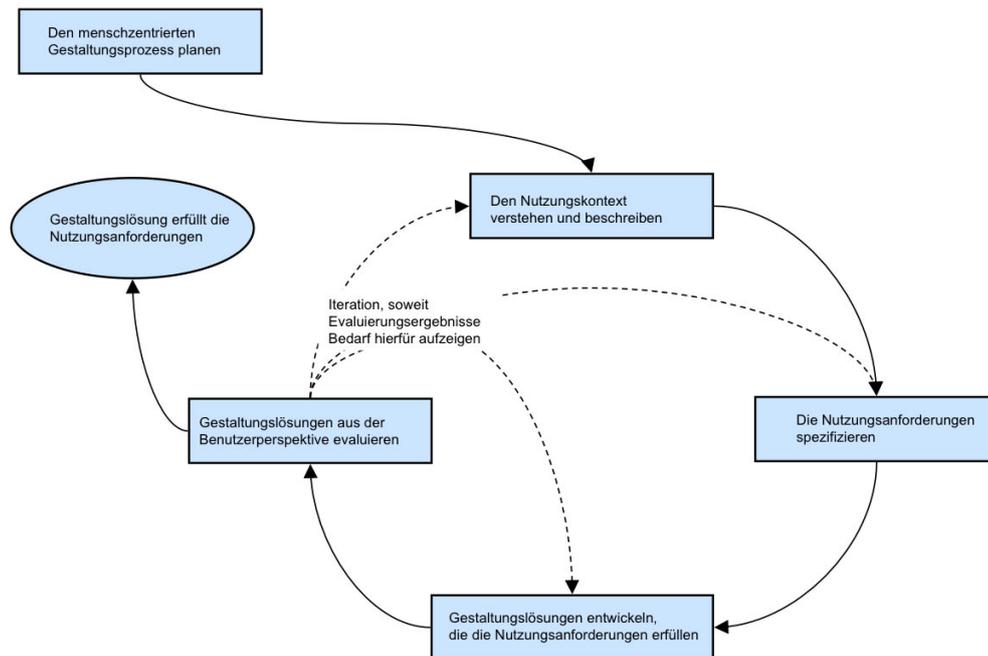


Abbildung 3.3: Benutzerzentrierter iterativer Designprozess nach ISO 9241-210 [134]

Der Grundgedanke, Feldstudien durchzuführen und Produkte speziell an die BenutzerIn anzupassen, entstand bereits in den frühen Anfängen des industriellen Designs (1955-1964) [56]. Die Philosophie des UCD findet sich auch in den anderen, eng verwandten Methoden wie beispielsweise im *Goal-Directed Design*, *Participatory Design* und *Human-Centered Design* wieder [57] [134] [58]. Unter letzterem Namen ist der ISO Standard für benutzerzentriertes System Design bekannt. Der ISO Standard ISO 9241-210 *Human-Centred Design For Interactive Systems* ersetzte 2010 seinen 11 Jahre alten Vorgänger ISO 13407. In der neuen Fassung wurden die grundlegenden Inhalte weiter konkretisiert. Neben der Gestaltung durch ein Verständnis der BenutzerIn, ihrer Aufgaben und ihrem Umfeld wird auch die Anwendung eines iterativen Prozesses (siehe Abschnitt 3.1) als essenziell erachtet. Dies ist notwendig, um die Lösungen basierend auf den Ergebnissen der Evaluierung weiterzuentwickeln. Abbildung 3.3 zeigt den iterativen Designprozess nach ISO 9241-210. Die BenutzerInnen sollen in den Design- und Entwicklungsprozess aktiv miteinbezogen werden und mit interdisziplinären Designteams zusammenarbeiten [134].

Im Folgenden werden unterschiedliche Methoden des *User-Centered Design* vorgestellt und im Detail betrachtet. Anschließend werden mit Interviews (Abschnitt 3.6) und Beobachtungen (Abschnitt 3.7) zwei wichtige Techniken der Feldforschung näher beleuchtet.

3.3 Personas

Der Begriff ‘Persona’ wurde von Alan Cooper [59] eingeführt. Er ist vor allem für seine zahlreichen Bücher und Publikationen im Bereich des Interaktionsdesign bekannt. Seine Methoden fasst er unter dem Titel ‘Goal-Directed Design (GDD)’ zusammen [58]. Bei Personas handelt es sich um fiktionale Personen, die als Design- und Entwicklungswerkzeug verwendet werden. Der Gebrauch von Benutzerprofilen entstand jedoch bereits früher und fand seine Anwendung vor allem im Bereich des Marketing [60].

Bei einer Persona handelt es sich keinesfalls bloß um die durchschnittliche BenutzerIn eines Systems, den sogenannten *Elastic User*. Die elastische BenutzerIn ist eine fiktive BenutzerIn, die je nach Bedarf Eigenschaften zugesprochen bekommt, um Designentscheidungen zu rechtfertigen. Des Weiteren sollen Personas dabei helfen, Randfälle zu identifizieren und *Self-Referential Design* zu verhindern. *Self-Referential Design* (deutsch: selbstbezogen) entsteht typischerweise, wenn DesignerInnen Systeme nach ihren eigenen Bedürfnissen und Zielen designen. Dadurch können Systeme entstehen, die für andere Benutzergruppen nur schwer zu verstehen und zu benutzen sind [58].

Personas bieten den DesignerInnen und EntwicklerInnen ein konkreteres Bild darüber, für wen sie das System entwickeln und für wen nicht. Außerdem können sie den Projektbeteiligten eine Gesprächsbasis bieten. Personas sollen möglichst lebendig, glaubwürdig und detailliert beschrieben werden, um wie reale Personen wahrgenommen werden zu können. Dazu gehört unter anderem auch die Angabe von Alter, Geschlecht, Aussehen und privatem Umfeld neben deren Bedürfnissen und Zielen [61].

Personas können in jedem Umfeld eingesetzt werden, um bestehende Methoden und Prozesse zu bereichern und benutzerzentriertes Denken zu fördern [62]. Anders Toxboe empfiehlt in *Introducing User-Centered Design to eXtreme Programming* [63] beispielsweise die Priorisierung der *User Stories* (siehe Abschnitt 3.4) für die Implementierung mit Hilfe von Personas [63].

Für jedes Projekt werden 3-12 Personas erstellt und je nach Wichtigkeit und Rolle ein Typ zugewiesen [64]. Primäre Personas stellen die primäre Zielgruppe dar, deren Bedürfnisse es unbedingt zu befriedigen gilt. Es kann maximal eine primäre Persona pro Benutzerschnittstelle existieren. Im Falle von mehreren spezialisierten Schnittstellen kann es jedoch mehrere primäre Personas geben. Für sekundäre Personas wird keine spezielle Benutzerschnittstelle erstellt. Sie werden dazu benutzt, zusätzliche spezielle Bedürfnisse zu verdeutlichen. Zu viele sekundäre Personas können jedoch auch ein Zeichen für einen zu breit gestreuten Fokus sein. Ergänzende Personas stellen eine Kombination der Bedürfnisse von primären und sekundären Personas dar. Negative Personas sollen verdeutlichen, für wen das System nicht entworfen wird. Es können jedoch auch Personas eingesetzt werden, die nicht zu den möglichen BenutzerInnen des Produkts zählen. *Customer Personas* befassen sich mit den Bedürfnissen der KundInnen. *Served Personas* stehen in direktem Einfluss des Produkts, wie etwa PatientInnen die mit einem Gerät zur Therapie behandelt werden [58].

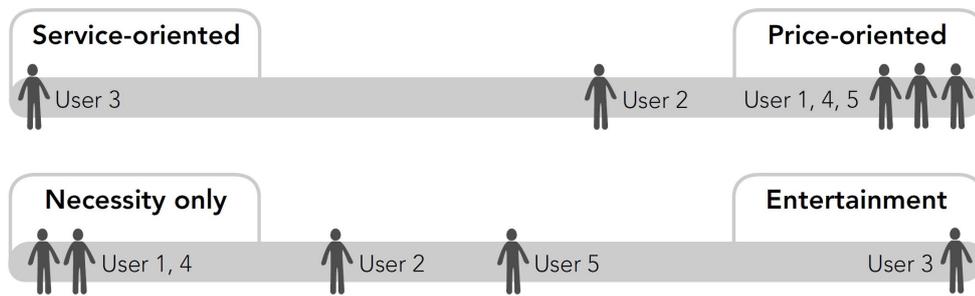


Abbildung 3.4: Darstellung der BenutzerInnen auf den Verhaltensachsen [58]

Cooper beschreibt in [58] einen systematischen Weg zur Erstellung von Personas. Dazu werden verschiedene Verhaltensvariablen (*Behavioral Variables*) definiert, die relevant für das Produkt sind. Demographische Variablen wie Alter, Geschlecht oder Herkunft sollen dafür jedoch nicht verwendet werden. Informationen wie beispielsweise die Erwartungen an das Produkt, unterschiedliche Motivationen zur Verwendung oder spezielle Fähigkeiten der BenutzerInnen erweisen sich als viel aussagekräftiger. Anschließend werden Interviews mit der Zielgruppe geführt und die TeilnehmerInnen bezüglich der Variablen eingeschätzt. Dafür eignet es sich besonders, die Ergebnisse auf einer Verhaltensachse darzustellen (siehe Abbildung 3.4). Das Abstandsverhältnis zwischen den Personen ist hierbei wichtiger als eine möglichst genaue Positionierung zu finden. Mit den so entdeckten Häufungspunkten werden schließlich entsprechende Rollen mit Zielen definiert. Anschließend werden sie mit textuellen Beschreibungen möglichst lebensecht gestaltet und einem Typ zugewiesen [58].

3.4 Szenarios

Jakob Nielsen definiert ein Szenario als eine in erzählender Form geschriebene Geschichte, über eine Person, die in einem Kontext und in einer spezifischen Situation ein Produkt verwendet, um ein bestimmtes Ziel zu erreichen [65]. Sie sind ein effektives Werkzeug zur Dokumentation von Anforderungen an ein System. Nach John M. Carroll besteht ein Szenario aus den folgenden Elementen [66]:

- Situation
- Akteure
- Ziele
- Handlungen
- Ereignisse
- Objekte

Die Methode des *Scenario-Based Design* wurde ebenfalls von Alan Cooper aufgegriffen und in seine Methodensammlung *Goal-Directed Design* integriert. Er empfiehlt ausdrücklich die Verwendung von Personas in Kombination mit Szenarios. Er kritisiert, dass die Verwendung von Rollen als Akteure nicht ausreichend konkret sei, um ein Verständnis für die BenutzerIn aufbringen zu können. Die ursprüngliche Methodik von Carroll schlägt zwar die Definition von Zielen vor, jedoch nur im Zusammenhang mit den Veränderungen, die ein Akteur in der konkreten Situation anstrebt. Dabei werden persönliche Ziele und Motivationen zu sehr außer Acht gelassen [58].

In dem Buch *Discovering Requirements - How to Specify Products and Services* von Alexander und Beus-Dukic [67] wird der Begriff Szenario nicht als spezifische Dokumentationsart aufgefasst. Laut ihrer Definition umfasst er verschiedene Arten zur effektiven Dokumentation der Benutzung eines Systems und dessen Anforderungen. Folgende vier Formen werden dabei genannt [67]:

- *User Stories*: Erzählende Beschreibung
- *Storyboards*: Graphische Darstellung
- *Operational Scenarios*: Detaillierte, sequentielle und nummerierte Beschreibung der einzelnen Handlungen
- *Use Cases*: Umfassen mehrere Szenarios (meist *operational-scenarios*) um einen Anwendungsfall möglichst umfassend zu dokumentieren

In weiterer Folge werden *Use Cases* näher betrachtet.

3.5 Use-Cases

Bei *Use Cases* (deutsch: Anwendungsfälle) handelt es sich um eine spezielle Form von Szenarios (siehe Abschnitt 3.4). Mit ihnen können die funktionalen Anforderungen an ein System sehr konkret und umfassend dokumentieren werden. Dabei wird das Systemverhalten durch eine genaue Definition von Aktionen und Eingaben der BenutzerIn und dazugehörigen Ausgaben beschrieben [58]. Abbildung 3.5 veranschaulicht, dass ein *Use Case* nicht nur einen einzigen der möglichen Handlungsstränge beinhaltet, sondern auch alternative Wege und Fehlerfälle beschreibt. Meist werden zur Beschreibung des Ablaufs *Operational Scenarios* (siehe Abschnitt 4.2.4) verwendet. Die Verwendung von *User Stories* und *Storyboards* ist aber nicht ausgeschlossen [67]. Sogar Kontrollflussdiagramme, Sequenzdiagramme, Petri-Netze und Programmiersprachen können zur Beschreibung des Ablaufs eingesetzt werden. Eine einfache textuelle Form wird jedoch empfohlen, da sie kein spezielles Vorwissen erfordert, um möglichst gut verstanden werden zu können [68].

Für die Dokumentation empfiehlt es sich eine Vorlage, wie etwa die aus dem Buch *Writing Effective Use Cases* von Alistair Cockburn [68] zu verwenden [67]. Das Buch gibt einen genauen Einblick in das Thema *Use Cases* und beschreibt anerkannte Praktiken zur Erstellung und

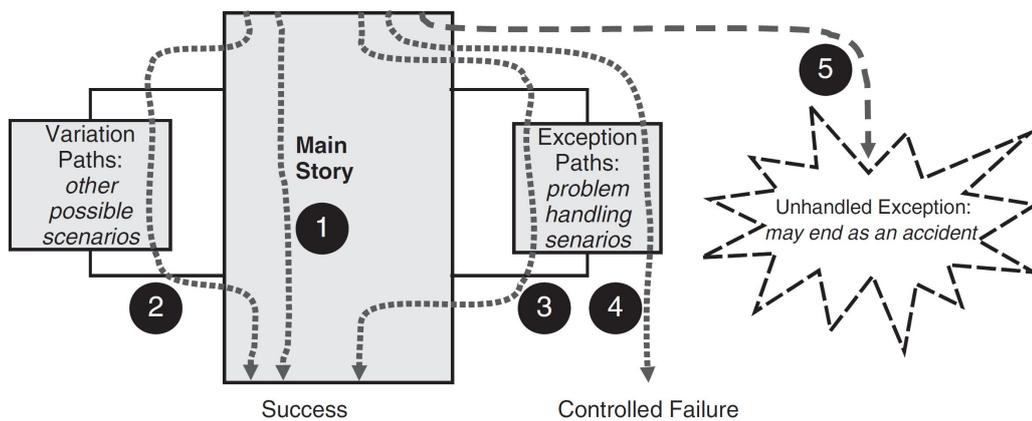


Abbildung 3.5: Use Case Diagramm [67]

Verwendung. Die Maximalversion seiner *Use Case* Vorlage enthält 20 zu adressierende Punkte. Darunter finden sich beispielsweise Felder wie:

- Akteure
- Interessen der Akteure und beteiligter Personen/Unternehmen
- Ziele
- Detailgrad
- Vorbedingungen
- Hauptszenario
- Alternative Pfade (sogenannte *Extensions*)
- Priorität

Es wird jedoch ausdrücklich empfohlen, das Formular gegebenenfalls zu adaptieren. Dazu soll auf Projektbasis über eine Form, einen Detailgrad und die Toleranz für Variationen in der Gestaltung entschieden werden. Eine zu formale und starre Vorlage kann dem Projekt wertvolle Zeit und Energie kosten [68].

3.6 Interviews

‘Unter Interview als Forschungsinstrument sei hier verstanden ein planmäßiges Vorgehen mit wissenschaftlicher Zielsetzung, bei dem die Versuchsperson durch eine Reihe gezielter Fragen oder mitgeteilter Stimuli zu verbalen Informationen veranlasst werden soll.’ [69, S. 71]

Bei Interviews handelt es sich um eine soziale Situation die über verbale Kommunikation strukturiert wird [70]. Durch das Ausmaß an Strukturierung lassen sich unterschiedliche Typen von Befragungen definieren. Mit Struktur ist in diesem Zusammenhang die Menge an Kontrolle gemeint, die durch die Interviewer auf die Kommunikation ausgeübt wird, indem eine vorgefertigten Liste an Fragen abgearbeitet wird [44]. Man unterscheidet zwischen den drei Formen **wenig strukturiert**, **teilstrukturiert** und **stark strukturiert** [70]. Die verschiedenen Typen von Interviews haben Vor- und Nachteile und eignen sich daher mehr oder weniger gut zum Erreichen bestimmter Ziele. Die Wahl der Methode wird je nach Zielsetzung getätigt [71]. Zur Dokumentation von Interviews empfiehlt es sich, textuelle Notizen zu machen sowie Tonaufnahmen durchzuführen [67] [58].

Man unterscheidet außerdem zwischen geschlossenen und offenen Fragen. **Geschlossene Fragen** verlangen eine präzise Antwort, wie beispielsweise Fragebögen mit vorgegebenen Antwortmöglichkeiten. Sie sollen möglichst kurz, klar und eindeutig gestellt werden. **Offene Fragen** hingegen geben den Interviewten viele Möglichkeiten, auf eine Frage zu antworten [44].

Beim **wenig strukturierten** Interview nimmt der Interviewer eine passive Rolle ein und die befragte Person bestimmt den Lauf des Gesprächs. Diese Technik wird vor allem in frühen Phasen von Untersuchungen angewendet, in denen konkrete Themengebiete und dazugehörige Fragestellungen noch nicht identifiziert wurden [70]. Den Interviewenden stehen zur Durchführung fallweise Gesprächsleitfäden zur Verfügung, die das Ziel und relevante Themengebiete beschreiben. Das Interview soll dabei sowohl in die Breite als auch in die Tiefe gehen und wird deshalb auch Tiefen- und Intensiv-Interview genannt [71]. Durch die große Freiheit bei der Durchführung hat der Interviewende die Möglichkeit, die Fragestellungen und den Gesprächsverlauf individuell zu gestalten. Solche informellen Befragungen können sowohl mit Einzelpersonen als auch mit Gruppen durchgeführt werden [70]. Die Ergebnisse solcher Studien sind nicht standardisiert und daher auch nicht vergleichbar. Die Dokumentation und Auswertung eines solchen Interviews ist aufwändiger als bei anderen Techniken [71].

Interviews auf Grundlage von fest vorgegebenen, standardisierten, geschlossenen Fragen werden als **stark strukturierte** Interviews bezeichnet [70]. Solche Befragungen können auch mit schriftlichen Fragebögen durchgeführt werden. Sie eignen sich besonders, wenn die Studie klare Ziele verfolgt und zu spezifischen Themen Klärungsbedarf besteht [44]. Allen Interviewten werden die selben Fragen gestellt, wodurch die Ergebnisse standardisiert und vergleichbar sind [71].

Das **teilstrukturierte** Interview verwendet sowohl offene Fragen als auch geschlossene Fragen. Dabei wird typischerweise ein Gesprächsleitfaden verwendet, der vordefinierte Fragen enthält. Anschließend an die Beantwortung einer Frage wird versucht, dem Interviewten tiefer gehende Informationen zu entlocken [44]. Die Ergebnisse einer solchen Studie sind durch ihre geringe Standardisierung nur selten vergleichbar [71].

3.7 Beobachtungen

Beobachtungen können unter kontrollierten Laborbedingungen oder in der natürlichen Umgebung des Beobachteten durchgeführt werden. Wie bei Interviews gibt es Vielzahl unterschiedlicher Techniken, aus denen es je nach Ziel, Fragestellung und Einschränkungen auszuwählen gilt [44]. Zusätzlich zu den Eigenschaften **Strukturiertheit** und **Offenheit** unterscheidet man Beobachtungstechniken auch Anhand der **Teilnahme** des Beobachters [72]. Zur Dokumentation von Beobachtungen empfiehlt es sich ebenfalls, Bild- und Tonmaterial zu sammeln. [67]

Die **Offenheit** ist dabei ein Maß für die Transparenz der Beobachtungssituation für die Beobachteten. Bei verdeckten Beobachtungen wissen die beobachteten Personen nicht, dass sie beobachtet werden. Der Beobachter arbeitet dabei mit einer Tarnung oder nutzt räumliche Gegebenheiten. Die verdeckte Beobachtung gilt jedoch als forschungsethisch umstritten, da sie das Vertrauensverhältnis zwischen Beobachter und den Beobachteten missbraucht. Bei offenen Beobachtungen wissen die Betroffenen von der Beobachtung. Die anfänglichen Verhaltensveränderungen, die durch die Beobachtungssituation entstehen, sind meist vernachlässigbar, da sie nach einiger Zeit wieder abklingen.

Bei einer **strukturierten Beobachtung** sind die zu beobachteten Ereignisse klar definiert und die anfallenden Daten dadurch standardisiert. Im Fall von hoch-strukturierten Beobachtungen spricht man auch von quantitativer Beobachtung. Die **unstrukturierte Beobachtung** hingegen folgt nur den Fragestellungen der Forschung und ist inhaltlich nicht limitiert.

Die **Teilnahme** des Beobachters ist ebenfalls ein wesentlicher Punkt. Man unterscheidet dabei zwischen aktiv und passiv teilnehmender Beobachtung sowie nicht-teilnehmender Beobachtung. Bei einer aktiven Teilnahme ist die soziale Interaktion des Beobachters in der beobachteten Gruppe Bestandteil der Methode. Ein passiver Beobachter beschränkt sich jedoch ausschließlich auf die Beobachterrolle [72].

Preece et al. unterscheiden in dem Buch *Interaction design: beyond human-computer interaction* [44] außerdem zwischen direkten und indirekten Beobachtungstechniken. Neben *Interaction Logs* fallen in ihrer Definition *Diaries* (siehe Abschnitt 3.8) in die Kategorie der indirekten Beobachtungen [44].

3.8 Diary studies

Nach Preece et al. [44] gehören *Diary Studies* zu den indirekten Beobachtungsmethoden (siehe Abschnitt 3.7) [44]. *Diary Studies* sind eine Methode, um Aktivitäten wie die Verwendung eines Produkts und die Gedanken des Benutzers zu dokumentieren [73]. Sie können bereits in sehr frühen Phasen eines Projektes eingesetzt werden, um Abläufe besser verstehen zu können und Anforderungen zu erarbeiten. In späten Phasen können sie dazu dienen, die Verwendung eines bestehenden Produkts zu erforschen und als Usability Test eingesetzt werden. Besonders eignen sie sich bei der räumlich verteilten Verwendung eines Systems oder für Langzeitstudien, bei

denen die Anwendung anderer Methoden zu kosten- und zeitintensiv oder schlichtweg nicht möglich ist. Steve August schlägt die Unterscheidung in vier verschiedene Arten von *Diary Studies* vor [73]:

- *Usage Diary*: Dokumentation der Interaktion mit einem System. Dies kann nach Bedarf auch eine Beschreibung der Aktivitäten außerhalb der direkten Benutzung des Systems beinhalten.
Beispiel: Dokumentation der Verwendung eines elektronischen Fahrkartenschalters von der Ankunft am Bahnhof bis zur Abfahrt.
- *Spotter Diary*: Wann, wo und wie ist das Systems im Leben des Benutzers präsent.
Beispiel: Zeitpunkte und Orte, an denen sich die BenutzerIn mit einem Produkt beschäftigt, indem sie es benutzt, darüber spricht oder daran denkt.
- *Process Diary*: Dokumentation eines längeren, zeitlich und räumlich verteilten Prozesses. Enthält typischerweise längere Erzählungen.
Beispiel: Dokumentation einer kompletten Hochzeitsplanung.
- *Behaviour Diary*: Dokumentation des Verhältnis zu einem bestimmten Thema.
Beispiel: Dokumentation des Verhältnis und der Gedanken zum Thema Geld über die Weihnachtszeit.

Die Unterscheidung gibt einen groben Überblick über die Möglichkeiten von *Diary Studies*. Die Gestaltung einer solchen Studie sollte sich an den Fragen, die man dadurch beantworten möchte, orientieren. Es empfiehlt sich daher, den Ablauf der Studie genau zu definieren und festzulegen. Dazu gehört die Definition von Aufgaben, wann diese von den TeilnehmerInnen durchgeführt werden sollen, die Art der Dokumentation und ein zeitlicher Rahmen. Die Art der Dokumentation kann von strukturierten Daten wie Tabellen und Fragebögen bis hin zu unstrukturierten Daten wie Erzählungen und Fotos reichen [73].

Cultural Probes sind eine spezielle Form der *Diary Study* [73]. Dabei handelt es sich um Pakete, die an die TeilnehmerInnen der Studie ausgehändigt werden. Klassische Beispiele für den Inhalt einer *Cultural Probe* sind Utensilien wie Postkarten, Stadtpläne, Notizblöcke und Einwegkameras. Mit den Materialien sollen anschließend bestimmte Aufgaben von den TeilnehmerInnen durchgeführt werden, wie zum Beispiel:

- Mach ein Foto von der ersten Person, die du morgen siehst.
- Schreibe auf die Postkarte, welches Gerät du am liebsten hast.
- Markiere Orte auf der Karte, die dich zum Tagträumen einladen.

Die Ergebnisse *Cultural Probes* sind nicht analysierbar oder eindeutig interpretierbar. Sie haben zum Zweck der DesignerIn Inspiration zu geben und zum Nachdenken anzuregen, wer die BenutzerInnen des zu entwickelnden Systems sind.

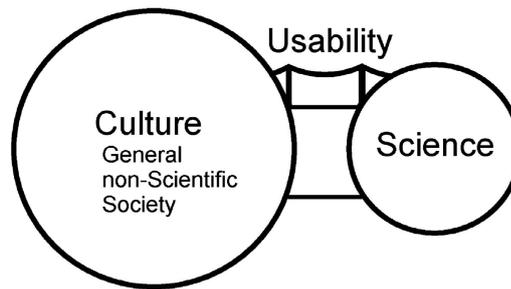


Abbildung 3.6: Usability als Brücke zwischen Kultur und Wissenschaft [74]

3.9 Usability

Jakob Nielsen schrieb 1993, dass 48% des Programmcodes in moderner Software für die Benutzerschnittstelle zuständig ist. Im Gegensatz dazu würden gewöhnliche Usability (deutsch: Benutzbarkeit) Aufwendungen nur 6% des Budgets kosten. Methoden zur Verbesserung der Usability könnten daher einen großen Teil der Software positiv beeinflussen [75].

Im Folgenden werden ein paar Definitionen des Begriffs Usability erläutert. Der Standard ISO 9241-11 erwähnt neben den Eigenschaften *Effectiveness* (deutsch: Effektivität) und *Efficiency* (deutsch: Effizienz) auch die Zufriedenheit des Benutzers als wichtigen Faktor für Usability. Die dort vorgefundene Definition des Begriffs lautet wie folgt:

'The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use'
[76, S. 4]

Der Standard ISO/IEC 9126-1 befasst sich mit der Definition von Qualität in Software im Allgemeinen. Hier ist Usability neben *Functionality* (deutsch: Funktionalität), *Reliability* (deutsch: Verlässlichkeit), *Efficiency* und *Maintainability* (deutsch: Wartbarkeit) als eines der Qualitätsmerkmale vorzufinden. Für die Begriffe werden außerdem Untereigenschaften definiert. Demnach besteht Usability aus Verständlichkeit, Erlernbarkeit, Bedienbarkeit, Attraktivität und Übereinstimmung mit gebräuchlichen Konzepten [77].

Die IEEE definiert den Begriff in ihrem *Glossary Of Software Engineering Terminology*. Die dort vorgefundene Definition nennt die Leichtigkeit, mit der die Bedienung eines Systems erlernt werden kann, sowie die Beschaffenheit der Eingabe- und Ausgabemöglichkeiten als die wichtigsten Kriterien. Der Begriff wird folgendermaßen umschrieben:

'The ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component.' [78, S. 161]

Aus den Definition ist ersichtlich, dass Usability zu den Zielen von UCD (siehe Abschnitt 3.2) gehört. Das Hauptaugenmerk liegt bei der BenutzerIn. Johnson et al. [74] geben eine philosophischere Definition der Ziele von Usability (siehe Abbildung 3.6). Demnach ist Usability die

Brücke zwischen Kultur und Wissenschaft und UCD ist das *Adjunct* (deutsch: Gehilfe, Ergänzung) von Usability [74].

In der Literatur existieren eine Vielzahl weiterer Definitionen für den Begriff Usability, die ihre eigenen Konzepte und Schlagwörter definieren. In vielen Fällen beschreiben unterschiedliche Begriffe die gleichen Eigenschaften, was zu Verwirrung von ForscherInnen und EntwicklerInnen führt [78]. Quesenbery [76] definierte die 5Es, um sich die Dimensionen von Usability leichter merken zu können. *Effectiveness* beschreibt, ob eine Aufgabe erfolgreich durchgeführt wurde. Dabei gilt es auch die Definition von Erfolg aus Sicht der BenutzerIn zu beachten. *Efficiency* bezieht sich auf die zeitliche Effizienz und die rasche Ausführung. *Engaging* (deutsch: einnehmend, bezaubernd) steht für eine ansprechende Gestaltung der optischen Darstellung sowie der Interaktionen. Die Zufriedenheit der BenutzerInnen ist hierfür maßgebend. Fehler im Programm und in der Bedienung zu verhindern, machen ein System *Error Tolerant* (deutsch: Fehlertolerant). Im Fehlerfall soll die Software der BenutzerIn geeignete Mittel geben, um den Fehler zu beheben. *Easy To Learn* (deutsch: leicht zu erlernen) bezieht sich auf den schnellen Einstieg sowie das Erlernen komplexerer Programmfunktionen. Auch die Einprägsamkeit der Verwendung ist ein Kriterium dafür [76].

Im Folgenden werden mit *Affordance* (Abschnitt 3.9.1) und *Metapher* (Abschnitt 3.9.2) zwei Faktoren von Benutzerschnittstellen näher betrachtet, durch die deren Usability stark beeinflusst werden kann. In Abschnitt 3.9.3 werden Metriken zur Messung von Usability-Kriterien durch strukturierte Forschungsmethoden (siehe Abschnitt 3.6 und 3.7) vorgestellt.

3.9.1 Affordance

Der Begriff *Affordance* wurde 1977 von James J. Gibson in dem Artikel *The Theory Of Affordances* [79] eingeführt. Das Nomen ist von Gibson frei erfunden und kommt von *to afford sth.* (deutsch: etwas ermöglichen/bieten/erlauben) [79]. Donald A. Norman, Mitbegründer der Nielsen Norman Group, griff den Begriff in seinem Buch *The Design Of Everyday Things* [80] erneut auf und popularisierte somit seine Verwendung in den Bereichen HCI (*Human-Computer-Interaction*) und Usability Engineering [80]. *Affordance* beschreibt, welche Handlungen und Verwendungen ein Objekt einem Menschen oder einem Tier anbieten. Die Art, wie etwas verwendet wird, ist dadurch also selbsterklärend. *Affordance* ist jedoch kein Attribut eines Gegenstandes selbst, sondern ist abhängig vom Betrachter. Die Oberfläche von Wasser bietet sich einem Wasserläufer an, um darauf zu laufen, während sie einen Menschen dazu einladen kann, sich zu waschen [79].

Das Vorhandensein von *Affordance* ist unabhängig davon, ob sie wahrgenommen wird. Deshalb unterscheidet William W. Gaver [81] zwischen vier verschiedenen Arten von *Affordance* [81]. Abbildung 3.7 veranschaulicht die Unterscheidung. Beim Design technischer Systeme gilt es also, *Affordance* wahrnehmbar zu machen und somit *Hidden Affordance* zu vermeiden. Ebenso soll *False Affordance* vermieden werden. Dabei handelt es sich um funktionslose Elemente, die fälschlicherweise eine Funktion vermuten lassen.

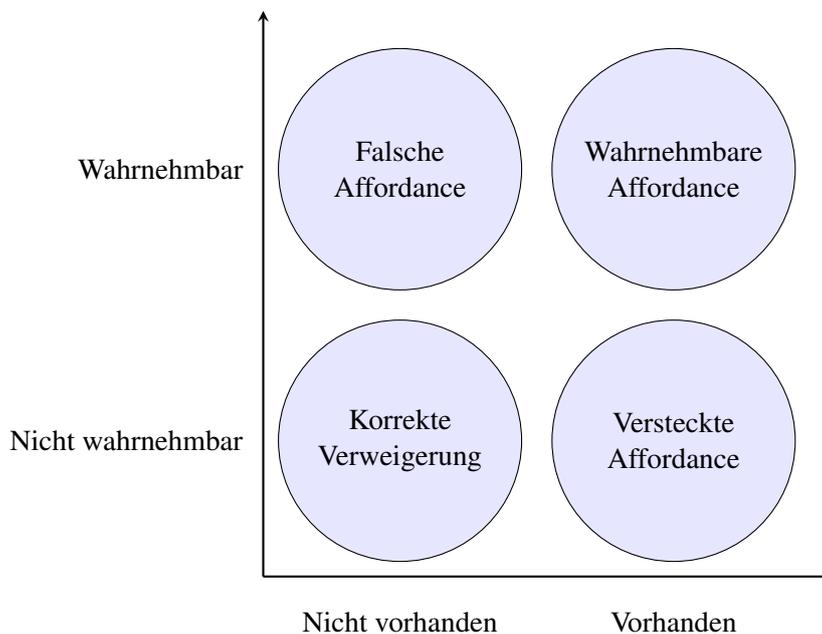


Abbildung 3.7: Arten von Affordance (reproduziert nach [81])

Ein Problem von Applikationen die auf NFC basieren ist, dass NFC-Tags selbst keinerlei *Affordance* aufweisen. NFC-Tags gibt es zwar in den verschiedensten Formen und Größen (siehe Abschnitt 2.2), sie bieten sich jedoch nicht speziell dazu an, mit dem eigenen Telefon berührt zu werden [15].

3.9.2 Metapher

Der Begriff ‘Metapher’ kommt vom griechischen *metaphorá* (Übertragung). Man spricht umgangssprachlich von einer Metapher, wenn ein Wort im übertragenen Sinn verwendet wird. Zwischen dem verwendeten Wort und dem gemeinten besteht dabei eine Ähnlichkeit.

Im Design von Interaktionen und Benutzerschnittstellen werden Metaphern dazu verwendet, um den BenutzerInnen zu ermöglichen, bereits erworbenes Wissen einzusetzen, um Unbekanntes zu verstehen. Dabei kann es sich sowohl um die Art der Verwendung einer Schnittstelle als auch dessen Funktion handeln. Die wohl bekannteste Metapher bei Mensch-Maschine Schnittstellen ist die des *Desktop*. Der Bildschirm zeigt dabei einen virtuellen Schreibtisch mit Grafiken bekannter Alltagsgegenstände wie Dokumente, Ordner und einem Mistkübel. Durch den bekannten Verwendungszweck im Alltag ist die damit verbundene Funktion und die Verwendung selbsterklärend [82].

Metaphern als Designwerkzeug wurden jedoch auch häufig kritisiert. Alan Cooper [58] bezeichnet Metaphern beispielsweise als zu einschränkend. Ihre Verwendung würde die DesignerInnen dazu bringen, ihre Schnittstellen unnötigerweise zu sehr an der realen Welt zu orientieren. Au-

ßerdem sei es ungewiss, ob die Metapher von der BenutzerIn erkannt wird [58]. Im Folgenden werden zuerst die von Saffer [82] gesammelten Kritikpunkte gegenüber Metaphern, und anschließend der Leitfaden zur Verwendung näher betrachtet.

Das Objekt, welches als Metapher verwendet wird, kann mehr oder weniger Funktionalität und Benutzungsmöglichkeiten bieten als sein digitales Pendant. Dadurch können sie einschränkend wirken, indem sie die DesignerIn daran hindern, dem abstrakten Objekt Funktionen und Eigenschaften zu geben, die das Objekt in der realen Welt nicht besitzt. Ebenso können unvorhergesehene Anforderungsänderungen oder neue Funktionalitäten möglicherweise nur schwer an die Metapher angepasst werden. Oft skalieren Metaphern auch nicht wie gewünscht [82]. In einen digitalen Ordner kann man beliebig Ordner ineinander sortieren. Diese Funktionalität hätte mit der Metapher eines Kuverts schlechter harmoniert und mit Kisten möglicherweise sogar besser.

Eine Metapher sollte daher nie das Konzept bestimmen, sondern es lediglich unterstützen. Es ist natürlich dass Metaphern im Laufe der Zeit in den Hintergrund rücken und der Erkennungswert sinkt. Dieser Umstand wird oftmals auch als Kritikpunkt gesehen, kann aber auch durchaus positiv gewertet werden, da es bedeutet, dass die Metapher nicht mehr notwendig ist, um die Bedienung zu verstehen. Die Metapher hilft Neulingen weiterhin beim Einstieg, aber limitiert erfahrene BenutzerInnen nicht. Daher ist es wichtig, auch von Metaphern loslassen zu können und sie notfalls auch komplett zu verwerfen. Es ist außerdem wichtig, kulturelle und kontextuelle Aspekte bei der Auswahl der Metaphern zu beachten [82].

3.9.3 Usability Metriken

'A metric is a way of measuring or evaluating a particular phenomenon or thing. We can say something is longer, taller, or faster because we are able to measure or quantify some attribute of it...' [83, S. 7]

Metriken werden verwendet, um Attribute und Eigenschaften zu messen, in Zahlen zu fassen und somit vergleichbar zu machen. Mit Hilfe von Usability Metriken kann also die Benutzbarkeit eines Systems zahlenmäßig erfasst werden [83].

Wie bereits in Abschnitt 3.9 festgestellt wurde, umfasst Usability eine Vielzahl an Dimensionen und bietet kein klar definiertes Konzept. Die Messung von Usability wurde deshalb auch oft kritisiert, da sie fälschlicherweise Verständnis für eine zu komplexe und subtile Materie suggerieren kann [84]. Sie eignen sich jedoch, um verschiedenste Aspekte unterschiedlicher Lösungen oder Änderungen an einem System vergleichen zu können, ohne sich dabei ein Gefühl verlassen zu müssen. Sie können selbst mit einer geringen Anzahl an TeilnehmerInnen brauchbare Daten liefern und ihre Ergebnisse können leicht vom Management verstanden werden [83].

Es existieren eine Vielzahl an Metriken, um unterschiedliche Eigenschaften eines Systems zu erfassen. Man unterscheidet im Allgemeinen zwischen *Preference Metrics*, *Performance Metrics* und *Predictive Metrics*. *Preference Metrics* (deutsch: Präferenz) messen die Zufriedenheit der BenutzerInnen. Die im Folgenden vorgestellten Metriken SUMI und SUS, gehören zu dieser

Kategorie. **Performance Metrics** (deutsch: Arbeitsleistung, Erfüllung) beschäftigen sich mit der Messung des Verhaltens und des Erfolgs der BenutzerInnen. Zu typischen Messwerten gehört die Ausführungszeit, das Verhältnis zwischen erfolgreichen und fehlerhaften Interaktionen oder die Zeit, welche die BenutzerInnen zum Behandeln von Fehlern benötigt [84]. **Predictive Metrics** sind auch unter dem Begriff *Design Metrics* bekannt. Sie geben die Möglichkeit zur Abschätzung der Auswirkungen von Änderungen auf die Usability eines Systems [78]. Im Folgenden werden mit MUSiC, SUMI und SUS, Metriken aus den Kategorien **Performance Metrics** und **Preference Metrics** näher betrachtet.

MUSiC - Metrics For Usability Standards In Computing

MUSiC ermöglicht die Messung von Attributen und Eigenschaften aus den Bereichen **Performance Metrics** und **Preference Metrics**. Im Folgenden werden die verschiedenen Messwerte im Detail besprochen.

Die **Effektivität** ist ein Maß der Vollständigkeit und Präzision mit der eine Aufgabe gelöst wurde. Zu diesem Zweck gilt es Auswertungskriterien zu definieren, um durch deren Anwendung die Quantität (Vollständigkeit) und Qualität (Korrektheit) einer Lösung zu erhalten. Die Effektivität ist also als Funktion mit den Parametern Quantität und Qualität definiert.

$$\text{Effektivität} = f(\text{Quantität}, \text{Qualität})$$

Unter dem Begriff **Effizienz** wird im Ingenieurwesen im Allgemeinen das Verhältnis zwischen Energieaufwand und daraus resultierender Leistung verstanden. Zur Berechnung können als Ressource verschiedenste Arten von Aufwendungen wie Zeit oder Finanzen eingesetzt werden. Dadurch ergeben sich unterschiedliche Effizienzmaße wie temporale- oder ökonomische **Effizienz**. Die Definition von MUSiC verwendet die Zeit zur Durchführung von Aufgaben als Ressource in der Berechnung.

$$\text{Effizienz} = \frac{\text{Effektivität}}{\text{Durchführungszeit}}$$

Die **Relative Benutzereffizienz** dient als Maß der Erlernbarkeit eines Systems. Dabei wird die Effizienz eines Experten mit der Effizienz eines Einsteigers bei der Benutzung im selben Kontext verglichen. Alternativ dazu kann auch die Zunahme von Effektivität oder Effizienz, nach mehrmaliger Durchführung als Maß für Erlernbarkeit herangezogen werden.

$$\text{RelativeBenutzereffizienz} = \frac{\text{EffizienzBenutzerIn}}{\text{EffizienzExperte}}$$

Der **Produktivitätszeitraum** zeigt, wie viel Zeit die BenutzerIn im Verhältnis zur gesamten Durchführungszeit damit verbracht hat, aktiv an der Lösung der Aufgabe zu arbeiten. Aktionen die geforderte Ausgaben produzieren, das Sammeln von Informationen oder die Durchführung von Handlungen, die zur Erzeugung der Ausgabe notwendig sind, werden als produktiv gewertet. Alle anderen Aktionen werden den Kategorien *Help Actions*, *Search Actions* und *Snag Actions* (deutsch: unerwartete Probleme) zugeordnet.

$$PP = \frac{\text{Durchführungszeit} - \text{HelpTime} - \text{SearchTime} - \text{SnagTime}}{\text{Durchführungszeit}}$$

Zur Erfassung der Werte empfiehlt es sich, den Testverlauf mit Videoaufnahmen zu dokumentieren. Die Analyse der gesammelten Daten kann, mittels spezieller Software wie beispielsweise DRUM (*Diagnostic Recorder For Usability Measurement*), wesentlich vereinfacht werden. DRUM unterstützt die BenutzerIn bei der Markierung und Verwaltung von Ereignissen im Testverlauf und bei der Berechnung von Metriken [85] [86] [87].

Um die Evaluierung der Benutzerzufriedenheit nicht außer Acht zu lassen, wurde als Teil von MUSiC, der Fragebogen SUMI (*Software Usability Measurement Inventory*) entwickelt. Er besteht aus 50 Fragen und ermöglicht die Erhebung der Messwerte 'Effizienz', 'Affekt', 'Hilfe und Unterstützung', 'Kontrollierbarkeit' und 'Erlernbarkeit' aus der subjektiven Sicht der BenutzerIn. Die Fragen sind hierbei mit den drei Möglichkeiten 'stimme zu', 'stimme nicht zu' und 'weiß nicht' zu beantworten. Die fünf zuvor genannten Messwerte werden durch jeweils zehn der Fragen abgedeckt. Zusätzlich dazu lässt sich aus 25 der Fragen ein globaler Wert für die Benutzerzufriedenheit errechnen. Die Ergebnisse der SUMI Metrik können anschließend mit einem Durchschnittswert verglichen werden, um weitere Anhaltspunkte für die Analyse zu liefern. Zu diesem Zweck bietet SUMI eine Datenbank mit den Testergebnissen einer Vielzahl von Standardanwendungen [135] [86].

SUS - System Usability Scale

Bei SUS handelt es sich um einen wesentlich kürzeren Fragebogen zur Erhebung der Benutzerzufriedenheit. Er besteht aus 10 Fragen, welche mit einer Skala von 1-5 bewertet werden. Die Antwort 1 steht hierbei für 'stimme überhaupt nicht zu' und 5 bedeutet 'stimme absolut zu'. Im Gegensatz zu SUMI ist das Ergebnis von SUS eine einzige Zahl zwischen 0 und 100. Die Punktzahl korreliert jedoch stark mit anderen Messungen, wie etwa der globalen Punktzahl von SUMI.

Die Fragen werden bei SUS abwechselnd positiv und negativ gestellt, um die Aufmerksamkeit der BenutzerIn zu erhöhen. Deshalb werden zur Berechnung der SUS Punktzahl, die Ergebnisse der Fragen mit ungeraden Indizes um 1 verringert und die der Fragen mit geraden Indizes von 5 subtrahiert. Somit ergibt sich für jede Frage ein Wert zwischen 0 und 4. Die einzelnen Werte werden anschließend addiert und mit 2.5 multipliziert, um eine Skala von 0 bis 100 zu erhalten [88]. Trotz des maximal erreichbaren Wertes von 100 ist das SUS Ergebnis jedoch nicht als Prozentwert zu interpretieren [89]. Wie bereits am Anfang dieses Abschnitts erwähnt, eignen sich Metriken zum Vergleich. Da also keine absolute Messung von Usability möglich ist, muss zur Interpretation ein Vergleichswert herangezogen werden [89].

Bangor, Kortum, und Miller [[89]] haben die Ergebnisse von 3500 Fragebögen aus 273 Studien zusammengefasst. Die Ergebnisse stammen aus Untersuchungen unterschiedlicher Software und Hardware wie Webapplikationen, Mobiltelefonen und Fernsehern. Dabei ermittelten

sie einen durchschnittlichen Wert von 69.5 Punkten. Alles unter 70 Punkten ist demnach als unterdurchschnittlich und mangelhaft zu beurteilen [89].

3.10 Technology Adoption Lifecycle

Die Grundlagen für den *Technology Adoption Lifecycle* (deutsch: Technologie Übernahme Lebenszyklus) stammen aus *The Diffusion Process* von George M. Beal und Joe M. Bohlen [90]. Die Publikation dokumentiert die Ergebnisse ihrer Studie, über die Bereitschaft von LandwirtInnen zur Integration von Innovationen in ihren Arbeitsprozess.

Das Ergebnis ihrer Arbeit waren folgende drei grundlegende Feststellungen. Das Übernehmen und Integrieren von Innovationen ist ein mehrstufiger, mentaler Prozess. Personen welche diesen Prozess durchmachen können die Phasen und deren zeitliche Abfolge klar voneinander unterscheiden. Dabei handelt es sich um folgende Stufen [90]:

1. Awareness Stage: Person erfährt von einer Innovation aber weiß nichts näheres darüber
2. Interest Stage: Person will mehr über die Innovation im Detail wissen
3. Evaluation Stage: Person denkt über die Konsequenzen der Integration nach
4. Trial Stage: Person testet die Innovation
5. Adoption Stage: Person übernimmt die Innovation

Die möglichen KundInnen für ein Produkt können abhängig von der Größe ihres Interesses an Neuerungen und dem Zeitpunkt, an welchem sie bereit sind, diese zu integrieren, kategorisiert werden. Sie erkannten dabei die Gruppierungen der *innovators*, *early adopters*, *early majority*, *majority* und *non-adpoters*.

Sie stellten außerdem fest, dass es verschiedene Arten von Innovationen gibt. Manche Neuerungen bedeuten bloß eine Veränderung der Ausrüstung und des Materials. Andere hingegen bringen eine Änderung in der Art der Verwendung mit sich [90].

Geoffrey Moore [60] griff die Ergebnisse von Beal und Bohlen [90] wieder auf und veröffentlichte seine weiterführenden Forschungen in dem Buch *Crossing the Chasm* (deutsch: Die Kluft überqueren). Er bezeichnet eben beschriebene Innovationen, die eine Änderung in der Art der Verwendung bedeuten, als *discontinuous innovation* (deutsch: unterbrechende Innovation).

Abbildung 3.8 zeigt ein Diagramm zur Veranschaulichung des *Technology Adoption Lifecycle*. Bei diesem Modell gilt es sich von links nach rechts vorzuarbeiten. Die Darstellung des Sachverhalts durch die lückenhafte Glockenkurve bezieht sich auf *discontinuous innovations* im Speziellen. Die Flächen unter den Kurvensegmenten repräsentiert die Anzahl der KundInnen in einer gewissen Kategorie. Die Größe der Lücken stellt dar, wie schwer es für eine Gruppe ist eine Lösung zu adoptieren, die speziell für die Gruppe zu ihrer Linken geschaffen wurde. Die

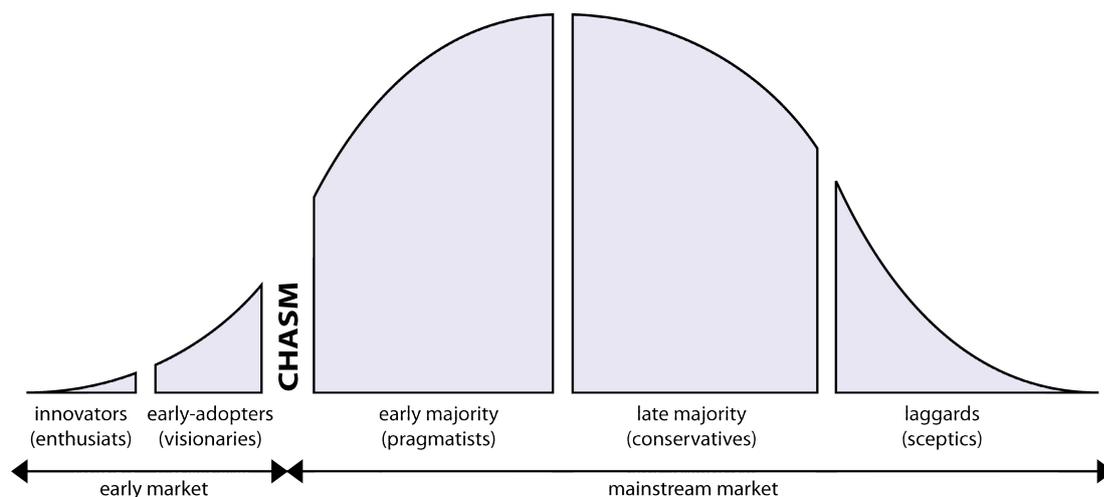


Abbildung 3.8: Technology Adoption Lifecycle (abgeändert, reproduziert nach [136])

Lücke zwischen dem vergleichsweise kleinen *early market* (deutsch: früher Markt), bestehend aus den ErstanwenderInnen, und dem Massenmarkt ist auffallend groß. Bei ErstanwenderInnen handelt es sich um Personen, die sich besonders für Neuerungen in dem betrachteten Bereich besonders interessieren. Sie investieren Zeit und Geld in die Anschaffung neuer Technologien und nehmen Probleme bei der Verwendung in Kauf. Das Vertrauen der ErstanwenderInnen ist essenziell, um die weitere Marktsegmente erreichen zu können. Die wichtigsten Kriterien für die *early majority* sind jedoch gute Referenzen sowie der praktische Nutzen und die Vorteile die aus einem Umstieg resultieren würden. Außerdem erwarten sie sich ein fehlerfrei funktionierendes Produkt. Im Gegensatz zu den ErstanwenderInnen wollen sie Evolution anstatt Revolution. Durch die Inkompatibilität der grundlegenden Einstellungen von *early market* und *early majority* gelten die ErstanwenderInnen jedoch nicht als brauchbare Referenz für den Massenmarkt [60].

Entwicklung

Die in der Theorie gewonnen Erkenntnisse werden im folgenden Abschnitt zum Teil wieder aufgegriffen und für die Entwicklung verwendet. Zuerst wird der gewählte Entwicklungsprozess näher betrachtet und ein genauer Einblick in die Vorgehensweise. Anschließend wird die Dokumentation, aller Entwicklungsarbeiten die im Rahmen dieser Arbeit durchgeführt wurden, präsentiert.

4.1 Der Entwicklungsprozess

Die Entwicklungen wurden unter der Verwendung von *User Centered Design* Methoden (siehe Abschnitt 3.2) in einem iterativen Prozess (siehe Abschnitt 3.1) durchgeführt. Das wiederholte Durchlaufen der einzelnen Phasen ist fester Bestandteil benutzerzentrierter Methoden. Jede Iteration besteht aus den Abschnitten **Spezifikationsphase**, **Entwicklungsphase**, **Testphase** und **Evaluierungsphase**. In der **Spezifikationsphase** werden Anforderungen und Ziele für die aktuelle Iteration definiert, die es anschließend in der **Entwicklungsphase** zu erfüllen gilt. Die daraus entstandene Lösung wird in der **Testphase** mit BenutzerInnen getestet. Die Ergebnisse der Tests werden in der anschließenden **Evaluierungsphase** analysiert und dienen als Ausgangspunkt für die weiteren Entwicklungen. Für die Entwicklungen im Rahmen dieser Arbeit wurden zwei Iterationen durchgeführt. Das darauf folgende Kapitel bietet eine zusammenfassende **Analyse** sowie einen Ausblick für zukünftige Entwicklungsmöglichkeiten.

4.2 Erste Anforderungsphase

Zu Beginn der Entwicklungen stand die Erarbeitung grundlegender **Anforderungen an die Interaktion mit NFC-Tags**. Anschließend wurden mit Hilfe des *Technology Adoption Lifecycle* (siehe Abschnitt 3.10) **Rollen und Ziele** identifiziert. Aus den daraus gewonnenen Daten wurden **Personas** (siehe Abschnitt 3.3) erstellt, die den gesamten weiteren Entwicklungsprozess begleiten. Den Abschluss der Spezifikationsphase bildet die Erstellung von **Szenarios**, in denen die

Personas und ihr Umfeld als Akteure auftreten. Alle in dieser Phase gewonnenen Informationen dienen als Anforderungen für die darauf folgende Entwicklungs- und Testphase.

4.2.1 Anforderungen an die Interaktion mit NFC-Tags

Ohne Rücksicht auf möglicherweise bestehende Limitierungen der Android Implementierung wurden zu Beginn Anforderungen an die Interaktion mit NFC-Tags erarbeitet. Das Verhalten der Standardumsetzung und die technische Machbarkeit der Anforderungen wurde anschließend in der Entwicklungsphase analysiert und bestmöglich umgesetzt. Folgende grundlegende Eigenschaften wurden definiert:

1. Die Interaktion mit einem NFC-Tag muss mit einer einfachen Handbewegung durchgeführt werden können.
2. Die Bedienung mittels NFC soll möglichst wenig Aufmerksamkeit und Konzentration benötigen.
3. Das Gerät soll auch auf einem NFC-Tag liegen gelassen werden können, um später beim Entfernen des Geräts eine weitere Aktion auslösen zu können.
4. Die Bedienung soll aktivitätsbasiert sein (siehe Logitech Harmony in Abschnitt 2.4), um möglichst wenige Interaktionen durchführen zu müssen.
5. Zusätzlich zu den NFC-Tags soll die Bedienung auch über den Touchscreen möglich sein.
6. Beim Auslösen einer Funktion per NFC-Tag soll eine passende Touchscreen Fernbedienung angezeigt werden können.

4.2.2 Rollen und Ziele

Für die Erstellung von Personas wurden zuerst Benutzerrollen und deren Ziele definiert. Bei dem angestrebten System handelt es sich um eine *discontinuous innovation* (siehe Abschnitt 3.10), denn die Interaktion mit den NFC-Tags unterscheidet sich grundlegend von der Bedienung per Fernbedienung oder Gerätetasten. Daraus ergab sich die Unterscheidung zwischen **ErstanwenderInnen** und der breiten Masse an **normale BenutzerInnen**. Um nicht schon in frühen Phasen Designfehler zu begehen, wurde es daher als besonders wichtig erachtet, beide Rollen voneinander zu unterscheiden und miteinzubeziehen. Zusätzlich dazu wurde die Rolle der **VerkäuferIn** definiert, die lediglich zur Abgrenzung dient. Durch die Heterogenität der zu steuernden Hardware entsteht ein nicht zu unterschätzender Konfigurationsaufwand, den es im kommerziellen Bereich zu minimieren gilt. Wie bereits im Abschnitt 2.5 über Heimautomation erwähnt wurde, gehört ein möglichst niedriger Aufwand bei Installation und Einrichtung zu den wichtigsten Kriterien für den Endanwender. Deshalb gilt es für die **VerkäuferIn** die Kosten minimal zu halten. Diese Bedürfnisse sollen jedoch absichtlich außer Acht gelassen werden. Um eine einfache Konfiguration durch den Endanwender zu gewährleisten, wäre eine Gerätedatenbank notwendig, deren Erstellung und Wartung außerhalb des Fokus dieser Arbeit liegt. Die Möglichkeit zur Fernwartung ist ebenso ein Anliegen der **VerkäuferIn**.

Die folgenden drei Rollen wurden definiert:

1. **Normale BenutzerIn:** Übernimmt *discontinuous innovations* nur Aufgrund von tatsächlichen Verbesserungen oder aus Notwendigkeit.
2. **ErstanwenderIn:** Ist besonders technikbegeistert und probiert gerne neue Technologien aus. Nimmt Probleme durchaus in Kauf.
3. **VerkäuferIn:** Möchte das System gewinnbringend vertreiben.

Ziele der **ErstanwenderIn:**

- Freude bei der Verwendung haben
- Weniger Fernbedienungen zu benötigen

Ziele der **normalen BenutzerIn:**

- Erleichterung im Alltag (Zeit und Mühen sparen)
- Kein Kontrollverlust
- Einfache Bedienung
- Weniger Fernbedienungen zu benötigen

Die **VerkäuferIn** verfolgt alle Ziele der beiden anderen Gruppen und darüber hinaus:

- Niedrige Konfigurationskosten
- Möglichkeit zur Konfiguration durch den Endkunden
- Niedrige Wartungskosten
- Möglichkeit zur Fernwartung

4.2.3 Personas

Mit Hilfe der zuvor identifizierten Rollen und Ziele wurden anschließend Personas erschaffen. Sie sollen ein besseres Verständnis für die BenutzerIn im Entwicklungsprozess schaffen und stellen die Akteure in den Szenarios dar.

Primäre Persona:

Name: Kerstin Ammer

Kerstin ist 43 Jahre alt. Sie ist Lehrerin an einem Gymnasium und unterrichtet dort die Fächer Biologie und Geschichte. Sie wohnt in einer schönen Wohnung nicht weit von der Stadt, zusammen mit ihrem sehr aufgeweckten, 13-jährigen Sohn Ralf und einem kleinen Beagle namens Bart. Ihr Ehemann Werner ist beruflich sehr viel unterwegs und nur am Wochenende zuhause. Sonntags trifft sie sich meist ihren Freundinnen zum Nordic Walking und anschließendem Kaffee mit Kuchen. Diese Zeit nützt der technikbegeisterte Werner dann meist geschickt, um gemeinsam mit seinem Sohn mit etwaigen neuen technischen Errungenschaften herumzuspielen. Kerstin kann die Begeisterung für Technik mit ihren Männern jedoch nicht teilen. Wenn sie sich ein elektronisches Gerät kauft, dann nie grundlos. Sie hat durchaus ein bisschen Angst, dass sie mit einem neuen Gerät nicht zurecht kommen könnte. Außerdem ist sie davon überzeugt, dass etwas Neues nicht unbedingt besser ist, auch wenn die Werbung uns das manchmal suggeriert.



Abbildung 4.1: Primäre Persona: Kerstin Ammer

Sekundäre Persona:

Name: Armin Buchinger

Armin ist ein 32 Jahre alter Elektrotechniker. Er hat an der TU-Wien studiert und arbeitet seither in den Entwicklungsabteilungen diverser namhafter Unternehmen. Einen großen Teil des überschüssigen Verdienstes nach Abzug der Fixkosten investiert er regelmäßig in neue technische Spielereien. In seiner Freizeit veranstaltet er gerne Filmabende mit seinen Freunden. Die dauern meist bis spät in die Nacht, genauso wie seine gelegentlichen online Videospiele. Er schläft daher generell gerne länger und braucht als erstes einen Kaffee nach dem Aufstehen. Armin hält sich auch gerne in der Natur auf wenn es das Wetter passt. Wenn es kalt ist oder regnet, nützt er die höhere Gewalt gerne als Ausrede, um das Haus nicht für Freizeitaktivitäten verlassen zu müssen.



Abbildung 4.2: Sekundäre Persona: Armin Buchinger

Anti Persona:

Name: Heinz Hübl

Heinz ist 37 Jahre alt und Unternehmer. Seine kleine Firma verkauft Heimautomationsysteme, Alarmanlagen und Videoüberwachung an KundInnen im In- und Ausland. Bei seinen Aufträgen handelt es sich von kleinen Installationen und Wartungsarbeiten bis hin zu ausgefallenen Wünschen Privatpersonen im Ausland. Solange der Preis stimmt, kann man von ihm fast alles haben. Allzu hohe Wartungs- und Konfigurationskosten werden jedoch von KundInnen meist nicht toleriert. Deshalb sind die Möglichkeit zur Fernwartung sowie die Änderung der Konfiguration durch den Benutzer für ihn ausschlaggebende Argumente. Seine Hauptziele beim Einsatz neuer Technologien sind: Kostenersparnis, Innovation und Kundenzufriedenheit. Heinz ist leidenschaftlicher Kartenspieler. Wenn das Wetter passt, nützt Heinz jedes Wochenende, um den alten Mercedes seines Großvaters auszuführen. Dann macht er gerne auch mehrtägige Ausflüge, um die Umgebung zu erkunden.

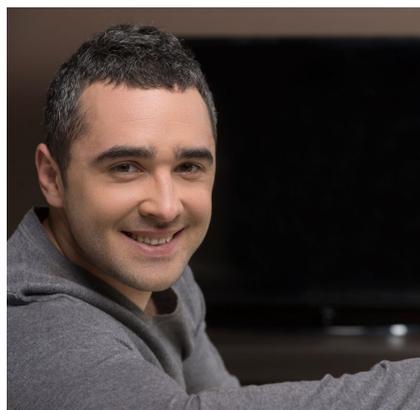


Abbildung 4.3: Anti Persona: Heinz Hübl

4.2.4 Szenarios

Die Szenarios beschreiben typische Anwendungsfälle in erzählender Form. Hierbei werden gemäß der Empfehlung von Alan Cooper (siehe Kapitel 4.2.4) die zuvor erstellten Personas als Akteure eingesetzt. Die Szenarios dienen zur Dokumentation von Anforderungen der geplanten Funktionalität und bilden in weiterer Folge die Grundlage für die Benutzertests.

Szenario 1 - Zuhause ankommen

Voll bepackt mit Einkäufen für die nächsten Tage kommt Kerstin Zuhause an. Um ihren Schlüssel aus der Tasche zu kramen, stellt sie ihre Einkäufe vor der Tür ab. Während sie die Tür aufsperrt, hält sie ihr Smartphone neben die Türklinke. Das Licht in den Räumen am Weg zur Küche geht an und ihr Lieblings-Radiosender wird gespielt. Gut gelaunt betritt sie die Wohnung und schubst hinter sich die Tür zu.

Szenario 2 - Wechsel zwischen Unterhaltungsmedien

Nach dem Abendessen und einem harten Arbeitstag möchte Kerstin am liebsten nur noch einen Film in ihrem Heimkino genießen. Ralf möchte aber lieber zuerst noch seine gestrige Niederlage in einem Videospiel wieder gutmachen. Kerstin gibt sich geschlagen, allerdings unter der Bedingung, dass nach einer erneuten Niederlage ihres Sohnes trotzdem Schluss ist. Sie hält ihr Smartphone an die Spielekonsole. Die Leinwand fährt in Position, die AV-Geräte werden eingestellt, das Licht wird gedämpft und das Spiel kann beginnen. Um schnell zu ihrem Film zu kommen, lässt Kerstin ihren Sohn unauffällig gewinnen. Ralf freut sich zu sehr über den Sieg, um die Farce zu erkennen. Sie legt ihr Smartphone auf das Videosymbol ihres Tisches, legt den Datenträger ein, das Licht wird weiter gedämpft. Kerstin lehnt sich entspannt zurück und schickt Ralf zum Zähneputzen.

Szenario 3 - Filmeabend

Armin lädt seine Freunde zu einem gemeinsamen Kinoabend zu sich nachhause ein. Bei gedämpften Licht wird zu einem spannenden Actionfilm eine große Menge an Chips und Getränken verzehrt. Gerade jetzt wo die Spannung seinen Höhepunkt erreicht, muss Bernhard dringend auf die Toilette. Armin bemerkt das und legt sein Smartphone auf das Pause-Symbol auf dem Tisch. Der Film pausiert und das Licht wird heller, um Bernhard einen sicheren Weg zur Toilette zu bieten. Seine Freunde sind zwar nicht sehr erfreut über die ungeplante Unterbrechung, zumindest verpassen sie jedoch nichts, während Bernhard durch den Lichtkegel des Projektors spaziert. Nach seiner Rückkehr nimmt Armin das Smartphone wieder vom Tisch. Die der Film wird fortgesetzt und das Licht geht wieder aus.

Szenario 4 - Abendroutine

Nach dem gelungenen Filmeabend torkelt Armin quer durch die Wohnung zum Badezimmer, um sich auf die Nachtruhe vorzubereiten. Dabei schaltet er das ein oder andere Licht ein. Auch im Wohnzimmer sind noch einige Geräte im Betrieb. Nach seiner Körperpflege geht er sofort

und ohne Umwege ins Bett und legt sein Smartphone auf den Nachttisch. Alle Geräte und Beleuchtungen in seiner Wohnung gehen automatisch aus und Armins Augen fallen ermüdet zu.

Szenario 5 - Morgenroutine

Armin hat es morgens oft eilig. Heute will er absolut keine Zeit verlieren, denn er muss dringend zum Flughafen. Als der Wecker klingelt, setzt er sich auf und nimmt sein Smartphone vom Nachttisch. Es ist noch dunkel und am Weg zur Toilette und zur Küche geht automatisch das Licht an. Er macht sich auf dem Weg in die Küche, wo ihm sofort der herrliche Kaffeeduft in die Nase steigt. Nach seiner Morgenroutine verlässt er schnell das Haus. Am Weg nach draußen hält er noch schnell das Smartphone an die Türe. In einer Minute wird die Alarmanlage aktiviert, alle Lichter gehen aus und das Telefon informiert ihn nebenbei noch über die aktuelle Wetterlage.

4.3 Erste Entwicklungsphase

Der folgende Abschnitt beschreibt alle Design- und Entwicklungsarbeiten der ersten Iteration im Detail. Zu Beginn wurde das Verhalten und die Entwicklungsmöglichkeiten der NFC Schnittstelle des **Android Smartphone** analysiert. Anschließend wurden Modifikationen an der Software des Geräts vorgenommen, um die zuvor definierten Anforderungen an die Interaktion (siehe Abschnitt 4.2.1) bestmöglich zu erfüllen. Danach folgen Abschnitte über die Steuerung der **Beleuchtung**, den **Fenstern** und der **Kaffeemaschine**. Dabei kamen sowohl kommerzielle Lösungen als auch eigens entwickelte Hardwarelösungen zum Einsatz. Ein kurzer Abschnitt über **NFC-Tags** befasst sich mit der Auswahl passender Transponder. Anschließend wird die **Android Universalfernbedienung reYmote** vorgestellt und eine Vielzahl von Weiterentwicklungen beschrieben, die notwendig waren, um die Szenarios vollständig zu unterstützen.

4.3.1 Android Smartphone

Als Entwicklungs- und Testgerät wurde ein Galaxy Nexus Android Smartphone von Samsung verwendet. Zuerst wurden die Interaktionsmöglichkeiten untersucht. Anschließend wurde nach Möglichkeiten gesucht, um die Interaktionen möglichst gut umzusetzen. Die Resultate werden nun im Detail erläutert.

Das Android Betriebssystem liest NFC-Tags nur, wenn der Bildschirm eingeschaltet und das Gerät entsperrt ist. Wie bereits in Abschnitt 2.2 erwähnt wurde, wird die geringe Reichweite von NFC als Sicherheitsaspekt angesehen. Die Notwendigkeit, den Bildschirm einzuschalten und das Gerät zu entsperren, schützt die BenutzerIn zusätzlich vor ungewollten Interaktionen und schont den Akku des Geräts. Die Bildschirmsperre wurde mit Hilfe der Applikation *Delayed Lock* und dem dazugehörigen *Delayed Lock WiFi Plugin* im Heim-WLAN deaktiviert. Tests während der Entwicklung haben ergeben, dass sich das Einschalten des Bildschirms selben Handgriff erledigen lässt, in dem man das Smartphone aufnimmt und Richtung des Tags bewegt. Es kam jedoch häufig vor, dass der Zustand des Bildschirms nicht korrekt wahrgenommen wurde und die Interaktionen somit mehrere Anläufe brauchten. Diese Notwendigkeit an erhöhter Aufmerksamkeit zur Interaktion widersprach schlussendlich der Anforderung 2, die

eine möglichst niedrige Konzentration der BenutzerIn zur Durchführung der Interaktion fordert.

Eine Internetrecherche ergab, dass diverse alternative Firmwares einen Menüpunkt zur Verfügung stellen, um Tags mit gesperrtem und sogar ausgeschaltetem Bildschirm lesen zu können. Die Versionshistorie solcher alternativen Firmwares ist jedoch typischerweise nicht vollständig aktuell gehalten und lückenhaft, was die Suche nach einer passenden Firmware erheblich erschwerte. Gefunden wurde diese Funktionalität schließlich im AOKP (Android Open Kang Project) ab JB Build 40 (24.06.2012) [137]. Für diese Arbeit wurde der zu dem Zeitpunkt aktuelle AOKP Build 'JB Milestone 1' auf dem Samsung Galaxy Nexus installiert und der Menüpunkt *NFC Polling* auf *Screen Off* gesetzt. Mit dieser Änderung wurde schließlich das Einlesen von NFC-Tags ohne den Bildschirm zu aktivieren und zu entsperren ermöglicht. Im alltäglichen Gebrauch listete die Android Akku-Applikation die systeminternen Prozesse mit 2% Verbrauch. Der Akkuverbrauch der Dauerverwendung des NFC Moduls ist demnach vernachlässigbar.

Zwei verschiedene Signaltöne signalisieren den Erfolg und den Fehlschlag beim Einlesen von NFC-Tags. Das Entfernen eines Tags wird akustisch nicht signalisiert. Ebenso wird das Ereignis laut Dokumentation [138] nicht per Intent (siehe Abschnitt 2.1) gemeldet. Um die Interaktion von Szenario 3 und 5 dennoch umsetzen zu können, wurde vorerst an die Möglichkeit zur Nutzung einer alternativen bzw. *low level* Programmierschnittstelle in Erwägung gezogen. Die betrachtete Lösungsvariante hätte jedoch *polling* benötigt (ein Softwareprozess der laufend den Status des Tags abfragt) und sich deutlich negativ auf den Energieverbrauch auswirkt. Bei einer weiteren Internetrecherche wurde schlussendlich eine Lösung für das Problem im *XDA-Developers Forum* gefunden [139]. Mittels root Berechtigungen ('Administratorrechte' im Linux Jargon) lässt sich die original NFC Applikation des Betriebssystems gegen eine angepasste Version ersetzen. Diese modifizierte Version erstellt *Intents* mit der Aktion *android.nfc.action.TAG_LOST*. Durch die Registrierung eines *Intent Filters* kann die eigene Applikation auf das Ereignis reagieren. Nach dem Austausch des Programms signalisiert das Betriebssystem das Entfernen eines Tags ebenfalls mit einem Signalton.

4.3.2 Beleuchtung

Die Steuerung der Beleuchtung wurde mit von Z-Wave Geräten, in Form von Lichtschaltern und Zwischensteckern realisiert. Kapitel 2.5 gibt einen tiefen Einblick in das zugrunde liegende Protokoll.

Es ist zwar eine breite Palette von Z-Wave Geräten über das Internet erhältlich, verglichen mit der großen Masse an herkömmlichen Geräten wirkte die Auswahl jedoch stark begrenzt. Als wandmontierte Geräte Schalter und Dimmer wurden die Geräte der Firma Z-Wave.Me ausgewählt. Bei den Aktoren (Schalter ZME_06431, Dimmer ZME_06433) handelt es sich eigentlich um Geräte der Firma REV-Ritter mit einer alternativen Software. Dadurch bieten sie zusätzliche Konfigurationselemente wie etwa die Steuerung anderer Lampen per Doppelklick. Alle 16 Lichtschalter der Testumgebung wurden durch Z-Wave Geräte ersetzt. Bei einem der bestellten Schalter ist jedoch ein Fehler beim Einspielen der neuen Software unentdeckt geblieben, wodurch diese Features nicht verfügbar waren.



Abbildung 4.4: Einzelne Komponenten der Z-Wave Lichtschalter

Abbildung 4.4 zeigt einen solchen Dimmer in seinen Einzelteilen. Von links nach rechts sind das: Schaltwippe, Abdeckrahmen, Montageplatte und Unterputzeinsatz. Bei diesen und anderen Z-Wave Lichtschaltern ist speziell darauf zu achten, ob es sich am Einbauort um 2 oder 3 Drahtsysteme handelt, also ob ein Nullleiter vorhanden ist. Ist ein Nullleiter vorhanden, muss auch ein Schalter mit Nulleiter Anschluss gewählt werden, wobei dies im Falle von Z-Wave.Me der Dimmer ist. In Abbildung 4.5 ist selbiger bereits in einer Unterputzdose montiert, jedoch noch ohne Rahmen und Wippe.



Abbildung 4.5: Montierter Taster ohne Rahmen und Wippe

Bei den Schaltern handelt es sich eigentlich um Taster. Sie bleiben nicht wie Lichtschaltern üblich in einer Schaltstellung stehen, sondern verbleiben in der Mittelstellung. Zum Einschalten wird oben gedrückt, zum Ausschalten unten. Anstelle von Wechsel- und Kreuzschaltern zur Erstellung von Wechselschaltungen und Kreuzschaltungen werden batteriebetriebene Taster (in diesem Fall der ZME_06444 *Wall Controller*) eingesetzt. Mittels Assoziation werden die *Wall Controller* mit den Aktoren verknüpft, um diese zu direkt anzusteuern. Dazu mussten zuerst die bestehenden Leitungen in der nicht mehr notwendigen Schalter entsprechend kurzgeschlossen werden, dass nur mehr ein einziger Aktor die Schaltung der Stromquelle durchführt. An den leeren Dosen werden die batteriebetriebenen Taster angebracht.

Durch eine minimale Änderung der Maße in der Herstellung der Schaltwippen klemmten alle *Wall Controller* nach der Montage und konnten somit nicht bedient werden. Um das Problem zu beheben, wurden die Gummitasten der Geräte mit einem scharfen Bastelmesser ein wenig gekürzt. Die bestehenden Energiesparlampen mussten gegen Halogenbirnen getauscht werden, da die Dimmer mit so geringen Lasten nicht zurechtkommen. Dies äußerte sich in stroboskopartigen Lichteffekten nach einem längerem Betrieb sowie schlechter Reaktion beim Dimmen selbst.

Abbildung 4.6 zeigt einen Dimmer-Zwischenstecker. Die runde Plastiktaste an der Vorderseite dient zur Einbindung in das Netzwerk, als Statusanzeige und als manuelle Bedienungsmöglichkeit.



Abbildung 4.6: Z-Wave Dimmer-Zwischenstecker

Als Z-Wave Controller und Schnittstelle zu einem herkömmlichen IP-Netzwerk wurde ein 'Mi Casa Verde Vera 2 Home Controller' verwendet. Mittels Kabel oder WLAN wird der Controller in ein bestehendes IP Netzwerk integriert. Die Verbindung wurde mittels Kabel gelöst, um die Funkstrecke und deren inhärente Probleme zu vermeiden. Anschließend können über einen Browser Z-Wave Geräte verwaltet, konfiguriert, abgefragt, gesteuert und programmiert werden. Außerdem bietet der Controller eine HTTP Schnittstelle zur Ansteuerung und Abfrage von Z-Wave Geräten.

Eine HTTP GET Anfrage unter folgender URL führt beispielsweise die Szene mit Nummer 20 aus:

```
/data_request?id=lu_action&serviceId=urn:micasaverde-com:serviceId:HomeAutomationGateway1&action=RunScene&SceneNum=20
```

Die folgende Anfrage setzt die TargetValue des Geräts 25, welches den Service *SwitchPower* unterstützt, auf 1 (schaltet das Gerät ein):

```
/data_request?id=lu_action&DeviceNum=25&serviceId=urn:upnp-org:serviceId:SwitchPower1&action=SetTarget&newTargetValue=1
```

Die Ausführung sehr großer Szenen kann jedoch einige Zeit in Anspruch nehmen. Der Controller scheint den Status der Geräte nach der Ausführung noch einmal zu prüfen und notfalls zu korrigieren. Dabei kann es sogar zu deutlich wahrnehmbaren Verzögerungen der Ausführungszeiten von bis zu einer halben Minute kommen. Die HTTP Verbindung bleibt in der Zwischenzeit geöffnet und blockiert dabei den ausführenden *Thread*.

4.3.3 Fenster

Die Testumgebung befindet sich in einem Dachgeschoss und war bereits mit io-homecontrol (siehe Abschnitt 2.5) Dachflächenfenstern und Sonnenschutzlösungen von Velux ausgestattet. Der Einbau ist mit der maximalen Raumhöhe von 4.5 Meter zu rechtfertigen, wodurch selbst die niedrigeren Fenster bei geringer Körpergröße und Armlänge nur schwer bis gar nicht ohne Hilfsmittel manuell bedienbar sind. Zur Steuerung existieren zwei verschiedene Fernbedienungen. Die kleinere Variante ohne Display kann mit einem Tastendruck, alle ihr zugewiesenen Geräte gleichzeitig bedienen. Getätigte Zuweisungen können jedoch nicht verändert oder gelöscht werden. Die große Fernbedienung bietet die Möglichkeit, mehrere Geräte zu verwalten, und erlaubt die Erstellung von Programmen und deren datums- und zeitabhängige Ausführung. Abbildung 4.7 zeigt die zwei verschiedenen Geräte.

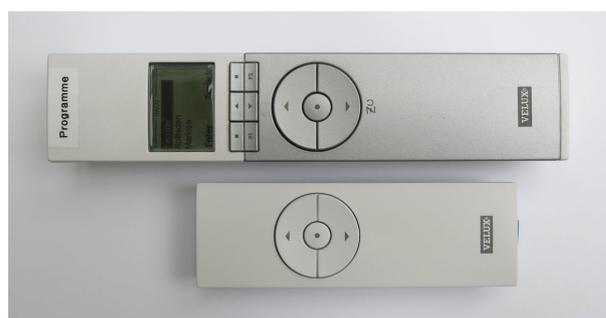


Abbildung 4.7: Originale Velux Fernbedienungen

Zum Zeitpunkt der Erstellung dieser Arbeit waren die Fenster nur direkt über das proprietäre Funkprotokoll von io-homecontrol steuerbar. Ein kommerzielles IP Gateway existierte nicht.

Laut der Aussage eines Technikers wäre jedoch ein dediziertes Android Tablet mit speziellem io-homecontrol Modul und in weiterer Folge ein IP Gateway in Entwicklung. Um die Fenster dennoch mit dem Smartphone über ein IP Netzwerk steuern zu können, wurde mit Hilfe einer original Fernbedienung und einem Arduino Ethernet eine Lösung entwickelt. Vorbild dafür war die Lösung des Users 'Mac Gyver' im *Homeseer Forum*. Abbildung 4.8 zeigt den dort veröffentlichten Schaltplan.

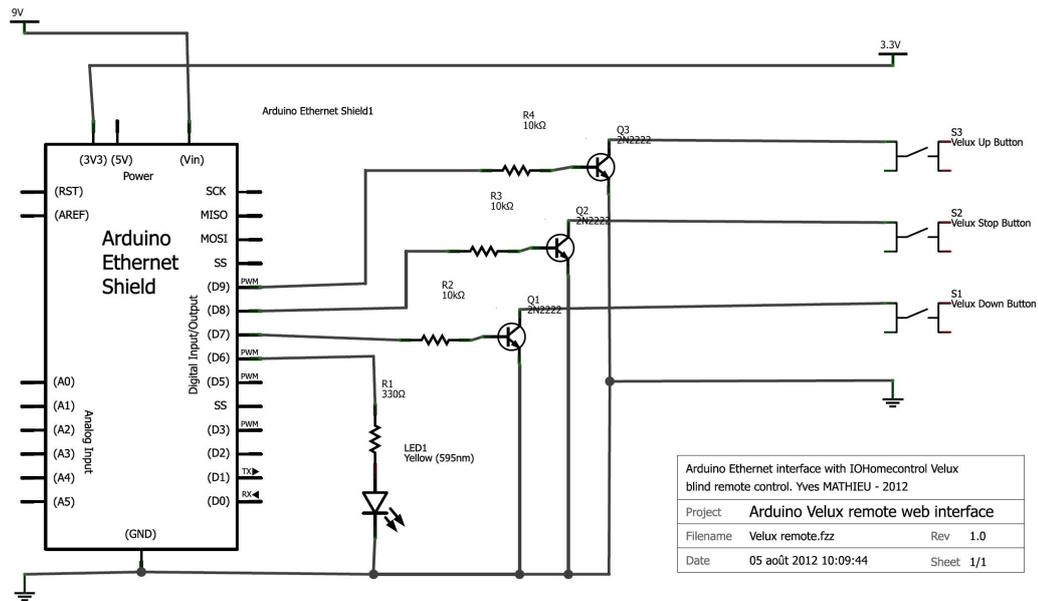


Abbildung 4.8: Schaltplan für Steuerung der Velux Fernbedienung

Folgende Komponenten wurden in dem IP Gateway verbaut:

- 1 x Arduino Ethernet
- 3 x NPN Transistor 2N2222
- 3 x 10k Widerstand
- 2 x LED, rot und gelb
- 1 x Gehäuse
- 1 x Fernbedienung
- 1 x Breadboard
- Diverse Steckbrücken

Zuerst wurde das Gehäuse der Fernbedienung geöffnet und die Platine der Fernbedienung herausgenommen. Anstatt der zwei AAA Micro Batterien dient der 3.3V Ausgang des Arduino als Stromversorgung für die Platine. Dazu wurden die Batteriekontakte entlötet und anschließend durch Steckbrückendraht ersetzt. Um den Aufwand der Lötarbeiten zu minimieren, wurde ein *Breadboard* (deutsch: Steckplatine) für den Aufbau des ersten Prototypen verwendet. Nach dem Ersetzen der Batteriekontakte wurde die Platine über das *Breadboard* mit der Stromversorgung und dem GND des Arduino verbunden und die Funktion anschließend geprüft.

Die Elektronik der Tasten besteht aus zwei konzentrischen, kreisförmigen Kontakten, die beim Tastendruck kurzgeschlossen werden. Dabei wird ein leitendes Material flach auf die Kontakte gedrückt, wodurch der Kontakt geschlossen wird. Abbildung 4.9 zeigt die Tasten der Fernbedienung mit und ohne leitender Abdeckung. Mit der Durchgangsprüfung eines Multimeters wurde anschließend verifiziert, dass der negative Pol der Taste direkt mit dem GND der Platine verbunden ist. Wenn der Kontakt geschlossen wird, schließt sich der Stromkreis und der vorgeschalteter Verbraucher verrichtet seine Arbeit.

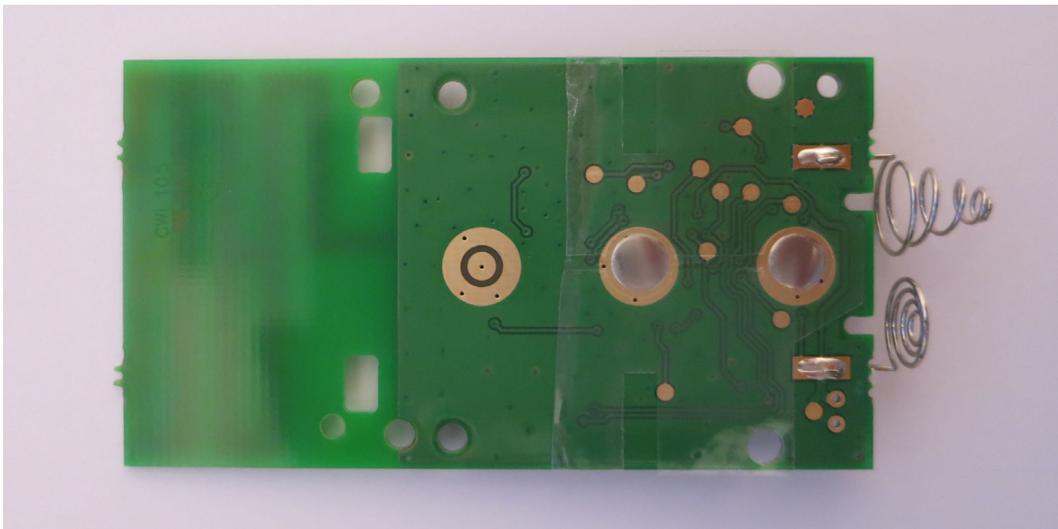


Abbildung 4.9: Platine der Velux Fernbedienung

Um die Tasten der Fernbedienung elektronisch auslösen zu können, wird auf Anfrage der positive Pol der Taste über einen NPN Transistor mit dem GND verbunden (siehe Schaltplan in Abbildung 4.8). Der Kollektor wird mit der Taste verbunden und der Emitter mit dem GND. Über den Basis Anschluss des Transistors steuert der Arduino mit einem digitalen Ausgang den Stromfluss von Kollektor zu Emitter. Dazu wurde je ein Steckbrückendraht an den positiven Pol der Taste gelötet. Der erste Versuch schlug fehl und eine Fernbedienung wurde dabei zerstört. Dank etwas Übung und besserem Werkzeug klappte es schlussendlich beim zweiten Versuch. Abbildung 4.10 zeigt den fertigen Aufbau. Mit Hilfe eines dünnen Bohrers könnte eine Durchsteckmontage realisiert werden und damit das verlöten um einiges einfacher gestaltet werden.

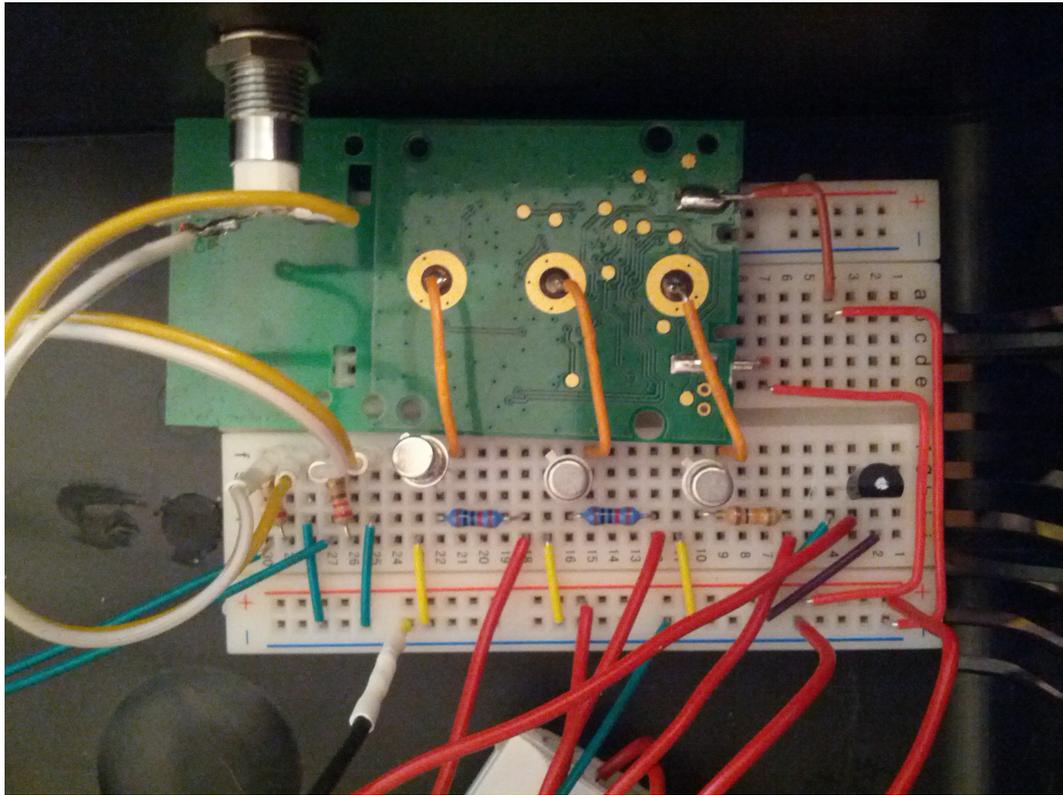


Abbildung 4.10: Arduino steuert Velux Fernbedienung

Die Elektronik wurde anschließend in ein geräumiges Gehäuse eingebaut, um zusätzlichen Platz für weitere Fernbedienungen zu bieten. Zwei LEDs an der Vorderseite stellen den aktuellen Status des Geräts dar und bieten damit eine einfache Möglichkeit zur Fehlersuche. Die Entwicklung eines prototypische *Sketches* (Arduino Programm) zur Steuerung gestaltete sich durch die Vorlage im *Homeseer Forum* und die große Auswahl an Anleitungen im Internet als sehr einfach. Das bloße Schalten eines digitalen Ausgangs gehört zu den Grundlagen in der Mikrocontroller-Programmierung. Zur Verarbeitung von Netzwerkanfragen wurde die Webduino Webserver Programmibliothek verwendet [140]. Zusätzlich zu den grundlegenden Funktionen Auf, Zu und Stopp wurde auch eine Zeitschaltung für 1, 5, 15 und 30 Minuten implementiert. Nach Ablauf der Zeit schließen die Fenster automatisch. Die Verarbeitung von Parametern in URLs stellte sich als ausgesprochen schwierig heraus, denn wie viele für Mikrocontroller ausgelegte Programm-bibliotheken ist auch Webduino sehr minimalistisch gehalten. Deshalb wurden für jedes Fenster eigene Endpunkte in Form von URLs erstellt. Mit den direkten Steuerungsmöglichkeiten Auf, Zu und Stopp sowie der Zeitsteuerung wurden also insgesamt sieben URLs angelegt. Der Befehl ‘Wohnzimmer Fenster Auf’ lässt sich beispielsweise über die URL http://192.168.1.46/WZ_UP aufrufen.

Um die Entwicklung der Zeitsteuerung zu vereinfachen, wurde die *TimedEvent* Programm-bibliothek verwendet [141]. Mit dessen Hilfe können Codeabschnitte zeitverzögert ausgeführt

werden. Die Programmierung des Timers ist dadurch im Vergleich zur herkömmlichen Konfiguration sehr einfach. Timer werden mit der Programmbibliothek folgendermaßen verwendet:

```
...
// Timer anlegen um nach 1000 Millisekunden die Funktion
// functionToCall aufzurufen und das Ereignis 1337 zu signalisieren
TimedEvent.addTimer(1337, 1000, functionToCall);

// Timer starten
TimedEvent.start(1337);
}

void loop()
{
// Dieser Aufruf in der loop Funktion des Arduino Sketches ist essenziell
TimedEvent.loop();
}

void functionToCall(TimerInformation* Sender, int param)
{
Serial.println("Timer called function with event " + String(param));

// Timer stoppen, neue Verzögerung auf 3000ms festlegen und erneut starten
TimedEvent.stop(1337);
TimedEvent.setNewDelay(1337, 3000);
TimedEvent.start(1337);
}
}
```

Der fertige Aufbau inklusive Gehäuse ist in Abbildung 4.11 zu sehen. Die grüne LED zeigt, dass das Gerät mit dem Netzwerk verbunden ist. Bei einer Tastendruck Aktion blinkt die gelbe LED einmal auf. Beim Verbindungsaufbau mit dem Netzwerk leuchtet die gelbe LED. Bei einer Pause zwischen Verbindungsversuchen blinkt sie.



Abbildung 4.11: io-homecontrol IP Gateway im Gehäuse

4.3.4 Kaffeemaschine

Die Fernsteuerung der Kaffeemaschine über ein IP Netzwerk wurde mit einem iTach WF2CC von Global Caché realisiert (siehe Abbildung 4.12). Das WF2CC besitzt drei Relais, die über eine TCP Schnittstelle geschaltet werden können, um den Stromfluss zu steuern. Während das IP2CC Modell über einen RJ45 Anschluss verfügt, bietet das WF2CC Modell eine 802.11b

WLAN Schnittstelle. Der WF2CC wurde zur Entwicklung der Lösung gewählt, da ein nicht benötigtes Gerät bereits vorhanden war und sich die WLAN Verbindung der Geräteserie als sehr praktisch und äußerst stabil erwies. Abbildung 4.12 zeigt das Gerät von hinten mit angebrachten Drähten.



Abbildung 4.12: iTach WF2CC von Global Caché

Als Kaffeemaschine kam ebenfalls ein ausgemustertes und nicht mehr benötigtes Gerät zum Einsatz. Die TK50N01 Nespresso Maschine von Siemens wird normalerweise über jeweils zwei Tasten links und rechts am Gerät bedient. Glücklicherweise dient eine der Tasten bloß zum Ein- und Ausschalten der Heizplatten für Tassen. Die drei Relais alleine reichten also vollkommen aus, um alle benötigten Funktionen von der Ferne auslösen zu können. Von der Platine der Kaffeemaschine 4.13 führen jeweils drei Kabel zu einer Taste. Mit dem Durchgangsprüfer des Multimeters wurde an den nicht isolierten Enden der Drähte auf der Tastenplatine überprüft, welche der Drähte durch einen Tastendruck kurzgeschlossen werden. Das andere Kabel dient lediglich dazu, die Status-LED in der Taste zu steuern. Die identifizierten Kabelpaare wurden dann an einer kleinen Stelle in der Nähe der Hauptplatine abisoliert und jeweils ein Draht eingelötet. Die zwei Kabel pro Taste wurden anschließend über ein Loch im Gehäuse nach außen geführt und mit je einem Relais verbunden. Per TCP Verbindung wird dann das jeweilige Relais für eine kurze Zeit geschlossen und wieder geöffnet, um einen Tastendruck zu simulieren. Abbildung 4.13 zeigt die Elektronik der Kaffeemaschine sowie den fertigen Aufbau.

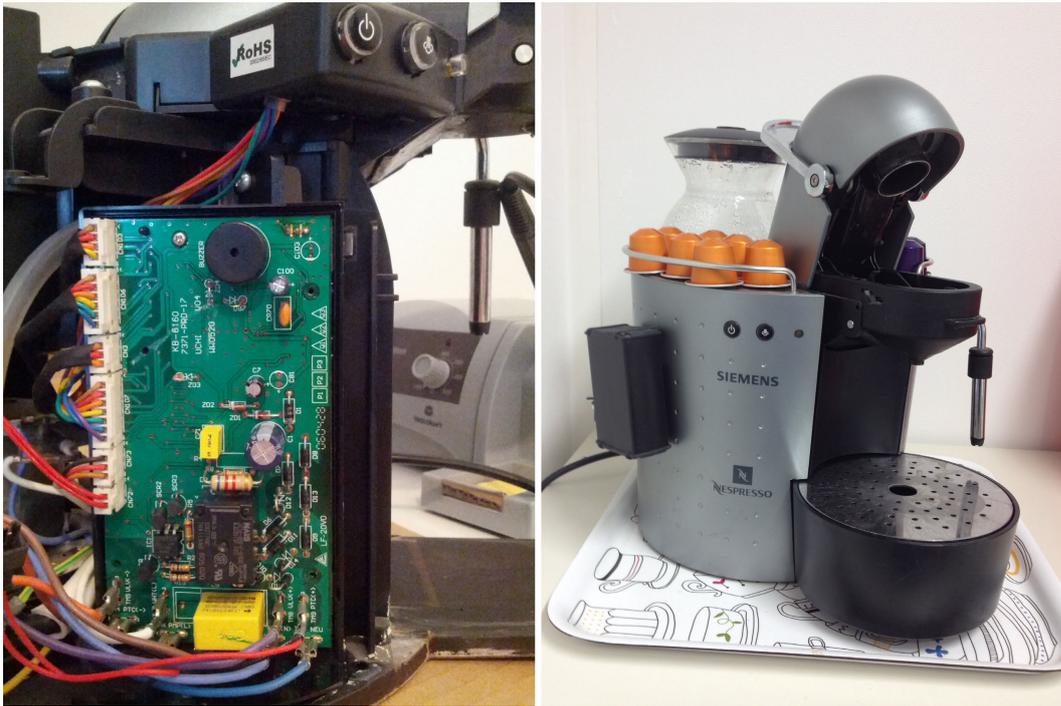


Abbildung 4.13: Modifizierte Kaffeemaschine

Wird der folgende Text an das Gerät übermittelt, wird Status des Relais mit der Nummer *<relaisnumber>* auf *<outputstate>* gesetzt:

```
setstate,1:<relaisnumber>,<outputstate>
```

Der iTach antwortet anschließend mit folgendem Text, um den aktuellen Status (*<currentstate>*) des Relais mit der Nummer *<relaisnumber>* zu signalisieren:

```
state,1:<relaisnumber>,<outputstate>
```

Die restlichen Funktionen der iTach CC API beschränken sich auf Statusabfragen der Relais und den gewöhnlichen iTach Funktionen, wie dem Auffinden des Geräts mittels UDP-Multicast und diverser Informationsabfragen.

4.3.5 NFC-Tags

Vorerst kamen quadratische 35mm 'MIFARE Classic' und 'MIFARE Ultralight C' Tag-Sticker aus einem NFC Starter-Kit zum Einsatz. Neben den zehn Aufklebern bot das Einsteigerpaket auch zwei Schlüsselanhänger und vier Scheckkarten. Nachdem der Vorrat an Aufklebern aufgebraucht und die Frage nach dem benötigten Speicherplatz geklärt war (siehe Implementierung in Abschnitt 4.3.7), wurden runde 35mm NTAG203 Sticker besorgt. Die 144 Byte sind mehr als genug Speicherplatz für mehrere UUIDs und die Sicherheitsfeatures teurer Tags werden in der geplanten Anwendung nicht benötigt.

4.3.6 Unterhaltungselektronik

Folgende Geräte der Unterhaltungselektronik waren in der Testumgebung vorhanden und wurden in die Smartphone Fernbedienung integriert:

- Videoprojektor: PT-AE900E
- Mehrkanal Hifi Verstärker: AV Receiver TX-SR605
- Spielkonsole: Microsoft Xbox 360
- HTPC: Intel Core 2 Duo E7400 (2x 2.8GHz), 4GB RAM, Nvidia 9500 GT passiv, Win 7
- XBMC: Xbox Media Center for Windows v12.1.0.0, Autostart auf HTPC
- Motorleinwand: IR Schnittstelle, Funktionen rauf/runter/stop, kein automatischer Stopp

4.3.7 Android Universalfernbedienung reYmote

Als Ausgangspunkt für die Fernbedienung der Geräte via Android Smartphone dient die Applikation reYmote. reYmote ist eine Software zur Erstellung von Universalfernbedienungen. Die Applikation wurde vom Autor dieser Arbeit entwickelt und wird seit Anfang 2012 kommerziell über Google-Play vertrieben. Im Folgenden wird zuerst der Zustand von reYmote zu Beginn dieser Arbeit erläutert. Anschließend werden zahlreichen Weiterentwicklungen und Konfigurationsarbeiten beschrieben, welche notwendig waren, um den Szenarios und den neuen Anforderungen gerecht zu werden.

Status v1.0.10

Zu Beginn der ersten Entwicklungsphase befand sich reYmote in der Version 1.0.10. Das zugrundeliegende Datenmodell bestand aus **Kommandos**, **Schnittstellen**, **Fernbedienungen** und **Tasten** (siehe Abbildung 4.14). Die Entitäten des Datenmodells sind über eine UUID eindeutig definiert. Dabei handelt es sich um eine zufällig generierte, hexadezimale ID, die in ihrer String Darstellung 37 Bytes benötigt (z.B.: 1b3c3386-9caf-4820-85af-9db850299bf0).

Bei **Schnittstellen** handelt es sich um IP Schnittstellen. Es existieren auch speziellere Typen von **Schnittstellen** in reYmote für die Steuerung des XBMC oder Geräte der iTach Familie (siehe Abschnitt 4.3.6). Ihre Verwendung erleichtert der BenutzerIn die Konfiguration der **Schnittstelle** und passender **Kommandos** und fügt eine Statusanzeige für das Gerät hinzu.

Kommandos übermitteln typischerweise eine Nachricht an eine **Schnittstelle**, um eine Aktion auszulösen. Dabei stehen unterschiedliche Kommandotypen für die Protokolle TCP, UDP und HTTP zur Verfügung. Es existieren jedoch auch **Kommandos**, die nicht mit einer **Schnittstelle** kommunizieren. Dabei handelt es sich um **Kommandos**, die beispielsweise mehrere andere **Kommandos** nacheinander ausführen oder die Anzeige einer anderen **Fernbedienung** auslösen.

Um die in Abschnitt 4.3.3 beschriebene Fenstersteuerung mit reYmote zu bedienen, konfiguriert

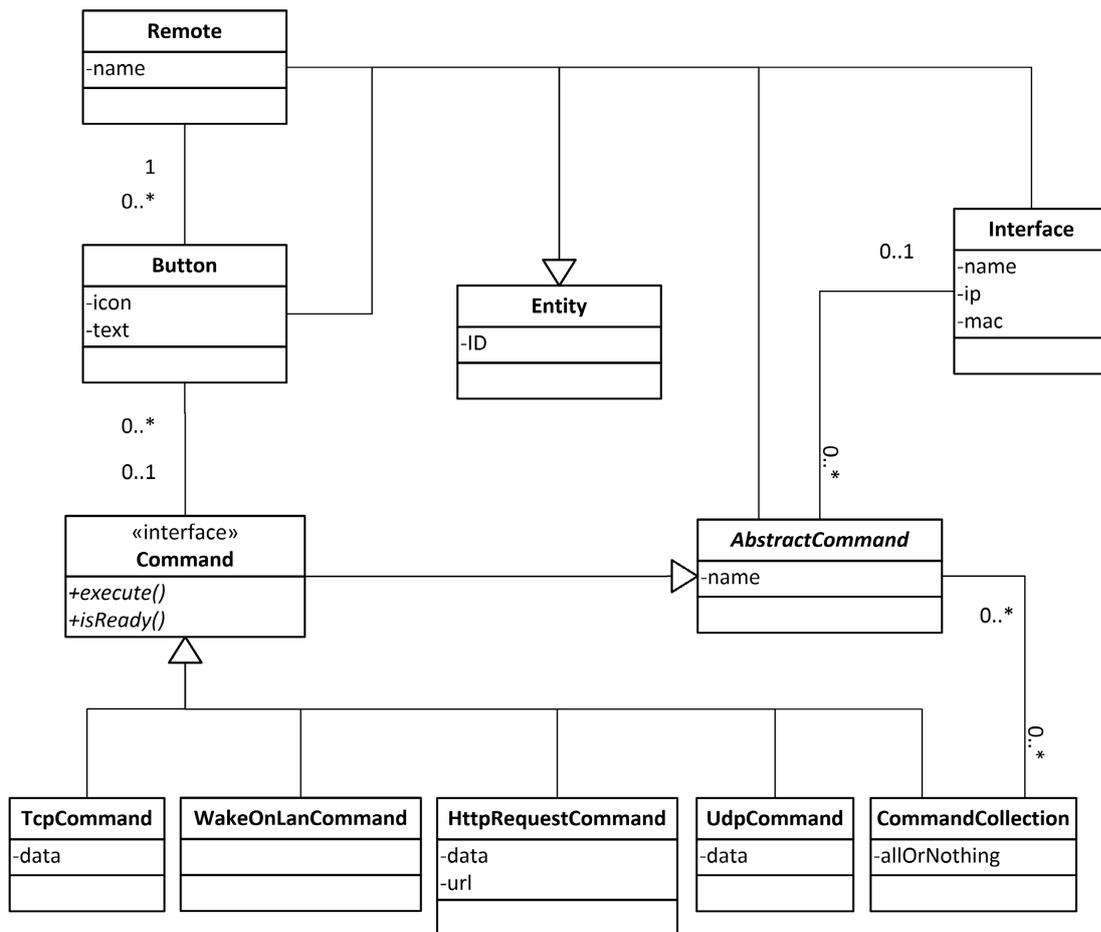


Abbildung 4.14: Datenmodell v1.0.10

man eine **Schnittstelle** mit der IP '192.168.1.46' und ein dazugehöriges HTTP **Kommando** mit dem URL-Suffix '/WZ_UP'. Die Ausführung des **Kommandos** führt eine HTTP GET Anfrage auf die URL 'http://192.168.1.46/WZ_UP' aus, wodurch die Fenster geöffnet werden.

Auszuführende **Kommandos** werden in eine *Queue* eingefügt und der Reihe nach abgearbeitet. Eine *Queue* ist eine Warteschlange die FIFO (First In - First Out) abgearbeitet wird. Es kommen also früher eingefügte Kommandos auch früher zur Ausführung.

Mit Hilfe eines grafischen Editors können eigene Benutzeroberflächen (**Fernbedienungen**) mit **Tasten** und **Hotspots** erstellt werden. Zuvor erstellte **Kommandos** müssen anschließend den UI-Elementen zugewiesen werden, um wiederum per Knopfdruck ausgeführt werden zu können. reYmote bietet jedoch auch eine XML Import- und Exportfunktion, um die Erstellung komplexer Konfigurationen zu erleichtern und sie leicht austauschen zu können.

Entwicklungen v1.1

In der ersten Iteration mussten zahlreiche Erweiterungen und Anpassungen der Android Applikation reYmote durchgeführt werden, um die Szenarios und Anforderungen bestmöglich zu unterstützen. In diesem Abschnitt werden die Änderungen im Detail beschrieben.

Zu Beginn der Weiterentwicklungen von reYmote wurden die Möglichkeiten zur Implementierung der NFC Funktionalität untersucht. Um ein Kommando durch das Lesen eines Tags ausführen zu können, muss dieser zuerst beschrieben werden. Für die zu schreibenden Daten boten sich zwei offensichtliche Möglichkeiten, die als XML serialisierten Kommandos oder ihre UUIDs. Die Datenmenge der serialisierten Kommandos und deren Schnittstellen ist jedoch viel zu variabel, um sicherstellen zu können, dass sie überhaupt auf einen Tag geschrieben werden können. Das Schema der XML Konfiguration sowie das Datenformat selbst sind für so geringen Speicherplatz nicht geeignet. Außerdem würde dies unter Umständen Sicherheitsrisiken mit sich bringen. Daher wurden die UUIDs der Kommandos als Daten für die NFC-Tags gewählt. Diese werden in ihrer 37 Byte String Darstellung auf den Tag geschrieben. Wenn zwei IDs auf den Tag geschrieben werden, wird ein Trennzeichen dazwischen eingefügt. Die zweite UUID ist notwendig, um beim Entfernen des Smartphones vom Tag, eine andere Aktion auslösen zu können. Somit bieten selbst NFC-Tags mit geringem Speicherplatz, wie etwa der Mifare Ultralight mit 48 Bytes beschreibbarem Speicher, genügend Platz für eine Referenz auf ein Kommando.

Um ein Kommando auf einen Tag schreiben zu können, wurde der Menüpunkt 'Write to NFC' (siehe Abbildung 4.15) im Untermenü der Kommandoliste hinzugefügt. Nach Auswahl der Option wird die BenutzerIn darum gebeten, das Gerät an den Tag zu halten. Ein Signalton vom Betriebssystem und visuelles Feedback in der Applikation geben Auskunft über den Erfolg des Vorgangs.

Liest man einen NFC-Tag mit Daten von reYmote ein, ist der Android interne Signalton ebenfalls zu hören und die Applikation zeigt ein GUI mit einer runden *Progressbar*, um der BenutzerIn zu vermitteln, dass gerade eine Aktion ausgeführt wird. Anschließend lädt reYmote das Kommando aus der Datenbank und reiht es zur Ausführung in die Warteschlange. Nach der Ausführung ertönt ein zum Ausführungsergebnis passender Signalton der Applikation und das GUI wird verlassen.

Um den effizienten Wechsel zwischen verschiedenen Aktivitäten durchführen zu können, wurde das Datenmodell erweitert und die Verarbeitung der Kommandos stark überarbeitet. Die Problematik und der Mangel an Funktionalität der bisherigen Implementierung lässt sich durch ein Beispiel aus der Praxis darstellen. Im folgenden wird ein einfacher und typischer Aufbau bestehend aus Fernseher und Spielkonsole zur Veranschaulichung herangezogen, mit denen die fiktive BenutzerIn die Aktivitäten **fernsehen** und **spielen** durchführt.

Um **fernsehen** oder spielen zu können, muss der Fernseher eingeschaltet sein und das passende Eingangssignal gewählt werden. Zusätzlich dazu muss zum **Spielen** die Spielkonsole gestartet werden. Um vom **Fernsehen** zum **Spielen** überzugehen, muss lediglich die Konsole eingeschalt-

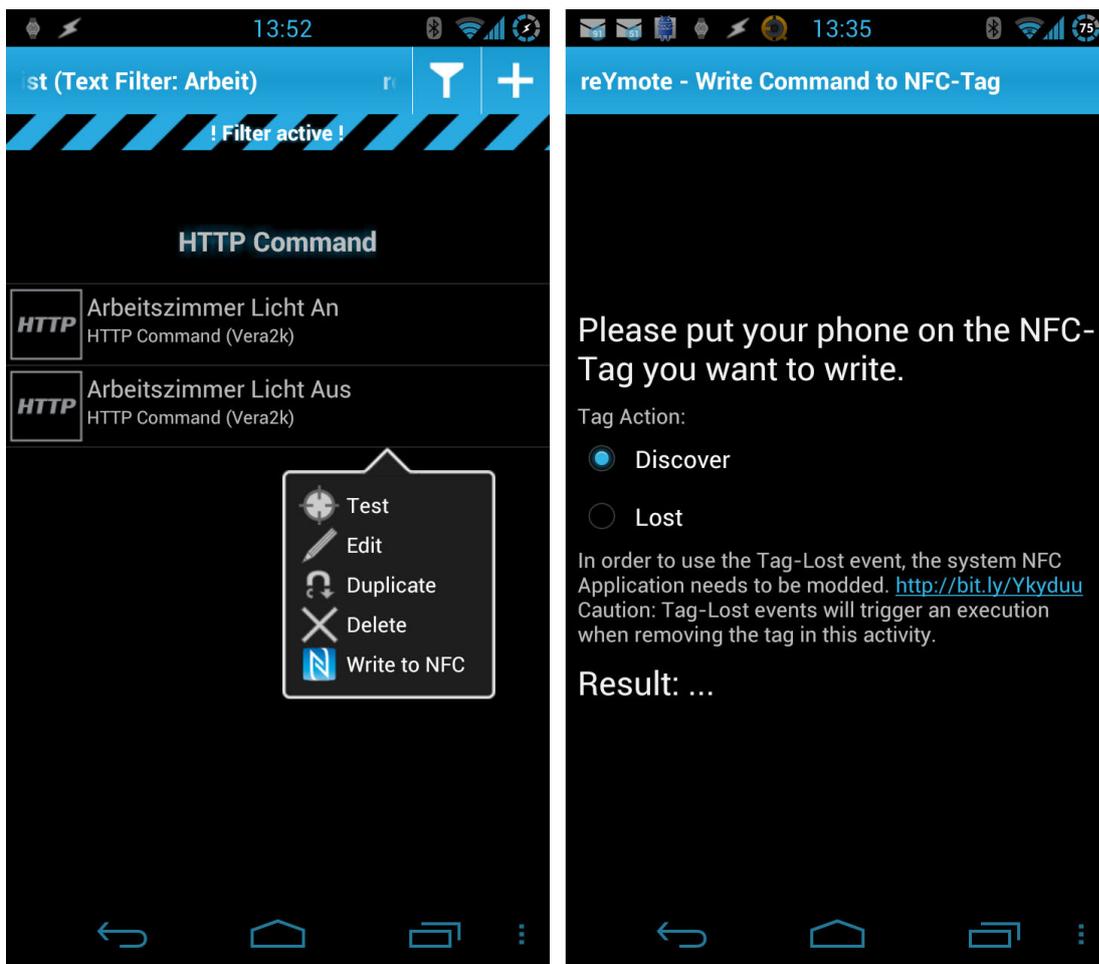


Abbildung 4.15: reYmote NFC Benutzerschnittstellen

tet und das Eingangssignal gewechselt werden. Wie bereits in Abschnitt 2.4 erwähnt werden die meisten herkömmlichen Geräte der Unterhaltungselektronik per Infrarot gesteuert. Für einen Großteil der Geräte existieren keine diskreten Kommandos für wichtige Funktionen wie etwa das Ein- und Ausschalten oder dem Wechsel von Eingängen. Um unter diesem Umstand von **Fernsehen** zu **Spielen** zu wechseln, ist es notwendig den Status des Fernseher zu kennen, um zu wissen, ob ein Kommando zum Einschalten an den Fernseher gesendet werden soll. Einige solcher Problemstellungen können umgangen werden, indem man andere Signale verwendet, die als diskrete Kommandos eingesetzt werden können. Auf vielen Fernsehgeräten könnte zum Beispiel die Taste '1' als diskretes Einschaltssignal verwendet werden.

Ein weiteres gutes Beispiel für die Notwendigkeit diskreter Kommandos ist die Motorleinwand, welche in der Testumgebung verwendet wird. Hierbei lassen sich durch Kombination einer 39-sekündigen Pause und der Kommandos 'Rauf', 'Runter' und 'Stopp', zusammengesetzte Kom-

mandos für ‘Leinwand-An’ und ‘Leinwand-Aus’ definieren. Das Kommando ‘Leinwand-An’ besteht dann aus:

1. Leinwand runter
2. 39 Sekunden warten
3. Leinwand stopp

Wie beim Beispiel des Fernsehers würde jedoch auch hier eine mehrfache Ausführung zu einem unerwünschten Ergebnis führen.

Aus diesem Grund wurde das Datenmodell um die Klassen **Aktivität** und **Gerät** erweitert (siehe Abbildung 4.16). Zu einem **Gerät** kann je ein Kommando für Ein- und Ausschalten sowie andere Kommandos, die den Status nicht verändern, hinzugefügt werden. Mit dieser Information kann die Applikation gezielt **Geräte** nach Bedarf ein- und ausschalten. Die Lösung eignet sich jedoch nur bedingt dazu, den Status der **Geräte** mitzuverfolgen. Dazu müsste die Applikation alle Kommandos kennen, die eine Statusänderung hervorrufen können. Von dieser Möglichkeit wurde abgesehen, da es den Konfigurationsaufwand unnötig erhöhen würde. Kommandos melden ein Ereignis vor und nach ihrer Ausführung. Wird dabei festgestellt, dass es sich um ein Kommando zum Ein- oder Ausschalten für ein **Gerät** handelt, wird der Status des **Geräts** verändert. Wenn sich das **Gerät** bereits im richtigen Zustand befindet, wird nichts unternommen. Das Ereignis vor der eigentlichen Ausführung bringt das Gerät in einen *Pending-On* oder *Pending-Off* Status der solange Anhält, bis die Ausführung vollständig und erfolgreich abgeschlossen wird.

Von der Konfiguration von Eingangskanälen wurde in dieser Iteration ebenfalls abgesehen. Das wäre zwar eine notwendige Änderung, um das Datenmodell besser an die Domäne anzupassen, für die Zwischenzeit wurde jedoch ein Workaround gewählt. Anstatt die reale Welt nachzubilden und ein **Gerät** mit x Kanälen zu konfigurieren, wurden x verschiedene **Geräte** erstellt. Die Einschaltkommandos sind in diesem Fall die Signale zum Kanalwechsel **Gerät**. Der verwendete Mehrkanal Hifi Verstärker schaltet sich beim Signal für den Kanalwechsel automatisch ein. Die Verwendung mehrerer solcher **Geräte** in einer **Aktivität** macht klarerweise keinen Sinn.

Eine **Aktivität** besteht aus einer Auswahl von **Geräten**. Beim Starten einer **Aktivität** werden zuerst alle nicht benötigten **Geräte** abgeschaltet. Alle in der Aktivität enthaltenen **Geräte** werden anschließend eingeschaltet. Bereits eingeschaltete **Geräte** bleiben eingeschaltet.

Manche Geräte benötigen eine gewisse Zeitspanne, um nach dem Einschalten auf weitere Befehle reagieren zu können. Mit bidirektionaler Kommunikation könnte beispielsweise der Status eines Geräts abgefragt werden, um darauf zu warten, bis das Gerät bereit ist. Die Infrarotschnittstellen herkömmlicher Unterhaltungselektronik bieten jedoch natürlich keinen Rückkanal. Deshalb wurde eine Einstellungsmöglichkeit geschaffen, um eine Einschaltdauer für **Geräte** spezifizieren zu können. Innerhalb dieser Zeit werden keine weiteren Kommandos des **Geräts** ausgeführt. Damit wird verhindert, dass zum Beispiel ein Kommando zum Wechseln des Eingangssignals beim Start verloren geht.

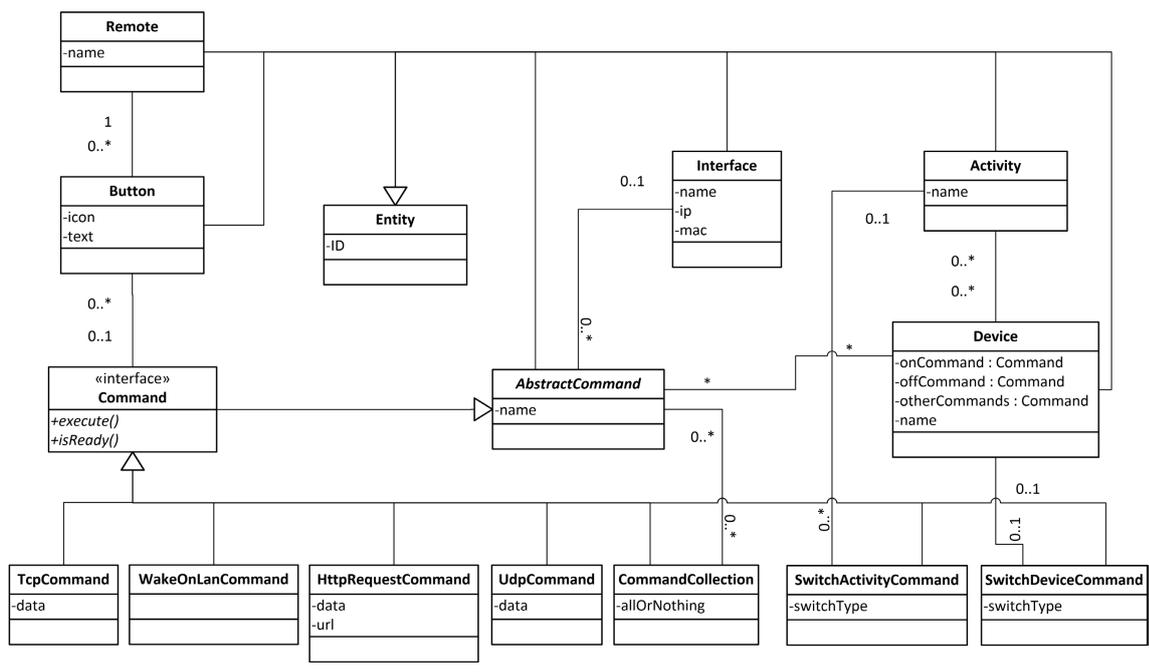


Abbildung 4.16: Datenmodell v1.1

In der vorigen Implementierung mussten später eingereichte Kommandos bedingungslos die Ausführung von früher eingereichten Kommandos abwarten. Um die Ausführungszeit der Starts von **Aktivitäten** zu verkürzen, wurde die Verarbeitung, die bis dahin single-threaded war, auf die Verwendung mehrerer *Threads* erweitert. Statt nur eines einzigen *Threads* für alle **Geräte** wird für jedes zu bedienende **Gerät** bei Bedarf ein eigener *Thread* gestartet. Maximal existieren jedoch nur fünf *Threads* gleichzeitig zur Abarbeitung, um Ressourcen zu schonen. Diese Änderung wirkte sich deutlich spürbar positiv auf die Ausführungszeit aus, wenn Kommandos Wartezeiten (wie etwa durch eine Einschaltverzögerung) auslösten. Kommandos, die keinem Gerät zugeordnet sind, werden in eine spezielle *NoDevice-Queue* eingefügt. Dazu gehören auch Kommandos, die **Aktivitäten** starten und stoppen. Alle Unterkommandos (Kommandos, die Teil eines anderen Kommandos sind oder von einem Kommando zur Laufzeit erzeugt werden) werden in dieselbe *Queue* eingefügt wie ihr Überkommando, sofern dieses bereits der *Queue* eines **Geräts** zugeordnet ist. Kommandos, die **Aktivitäten** starten, werden zwar in die *NoDevice-Queue* eingefügt, die Kommandos es zur Laufzeit generiert, werden jedoch in die entsprechende *Queue* des jeweiligen **Geräts** eingefügt.

Durch die Änderungen in der Kommandoausführung waren weitere, geringfügige Anpassungen notwendig. Wird eine Instanz des Apache HTTP-Client aus mehreren *Threads* heraus verwendet, muss der speziell dafür vorgesehene *ThreadSafeClientConnManager* eingesetzt werden. Die Notwendigkeit dessen ist jedoch sehr auffällig da andernfalls *Exceptions* (Fehler, deutsch: Ausnahmen) geworfen werden.

Während der Entwicklungsarbeit wurde die zugrunde liegende XML Konfiguration laufend an die neuen Möglichkeiten zur Konfiguration angepasst und erweitert. Um die Daten nicht jedes Mal manuell auf das Gerät kopieren zu müssen, wurde eine neue Einstellungsmöglichkeit geschaffen. Dadurch können Konfigurationen von einem Webserver geladen und direkt importiert werden. Diese Maßnahme erleichterte und beschleunigte einen kleinen Teil der Entwicklungsarbeit erheblich.

Konfiguration

Folgende Geräte wurden in reYmote für die erste Iteration konfiguriert:

1. AV Tuner
2. AV DVD
3. AV Game
4. HTPC
5. Kaffeemaschine
6. Kino Beleuchtung
7. Motorleinwand
8. Videoprojektor
9. Xbox 360

Die Wahl des Eingangssignals am **Mehrkanal Hifi Verstärker** wurde durch den in den Weiterentwicklungen (siehe Abschnitt 4.3.7) beschriebenen Workaround realisiert (Geräte 1, 2 und 3). Der **Verstärker**, die **Xbox 360** und der **Videoprojektor** werden mit einem iTach WF2IR IP-Infrarot Gateway bedient. Der **HTPC** wird mittels *Wake On LAN* eingeschaltet und mit der *Shutdown* Funktion der **XBMC** Schnittstelle heruntergefahren. Um das Ein- und Ausschalten der **Kaffeemaschine** übernimmt der iTach WF2CC (siehe Abschnitt 4.3.4). Das Gerät **Kino Beleuchtung** kümmert sich beim Starten von Aktivitäten die den **Videoprojektor** verwenden um stark gedämpftes Licht und sorgt für das erneute Einschalten der Lichter im Heimkino und den angrenzenden Räumen beim Stoppen. Die **Motorleinwand** wird mit einem irTrans IP-Infrarot Gateway über dessen UDP Schnittstelle gesteuert.

Die Konfiguration enthält die vier Aktivitäten **Radio**, **Lounge**, **Spielen** und **Film**. Tabelle 4.1 zeigt, welche Geräte in den Aktivitäten enthalten sind.

	Radio	Lounge	Spielen	Film
AV Tuner	X			
AV DVD		X	X	X
AV Game			X	
HTPC		X		X
Kino Beleuchtung		X	X	X
Motorleinwand			X	X
RGB LED Beleuchtung		X	X	X
Videoprojektor			X	X
Xbox 360			X	X

Tabelle 4.1: Konfiguration von Aktivitäten der ersten Iteration

Touchscreen-Fernbedienung

Um reYmote als Touchscreen-Fernbedienung verwenden zu können, wurden verschiedene Bildschirme konfiguriert. Im Folgenden wird die grafische Benutzerschnittstelle der Fernbedienung und ihre Funktionsweise im Detail erklärt. Die Abbildungen 4.17 und 4.18 zeigen die verwendeten Icons. Die Abbildung 4.19 zeigt die unterschiedlichen Bildschirme der Anwendung.

Direkt nach dem Start wird der der Hauptbildschirm (a) angezeigt. Von dort können die zwei Aktivitäten 'Spielen' und 'Film' direkt ausgeführt werden. Dabei wird automatisch auf die jeweilige Fernbedienung zur Steuerung diverser Funktionen, wie etwa die Steuerung der Lautstärke und der Wiedergabe, gewechselt.

Die 'Spielen' Fernbedienung (b) enthält Tasten zur Steuerung von Licht und Lautstärke. Außerdem findet sich ganz oben eine Taste, um die momentane Aktivität zu beenden.

Die 'Film' Fernbedienung (c) ist der 'Spielen' Fernbedienung sehr ähnlich. Sie enthält zusätzlich Tasten zur Navigation im XBMC und zur Mediensteuerung. Durch das Pausieren der Wiedergabe wird das Gerät 'Kino Beleuchtung' ausgeschaltet. Durch erneutes Drücken der Pausieren Taste wird die Beleuchtung wieder eingeschaltet.

Die 'Radio' Fernbedienung (d) gleicht der 'Spielen' Fernbedienung ebenfalls bis auf zusätzliche Elemente. Hier kann zwischen zwei Webradios und dem FM Radio gewählt werden. Außerdem gibt es zwei Tasten zum Wechseln der Radiosender.

Mit dem Haus Icon auf dem Hauptbildschirm (a) gelangt man zum Hauptschirm der Heimsteuerung (e). Von ihm aus gelangt man auf weitere Bildschirme. Die Fernbedienung 'PCs' wird bei den Tests nicht verwendet.

Mit der Fernsteuerung für die Kaffeemaschine (f) kann Kaffee gemacht werden. Wird ein Kaffee bestellt, ohne die Maschine vorher einzuschalten, wird sie automatisch eingeschaltet.

Icon	Icon	Verwendung
	Aktivität beenden	Beenden und zum Hauptbildschirm zurückkehren
	Radio	Radio Aktivität starten / FM-Tuner verwenden
	Gaming	Xbox 360 Aktivität starten
	Movie	Heimkino Aktivität starten
	Heimsteuerung	Zum Bildschirm für die Heimsteuerung
	Licht An	Schaltet ein Licht ein / Stellt Licht für Heimkino ein
	Licht Aus	Schaltet ein Licht aus / Beendet Heimkino Licht
	Webradio Fritz	Startet das Fritz-Webradio
	Webradio abc-lounge	Startet das Lounge-Webradio

Abbildung 4.17: Icons der Touchscreen-Fernbedienung 1/2

Die Fernbedienung für die Fenster (g) enthält zur Zeit nur die Wohnzimmerfenster. Mit ihr können die Fenster geöffnet und geschlossen werden. Mit der Taste zwischen den Pfeilen kann das Fenster gestoppt werden.

Die Fernbedienungen (h) und (i) ermöglichen die Steuerung aller Lampen in der Testumgebung. Die Pfeil Taste bei der Treppe ermöglichen den schnellen Wechsel ins andere Geschoss.

Icon	Icon	Verwendung
	Lauter	Erhöht die Lautstärke
	Leiser	Senkt die Lautstärke
	Hinauf/Öffnen	Zur Navigation und zum Öffnen der Fenster
	Hinunter/Schließen	Zur Navigation und zum Schließen der Fenster
	Links	Navigation
	Rechts	Navigation
	Vorspulen	Mediensteuerung
	Zurückspulen	Mediensteuerung
	Pause/Fortsetzen	Mediensteuerung
	Stop	Mediensteuerung
	Stop-Fenster	Fenster

Abbildung 4.18: Icons der Touchscreen-Fernbedienung 2/2



(a) Hauptbildschirm



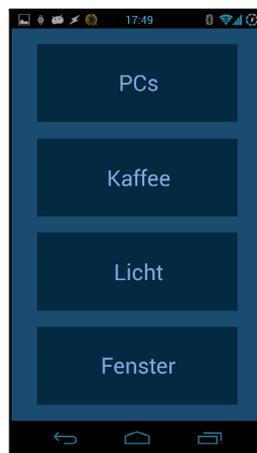
(b) Xbox 360



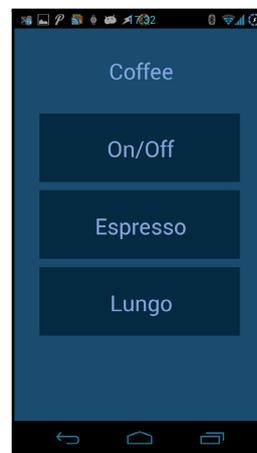
(c) Heimkino



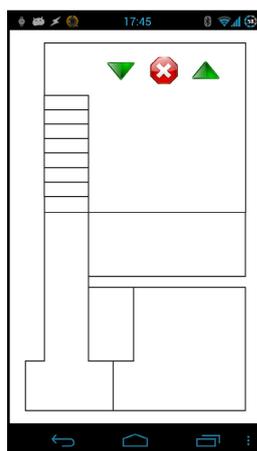
(d) Radio



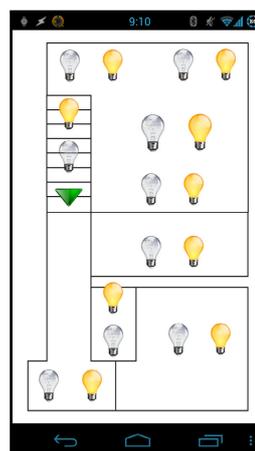
(e) Heimsteuerung



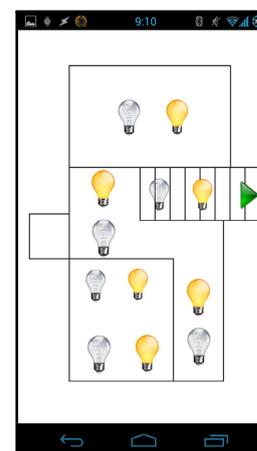
(f) Kaffemaschine



(g) Fenster



(h) Licht-EG



(i) Licht-OG

Abbildung 4.19: Bildschirmfotos der Touchscreen-Fernbedienung

NFC-Fernbedienung

Zur Interaktion durch Lesen von NFC-Tags wurden verschiedene Grafiken entworfen und ausgedruckt. Die NFC Sticker wurden an deren Rückseite angebracht. Die beklebten Ausdrücke wurden anschließend laminiert, ausgeschnitten und mit Klebeband oder Klebefolie befestigt. Somit sind sie vor Abnützungen geschützt und sehen ansehnlicher aus.

Einer der NFC-Tags befindet sich auf der Ablagefläche am Kopfteil des Betts (siehe Abbildung 4.20). Wenn die BenutzerIn schlafen geht, soll sie ihr Smartphone darauf platzieren. Anschließend werden alle Lichter (bis auf das im Schlafzimmer selbst) nach zwei Minuten ausgeschaltet. Entfernt man das Smartphone vom NFC-Tag, wird die Kaffeemaschine eingeschaltet und ein Kaffee gemacht.



Abbildung 4.20: NFC-Tags beim Bett

In der offenen Küche befinden sich NFC-Tags für die Fenstersteuerung der Fenster des angrenzenden Wohnzimmers sowie ein weiterer NFC-Tag, der die Wiedergabe des Webradios des Berliner Radiosenders 'Fritz' startet. Am Rande der Wohnküche befindet sich die Kaffeemaschine. An der Wand daneben befindet sich ein NFC-Tag zum Ausführen der Aktivität 'Radio'.

Der NFC-Tag zur Ausführung der Aktivität 'Spielen' wurde an der Rückseite des Gamepads angebracht. Da man es zum Spielen benötigt, schien dies eine sinnvolle Platzierung zu sein.

Ein Tag für die Aktivität 'Film' sowie ein weiterer zum Pausieren der Wiedergabe wurden am Couchtisch im Wohnzimmer angebracht (siehe Abbildung 4.21). Ebenso befindet sich dort ein Tag, um die aktuelle Aktivität zu beenden und noch ein Tag für die Radio Aktivität. Zusätzlich zur Wiedergabe des Webradiosenders 'ABC Lounge' dämpft er das Licht und aktiviert die RGB LED Beleuchtung, um eine angenehme und gemütliche Atmosphäre zu schaffen.



Abbildung 4.21: NFC-Tags auf Glastisch

4.4 Erste Testphase

In der Testphase wurde ein Testplan erstellt und Benutzertests durchgeführt. Der Testplan beschreibt die Ziele der Tests sowie die geplante Vorgehensweise. Die Ergebnisse wurden anschließend in der Evaluierungsphase ausgewertet und analysiert.

4.4.1 Ziele

Ziel der Benutzertests ist die Gegenüberstellung der verschiedenen Interaktionsmöglichkeiten zur Steuerung der Heimelektronik. Mit Hilfe von wenig strukturierten Interviews (siehe Abschnitt 3.6) und Beobachtungen (siehe Abschnitt 3.7) sowie der Analyse des Videomaterials soll die Benutzerakzeptanz evaluiert werden. Die Ergebnisse der SUS Fragebögen (siehe Abschnitt 3.9.3) gelten als Indikator für die subjektiv wahrgenommene Usability. Dadurch sollen die Ergebnisse der Benutzerakzeptanz Evaluierung weiter untermauert werden. Die Ergebnisse der MUSiC Metrik (ebenfalls in Abschnitt 3.9.3) für *Relative User Efficiency* sollen zeigen, wie lange die Testpersonen für die Durchführung der Aufgaben verglichen mit einer bereits geübten

Person brauchen. Die Metrik gilt als Indikator für die Erlernbarkeit eines Systems. Abschließenden Interviews sollen weitere Ideen für Verbesserungen und Erweiterungen aufbringen. Ein besonders wichtiges Ziel ist es, etwaige Probleme in der Verwendung der Interaktionen aufzudecken. Als weitere Zielsetzung gilt die Erfüllung der Wünsche und Ziele der Personas. Dazu werden die Testpersonen den jeweiligen Personas zugeordnet.

4.4.2 Testpersonen

Die Tests wurden mit fünf Personen durchgeführt. Die folgende Tabelle zeigt die Zuordnung zu den Personas. Außerdem wurde eine Einschätzung der Erfahrungen und Ambitionen in den Bereichen 'Benutzung von Smartphones und Tablets', 'Steuerung komplexer AV Einrichtungen' und 'Heimautomation' eingeholt. Tabelle 4.2 gibt einen Überblick über die TeilnehmerInnen. Hierbei steht das ✓ für 'Ja', das ✗ für 'Nein' und der ● für 'Wenig'.

Name	Persona	Smartphones	Komplexe AV Anlage	Interesse an HA
Christian	primär	✓	✗	●
Wilhelm	primär	✗	✗	✗
Christiane	primär	✗	✗	✗
Jeannette	primär	●	●	●
Konrad	sekundär	✓	✓	✓

Tabelle 4.2: TeilnehmerInnen der Benutzertests der ersten Iteration

4.4.3 Vorgehensweise

In jedem Testdurchlauf müssen die selben Aufgaben auf drei verschiedene Arten gelöst werden:

1. Auf herkömmliche Art und Weise mit den Steuerungsmöglichkeiten, welche die Geräte selbst bieten (original Fernbedienung, Tasten am Gerät)
2. Mit der Touchscreen-Fernbedienung
3. Mit den NFC-Tags und bei Bedarf mit der Touchscreen-Fernbedienung

Bei den Tests wird wie folgt vorgegangen:

1. Die Testperson wird um eine Einverständniserklärung für die Video- und Tonaufnahme des Testverlaufs und die Publikation der Ergebnisse gebeten.
2. Die Video- und Tonaufnahme wird gestartet und während des gesamten Testverlaufs durchgeführt.
3. Sicherstellen dass der Testbereich möglichst gut abgedunkelt ist (Jalousien, Markise).
4. Die Testperson darum bitten, ihre Handlungen zu kommentieren.
5. Durchführung der Aufgaben:

- a) Sicherstellen, dass die Kaffeemaschine einsatzbereit ist (Tasse, Wasser, Kapsel).
- b) Sicherstellen, dass das Licht im Wohnzimmerbereich brennt (Küche, Esstisch, Wohnzimmer).
- c) Kurze Erklärung der Interaktionsmöglichkeiten geben.
- d) Der BenutzerIn die Möglichkeit geben, sich kurz zu erholen, und etwas zum Trinken anbieten.
- e) Die BenutzerIn nach Unklarheiten fragen.
- f) Durchführung aller Aufgaben durch die Testperson.
- g) Die BenutzerIn nach Anmerkungen fragen.
- h) Die BenutzerIn den SUS Fragebogen ausfüllen lassen.

4.4.4 Testfälle

Die Testfälle orientieren sich an den in der Requirementsphase entstandenen Szenarios. Die Testspezifikation besteht aus mehreren Aufgaben, die sequenziell durchgeführt werden sollen. Nach der Durchführung aller Aufgaben mit einer der Methoden wird zur nächsten Methode übergegangen und wieder bei der ersten Aufgabe angefangen.

Aufgabe Nr.	1
Beschreibung	Die BenutzerIn möchte Radio hören.
Erfolgsbedingungen	Radio läuft.

Aufgabe Nr.	2
Beschreibung	Die BenutzerIn möchte Xbox 360 spielen.
Erfolgsbedingungen	1. Xbox 360 und alle notwendigen Geräte laufen (Ton und Anzeige). 2. Licht genügend abgedunkelt.

Aufgabe Nr.	3
Beschreibung	Die BenutzerIn möchte nun einen Film ansehen anstatt zu spielen.
Erfolgsbedingungen	1. Alle notwendigen Geräte laufen (Ton und Anzeige). 2. Licht genügend abgedunkelt. 3. Film Wiedergabe läuft.

Aufgabe Nr.	4
Beschreibung	Die BenutzerIn möchte die Wiedergabe pausieren um etwas aus der Küche zu holen.
Erfolgsbedingungen	1. Wiedergabe pausiert. 2. Licht in der Küche brennt.

Aufgabe Nr.	5
Beschreibung	Die BenutzerIn möchte die Wiedergabe fortsetzen.
Erfolgsbedingungen	1. Wiedergabe wird fortgesetzt. 2. Licht genügend abgedunkelt.

Aufgabe Nr.	6
Beschreibung	Die BenutzerIn möchte alle Geräte ausschalten (und schlafen gehen).
Erfolgsbedingungen	Alle Geräte der Unterhaltungselektronik sind ausgeschaltet.

Aufgabe Nr.	7
Beschreibung	Die BenutzerIn (wacht morgens auf und) möchte einen Kaffee trinken.
Erfolgsbedingungen	Kaffee ist fertig.

4.5 Erste Evaluierungsphase

Die Evaluierungsphase beschäftigt sich mit der Auswertung und Analyse der Benutzertests. Zuerst werden besondere Vorkommnisse der Benutzertests im Detail beschrieben und die Ergebnisse der Metriken präsentiert. Darauf folgt eine zusammenfassende Analyse der Tests. Die Ergebnisse aus dieses Abschnitts der Iteration werden anschließend dazu verwendet, weitere Ziele für die nächste Iteration zu definieren.

4.5.1 Benutzertests

Die Benutzertests wurden mit der herkömmlichen Interaktion begonnen. Dabei durften die Testpersonen ausschließlich die standardmäßig gegebenen Mittel zur Steuerung, wie Fernbedienungen und Gerätetasten verwenden. Die Durchführung des ersten Durchlaufs dauerte jeweils etwa 30 Minuten. Auffallend dabei war, dass alle BenutzerInnen anfangs mit den Infrarotfernbedienungen daneben zielten und sehr lange für die einzelne Schritte brauchten. Außerdem waren alle Testpersonen, die zur primären Persona gehörten, trotz initialer Erklärung von der Kanalauswahl des Mehrkanal Verstärkers verwirrt. Diese beiden Probleme zeigen deutlich den erhöhten Lern- und Konzentrationsaufwand bei der Bedienung komplexer AV Anlagen mit den herkömmlichen

Mitteln. Abbildung 4.22 zeigt alle Fernbedienungen die zur Durchführung des Tests verwendet werden durften. Von links nach rechts sind dies die Fernbedienungen für: Motorleinwand, Fenster, Videoprojektor, Mehrkanal Verstärker, Xbox 360 Gamepad und eine Drahtlose Maus-Tastatur Kombination.



Abbildung 4.22: Herkömmliche Fernbedienungen für die ersten Benutzertests

Die Lichtverhältnisse waren bei allen Tests deutlich suboptimal für den Betrieb des Videoprojektors. Die anfängliche Festbeleuchtung wirkte sich zusätzlich negativ auf die Bildqualität aus. Es hat jedoch keine einzige der Testpersonen aus Eigeninitiative das Licht abgedreht.

Die aktivitätsbasierte Bedienung war eine auffallend große Erleichterung für die Testpersonen. Die Durchführung der Aufgaben mittels NFC- und Touchscreen-Fernbedienung dauerte jeweils nur etwa zehn Minuten. Die User waren durchwegs begeistert davon, mit nur wenigen Interaktionen das komplette System bedienen zu können. Besonderen Gefallen fand die automatische Steuerung des Lichts, passend zur jeweiligen Aktivität.

Das Layout der Touchscreen-Fernbedienungen war verständlich und bereitete den BenutzerInnen nahezu keine Schwierigkeiten. Nur die Notwendigkeit vom Hauptschirm (a) aus zweimal auf das Radio Symbol tippen zu müssen, um FM Radio zu hören, stiftete Verwirrung. Selbst die in der Bedienung von Touchscreen Geräten ungeübten Testpersonen konnten die Aufgaben mit dieser Methode in kürzester Zeit erledigen. Drei der TeilnehmerInnen fanden auffallend viel Spaß daran, die Lampen der Testumgebung per Smartphone zu bedienen. Dabei wirkte sich eine Unzulänglichkeit in den Änderungen der Apache HTTP-Client Konfiguration negativ aus. Nach

der Ausführung mehrerer Kommandos zur Lichtsteuerung traten massiv erhöhte Verzögerungen in der Ausführung auf. Dabei kam es selbst bei der Steuerung einzelner Lampen zu einer Ausführungsdauer von bis zu einer Minute. Das Problem wurde gegebenenfalls durch einen Neustart der Applikation behoben.

Bei der freien Steuerung der diversen Lampen wurden kleine Fehler in der Zuweisung der Tasten aufgedeckt. Die Probleme wurden jedoch schon nach dem zweiten Benutzertest vollständig behoben.

Die TeilnehmerInnen waren mit der NFC-Fernbedienung nahezu gleich effizient wie mit der Touchscreen-Fernbedienung. Trotz initialer Erklärung über den Lesebereich des Smartphones hatten jedoch ausnahmslos alle Schwierigkeiten dabei, den Lesebereich des Smartphones auf Anhieb korrekt auf dem Tag zu platzieren. Nach dem Überwinden dieser Hürde waren sie aber vom Anzeigen der passenden Fernbedienung auffallend positiv überrascht. Nach ungefähr drei bis fünf Übungsversuchen waren die TeilnehmerInnen viel sicherer beim Einlesen der NFC-Tags und die anfänglichen Unsicherheiten verschwanden.

Das Pausieren der Wiedergabe durch das Liegenlassen des Geräts auf dem entsprechenden Tag, wurde nur von einer Testperson auf Anhieb richtig ausgeführt. Nach der Erklärung, dass die Wiedergabe beim Entfernen des NFC-Tags fortgesetzt wird, waren die Interaktion und der Mechanismus jedoch für alle verständlich und nachvollziehbar. Allen bis auf einer TeilnehmerIn gefiel diese Art der Interaktion sehr. Sie pausierten die Wiedergabe mehrmals unaufgefordert und es schien ihnen offensichtlich Freude zu bereiten. Die verbleibende TeilnehmerIn kritisierte, dass sie ihr Smartphone dann liegen lassen müsste und keine weiteren Tags damit einlesen könnte.

Trotz des Fehlers bei der Verwendung des NFC-Tags für Pause und der anschließenden Erklärung der Funktion zeigte sich das selbe Problem bei der Benutzung des Tags beim Bett. Nur zwei der Testpersonen haben das Smartphone tatsächlich dort liegengelassen. Dadurch wäre der Kaffee direkt beim Schlafengehen aus der Kaffeemaschine geflossen anstatt erst morgens.

Die Steuerung der Kaffeemaschine erheiterte die Testpersonen und animierte zu weiteren Ideen für die Heimsteuerung. Ein NFC-Tag im Auto könnte das Garagentor öffnen. Ein anderer im Büro könnte den Aufbruch in Richtung Heim signalisieren und diverse Aktionen auslösen, wie etwa die Heizung passend einstellen.

Zwei der Testpersonen, welche der primären Persona zugeordnet sind, stellten bei den abschließenden Gesprächen die selbe Frage: 'Wozu brauche ich die NFC-Tags, wenn ich doch das selbe mit dem Touchscreen machen kann?'. Es wurde versucht, dies folgenderweise zu begründen: 'Dabei handelt es sich um eine zusätzliche, günstige Erweiterung zur Interaktion. Die BenutzerIn platziert die NFC-Tags passend zu ihren Lebensgewohnheiten und häufigen Aktivitäten im eigenen Heim. Dadurch sollen die BenutzerInnen diese immer wiederkehrenden, mit Orten oder Gegenständen gekoppelten Steuerungsaufgaben, schnell und mit möglichst wenig Aufmerksam-

keit durchführen können.'. Die zwei zweifelnden Testpersonen blieben jedoch trotz der Erklärung nur mäßig befriedigt und von der Notwendigkeit der zusätzlichen Interaktionsmöglichkeit nicht vollkommen überzeugt.

Zwei TeilnehmerInnen (die TeilnehmerIn der sekundären Persona und die der primären Persona) fanden besonders gefallen an der NFC Interaktion und konnten sich gut vorstellen, ein solches System im eigenen Heim einzusetzen. Den verbleibenden Testperson gefiel die zusätzliche Interaktionsmöglichkeit, sie schienen jedoch das Vorhandensein einer aktivitätsbasierten Touchscreen Variante als einen deutlich wichtigeren Schritt im Bedienkomfort anzusehen.

Eine BenutzerIn bemängelte, dass die Icons der Touchscreen-Fernbedienung nicht mit denen der NFC-Tags übereinstimmten. Außerdem erwähnte sie, dass die Grafiken der Touchscreen-Fernbedienung leichter verständlich wären. Die Inkonsistenz der gewählten Grafiken wurde nur von einer Person negativ angemerkt, der niedrigere Erkennungswert war jedoch wahrnehmbar.

Während der einzelnen Testdurchläufe kamen unterschiedliche Kameras zum Einsatz. Leider wurde die Videoaufnahme in mehreren Fällen unerwartet unterbrochen, wodurch manche Sitzungen nur lückenhaft aufgezeichnet wurden. Aufgrund der Wettersituation kam die Fenstersteuerung nicht zum Einsatz.

4.5.2 SUS

Die SUS Metrik ist ein Maß für die Zufriedenheit der BenutzerInnen. Die Auswertung zeigt die Touchscreen-Fernbedienung als knappen Gewinner. Sowohl die NFC- als auch die Touchscreen-Fernbedienung liefern ein nahezu perfektes Ergebnis. Obwohl also die Sinnhaftigkeit der NFC basierten Interaktion von zwei BenutzerInnen in Frage gestellt wurde, bewerteten sie das System dennoch durchwegs als generell einfach und angenehm zu bedienen. Abbildung 4.23 zeigt ein Balkendiagramm mit den Ergebnissen des SUS Fragebogens.

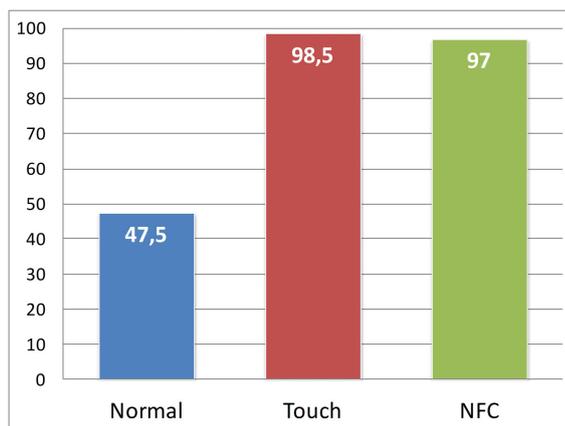


Abbildung 4.23: SUS Punkte

4.5.3 MUSiC

Die relative Benutzereffizienz der MUSiC Metrik ist ein Indikator für die Erlernbarkeit eines Systems. Zur Berechnung wird die Effizienz eines Experten zu den Testergebnissen der TeilnehmerInnen in Relation gestellt. Das beste Ergebnis lieferte die Touchscreen-Fernbedienung. Die etwas niedrigere Punktzahl der NFC-Fernbedienung entstand aus den anfänglichen Lese-problemen und den Unklarheiten bei der Interaktion durch Liegenlassen des Smartphones. Die Fehlbedienungen schlugen sich entsprechend der Schwere ihrer Auswirkungen auf die Effektivität aus. Den Kaffee, schon beim Schlafengehen zu machen, hatte eine Effektivität von 0.5 zur Folge. Die Fehlbedienung der Pause Funktion wurde mit Effektivität 0.8 gerechnet. Ohne diesen Malus hätte die NFC Variante eine relative Benutzereffizienz von 0.72 gehabt. Abbildung 4.24 zeigt ein Balkendiagramm mit den Ergebnissen der Metrik.

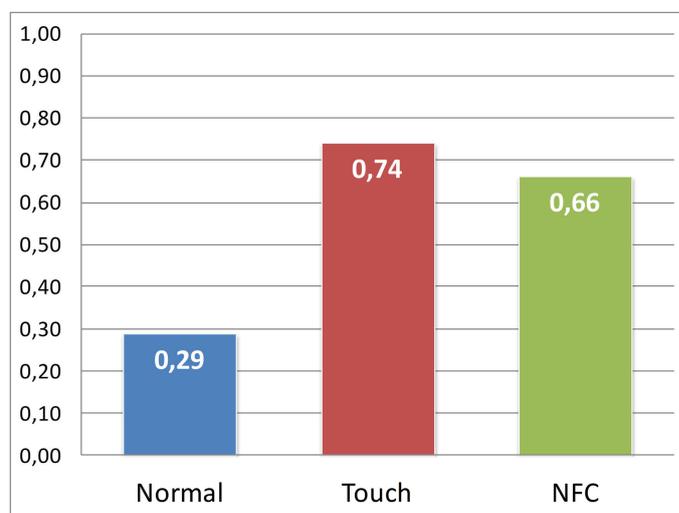


Abbildung 4.24: Relative Benutzereffizienz der MUSiC Metrik

4.5.4 Analyse und Zielsetzung

Die Ergebnisse der Benutzertests zeigten ein sehr positives Bild. Die Benutzerakzeptanz ist in beiden Benutzergruppen merklich vorhanden und die persönlichen Ziele der Personas gelten größtenteils als erfüllt. Einzig und allein das Ziel *kein Kontrollverlust* war durch die teils unausgereifte Implementierung gefährdet. Mit Hilfe der Beobachtungen und Interviews wurden Bereiche gefunden, die es in der nächsten Iteration zu verbessern gilt.

Die herkömmlichen Mittel und Wege zur Heimsteuerung bewiesen sich als wesentlich ineffizienter und mit weit höherem Lern- und Konzentrationsaufwand verbunden sind, als die aktivitätsbasierte Steuerung mit einer Universalfernbedienung.

Der zielgenaue Umgang mit dem NFC Lesegerät des Smartphones ist erst nach ein paar Übungsversuchen möglich. In der nächsten Iteration soll daher der Lesebereich des Geräts weiter unter-

sucht werden. Möglicherweise kann das Auslesen durch die Verwendung größerer Tags erleichtert werden.

An der Bedienung mittels aktivitätsbasierter NFC- und Touchscreen-Fernbedienung hatten die BenutzerInnen sichtlich Freude und sie war mit deutlich weniger Aufwand verbunden. Die Sinnhaftigkeit der NFC Variante wurde jedoch auch kritisiert. Zwei BenutzerInnen war nicht klar wozu sie die Tags benötigen, wenn sie das selbe Ergebnis auch mit der Touchscreen-Fernbedienung erreichen können. Dies ist möglicherweise auf die Testsituation und das starre Konstrukt an vorgefertigten Aufgaben zurückzuführen. Der Komfort den Bildschirm nicht einschalten zu müssen und sich ein paar Tastendrucke zu sparen, konnte so nicht gut zur Geltung kommen. Der nächste Benutzertest soll daher untersuchen, ob die Interaktion mittels NFC-Tags für die Testpersonen einen deutlichen Mehrwert bringen kann. Die Art der Interaktion soll dafür frei gewählt werden können und in verschiedensten Situationen stattfinden. Um einem freien Test ausreichend standzuhalten, ist es jedoch notwendig, die Implementierung der Kommando Ausführung zu finalisieren und technische Probleme zu beseitigen.

Die Tags, auf denen das Smartphone liegengelassen werden kann und soll, wurden größtenteils nicht richtig erkannt und verwendet. Trotz eines Hinweises zu Beginn des Tests und einer Erklärung nach der falschen Benutzung haben drei von fünf BenutzerInnen den selben Fehler beim NFC-Tag am Nachttisch gemacht. Tags, auf denen das Lesegerät liegengelassen werden soll, müssen daher speziell gekennzeichnet sein und klar von den herkömmlichen Tags zu unterschieden werden zu können. Der Erkennungswert der Grafiken auf den NFC-Tags war deutlich niedriger als jener der Touchscreen-Fernbedienung. Die Icons der Touchscreen-Fernbedienung sollen daher auch auf den NFC-Tags Verwendung finden.

Der Kritikpunkt, dass man durch die Interaktion durch Liegenlassen des Smartphones keine weiteren Tags mehr einlesen könne, wird nicht behandelt. Der Umstand ist zu vernachlässigen, da die Benutzerin die Touchscreen-Fernbedienung stattdessen benützen könnte. Eine weitere Möglichkeit wäre es, die Pause Funktion nur beim Einlesen des NFC-Tags auszuführen und nicht ebenso beim Verlassen. Diese Funktion muss also je nach Gewohnheiten und Vorlieben der BenutzerIn konfiguriert werden.

Durch die Fehlbedienung mit dem NFC-Tag am Nachttisch wäre der Kaffee direkt beim Schlafen gehen gemacht worden. Daher ist auch die Möglichkeit zur kontextsensitiven Konfiguration notwendig, um die aktuelle Uhrzeit bei der Auswahl des Kommandos zu berücksichtigen und Fehler zu vermeiden.

Die zuvor definierten Anforderungen gelten somit als größtenteils erfüllt. Einzig und allein die zweite Anforderung *Die Bedienung mittels NFC soll möglichst wenig Aufmerksamkeit und Konzentration benötigen* ist durch die Inkonsistenz der Grafiken zwischen NFC- und Touchscreen-Fernbedienung gefährdet.

4.6 Zweite Anforderungsphase

Die Anforderungen für die zweite Iteration ergaben sich aus der Analyse der Benutzertests. Im Folgenden werden die aus den Erkenntnissen der ersten Iteration entstandenen Anforderungen beschrieben. Eine detaillierte Behandlung der Themen wird in den folgenden Phasen vorgenommen.

4.6.1 Gestaltung des Benutzertests

Bei den Benutzertests der vorigen Iteration hatten die möglichen Vorteile der zusätzlichen Interaktionsmöglichkeit durch NFC-Tags, keine Chance zur Geltung zu kommen. Die Definition der verschiedenen Aufgaben hatte zwar durchaus seinen Nutzen, aber verhinderte zugleich die Untersuchung der Frage, ob das System eine zusätzliche Erleichterung im Alltag bieten kann. Der Großteil der TeilnehmerInnen der ersten Iteration hat zwar zwischen den Aufgaben die Möglichkeit genutzt, alle Möglichkeiten zur Interaktion auszuprobieren und spielerisch zu erkunden, dabei verfolgten sie jedoch kein persönliches Ziel. Es war also nicht möglich, einen gefühlten Mehrwert festzustellen. Die Benutzertests der zweiten Iteration sollen daher freier gestaltet sein und über einen längeren Zeitraum durchgeführt werden. Die BenutzerIn soll ihre Handlungen selbst entscheiden können und ihre eigenen Ziele verfolgen.

4.6.2 Ausführung von Kommandos

Um einem freien Test zu durchführen zu können, muss die Abarbeitung und Ausführung von Kommandos grundlegend überarbeitet werden. Die mehrmalige, aufeinanderfolgende Ausführung von Aktionen führt in der momentanen Implementierung zu Problemen. Ebenso gibt es zusätzliche Probleme, wenn Kommandos aufgrund eines Netzwerkfehlers oder einer Fehlkonfiguration fehlschlagen. Um die BenutzerIn bestmöglich zu unterstützen, muss die Implementierung weiter finalisiert werden.

4.6.3 NFC Lesebereich

Alle TeilnehmerInnen der Benutzertests hatten anfangs Probleme dabei den Lesebereich des Smartphones korrekt auf den NFC-Tags zu platzieren. Daher sollen Möglichkeiten zur Vereinfachung des Einlesens untersucht werden. Dazu zählen Möglichkeiten zur Gestaltung der Tags sowie die Verwendung anderer Formate.

4.6.4 Kontextsensitive Steuerung

Die Steuerung der Kaffeemaschine soll die aktuelle Uhrzeit beachten und Kaffee nur tagsüber zubereiten. Es soll also möglich sein, das Smartphone nachts vom NFC-Tag zu entfernen (z.B. um einen Anruf entgegen zu nehmen), ohne das Kaffee gemacht wird.

4.6.5 Konsistente Grafiken

Die Icons auf den NFC-Tags sollen durch die der Touchscreen-Fernbedienung ersetzt werden. Die Inkonsistenz fiel während der Benutzertests negativ auf und widerspricht teilweise der Anforderung 'Die Bedienung mittels NFC soll möglichst wenig Aufmerksamkeit und Konzentration benötigen' (siehe Abschnitt 4.2.1). Ebenso soll das Radio Symbol auf der Hauptfernbedienung gegen ein weniger spezifisches Symbol für Musik ausgetauscht werden.

4.6.6 HTTP Probleme

Während der Benutzertests kam es zu Problemen mit häufigen HTTP Anfragen. Sie äußerten sich in extrem hohen Latenzen von bis zu einer Minute nach der Durchführung einer gewissen Anzahl an Anfragen an einen Host. Da der Benutzertest dieser Iteration der BenutzerIn sehr viel Freiheit gibt, ist die Durchführung mit diesem Fehler nicht möglich. Die Probleme mit HTTP Anfragen müssen daher beseitigt werden.

4.6.7 Logging

Um den freien Benutzertest ohne Videoaufnahmen dokumentieren zu können, soll ein Mechanismus zur Protokollierung von Interaktionen implementiert werden. Damit soll die Anzahl der Interaktionen und deren Uhrzeit festgehalten werden.

4.7 Zweite Entwicklungsphase

Die zweite Entwicklungsphase beschäftigte sich damit, die in der ersten Iteration erkannten Probleme zu beseitigen und die Implementierung zu finalisieren. Im Folgenden werden alle Entwicklungsarbeiten der zweiten Iteration detailliert beschrieben.

4.7.1 Ausführung von Kommandos

Die Erweiterungen der ersten Iteration an der Ausführung von Kommandos waren ein wichtiger Schritt, um die Ausführungszeit komplexer Aktivitäten drastisch zu reduzieren. Die Implementierung reichte zwar für die Benutzertests der ersten Iteration aus, es war jedoch nicht sonderlich schwer Fälle zu finden, in denen das System nicht so wie gewünscht reagiert. Der Status eines Geräts ändert sich beispielsweise von *Pending* auf den finalen Status nur, wenn das Kommando erfolgreich ausgeführt wurde. Wenn die Ausführung fehlschlägt, bleibt das Gerät somit im Status *Pending*. Der Status kann zwar über das UI ausgebessert werden, aber dieser Arbeitsschritt ist mühselig. Versucht die BenutzerIn stattdessen einfach das Aktivitäts- oder Geräte-Kommando erneut auszuführen, wird keine Aktion für das Gerät unternommen da die Software den Fehler nicht richtig behandelt. Die Information über den Fehlschlag wird momentan einfach verworfen.

Die Ausführungsschicht hatte zudem noch viele Synchronisationsprobleme, die während eines freien Tests sicherlich Probleme hervorrufen würden. Ein gutes Beispiel dafür ist die Ausführung von Kommandos zum Starten und Stoppen von Aktivitäten. In der ersten Implementierung

blockieren diese Kommandos den *Thread* der die *NoDevice-Queue* abarbeitet nicht. Das Kommando entscheidet lediglich welche Geräte ein- oder ausgeschaltet werden müssen und fügt dementsprechend Kommandos in andere *Queues* ein. Dadurch kommt es zu unerwartetem Verhalten, wenn man rasch zwischen Aktivitäten wechselt und weil man sich beispielsweise überraschend umentscheidet.

Als Beispiel dafür wird eine Aktivität mit einem einzigen Gerät angenommen. Führt man den Start der Aktivität durch, wird das Gerät als ausgeschaltet erkannt und anschließend eingeschaltet. Dadurch befindet sich das Gerät kurze Zeit im Zustand *Pending-On*. Wenn man nun die Aktivität stoppt, wird das Gerät ausgeschaltet, sobald es den *Pending-On* Status verlässt und somit als vollkommen eingeschaltet gilt. Versucht man in der Zwischenzeit die Aktivität wieder zu starten, wird das Gerät jedoch immer noch als *Pending-On* erkannt und es wird keine weitere Aktion unternommen. Die bevorstehende Zustandsänderung in den ausgeschalteten Status wird also nicht beachtet. Nach Abarbeitung der Kommandos bleibt das Gerät dadurch fälschlicherweise ausgeschaltet.

Dieses minimalistische Szenario scheint recht unwahrscheinlich, der Fehler tritt jedoch natürlich auch bei komplexeren, realistischeren Szenarien auf. Ein besseres Beispiel dafür wäre das folgende Szenario:

1. Starte Aktivität 'Film'
2. Beende Aktivität 'Film'
3. Starte Aktivität 'Spiel'

Die Geräte, welche in beiden Aktivitäten enthalten sind und sich bei der Durchführung von Schritt 3 immer noch im Status *Pending-On* befinden, werden nicht eingeschaltet. Dieses Szenario ist weitaus realistischer, da es in der Praxis durchaus vorkommen wird, dass sich die BenutzerIn bei der Wahl der Aktivität spontan umentscheidet.

Zusätzlich zur Behebung der Synchronisationsprobleme durch Lock-Mechanismen wurde ein *QueueManager* implementiert. Er übernimmt das tatsächliche Einfügen der Kommandos in die *Queues*. Sobald eine *Queue* geleert wurde, wird der abarbeitende *Thread* terminiert, um die limitierten Ressourcen des Geräts zu schonen. Der *QueueManager* erkennt dies und startet gegebenenfalls einen neuen *Thread* sobald neue Kommandos zur Abarbeitung vorhanden sind.

4.7.2 NFC Lesebereich

Zur möglichen Erleichterung des Einlesens von NFC-Tags wurde zuerst eine Recherche im Internet durchgeführt. Sie ergab, dass die Verwendung größerer Tags keine Verbesserung bringen würde und die momentan verwendete Größe optimal sei [142]. Um dies zu überprüfen, wurden NFC-Tags im Scheckkarten Format (85,60 mm x 53,98 mm) mit den runden 35mm Tags verglichen. Die NFC Antenne des Smartphones befindet sich im Akku des Geräts. Es wird dabei angenommen, dass sie darin die größtmögliche Fläche einnimmt, da sie an allen vier Kanten das

selbe Verhalten zeigte. Somit ist sie nur wenig kleiner als der Akku selbst ist.

Abbildung 4.24 zeigt einen Vergleich des Lesebereichs der zwei unterschiedlichen Formate. Bei der abgebildeten Platzierung gelten beim Scheckkartenformat die durchgezogenen grünen Linien als Lesebereich und beim runden Aufkleber die blauen. Wie daraus ersichtlich ist, muss auch bei größeren Tags ein Großteil der Antenne innerhalb der Antenne des Lesegeräts platziert werden. Schiebt man den rechts überstehenden Rand des Aufklebers innerhalb den Antennenbereich des Akkus, gilt die durchgezogene grüne Linie und der Tag erlaubt eine größere Toleranz in vertikaler Richtung. Die gestrichelten Linien auf der Scheckkarte zeigen das Verhalten, welches eintritt, wenn man den Tag um 90 Grad drehen würde.

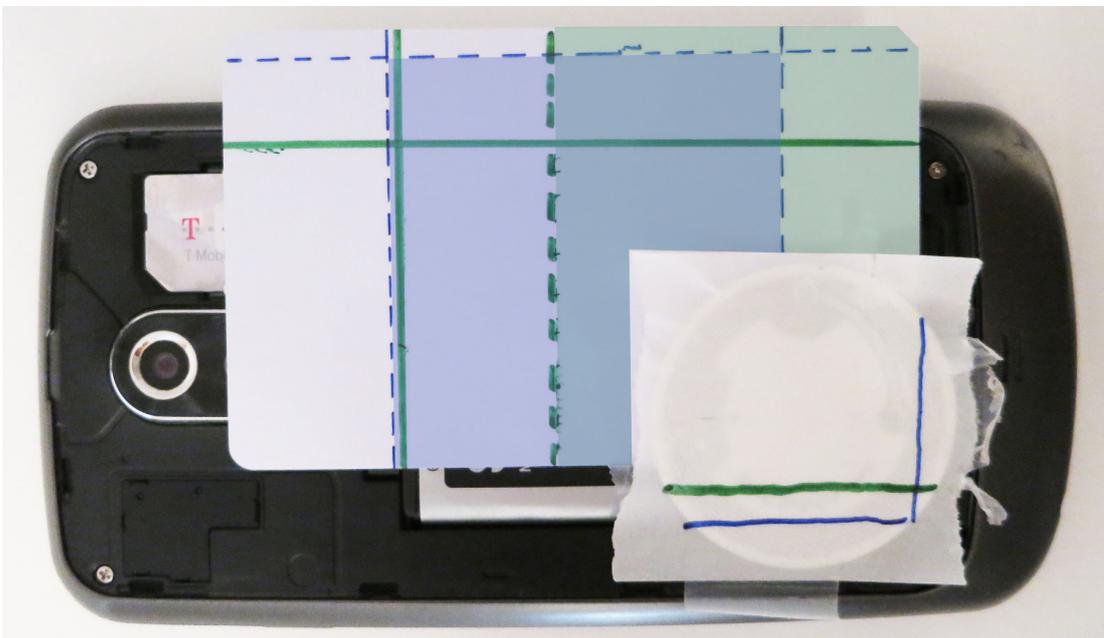


Abbildung 4.25: NFC-Tag Lesebereich Vergleich

Eine Möglichkeit zur Gestaltung wäre es, die Verpackungen der Tags in der selben Form und Größe wie das Gerät selbst zu erstellen. Diesen Weg hat Google eingeschlagen, um die Platzierung ihrer Geräte bei Demos im Rahmen der Google I/O zu erleichtern (siehe Abbildung 4.26). Beim linken Smartphone handelt es sich um ein platziertes Gerät. Das rechte Smartphone ist bloß ein Aufdruck, um die Positionierung eines zweiten Gerätes zu erleichtern. Die zwei Aufdrucke enthalten jedoch keine NFC-Tags, sondern erleichtern die Demonstration von *Peer-To-Peer* Anwendungen mit NFC-fähigen Android Smartphones.

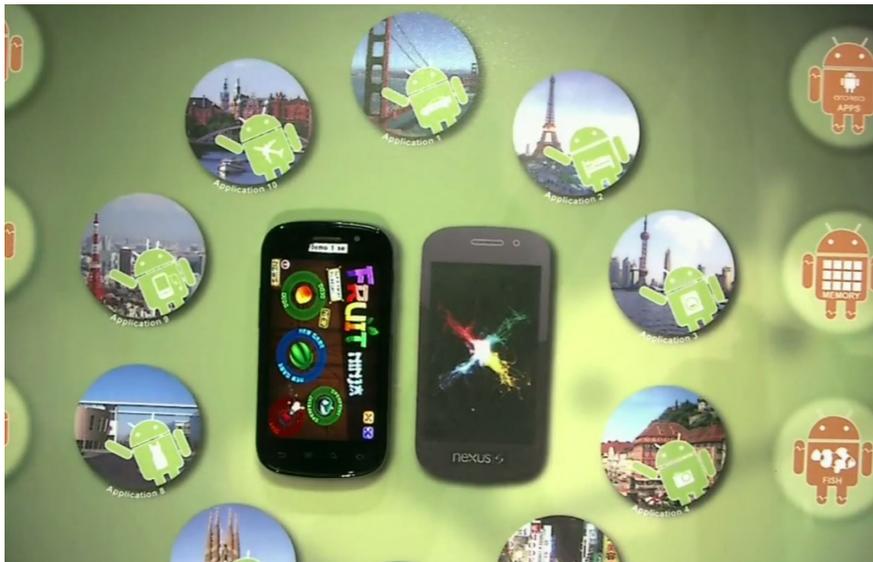


Abbildung 4.26: Google IO NFC Demo [143]

Die Darstellung in Größe und Form des verwendeten Smartphones stellt jedoch auch keine universelle Lösung des Problems dar. Die verschiedenen Formfaktoren und unterschiedlich platzierten Antennen würden bewirken, dass die zu vermutende Ausrichtung nur bei der Verwendung bestimmter Geräte richtig ist. Würde man nur ein einziges Gerät zur Fernsteuerung per NFC verwenden, würde ein Wechsel zu einem neuen Gerät den Vorteil möglicherweise wieder zunichte machen.



Abbildung 4.27: NFC Antennen der zwei Smartphones

Abbildung 4.27 zeigt die Positionierung der Antennen im Testgerät (Galaxy Nexus) und im Samsung Galaxy Note 2. Der Größenunterschied ist deutlich und die Positionierung der Antennen ist sehr unterschiedlich. Im Gegensatz zum Galaxy Nexus befindet sich die NFC Antenne des wesentlich größeren Geräts im Akkudeckel. Die grüne Umrandung zeigt die Antenne und die Verbindungskontakte.

Der Lernaufwand für das Finden des Lesebereichs kann also nicht vermieden werden. Die einzige Möglichkeit zur Erleichterung der Interaktion wäre die Darstellung in Form und Größe des verwendeten Geräts, die aber nur für einen Teil der Geräte universell und richtig wäre. Deshalb wurde von einer grundlegenden Formänderung und Größenänderung der Tags und deren Verpackung wieder abgesehen.

4.7.3 Kontextsensitive Steuerung

Die kontextsensitive Steuerung wurde mit Hilfe der Android Applikation *Tasker* realisiert. Durch das Ausnützen der Modularität von Android konnte der Aufwand für die Implementierung stark reduziert werden. *Tasker* ermöglicht die Ausführung unzähliger Aktionen am Smartphone und die Reaktion auf verschiedenste Ereignisse. Anwendungen von Dritthersteller, welche die *locale-plugin* Architektur implementieren, können als Quelle für Aktionen dienen. *Tasks* (deutsch: Aufgaben) können außerdem von fremden Applikationen zur Ausführung gebracht werden. *Tasker* bietet einen grafischen Editor für Benutzeroberflächen und erlaubt die Erstellung von Skripten, um beispielsweise weitere Bedingungen zu definieren.

Um die Zusammenarbeit mit der *Tasker* Applikation zu ermöglichen und als *locale-plugin* erkannt zu werden, musste reYmote entsprechend adaptiert werden. reYmote kann sich dadurch die bereits ausgereiften Möglichkeiten zum Definieren von Bedingungen zunutze machen. Wenn das Smartphone vom NFC-Tag am Nachttisch entfernt wird, löst ein Kommando in reYmote einen *Tasker Task* aus. Der *Task* überprüft anschließend die Gültigkeit der Bedingung gegenüber der Uhrzeit. Sind die Bedingungen erfüllt, wird in weiterer Folge vom *Tasker Task* das reYmote Kommando ausgelöst, um Kaffee zuzubereiten.

Abbildung 4.28 zeigt die kontextsensitive Konfiguration. Da *Tasker* mehrfache Bedingungen in If-Ausdrücken nicht unterstützt, wurden zwei einzelne Elemente geschachtelt. Durch diese Konfiguration wird Kaffee also nur zwischen 5 Uhr früh und 16 Uhr zubereitet.

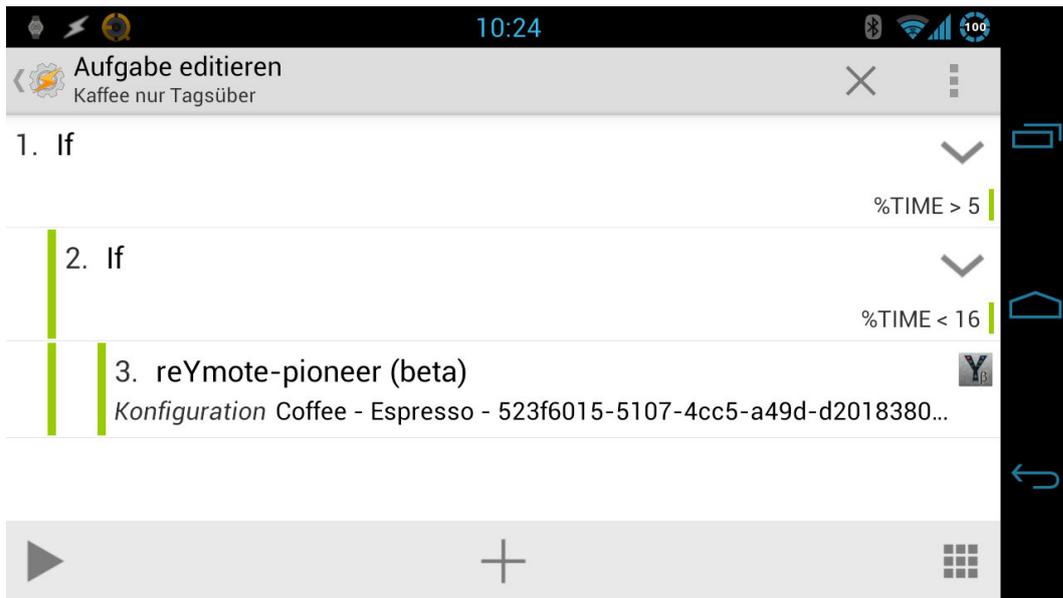


Abbildung 4.28: Tasker Task Konfiguration

Um *Tasker Tasks* durch reYmote ausführen zu können, wurde ein weiteres Kommando als Konfigurationselement erstellt. Die Auswahl des auszuführenden *Tasks* geschieht über einen Dialog der von *Tasker* erzeugt wird (siehe Abbildung 4.29).

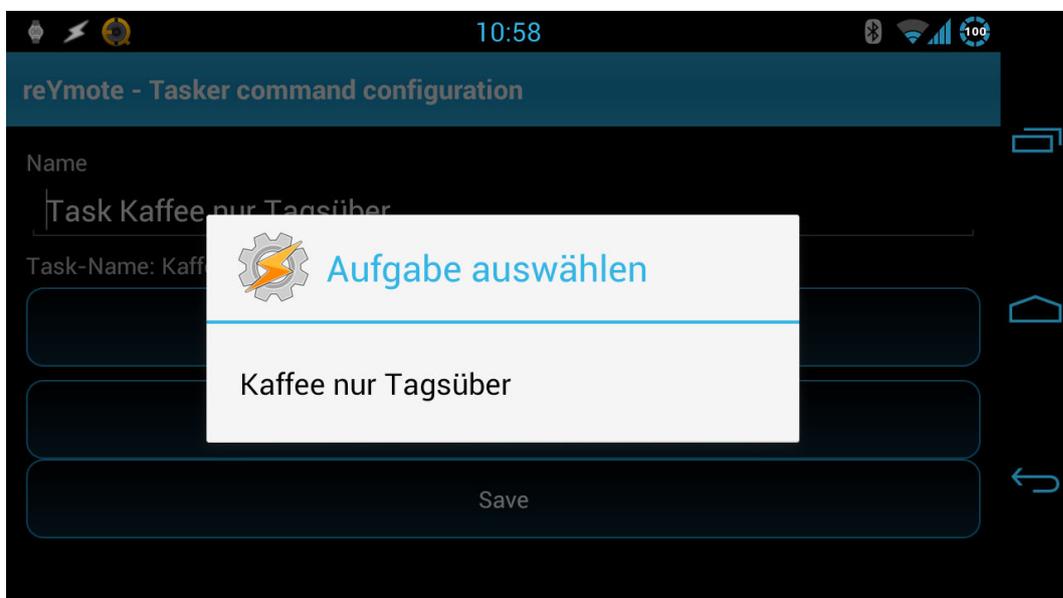


Abbildung 4.29: reYmote Tasker-Kommando Konfiguration

4.7.4 Konsistente Grafiken

Die Verpackungen der NFC-Tags wurden neu entworfen und in der selben Art hergestellt wie in der ersten Iteration. Dies geschah um die Konsistenz der Darstellung zwischen den Fernbedienungsvarianten zu erhöhen und den höheren Erkennungswert der Icons der Touchscreen-Fernbedienung zu nützen. Für die Fenstersteuerung wurden neue Grafiken entworfen, da dies in der Touchscreen-Fernbedienung bloß Pfeile an der jeweiligen Position im Grundriss waren.

Die neuen Ausdrücke der Tags sind quadratisch mit einer Seitenlänge von ungefähr 650 mm. Insgesamt wurden in dieser Phase 25 NFC-Tags mit den dazugehörigen Grafiken versehen und laminiert. Dabei wurden die meisten der Tags aus den alten Verpackungen wiederverwendet.

Die NFC-Tags wurden auch farblich eindeutig gekennzeichnet, um Problemen aus der ersten Iteration vorzubeugen. Tags, die eine Aktivität starten haben einen grünen Rahmen. Tags die Aktivitäten beenden haben einen roten Rahmen. Tags die einfach nur Aktionen auslösen, haben einen weißen Rahmen. Die Tags, auf denen das Telefon liegen gelassen werden soll wurden in der ersten Iteration nicht intuitiv erkannt. Daher haben Tags, auf denen das Lesegerät liegen gelassen werden soll, einen blauen Rahmen. Abbildung 4.30 zeigt die neuen Tags und das Smartphone.

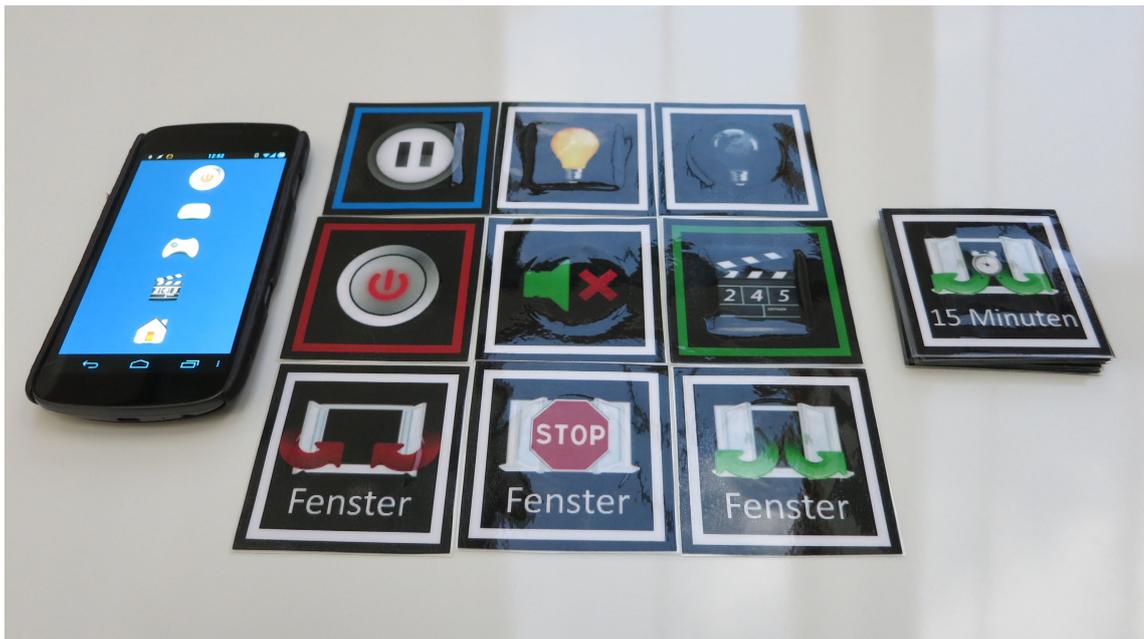


Abbildung 4.30: NFC-Tag Sammlung für die zweite Iteration

Das Radiosymbol auf der Hauptfernbedienung wurde nicht verändert. Durch Auswahl des Radiosymbols wird automatisch die Aktivität gestartet und die Radiofernbedienung angezeigt. Die Webradiosender und das extra Radiosymbol wurden von der Benutzeroberfläche entfernt. Abbildung 4.31 zeigt den Hauptschirm und die neue Radiofernbedienung.

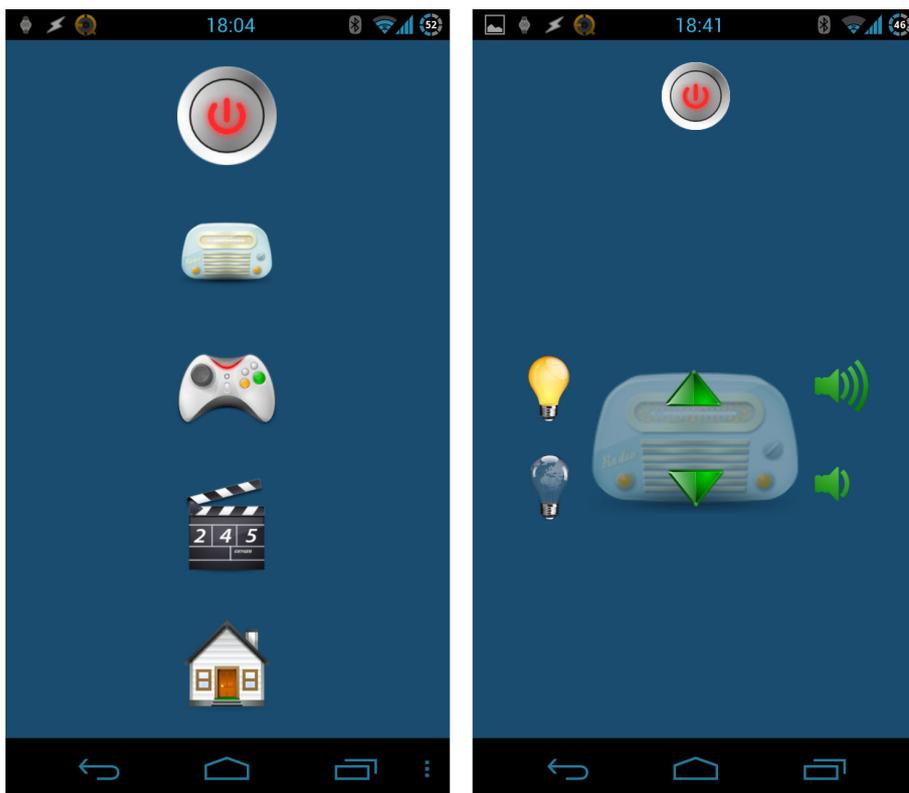


Abbildung 4.31: Hauptschirm und neue Radiofernbedienung

4.7.5 HTTP Probleme

Die Performance Probleme mit HTTP Anfragen hatte ihren Ursprung in der Umstellung der Ausführung auf Nebenläufigkeit und der dadurch notwendigen Konfigurationsänderungen des Apache-Http-Client. Der *ThreadSafeClientConnManager* verlangt, dass Antworten auf Anfragen konsumiert werden, um Netzwerkressourcen wieder freizugeben und neue Verbindungen herstellen zu können. Durch das Lösen des Problems konnte die Performance der HTTP Anfrage wieder auf ein zufriedenstellendes Niveau gebracht werden.

4.7.6 Logging

In dieser Iteration wird der Benutzertest nicht per Video dokumentiert. Die geplante Gestaltung und Länge des Tests würde möglicherweise häufige Ortswechsel der Ausrüstung benötigen und einen enormen Arbeitsaufwand beim Sichten des Videomaterials bedeuten. Um die Interaktionen der BenutzerIn trotzdem nachvollziehen zu können, wurde der Logging-Mechanismus von reYmote erweitert. Daraus ist ersichtlich, wann die BenutzerIn welche Aktionen durchgeführt hat.

4.8 Zweite Testphase

Die Testphase beschreibt die neue Vorgehensweise und Zielsetzung für die Benutzertests. Die Durchführung der Tests gehört ebenfalls zu dieser Phase. Die Dokumentation und Analyse wird in der darauf folgenden Evaluierungsphase präsentiert.

4.8.1 Ziele

Diese Testphase soll weiteren Aufschluss über die Frage geben, ob die NFC-Fernbedienung für die BenutzerIn eine Erleichterung im Alltag darstellt und zusätzlichen Komfort bei der Bedienung bietet. Beobachtungen, Gespräche und die Analyse der gesammelten Daten sollen anschließend nähere Informationen über den von der BenutzerIn subjektiv wahrgenommenen Nutzen liefern. Die Ziele gelten als erfüllt, wenn die BenutzerIn aus freiem Willen NFC-Tags einsetzt, um persönliche Ziele zu erreichen und ihnen einen Nutzen zuspricht.

4.8.2 Testpersonen

Für diesen Benutzertest wurde vorerst nur eine einzige Testperson ausgewählt. Sie muss zur Kategorie der primären Persona gehören, denn die Zweifel an der Sinnhaftigkeit der NFC Benutzerschnittstelle stammen von dort. Diese Kategorie repräsentiert außerdem die größere Masse an Benutzern und ist laut Definition ihrer Ziele schwerer zu befriedigen. Sollte in der ersten Durchführung kein Nutzen festzustellen sein, wird der Test mit einer zweiten TeilnehmerIn erneut absolviert.

4.8.3 Vorgehensweise

Die BenutzerIn soll die Testumgebung einen Tag lang frei nutzen können. Die Möglichkeiten der Fernbedienung sollen in einer gemütlichen und ungezwungenen Atmosphäre gemeinsam mit dem Testleiter genützt und erweitert werden. Es handelt sich also um eine strukturierte und aktiv teilnehmende Beobachtung. Dabei soll beobachtet werden, ob die BenutzerIn die Tags zur Erfüllung persönlicher Bedürfnisse einsetzt. Solche beobachteten Ereignisse werden anschließend in Gesprächen näher erkundet, um die Gedanken des BenutzerIn diesbezüglich zu erfahren.

Der Testleiter soll den Verlauf schriftlich dokumentieren. Die NFC-Tags werden vorerst entfernt und stehen der BenutzerIn frei zur Verfügung. Neue Tags werden vom Testleiter nach Bedarf konfiguriert. Dazu wird die BenutzerIn gebeten, Wünsche betreffend der Konfiguration von NFC-Tags zu äußern. Vor Beginn des Tests wird der TeilnehmerIn noch eine kurze Erklärung über die verschiedenen Steuerungsmöglichkeiten und die steuerbaren Geräte gegeben.

4.9 Zweite Evaluierungsphase

Die zweite Evaluierungsphase bietet die Dokumentation des letzten Benutzertests. Anschließend werden die Analyse und die daraus gewonnenen Erkenntnisse präsentiert.

4.9.1 Benutzertest

Der Benutzertest wurde an einem warmen und sonnigen Nachmittag begonnen. Die Aktivitäten zogen sich bis in die späten Abendstunden und die Steuerung konnte in vielen Situationen getestet werden. Insgesamt wurden knapp über 100 Interaktionen durchgeführt. Ein Großteil davon wurde klarerweise dabei über die Touchscreen-Fernbedienung ausgelöst, da Detailfunktionen wie Lautstärke und Navigation nur so bedient werden. Dennoch wurde die NFC-Fernbedienung insgesamt 29 mal benutzt. Auch bei diesem Test klappte die Interaktion mit den NFC-Tags wie erwartet nicht auf Anhieb reibungslos. Nach ein paar Versuchen war der Lesebereich jedoch gefunden und die weiteren Interaktionen waren erfolgreich.

Zu Beginn wurde im Wohnzimmer bei einem Glas Wasser, ein gemütliches Gespräch über den bisherigen Verlauf der Arbeit geführt. Die NFC-Tags gefielen der BenutzerIn und sie machte sich mit den nicht zugewiesenen vertraut. Anschließend erkundete sie die Touchscreen-Fernbedienung. Nachdem viele Aktionen getestet wurden, probierte sie den NFC-Tag zum Schließen der Fenster. Sie hatte sie zuvor per Touchscreen geöffnet und wollte sie nun doch wieder schließen. Anschließend platzierte sie die Tags für die Fenster an der vertikalen Seite eines Balkens zwischen den Dachfenstern (siehe Abbildung 4.32). Im Gespräch merkte sie an, dass sie somit vom Tisch und von der Couch einfach zu erreichen sind.



Abbildung 4.32: Tags zur Fernbedienung der Fenster

Die BenutzerIn versuchte auch eifrig, Verwendung für die Interaktion mit Liegenlassen zu finden. Folgende Steuerungen wurden hierbei konfiguriert und getestet:

- Fenster geöffnet lassen bis Lesegerät entfernt wird
- Radio hören bis Lesegerät entfernt wird

Im Gespräch nannte sie eine Funktion zum Duschen, die sie im Badezimmer platzieren würde, wenn es ihre Wohnung wäre. Solange das Lesegerät auf dem Tag liegt, bleibt das Licht im Badezimmer eingeschaltet. Nach dem Duschen entfernt man das Gerät von Tag und es wird automatisch für 10 Minuten gelüftet. Es sollte jedoch möglich sein, die Dauer wesentlich zu verkürzen, wenn es Winter ist oder es im Freien sehr kalt ist. Sie merkte außerdem an, dass man das Telefon beim Duschen klarerweise nicht verwenden kann und es somit auf dem NFC-Tag gut aufgehoben ist.

Der Nachmittag wurde mit einer Partie Schach und anschließendem Entspannen bei Musik verbracht. Bei der Schachpartie kam der NFC-Tag zum Lüften per Liegenlassen regelmäßig zum Einsatz. Während dem Musik hören, wünschte sich die BenutzerIn einen Tag zum Wechseln des Radiokanals. Die BenutzerIn entspannte weiter auf dem Couch und der Testleiter bereitete in der Zwischenzeit den NFC-Tag vor. Wenige Minuten später wurde der Tag an einer geeigneten Position am Rand des Couchtischs angebracht. Die BenutzerIn konnte dadurch mit einem einfachen Handgriff direkt aus ihrer Liegeposition den Kanal wechseln. Sie empfand die Interaktion als sehr angenehm und benötigte auffallend wenig Aufmerksamkeit zur Durchführung. Abbildung 4.33 zeigt die Anordnung der Tags.

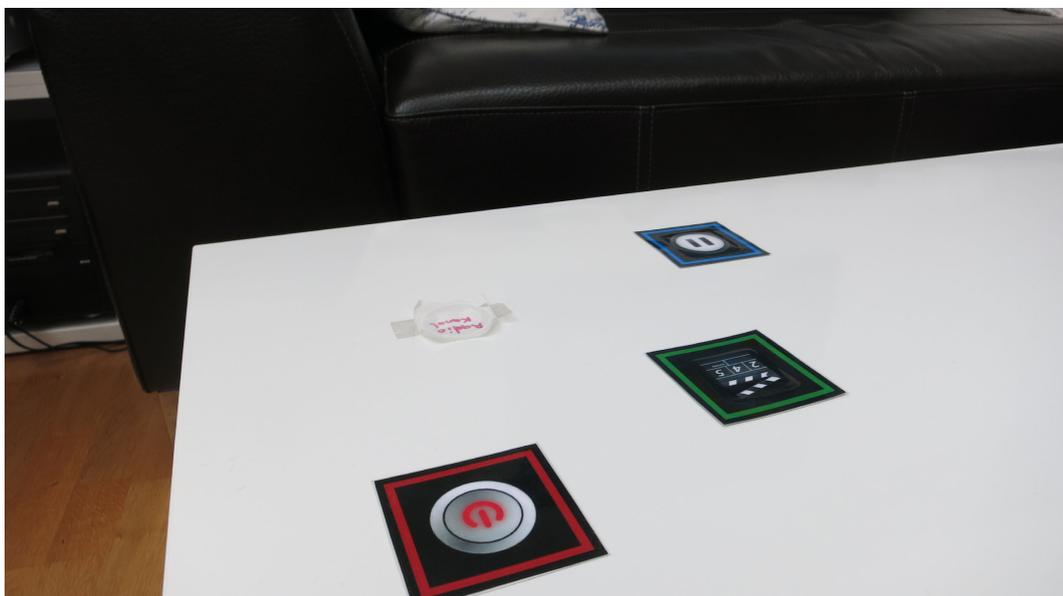


Abbildung 4.33: Tag zum Wechsel des Radiosenders

Abends wurde ein Film angeschaut. Einen besonderen Vorteil bot die NFC-Fernbedienung als die BenutzerIn angerufen wurde. Durch einen einfachen Handgriff pausierte sie die Wiedergabe, indem sie das Gerät auf dem NFC-Tag liegen lies. Die BenutzerIn erwähnte erneut, dass sie die Steuerung als sehr angenehm empfand. Sie merkte jedoch auch an, dass sie die Funktion nicht benutzt hätte, wenn jemand auf der Fernbedienung angerufen hätte. Selbst eine Konfiguration mit einer einfachen Interaktion ohne Liegenlassen hätte sie persönlich in diesem Fall als weniger angenehm empfunden als die Steuerung mit einem separaten Gerät.

Die BenutzerIn meinte abschließend, dass sich die Interaktion mit den NFC-Tags auf Grund ihrer Größe und örtlichen Gebundenheit ähnlich zu dem Betätigen eines Lichtschalters anfühlte.

4.9.2 Analyse

Der Benutzertest der zweiten Iteration zeigte ebenfalls ein durchwegs positives Bild bezüglich Benutzerakzeptanz und der Frage nach dem Nutzen der NFC-Fernbedienung. Die anfänglichen Probleme mit dem Einlesen der NFC-Tags und die notwendige Eingewöhnungsphase waren erwartungsgemäß auch in diesem Test eindeutig vorhanden.

Die neuen Designs der NFC-Tags gefielen der BenutzerIn. Sie fand innerhalb kurzer Zeit Situationen und Orte, an denen ihr die Steuerung per NFC einen zusätzlichen Komfort bot. Sie nannte außerdem mögliche Anwendungen, die sie in ihrem eigenen Heim als äußerst sinnvoll erachten würde.

Vor allem während Gesprächen oder anderen Aktivitäten, wurde die Interaktion von der BenutzerIn als besonders angenehm empfunden. Die Beobachtungen und Interviews mit der BenutzerIn ergaben, dass die Interaktion mit den NFC-Tags oft weniger Aufmerksamkeit benötigt als die anderen Methoden. Dazu müssen sie jedoch geeignet platziert werden. In bestimmten Situationen, wie zum Beispiel während eines Anrufs, war der Unterschied besonders deutlich zu erkennen. Die Ausführung der Interaktion wirkte dabei besonders einfach durchzuführen, ohne dabei vom Gespräch abzulenken.

Wie das Beispiel des Anrufs während eines Films ebenfalls zeigt, kann es von Vorteil sein, ein dediziertes Gerät als Fernbedienung zu benutzen. Mittlerweile werden auch immer mehr günstige Android Smartphones mit NFC Schnittstelle ausgestattet. Der Anschaffungspreis eines dedizierten Geräts liegt also weit unter dem einer High-End Universalfernbedienung. Dafür wäre es jedoch wichtig, die Applikation zu erweitern, um mehrere Geräte zu unterstützen und einen transparenten Wechseln zwischen den Geräten durchführen zu können.

Die BenutzerIn fand auch Verwendungen für die Interaktion mit Liegenlassen. In vielen Situationen kann auch hierbei ein dediziertes Gerät von Vorteil sein. Einige Situationen und Steuerungen würden sich jedoch möglicherweise auch mit weiterer Kontextsensitivität der Steuerung absichern lassen. Beispielsweise könnte das Ereignis beim Verlassen des NFC-Tags einfach unbeachtet bleiben, wenn ein Anruf eingeht. Sobald man das Telefonat beendet hat, könnte man ebenfalls, ohne eine weitere Interaktion auszulösen, das Gerät wieder auf dem Tag platzieren.

Die BenutzerIn verglich die Interaktion mit dem Betätigen eines Lichtschalters. Die Steuerungen lassen sich natürlich auch mit Funklichtschaltern realisieren. Preislich übersteigt diese Lösung jedoch die Kosten der NFC-Tags bei weitem.

Analyse

Dieses Kapitel beschäftigt sich mit der abschließenden Analyse des kompletten Forschungsprozesses sowie dessen Ergebnisse. Der darauf folgende Abschnitt gibt einen Ausblick über weitere Entwicklungsmöglichkeiten zur Verbesserung des Systems und verwendeter Technologien sowie der Interaktion per NFC.

5.1 Entwicklung

Im Laufe dieser Arbeit wurde mit einem speziellen Fokus auf die Anwendung benutzerzentrierter Methoden, durch einen iterativen Prozess ein System entwickelt, das den Komfort bei der Bedienung von Haushaltsgeräten erhöhen und den dazu notwendigen Aufwand verringern kann. Die technische Grundlage dieser Arbeit bildete die Android Applikation reYmote, die vom Autor dieser Arbeit erstellt wurde und weiterentwickelt wird.

Zu Beginn dieser Arbeit stand die Definition von Anforderungen an die geplante Interaktion mit dem Smartphone und NFC-Tags. Um die Fragestellung bestmöglich untersuchen zu können, war es besonders wichtig, eine möglichst hohe Qualität der Interaktion anzustreben.

Um in frühen Entwicklungsphasen keine Designfehler zu begehen, wurde mit Personas gearbeitet. Bei ihrer Erstellung halfen die Erkenntnisse, die man aus dem *Technology Adoption Lifecycle* von [90] und [60] gewinnen kann. Zur Veranschaulichung dient die Lückenhafte Glockenkurve (siehe Abbildung 5.1) Daraus ergab sich die Unterscheidung der Benutzergruppe der ErstanwenderInnen und die der normalen BenutzerInnen. Die ErstanwenderInnen interessieren sich für die Art der betrachteten Entwicklung speziell. Sie sind leichter zu begeistern und nehmen Fehler in Kauf. Normale BenutzerInnen nutzen das System nur aufgrund tatsächlicher Vorteile. Da die ErstanwenderInnen eine kleine Gruppe bilden und andere, teilweise niedrigere Ansprüche stellen, wurde die Gruppe der normalen BenutzerInnen priorisiert. Mit Hilfe der identifizierten Benutzergruppen wurden schließlich deren persönliche Ziele definiert.

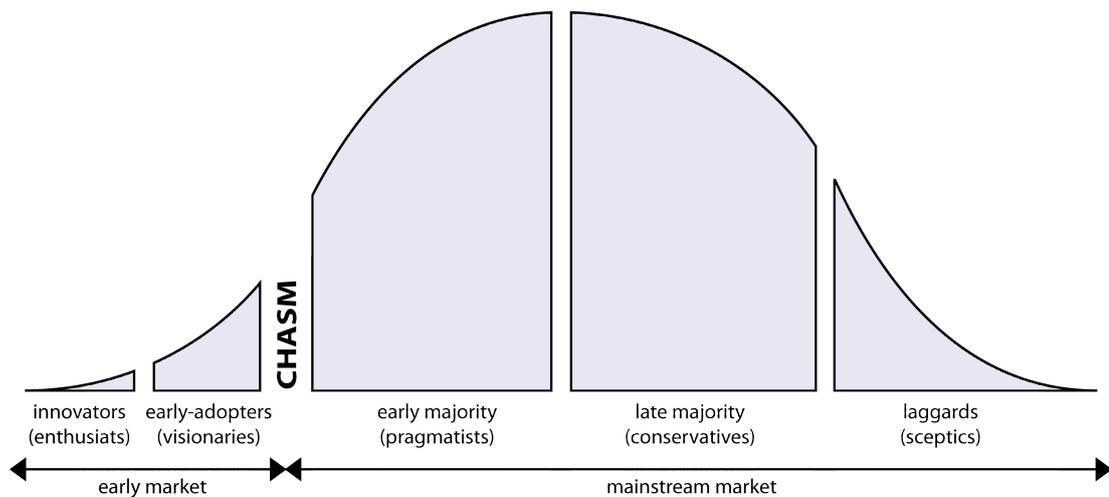


Abbildung 5.1: Technology Adoption Lifecycle (abgeändert, reproduziert nach [136])

Aus den Benutzergruppen und deren Ziele wurden anschließend Personas erstellt. Abbildung 5.2 zeigt die verwendeten Fotos der Personas. Die primäre Persona gehört zur Benutzergruppe der ErstanwenderInnen und die sekundäre zu den normalen BenutzerInnen. Um eine deutliche Abgrenzung für diese Arbeit zu treffen, wurde eine negative Persona eingeführt, wie von [59] empfohlen wird. Ihr Ziel ist die Minimierung des Konfigurationsaufwands und die Gewinnmaximierung. Bestimmte Funktionalitäten wie die Möglichkeit zur Fernwartung und eine besonders simple Konfiguration werden von ihr gefordert. Diese Fragen sollten in dieser Arbeit jedoch nicht behandelt werden.



Abbildung 5.2: Fotos der verwendeten Personas

Anschließend wurden Szenarios erstellt, um Entwicklungsziele zu definieren. Die Personas wurden, wie von [58] vorgeschlagen wird, als Akteure in den Szenarios eingesetzt, um ein besseres Verständnis für die BenutzerInnen im Entwicklungsprozess zu bekommen. Die Szenarios und die Personas waren eine große Hilfe, dabei den Fokus zu wahren und Ziele für die Entwicklungen zu definieren.

In der ersten Entwicklungsphase waren zahlreiche Entwicklungsarbeiten notwendig, um eine möglichst vollständige Testumgebung und Fernbedienung bieten zu können. Dies wird von [46] zur Entwicklung eines solchen Systems empfohlen. Hauptbestandteil der Entwicklungsaktivitäten waren die notwendigen Weiterentwicklungen von reYmote, um eine aktivitätsbasierte Fernsteuerung bieten zu können. Abbildung 5.3 zeigt das Datenmodell nach der ersten Iteration. Neben der Installation von zahlreichen Z-Wave Lichtschaltern wurden auch eigene Lösungen entwickelt, um die elektronischen Fenster in der Testumgebung und eine Kaffeemaschine über das Smartphone steuern zu können. Um die für die Interaktion definierten Anforderungen möglichst gut umsetzen zu können, musste das verwendete Android Testgerät zusätzlich modifiziert werden. Darauf folgte die Erstellung der Konfiguration für die komplette Testumgebung und die Erstellung von Benutzerschnittstellen.

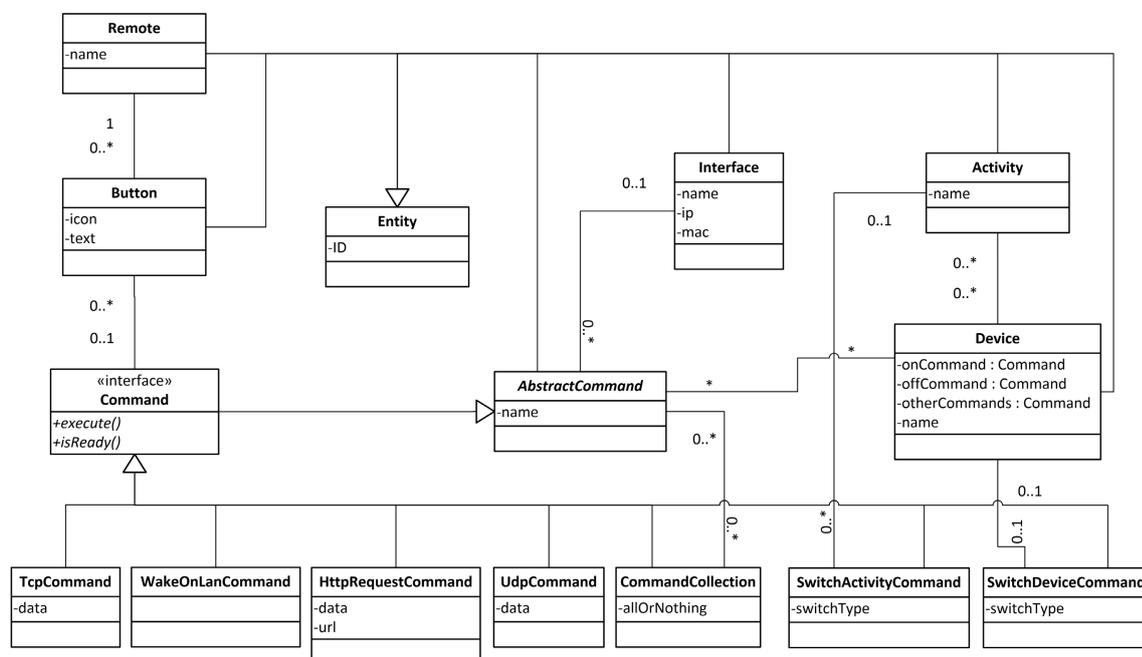


Abbildung 5.3: Datenmodell nach der ersten Iteration

Die eingesetzten NFC-Tags wurden hinter einem Ausdruck laminiert. So lassen sie sich leicht in die Umgebung integrieren. Die optische Gestaltung der NFC-Tags wurde in der zweiten Iteration jedoch grundlegend geändert und verbessert. Die hergestellten NFC-Tags wurden schließlich in der Testumgebung an Orten angebracht, an denen die Personas bestimmte Aktionen durchführen würden. Abbildung 5.4 zeigt ein paar der Tags der ersten Iteration.



(a) Couchtisch

(b) Bett

Abbildung 5.4: Ein paar Tags der ersten Iteration

Anschließend wurden Benutzertests mit fünf Benutzern durchgeführt, um den Nutzen des Systems zu erproben. Abbildung 5.5 zeigt ein paar Bilder von den Benutzertests. Die TeilnehmerInnen mussten die vorher definierten Szenarios in leicht abgeänderter Form durchspielen. Dabei wurden auch die Usability Metriken SUS und MUSiC eingesetzt. Die Benutzertests zeigten ein positives Gesamtbild und boten zahlreiche neue Erkenntnisse für weiterführende Entwicklungen.



Abbildung 5.5: Bilder der ersten Benutzertests

Da während der Tests die Frage nach der Notwendigkeit und der Sinnhaftigkeit einer solchen zusätzlichen Schnittstelle aufkam, wurde der Fokus des nächsten Benutzertests auf diese Fragestellung gelegt. In weiterer Folge musste jedoch zuerst die Implementierungen aus der ersten Iteration finalisiert werden, um einen freien Benutzertest zu erlauben.

Der Fokus in der Gestaltung der NFC-Tags der ersten Iteration lag eher auf einer unauffälligen, optisch ansprechenden Gestaltung. Dabei kam jedoch anderen wichtigen Aspekten weniger Beachtung zu. Dies hatte einen niedrigeren Erkennungswert zur Folge. Die Verwendung unterschiedlicher Grafiken auf der Touchscreen- und der NFC-Fernbedienung bot eine zusätzliche Quelle für Verwirrungen. Außerdem waren die Tags, auf denen das Smartphone liegengelassen werden sollte, von den anderen durch optische Merkmale nicht klar zu unterscheiden. Der Fokus der Gestaltung für die zweite Iteration lag daher auf der Herstellung von Konsistenz zwischen den Varianten sowie der farblichen Kennzeichnung der verschiedenen Kategorien. Abbildung 5.6 zeigt einige der Tags der zweiten Iteration.



Abbildung 5.6: Einige Tags der zweiten Iteration

Die BenutzerInnen hatten ausnahmslos alle anfängliche Probleme beim Einlesen der NFC-Tags. Sie konnten den Lesebereich trotz initialer Aufklärung nicht auf Anhieb finden. In [15] wurde das selbe Phänomen beobachtet. Deshalb wurden in der zweiten Iteration Möglichkeiten zur Erleichterung untersucht. Diese Untersuchungen ergaben jedoch, dass es nicht möglich ist, durch die Verwendung anderer NFC-Tags oder durch Mittel der Gestaltung eine universelle Lösung zu schaffen. Abbildung 5.7 zeigt ein paar Bilder der Untersuchungen.

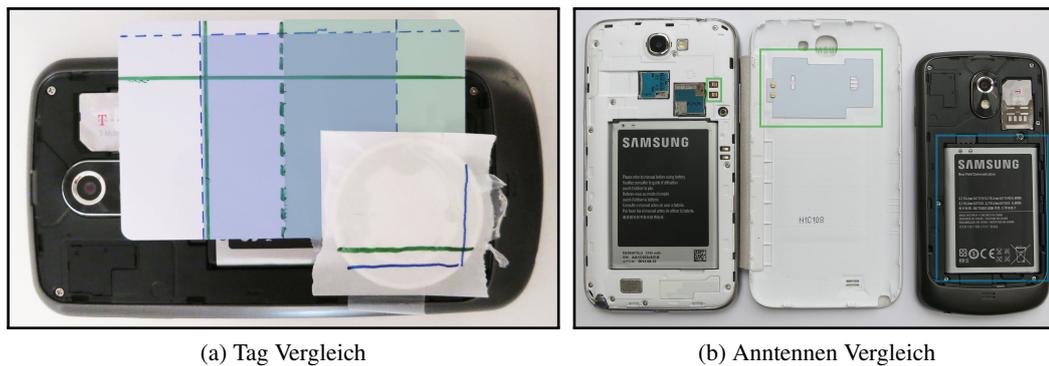


Abbildung 5.7: NFC Lesebereich Untersuchung

Um die BenutzerInnen noch besser unterstützen zu können, wurde anschließend eine Möglichkeit zur kontextsensitiven Steuerung implementiert. Dazu wurde das Prinzip der Modularität durch Android *Intents* ausgenutzt. Es wurden Schnittstellen implementiert, um eine Zusammenarbeit mit der Applikation *Tasker* zu erlauben und dadurch Bedingungen für Aktionen definieren zu können.

Das starre Konstrukt an zu lösenden Aufgaben in Kombination mit Beobachtungen und wenig strukturierten Interviews ergab eine gute Mischung, um erste Fragen zu klären. Zur Beantwortung der Frage nach der Sinnhaftigkeit, dem eigentlich Nutzen und einer möglichen Erleichterung im Alltag, wurden in der zweiten Iteration freiere Benutzertest eingesetzt.

Bei den Tests der zweiten Iteration wurden aktiv teilnehmende Beobachtungen durchgeführt. Vorerst war die Verwendung von passiv teilnehmender Beobachtung in Kombination mit *Diary Studies* geplant, dies wurde jedoch verworfen. In diesem Fall wäre die Verwendung von *Diary Studies* mit höheren Kosten verbunden gewesen und hätte schlechtere Ergebnisse erbracht. Andere Methoden eigneten sich schlichtweg besser für diesen Benutzertest. Durch die aktiv teilnehmende Beobachtung ergaben sich interessante Gespräche und die Funktionalitäten konnten in einer entspannten sozialen Situation erprobt werden.

Der abschließende Benutzertest zeigte erneut ein sehr positives Bild über die Benutzerakzeptanz und die Usability des Systems. Die BenutzerIn fand zahlreiche Möglichkeiten, die Tags einzusetzen, und nutzte sie sinnvoll, um ihre persönlichen Ziele zu erreichen. Sie erleichterten ihr in gewissen Situationen die Steuerung der Heimelektronik immens und benötigte weniger Aufmerksamkeit und Konzentration als die herkömmlichen Arten der Interaktion zur Bedienung der Geräte. Die eingesetzten Methoden eigneten sich äußerst gut zur Beantwortung der Forschungsfrage. Die angenehme Atmosphäre und die verschiedenen frei gewählten Szenarien boten viele interessante Gespräche und Einblicke in die Benutzung des Systems. Abbildung 5.8 zeigt ein paar Bilder der Platzierungen von NFC-Tags im zweiten Benutzertest.



(a) Fenstersteuerung

(b) Mediensteuerung

Abbildung 5.8: Platzierungen der NFC-Tags im zweiten Benutzertest

Die Benutzerakzeptanz hängt jedoch auch von der optischen Gestaltung der Tags ab. Wenn sie keinen Gefallen finden, werden sie nicht platziert und somit auch nicht sinnvoll verwendet. Ebenfalls ist eine große Anzahl an zu steuernden Geräten vorteilhaft, wie [46] feststellten. Eine BenutzerIn die nur ein Mindestmaß an Geräten besitzt die sich fernsteuern ließen, würde die Sinnhaftigkeit eines solchen Systems ebenfalls weiter in Frage stellen. Tomitsch et al. [15] stellten außerdem fest das NFC-Tags keinerlei Affordance bieten. Möglicherweise kann sich dieser Umstand mit der starken Verbreitung von NFC Anwendungen ändern, wenn es normal für die BenutzerIn wird, Gegenstände mit einem persönlichen Mobilgerät zu berühren.

5.2 Fazit

Um ein System zur Fernbedienung einer Vielzahl von Haushaltsgeräten per Smartphone und NFC-Tags zu entwickeln, ist es notwendig, sich mit den verwendeten Technologien im Detail auseinanderzusetzen. Die große Anzahl verschiedener Technologien zur Fernsteuerung und der Tatsache, dass viele Geräte nicht mit einer Fernbedienung geliefert werden, lassen die Vision vom *Smart Home* und von *Ubiquitous Computing* für den Endverbraucher unerreichbar erschei-

nen. Zur Erstellung des Systems mussten daher eine Vielzahl an Entwicklungsarbeiten und Konfigurationstätigkeiten durchgeführt werden.

Durch Modifikationen an der Software des Android Testgeräts konnte die Interaktion mit den NFC-Tags weitgehend verbessert werden. Die initiale Einlernphase zum Finden des Lesebereichs konnte jedoch nicht vermieden werden.

Die BenutzerInnen bevorzugten eine konsistente Gestaltung der Grafiken bei den NFC-Tags und der Touchscreen-Fernbedienung. Dadurch bieten sie der BenutzerIn einen Wiedererkennungswert und die dahinter stehenden Funktionen werden schneller identifiziert. Die schlichte Gestaltung durch farbliche Rahmen hilft, die Art der Interaktion und die Funktion noch besser erkennen zu können.

Abgesehen von der Einlernphase bietet die Bedienung per NFC-Tag eine ähnliche Effizienz und Effektivität wie die Touchscreen-Variante. In gewissen Situationen bietet sie jedoch ein zusätzliches Maß an Komfort, wie beispielsweise während eines Gesprächs.

Die Verwendung des Systems war einfach zu erlernen. Verglichen mit herkömmlichen Methoden zeigte die Universalfernbedienung große Vorteile. Unter Miteinbeziehung der Interaktion mit den NFC-Tags stieg der Lernaufwand insgesamt nur marginal.

Die Verwendung einer Universalfernbedienung steigerte die Benutzerzufriedenheit enorm. Das Gefallen an der NFC Interaktion hängt jedoch von wesentlich mehr Faktoren ab. Neben dem optischen Aspekt ist es auch notwendig, Aktivitäten und dazugehörige Orte zu identifizieren um sinnvolle Platzierungen für Tags zu finden. Personen, die kaum elektronische Geräte besitzen und diese auch noch selten benutzen, werden kein großes Gefallen und Tags in ihrer Umgebung finden.

Der mögliche Mehrwert und der subjektiv wahrgenommene Nutzen hängen unter anderem von den zuvor genannten Faktoren ab. Die Vision des *Smart Home* und *Ubiquitous Computing* wird von vielen Menschen sehr positiv aufgenommen, in der Realität sind die vorhandenen Lösungen jedoch oft zu teuer, zu komplex und zu aufwendig. Dennoch ist eine generelle Benutzerakzeptanz vorhanden und es existiert eine Zielgruppe für ein solches System.

5.3 Weiterführende Entwicklungen

Dieser Abschnitt beschreibt mögliche Weiterentwicklungen verschiedenster Art. Dabei handelt es sich neben Erweiterungen der Applikation reYmote auch um die zukünftige Entwicklung neuer und bestehender Systeme.

5.3.1 Mehrbenutzerbetrieb durch Synchronisierung

Während der Benutzertests dieser Arbeit (siehe Abschnitt 4.9.1) wurde festgestellt, dass die Verwendung eines dedizierten Smartphones zur Steuerung per NFC bei gewissen Aktivitäten Vorteile bieten kann. Dazu ist es jedoch notwendig, dass Datenmodell der Fernsteuerungsapplikation zwischen den verschiedenen Geräten zu synchronisieren, um den Status der gesteuerten Geräte richtig behandeln zu können. Durch diese Erweiterung könnten Familien gemeinsam das Heim steuern und jedes beliebige Smartphone oder Tablet kann die Steuerung übernehmen.

5.3.2 Erleichterung der Konfiguration

In der ersten Iteration wurde absichtlich eine Abgrenzung definiert, um nicht näher auf den Aspekt der Konfiguration durch die unerfahrenen BenutzerIn einzugehen. Die Applikation *reYmote* leidet momentan durchaus unter *Self-Referential Design*. Die Konfiguration der Fernbedienung verlangt, dass sich die BenutzerIn mit dem Datenmodell auseinandersetzt, um ein sichtbares Ergebnis zu erzielen zu können. Um die Inbetriebnahme zu erleichtern, wäre es notwendig Programmlogik, Benutzerschnittstellen und Gerätedatenbanken zu kombinieren und somit ein abgerundetes Bild zu bieten.

5.3.3 Gerätedatenbanken

Die Planung, Erstellung, Verwaltung und Aktualisierung von Datenbanken die Geräte- und Protokollinformationen zur Fernsteuerung beinhalten, ist sehr zeit- und kostenintensiv. Auf Anfrage bei *onlyoneremote.com* wurde ein Lizenzpreis von mehreren Tausend Dollar pro Jahr genannt. Dabei handelt es sich um eine exzellent gewartete IR Datenbank. Sie enthält jedoch keine Informationen zu anderen Steuerungsmöglichkeiten. Die einzige Möglichkeit, ein solches System zu verwalten und aktualisieren, wäre mit Hilfe von *Crowd Sourcing*. Die Entwicklung einer Web-Plattform als Schnittstelle zu dem System und ein gutes Datenbankdesign wären jedoch notwendig, um dieses Projekt ins Leben zu rufen.

5.3.4 Android NFC im Standby

Um für die Heimsteuerung besser geeignet zu sein, sollte das Android Betriebssystem die Verwendung des NFC Moduls im Standby erlauben. Laut der Darstellung des Stromverbrauchs am Testgerät wirkte sich die Modifikation der original NFC Applikation nicht merklich negativ aus. Um den Sicherheitsaspekt dieser Einschränkung jedoch nicht zu gefährden, sollte diese Option unbedingt den aktuellen Ort des Geräts miteinbeziehen. Die einfachste Möglichkeit wäre die Konfiguration durch die Definition von Heimnetzwerken. Wenn das Gerät zu einem der konfigurierten WLANs verbunden ist, könnte die Funktion bedenkenlos aktiviert werden. Um dennoch weiter Strom zu sparen, könnte dies in Verbindung mit dem Beschleunigungssensor des Smartphones gekoppelt werden. Dies geschieht unter der Annahme, dass die BenutzerIn das Gerät zum Tag bewegt und nicht umgekehrt. Somit würde das NFC Modul nur aktiviert werden, wenn die BenutzerIn das Telefon bewegt und sie zuhause ist.

5.3.5 Positionierung der NFC Antenne

Zu Beginn jedes Benutzertests, hatten die TeilnehmerInnen Probleme dabei den NFC Lesebereich des Smartphones auf Anhieb richtig zu verwenden. Eine initiale Erklärung über die Positionierung der Antenne änderte nichts daran. Tomitsch et al. [15] beschrieben das selbe Problem und merkten an, dass NFC-Tags keinerlei *Affordance* bieten. Es wäre daher empfehlenswert, die Positionierung der Antenne des NFC Moduls für bestimmte Gerätetypen zu standardisieren und entsprechend zu kennzeichnen, um Bedienfehler zu minimieren.

5.3.6 Wearable NFC Lesegerät

Eine Armbanduhr oder ein Ring als NFC Lesegerät wäre ebenfalls eine interessante Entwicklung. Somit wäre der Griff zum Smartphone obsolet und das Lesegerät könnte drahtlos das Smartphone über die eingelesenen Tags informieren. Eine solche Interaktion, wäre auch mit dem bereits entwickelten NFC-Tag Ring möglich. Dafür müsste man jedoch aktive NFC Lesegeräte in der Umgebung integrieren, die Reaktionen aufgrund von Kontakt mit dem Ring auslösen. Dies wäre mit erheblich höheren Kosten verbunden. Die Lesegeräte brauchen außerdem eine Stromversorgung und lassen sich nicht so leicht an beliebigen Objekten anbringen wie NFC-Tags. Im Projekt von [46] wird ein beispielsweise ein NFC Lesegerät in einem Korb beim Eingangsbereich eingesetzt. Legt die BenutzerIn ihren Schlüsselbund inklusive passivem Tag darin ab, werden entsprechende Steuerungen ausgelöst. Mit Hilfe der Lösung mit einem Ring würden sich neue Möglichkeiten zur Schaffung von *Affordance* bieten, da mit der Hand selbst bedient wird und nicht mit einem Smartphone.

Zusammenfassung

Im Rahmen dieser Arbeit wurde ein System entwickelt und evaluiert, welches die Heimsteuerung mittels NFC-Tags und Android Smartphone erlaubt. An die Einfachheit der Interaktionen wurde dabei ein Höchstmaß an Anforderungen gestellt, um der BenutzerIn ein möglichst positives Erlebnis bieten zu können. Unter Zuhilfenahme von benutzerzentrierten Methoden wurden weitere Ziele für die Entwicklung definiert und evaluiert.

Im Laufe ausgedehnter Entwicklungsarbeiten wurde eine Testumgebung geschaffen, in der eine Vielzahl an Aktionen per Fernbedienung ausgelöst werden können. Die Räumlichkeiten dienten anschließend als Grundlage für die verschiedensten Benutzertests, die zur Beantwortung der Forschungsfragen durchgeführt wurden.

Durch die Anwendung eines iterativen Entwicklungsprozesses konnten die Entwicklungen effektiv vorangetrieben werden. Das Feedback der BenutzerInnen und die Erkenntnisse aus Beobachtungen brachten wertvolle Informationen für Weiterentwicklungen und Verbesserungen.

Die Untersuchungen zeigten, dass die Interaktion mit Haushaltsgeräten per NFC und Smartphone durchaus Gefallen findet und einen gewissen Mehrwert bieten kann. In bestimmten Situationen fordert sie auffallend weniger Konzentration und Aufmerksamkeit von der BenutzerIn und kann somit die Benutzerzufriedenheit und Usability erhöhen.

Persönliches Resümee

Die Anwendung des iterativen Entwicklungsprozesses im Zuge dieser Arbeit bewies sich als äußerst sinnvoll. Die Beantwortung der Forschungsfragen wäre in einer einzigen Iteration nicht möglich gewesen. Außerdem hat sie ihren Zweck erfüllt, indem dadurch ein besseres System entstanden ist. Die Anwendung iterativer Methoden ist unumgänglich, um ein gut nutzbares System erschaffen zu können. Ebenso sind benutzerzentrierte Methoden unerlässlich.

Die aktiv teilnehmende Beobachtung über einen längeren Zeitraum erwies sich auch als guter Schachzug. Damit konnten die restlichen Forschungsfragen schlussendlich geklärt werden.

Die Personas waren ein interessantes Entwicklungswerkzeug. Sie hatten in Kombination mit den Szenarios durchaus ihren Nutzen. Ich bin jedoch sicher, dass sie in dieser Arbeit nicht ihre volle Wirkung entfalten konnten. Ich persönlich würde gerne ihren Einsatz in einem großen Team und einem großen Projekt miterleben.

Aus dieser Arbeit entstand ein System, das ich selbst gerne nutze. Die Faszination an den meisten der hier behandelten Themen war mit einer der größten Motivatoren, um diese Arbeit neben der Berufstätigkeit fertigzustellen.

Literaturverzeichnis

- [1] Logitech Inc., *Harmony 1100 User Manual*. Logitech Inc., 2008. [Online]. Available: <http://www.logitech.com/assets/44641/3/harmony1100-user-guide.pdf>.
- [2] M. Nosrati, R Karimi, and H. Hasanvand, "Mobile Computing: Principles, Devices and Operating Systems", *World Applied Programming*, vol. 2, no. 7, pp. 399–408, 2012. [Online]. Available: http://www.it.iitb.ac.in/arndg/dokuwiki/images/3/38/Android_wifi_working_principles.pdf.
- [3] C. Geisler, "When management becomes personal: An activity-theoretic analysis of Palm technologies", in *Writing Selves/Writing Societies*, 2003, pp. 125–158. [Online]. Available: http://wac.colostate.edu/books/selves_societies/geisler/.
- [4] Psion Inc., *PSION ORGANIZER II*. Psion Inc., 1987. [Online]. Available: <http://www.thecomputerarchive.com/archive/Computers/NonPCportables/PsionOrganiserII1987.pdf>.
- [5] J. Quain, "Simon sends - Simon PDA from BellSouth Cellular - Hardware Review - Evaluation", *Home Office Computing*, 1995. [Online]. Available: <http://research.microsoft.com/en-us/um/people/bibuxton/buxtoncollection/a/pdf/simonreview.pdf>.
- [6] H Bauge, "Secdroid: An Improved Alarm Distribution System", Master Thesis, University of Oslo, 2013. [Online]. Available: <http://home.ifi.uio.no/paalh/students/HaavardBauge.pdf>.
- [7] H. Sugiharto, *Current and Future Mobile Platforms*. Technical University Berlin, 2010. [Online]. Available: http://www.snet.tu-berlin.de/fileadmin/fg220/courses/SS10/snet-project/mobile-platforms_sugiharto.pdf.
- [8] D. Ehringer, *The dalvik virtual machine architecture*. Boston University, 2010. [Online]. Available: http://show.docjava.com/posterous/file/2012/12/10222640-The_Dalvik_Virtual_Machine.pdf.

- [9] S. Burkard, *Near Field Communication in Smartphones*. Berlin Institute of Technology, 2011. [Online]. Available: http://www.snet.tu-berlin.de/fileadmin/fg220/courses/WS1112/snet-project/nfc-in-smartphones_burkard.pdf.
- [10] S. Navya and S. Badugu, "Radio-Frequency Identification Incorporation with Wireless Sensor Networks and Devices", *International Journal of Computer Science and Telecommunications*, vol. 4, no. 2, pp. 111–114, 2013. [Online]. Available: <http://www.ijcst.com/vol42/1/b1446.pdf>.
- [11] P. Nikitin, "Leon Theremin (Lev Termen)", *Antennas and Propagation Magazine, IEEE*, vol. 54, no. 5, pp. 252–257, 2012. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6348173.
- [12] I. Singh, "A High-Precision Indoor Tracking System", Master Thesis, Dalhousie University, 2013. [Online]. Available: <http://dalspace.library.dal.ca/handle/10222/34642>.
- [13] S. Puglia and A. Vitaletti, "Alternative RFID based Architectures for Mobile HCI with Physical Objects", in *Mobile Interaction with the Real World (MIRW 2006) Workshop*, Espoo, 2006, pp. 34–38. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.190.2788&rep=rep1&type=pdf>.
- [14] H. Ailisto, T. Matinmikko, and A. Ylisaukko-Oja, *Physical browsing with NFC technology*. VTT Technical Research Centre of Finland, 2007, ISBN: 9789513869465. [Online]. Available: <http://vtt.fi/inf/pdf/tiedotteet/2007/T2400.pdf>.
- [15] M. Tomitsch, T. Grechenig, and R. Schlögl, *Real-World Tagging in the Wild: On the Usability and Accessibility of NFC-based Interactions*. Vienna University of Technology, 2008.
- [16] G. Gridling and B. Weiss, *Introduction to microcontrollers*. Vienna University of Technology, 2007. [Online]. Available: <http://ti.tuwien.ac.at/ecs/teaching/courses/mclu/theory-material/Microcontroller.pdf>.
- [17] R. Bannatyne and G. Viot, "Introduction to microcontrollers", *Wescon/97. Conference Proceedings*, pp. 564–574, 1997. [Online]. Available: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=632384&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D632384.
- [18] Atmel Corporation, *ATmega328P datasheet*. Atmel Corporation, 2009. [Online]. Available: <http://www.atmel.com/Images/doc8161.pdf>.
- [19] A. Plepys, "Environmental Implications of Product Servicising", PhD thesis, Lund University, 2004. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=2E60880B128C0E39C00D0F37876A2251?doi=10.1.1.118.6922&rep=rep1&type=pdf>.
- [20] M. Banzi, *Getting Started with arduino*. Maker Media Inc., 2009, ISBN: 9780596155513.

- [21] A. Gibb, “New media art, design, and the Arduino microcontroller: A malleable tool”, Master Thesis, School of Art and Design, 2010. [Online]. Available: <http://aliciagibb.com/wp-content/uploads/2013/01/New-Media-Art-Design-and-the-Arduino-Microcontroller-2.pdf>.
- [22] C. Shannon and W. Weaver, *The mathematical theory of communication*. University of Illinois Press, 1949. [Online]. Available: <http://www.magmamater.cl/MatheComm.pdf>.
- [23] C. Shannon, “A mathematical theory of communication”, *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 1–54, 2001. [Online]. Available: <http://dl.acm.org/citation.cfm?id=584093>.
- [24] T. Yamazaki, “Beyond the smart home”, *International Conference on Hybrid Information Technology (ICHIT'06)*, pp. 350–355, Nov. 2006. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4021238>http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4021238.
- [25] C. Pätz, *Z-Wave Grundlagen*. Books on Demand, 2011, ISBN: 3842334338. [Online]. Available: http://smarhome.fi/img/cms/PDF/ZWHB_en_1.0.pdf.
- [26] J. L. Ryan, “Home automation”, *Electronics Communication Engineering Journal*, pp. 185–192, 1989.
- [27] W. Kastner, “Communication systems for building automation and control”, *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1179–1203, 2005. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1435745.
- [28] H. Brandstetter, “Heimautomationssysteme - Theorie und Praxis”, Master Thesis, Vienna University of Technology, 2008. [Online]. Available: https://www.autotuwien.ac.at/bib/pdf_thesis/THESIS0003.pdf.
- [29] A. Dentel and U. Dietrich, “Thermische Behaglichkeit - Komfort in Gebäuden”, *rom-umwelt-stiftung.de*, pp. 1–37, 2013. [Online]. Available: http://rom-umwelt-stiftung.de/wp-content/uploads/2006/02/Dokumentation_Thermische_Behaglichkeit.pdf.
- [30] C. Reinisch, W. Kastner, G. Neugschwandtner, and W. Granzer, “Wireless Technologies in Home and Building Automation”, *2007 5th IEEE International Conference on Industrial Informatics*, pp. 93–98, Jul. 2007, ISSN: 1935-4576. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4384737>.
- [31] R. Torbensen, K. M. Hansen, and T. S. Hjorth, “My Home is My Bazaar - A Taxonomy and Classification of Current Wireless Home Network Protocols”, *2011 Second Eastern European Regional Conference on the Engineering of Computer Based Systems*, pp. 35–43, Sep. 2011. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6037512>.

- [32] D. Rye, “How it all started”, *HTI News*, vol. 3, no. 1, 1984. [Online]. Available: <http://hometoys.com/emagazine.php?url=/htinews/feb98/columns/x10/x10.htm>.
- [33] M. Galeev, “Catching the z-wave”, *Electronic Engineering Times India*, 2006. [Online]. Available: http://www.eetindia.co.in/ARTICLES/2006OCT/PDF/EEIOL_2006OCT30_RFD_EMS_TA.pdf.
- [34] SOMFY Group, *io-homecontrol - The residential radio protocol*. Pepper Studio, 2010. [Online]. Available: http://www.somfysystems.com/downloads/nam/guide_io_en.pdf.
- [35] G. Martin, *Wireless sensor solutions for home building automation - the successful standard uses energy harvesting*. EnOcean GmbH, 2007.
- [36] M. Weiser, “Some computer science issues in ubiquitous computing”, *Communications of the ACM*, vol. 36, no. 7, pp. 75–84, 1993. [Online]. Available: <http://dl.acm.org/citation.cfm?id=159617>.
- [37] —, “The computer for the 21st century”, *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 3, no. 3, pp. 3–11, 1991. [Online]. Available: <http://dl.acm.org/citation.cfm?id=329126>.
- [38] M. Mühlhäuser and I. Gurevych, “Introduction to Ubiquitous Computing”, in *Handbook of Research: Ubiquitous Computing Technology for Real Time Enterprises*, IGI Global, 2009, pp. 1–19. [Online]. Available: <http://www.igi-global.com/viewtitle.aspx?titleid=37772>.
- [39] A. Ferscha, *What is Pervasive Computing?* Johannes Kepler Universität Linz, 2003.
- [40] O. Shaer, “Tangible User Interfaces: Past, Present, and Future Directions”, *Foundations and Trends in Human-Computer Interaction*, vol. 3, no. 1-2, pp. 1–137, 2009, ISSN: 1551-3955. [Online]. Available: <http://www.nowpublishers.com/product.aspx?product=HCI&doi=1100000026>.
- [41] S. McConnell, “Daily build and smoke test”, *IEEE software*, vol. 13, no. 4, pp. 1–2, 1996. [Online]. Available: <http://ngis.computer.org/csdl/mags/so/1996/04/s4144.html>.
- [42] H. Ishii and B. Ullmer, “Tangible Bits : Towards Seamless Interfaces between People , Bits and Atoms”, in *CHI '97 Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, 1997, pp. 234–241, ISBN: 0897918029.
- [43] P. Väikkynen, *Physical Selection in Ubiquitous Computing*. VTT Technical Research Centre of Finland, 2007, p. 175, ISBN: 9789513870614.
- [44] J. Preece, Y. Rogers, and H. Sharp, *Interaction design: beyond human-computer interaction*. John Wiley and Sons Inc., 2002. [Online]. Available: <http://doc58.minibooks.org/helen-sharp-yvonne-rogers-jenny-preece-PDF-573818.pdf>.

- [45] B. Ullmer and H. Ishii, “Emerging frameworks for tangible user interfaces”, *IBM systems journal*, vol. 39, no. 3, pp. 1–15, 2000. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5387042.
- [46] M.-A. Llorente-Carmona, A. M.-C. Martínez-Cavero, and J.-P. Lázaro-Ramos, “Environment control platform based on KNX and NFC technologies to support independent daily life”, in *KNX Scientific Conference*, 2010.
- [47] C. Beckmann, “SiteView: Tangibly programming active environments with predictive visualization”, *Intel Research Tech Report*, pp. 167–168, 2003. [Online]. Available: http://intel-research.net/Publications/Berkeley/070920031125_143.pdf.
- [48] I. Sommerville, “Software process models”, *ACM Computing Surveys*, vol. 28, no. 1, pp. 269–271, Mar. 1996, ISSN: 03600300. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=234313.234420>.
- [49] P. Rannikko, “User-Centered Design in Agile Software Development”, Master Thesis, University of Tampere, 2011. [Online]. Available: <http://tutkielmat.uta.fi/pdf/gradu04854.pdf>.
- [50] M. Cusumano and R. Selby, “How Microsoft builds software”, *Communications of the ACM*, pp. 53–61, 1997. [Online]. Available: <http://dl.acm.org/citation.cfm?id=255698>.
- [51] The Standish Group International Inc., *The chaos report*. The Standish Group International Inc., 1995. [Online]. Available: <http://www.projectsmaart.co.uk/docs/chaos-report.pdf>.
- [52] P. Kruchten, *From Waterfall to Iterative Development – A Challenging Transition for Project Managers*. therationaleedge.com, 2001.
- [53] D. A. Norman, “Cognitive engineering”, *User centered system design*, pp. 32–61, 1986.
- [54] J. Gulliksen, B. Göransson, I. Boivie, S. Blomkvist, J. Persson, and A. s. Cajander, “Key principles for user-centred systems design”, in *Behaviour Information Technology*, 6, vol. 22, Nov. 2003, pp. 397–409, ISBN: 0144929031. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/01449290310001624329>.
- [55] L. Constantine and L. Lockwood, “Usage-centered engineering for Web applications”, *Software, IEEE*, vol. 1, no. 978, pp. 1–15, 2002. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=991331.
- [56] J. Arnold, *Big ideas: a history of field research in industrial design in the United States*. University of Art and Design Helsinki, 2005. [Online]. Available: <http://www2.uiah.fi/joiningforces/papers/Arnold.pdf>.
- [57] J. Gulliksen, A. Lantz, and I. Boivie, “User centered design in practice-problems and possibilities”, in *Participatory Design Conference*, Stockholm: Royal Institute of Technology, 1999. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.32.2003&rep=rep1&type=pdf>.

- [58] A. Cooper, *About Face 3: The Essentials of Interaction Design*. Wiley Publishing, Inc., 2007, ISBN: 9780470084113. [Online]. Available: <http://ivi.sagepub.com/lookup/doi/10.1057/palgrave.ivs.9500066>.
- [59] ———, *The inmates are running the asylum*. Macmillan Publishing Co. Inc., 1999, ISBN: 0672316498.
- [60] G. A. Moore, *Crossing the chasm*. HarperCollins Publishers, 1991, p. 174, ISBN: 0060189878. [Online]. Available: <http://www.ic-at.org/papers/01033.pdf>.
- [61] J. Pruitt, “Personas: practice and theory”, *Proceedings of the 2003 conference on Designing for user experiences*, pp. 1–15, 2003. [Online]. Available: <http://dl.acm.org/citation.cfm?id=997078.997089>.
- [62] T. Adlin and J. Pruitt, *The Essential Persona Lifecycle*. Morgan Kaufmann Publishers, 2005, ISBN: 9780123814180.
- [63] A. Toxboe, *Introducing User-Centered Design to eXtreme Programming*. Copenhagen Business School, 2005.
- [64] S. Blomkvist, “Persona - an overview Personas and goal-directed design”, *The User as a personality. Using Personas as a tool for design*, pp. 1–8, 2002.
- [65] L. Nielsen, *Engaging personas and narrative scenarios*. Copenhagen Business School, 2004. [Online]. Available: <http://personas.dk/wp-content/samlet-udgave-til-load.pdf>.
- [66] J. M. Carroll, “Five reasons for scenario-based design”, in *HICSS '99 Proceedings of the Thirty-Second Annual Hawaii International Conference on System Sciences-Volume 3*, 1999, p. 3051, ISBN: 0769500013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0953543800000230>.
- [67] I. Alexander and L. Beus-Dukic, *Discovering requirements: how to specify products and services*. John Wiley and Sons Ltd., 2009, ISBN: 9780470712405.
- [68] A. Cockburn, *Writing effective use cases*. Addison-Wesley Professional, 2001, ISBN: 0201702258.
- [69] E. Scheuch, “Das Interview in der Sozialforschung”, in *Handbuch der empirischen Sozialforschung*, GESIS - Leibniz-Institut für Sozialwissenschaften, 1973.
- [70] R. Schnell, P. Hill, and E. Esser, *Methoden der empirischen Sozialforschung*. Oldenbourg Wissenschaftsverlag, 2011. [Online]. Available: http://books.google.at/books?id=d37SHMEuJAKC&printsec=frontcover&dq=Methoden+der+empirischen+Sozialforschung&hl=en&sa=X&ei=ppF_UrubPJDAAtAbp-YCIAQ&redir_esc=y#v=snippet&q=beobachtung&f=false.
- [71] S. Grüttner, B. Egle, and L. Urich, “Interview”, Seminararbeit, Pädagogische Hochschule Freiburg, 2006. [Online]. Available: https://www.ph-freiburg.de/fileadmin/dateien/fakultaet3/sozialwissenschaft/Quasus/Hausarbeiten/EW2_HA-_Interview.pdf.

- [72] S. Schmid, “Teilnehmende Beobachtung: Offen oder verdeckt?”, Seminararbeit, Universität Luzern, 2009. [Online]. Available: <http://content.grin.com/document/v147540.pdf>.
- [73] M. Kuniavsky, *Observing the user experience*. Morgan Kaufmann Publishers, 2001, ISBN: 0123848695. [Online]. Available: http://dusk.geo.orst.edu/ethics/papers/Kuniavsky_2003.pdf.
- [74] R. R. Johnson, M. J. Salvo, and M. W. Zoetewey, “User-Centered Technology in Participatory Culture: Two Decades Beyond a Narrow Conception of Usability Testing”, *IEEE Transactions on Professional Communication*, vol. 50, no. 4, pp. 320–332, Dec. 2007, ISSN: 0361-1434. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4381241>.
- [75] J. Nielsen, *Usability Engineering*. Morgan Kaufmann Publishers, 1993, ISBN: 0125184069.
- [76] W. Quesenbery, “Balancing the 5Es : Usability set your mind at Es”, *Cutter IT Journal*, vol. 17, no. 2, pp. 4–11, 2004.
- [77] P. Botella, X. Burgués, J. Carvallo, X. Franch, G. Grau, J. Marco, and C. Quer, “ISO/IEC 9126 in practice: what do we need to know?”, in *Software Measurement European Forum 2004*, Rome, 2004, pp. 297–306. [Online]. Available: [http://www.essi.upc.edu/\\$\sim\\$webgessi/publicacions/SMEF'04-ISO-QualityModels.pdf](http://www.essi.upc.edu/\simwebgessi/publicacions/SMEF'04-ISO-QualityModels.pdf).
- [78] A. Seffah, M. Donyaee, R. B. Kline, and H. K. Padda, “Usability measurement and metrics: A consolidated model”, *Software Quality Journal*, vol. 14, no. 2, pp. 159–178, Jun. 2006, ISSN: 0963-9314. [Online]. Available: <http://www.springerlink.com/index/10.1007/s11219-006-7600-8>.
- [79] J. J. Gibson, *The Theory of Affordances*. Lawrence Erlbaum Associates Inc., 1977.
- [80] D. Norman, *The design of everyday things*. MIT Press, 1998, ISBN: 0262640376.
- [81] W. W. Gaver, “Technology affordances”, in *CHI '91 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1991, pp. 79–84. [Online]. Available: <http://dl.acm.org/citation.cfm?id=108856>.
- [82] D. Saffer, “The role of metaphor in interaction design”, Master Thesis, Carnegie Mellon University, 2005. [Online]. Available: http://www.odannyboy.com/portfolio/thesis/saffer_thesis_paper.pdf.
- [83] T. Tullis and B. Albert, *Measuring the user experience*. Morgan Kaufmann Publishers, 2008.
- [84] A. Fernández-Caballero, V. López-Jaquero, F. Montero, and P. González, “Usability metrics in adaptive agent-based tutoring systems”, *E-learning/E-teaching on the Web. International Conference on Web Engineering, ICWE 2003*, pp. 144–154, 2003. [Online]. Available: http://www.dsi.uclm.es/personal/AntonioFdez/nais/nais/investigacion/publicaciones/congresos_2003/HCII2003-usability.pdf.

- [85] N. Bevan and M. Macleod, “Usability measurement in context”, *Behaviour Information Technology*, vol. 13, no. 1-2, pp. 132–145, Jan. 1994, ISSN: 0144-929X. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/01449299408914592>.
- [86] N. Bevan, “Measuring usability as quality of use”, *Software Quality Journal*, vol. 4, no. 2, pp. 115–130, 1995. [Online]. Available: <http://link.springer.com/article/10.1007/BF00402715>.
- [87] M. Macleod and R. Bowden, “The MUSiC performance measurement method”, *Behaviour and Information Technology*, vol. 16, pp. 279–293, 1997. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/014492997119842>.
- [88] J. Brooke, “SUS-A quick and dirty usability scale”, *Usability evaluation in industry*, 1996.
- [89] A. Bangor, P. Kortum, and J. Miller, “Determining what individual SUS scores mean: Adding an adjective rating scale”, *Journal of usability studies*, vol. 4, no. 3, pp. 114–123, 2009. [Online]. Available: http://66.39.39.113/upa_publications/jus/2009may/JUS_Bangor_May2009.pdf.
- [90] G. Beal and J. Bohlen, “The diffusion process”, *Iowa Agricultural Extension Service*, 1957. [Online]. Available: <http://sites.google.com/site/polomacnenad/files/DiffusionProcess.pdf>.

Webseiten

- [91] Logitech Inc., *Logitech Study Shows Multiple Remote Controls Hindering Entertainment Experiences Around the Globe*. [Online]. Available: <http://www.logitech.com/en-us/press/press-releases/7748> (visited on 11/25/2013).
- [92] Wikipedia, *Image of IBM Simon*. [Online]. Available: http://upload.wikimedia.org/wikipedia/commons/9/9e/IBM_Simon_Personal_Communicator.png (visited on 11/04/2013).
- [93] —, *IBM Simon*. [Online]. Available: http://en.wikipedia.org/wiki/IBM_Simon (visited on 11/08/2013).
- [94] —, *Mobile OS Verkäufe*. [Online]. Available: http://upload.wikimedia.org/wikipedia/commons/8/83/World_Wide_Smartphone_Sales.png (visited on 11/07/2013).
- [95] Gartner Inc., *Smartphone Sales, 2013*. [Online]. Available: <http://www.gartner.com/newsroom/id/2573415> (visited on 11/07/2013).
- [96] Google Inc., *Nexus 5*. [Online]. Available: <http://www.google.com/nexus/5/> (visited on 11/08/2013).
- [97] A. Eromenko, *Battle of the Androids: Google Android vs. Samsung Android, 2013*. [Online]. Available: http://www.osnews.com/story/26751/Battle_of_the_Androids_Google_Android_vs_Samsung_Android (visited on 11/11/2013).
- [98] Wikipedia, *Android (Operating System)*. [Online]. Available: [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system)) (visited on 11/11/2013).
- [99] The Apache Software Foundation, *Apache License, Version 2.0*. [Online]. Available: <http://www.apache.org/licenses/LICENSE-2.0> (visited on 11/04/2013).
- [100] D. Gillmor, *Android's smartphone OS upgrade issues need more than a quick fix, 2012*. [Online]. Available: <http://www.theguardian.com/commentisfree/2012/oct/28/android-smartphone-os-upgrade> (visited on 11/08/2013).
- [101] A. Cunningham, *What happened to the Android Update Alliance?, 2012*. [Online]. Available: <http://arstechnica.com/gadgets/2012/06/what-happened-to-the-android-update-alliance/> (visited on 11/08/2013).

- [102] Say Media Inc., *iOS Versions*, 2013. [Online]. Available: [http://readwrite.com/2013/10/18/after-one-month-73-of-ios-users-have-adopted-ios-7#awesm=\\$\sim\\$omyw3vKQEYRwCI](http://readwrite.com/2013/10/18/after-one-month-73-of-ios-users-have-adopted-ios-7#awesm=\simomyw3vKQEYRwCI) (visited on 11/08/2013).
- [103] Google Inc., *Android Versions*. [Online]. Available: <http://developer.android.com/about/dashboards/index.html> (visited on 11/08/2013).
- [104] Oracle Corp., *Learn about Java*. [Online]. Available: <http://www.java.com/en/about/> (visited on 11/11/2013).
- [105] Google Inc., *Application Fundamentals*. [Online]. Available: <http://developer.android.com/guide/components/fundamentals.html> (visited on 11/11/2013).
- [106] —, *Android Activity Class*. [Online]. Available: <http://developer.android.com/reference/android/app/Activity.html> (visited on 11/04/2013).
- [107] —, *Intents and Intent Filters*. [Online]. Available: <http://developer.android.com/guide/components/intents-filters.html> (visited on 11/04/2013).
- [108] NFC-Forum, *NFC Forum N-Mark*. [Online]. Available: <http://www.nfc-forum.org/resources/N-Mark/> (visited on 11/03/2013).
- [109] —, *About NFC*. [Online]. Available: <http://www.nfc-forum.org/aboutnfc/> (visited on 11/18/2013).
- [110] Wikipedia, *Near Field Communication*. [Online]. Available: http://en.wikipedia.org/wiki/Near_field_communication (visited on 11/04/2013).
- [111] Popular Science, *Five Reasons Why I'm Excited About Google's Nexus Q*, 2012. [Online]. Available: <http://www.popsci.com/gadgets/article/2012-06/why-im-excited-about-googles-nexus-q> (visited on 11/04/2013).
- [112] CBS Interactive Inc., *The most practical, creative ways to use NFC with your Android device*, 2012. [Online]. Available: http://howto.cnet.com/8301-11310_39-57478334-285/the-most-practical-creative-ways-to-use-nfc-with-your-android-device/ (visited on 11/04/2013).
- [113] Kickstarter Inc., *NFC Ring*. [Online]. Available: <http://www.kickstarter.com/projects/mclear/nfc-ring> (visited on 11/04/2013).
- [114] Mikrocontroller.net, *AVR-Tutorial: Timer*. [Online]. Available: http://www.mikrocontroller.net/articles/AVR-Tutorial:_Timer (visited on 11/11/2013).
- [115] Atmel Corporation, *Atmel AVR 8-bit Microcontroller MTBF*. [Online]. Available: <http://support.atmel.com/bin/customer.exe?=&action=viewKbEntry&id=266> (visited on 11/01/2013).
- [116] Wikipedia, *List of Intel Microprocessors*. [Online]. Available: http://en.wikipedia.org/wiki/List_of_Intel_microprocessors.
- [117] RS Components Handelsges.m.b.H., *Atmega328P-PU im RS Components Webshop*. [Online]. Available: <http://at.rs-online.com/web/p/mikrocontroller/7380435/> (visited on 11/11/2013).

- [118] Arduino Website, *Arduino FAQs*. [Online]. Available: <http://www.arduino.cc/> (visited on 11/03/2013).
- [119] RS Components Handelsges.m.b.H., *Atmel STK500 im RS Components Webshop*. [Online]. Available: <http://at.rs-online.com/web/p/entwicklungskits-prozessor-mikrocontroller/0384935/> (visited on 11/05/2013).
- [120] —, *Atmel STK600 im RS Components Webshop*. [Online]. Available: <http://at.rs-online.com/web/p/entwicklungskits-prozessor-mikrocontroller/6962446/> (visited on 11/05/2013).
- [121] Arduino Website, *Arduino Webshop*. [Online]. Available: <http://store.arduino.cc/ww/index.php> (visited on 11/04/2013).
- [122] Atmel Corporation, *Atmel STK500*. [Online]. Available: <http://www.atmel.com/tools/STK500.aspx> (visited on 11/03/2013).
- [123] —, *Atmel Studio*. [Online]. Available: <http://www.atmel.at/tools/atmelstudio.aspx> (visited on 11/24/2013).
- [124] Arduino Website, *Arduino XBee-Shield*. [Online]. Available: <http://arduino.cc/en/Main/ArduinoXbeeShield> (visited on 11/11/2013).
- [125] Sindono.com, *Arduino mit Ethernet- und XBee-Shield*, 2010. [Online]. Available: <http://www.sindono.com/groen/wp-content/uploads/2010/10/foto-2.JPG> (visited on 11/11/2013).
- [126] Wikipedia, *Remote control*. [Online]. Available: http://en.wikipedia.org/wiki/Remote_control (visited on 11/09/2013).
- [127] Preisvergleich Internet Services AG, *Samsung Smartphones mit IR-Blaster*. [Online]. Available: [http://geizhals.at/?cat=umtsover&xf=158_Infrarot\\$\sim\\$1022_Samsung&sort=p](http://geizhals.at/?cat=umtsover&xf=158_Infrarot\sim1022_Samsung&sort=p) (visited on 11/09/2013).
- [128] T. Bielohlawek, *IrDude*. [Online]. Available: <https://github.com/rngtng/IrDude/blob/master/src/com/rngtng/irdude/MainActivity.java> (visited on 11/09/2013).
- [129] SAMSUNG, *Samsung Smart TV*. [Online]. Available: <http://www.samsung.com/at/microsite/smarttv/> (visited on 11/09/2013).
- [130] T. Quist, *SamyGo Remote*. [Online]. Available: <https://play.google.com/store/apps/details?id=de.quist.app.samyGoRemote&hl=en> (visited on 11/09/2013).
- [131] Ascendtechnology.net, *ZENSY INC*. 2013. [Online]. Available: <http://www.ascendtechnology.net/zensys.htm> (visited on 11/12/2013).
- [132] Io-homecontrol, *Io-homecontrol*. [Online]. Available: <http://io-homecontrol.com/> (visited on 08/20/2012).

- [133] N. Pennella, *Botfest 2003 - Marble Madness Color Sensor*, 2003. [Online]. Available: [http://www.cs.uml.edu/~sim\\$fredm/courses/91.548-spr03/student/npennell/botfest_writeup/everything.jpg](http://www.cs.uml.edu/~sim$fredm/courses/91.548-spr03/student/npennell/botfest_writeup/everything.jpg) (visited on 11/09/2013).
- [134] T. Geis, *Neue ISO 9241-210 'Prozess zur Gestaltung gebrauchstauglicher Systeme' ersetzt die ISO 13407*, 2010. [Online]. Available: <http://blog.procontext.com/2010/03/iso-9241210-prozess-zur-entwicklung-gebrauchstauglicher-interaktiver-systeme-ver%C3%B6ffentlicht.html> (visited on 10/30/2013).
- [135] HTW Chur, *SUMI (Software Usability Measurement Inventory)*. [Online]. Available: <http://www.cheval-lab.ch/index.php?id=653&type=98> (visited on 10/28/2013).
- [136] Marketingmuses.typepad.com, *Technology Adoption Lifecycle*, 2010. [Online]. Available: <http://marketingmuses.typepad.com/marketingmuses/2010/07/crossing-the-chasm-is-counterintuitive.html> (visited on 11/04/2013).
- [137] AOKP Forum, *Official AOKP Build 40 Release Thread*, 2012. [Online]. Available: <http://www.galaxynexusforum.com/forum/galaxy-nexus-roms-lte-cdma/6016-official-aokp-build-40-a-2.html> (visited on 08/17/2012).
- [138] Google Inc., *Android NFC Basics*. [Online]. Available: <http://developer.android.com/guide/topics/connectivity/nfc/nfc.html> (visited on 11/18/2013).
- [139] XDA Forum, *Detecting NFC Tag removal hacking XDA Forum Thread*, 2013. [Online]. Available: <http://forum.xda-developers.com/showthread.php?t=2122019> (visited on 12/08/2012).
- [140] Webduino, *Webduino: Simple and extensible web server for Arduino and Ethernet Shield*. [Online]. Available: <https://github.com/sirleech/Webduino> (visited on 12/09/2013).
- [141] R. Ferrei, *Event-Based Library for Arduino*. [Online]. Available: <https://code.google.com/p/ebl-arduino/> (visited on 09/02/2013).
- [142] Rapidnfc.com, *NFC Tag Sizes*. [Online]. Available: http://rapidnfc.com/nfc_tag_sizes (visited on 07/14/2013).
- [143] Google Inc., *Google I/O 2011: How to NFC*. [Online]. Available: <http://www.youtube.com/watch?v=49L7z3rxx4Q&feature=youtu.be&t=16m29s> (visited on 11/22/2013).

Abbildungsverzeichnis

2.1	IBM Simon [92]	5
2.2	Mobile OS Verkäufe in Tausenden (abgeändert, reproduziert nach [94])	7
2.3	NFC N-Mark [108]	9
2.4	Geöffneter NFC-Tag	11
2.5	NFC-Ring	12
2.6	Arduino Ethernet und Atmel STK500	15
2.7	Arduino mit Ethernet- und XBee-Shield [125]	16
2.8	Shannon-Weaver Kommunikationsmodell (abgeändert, reproduziert nach [22])	17
2.9	Nachbau der Marble Answering-Machine [133]	25
2.10	SiteView [47]	29
3.1	Klassisches Wasserfallmodell (reproduziert nach [49])	32
3.2	Iterativer Prozess (abgeändert, reproduziert nach [52])	33
3.3	Benutzerzentrierter iterativer Designprozess nach ISO 9241-210 [134]	34
3.4	Darstellung der BenutzerInnen auf den Verhaltensachsen [58]	36
3.5	Use-Case Diagramm [67]	38
3.6	Usability als Brücke zwischen Kultur und Wissenschaft [74]	42
3.7	Arten von Affordance (reproduziert nach [81])	44
3.8	Technology Adoption Lifecycle (abgeändert, reproduziert nach [136])	49
4.1	Primäre Persona: Kerstin Ammer	54
4.2	Sekundäre Persona: Armin Buchinger	55
4.3	Anti Persona: Heinz Hübl	55
4.4	Einzelne Komponenten der Z-Wave Lichtschalter	59
4.5	Montierter Taster ohne Rahmen und Wippe	59
4.6	Z-Wave Dimmer-Zwischenstecker	60
4.7	Originale Velux Fernbedienungen	61
4.8	Schaltplan für Steuerung der Velux Fernbedienung	62
4.9	Platine der Velux Fernbedienung	63
4.10	Arduino steuert Velux Fernbedienung	64
4.11	io-homecontrol IP Gateway im Gehäuse	65
4.12	iTach WF2CC von Global Caché	66
4.13	Modifizierte Kaffeemaschine	67

4.14	Datenmodell v1.0.10	69
4.15	reYmote NFC Benutzerschnittstellen	71
4.16	Datenmodell v1.1	73
4.17	Icons der Touchscreen-Fernbedienung 1/2	76
4.18	Icons der Touchscreen-Fernbedienung 2/2	77
4.19	Bildschirmfotos der Touchscreen-Fernbedienung	78
4.20	NFC-Tags beim Bett	79
4.21	NFC-Tags auf Glastisch	80
4.22	Herkömmliche Fernbedienungen für die ersten Benutzertests	84
4.23	SUS Punkte	86
4.24	Relative Benutzereffizienz der MUSiC Metrik	87
4.25	NFC-Tag Lesebereich Vergleich	92
4.26	Google IO NFC Demo [143]	93
4.27	NFC Antennen der zwei Smartphones	93
4.28	Tasker Task Konfiguration	95
4.29	reYmote Tasker-Kommando Konfiguration	95
4.30	NFC-Tag Sammlung für die zweite Iteration	96
4.31	Hauptschirm und neue Radiofernbedienung	97
4.32	Tags zur Fernbedienung der Fenster	99
4.33	Tag zum Wechsel des Radiosenders	100
5.1	Technology Adoption Lifecycle (abgeändert, reproduziert nach [136])	104
5.2	Fotos der verwendeten Personas	104
5.3	Datenmodell nach der ersten Iteration	105
5.4	Ein paar Tags der ersten Iteration	106
5.5	Bilder der ersten Benutzertests	106
5.6	Einige Tags der zweiten Iteration	107
5.7	NFC Lesebereich Untersuchung	108
5.8	Platzierungen der NFC-Tags im zweiten Benutzertest	109

Tabellenverzeichnis

2.1	ATmega328P Speicher	13
2.2	Vergleich RF-basierender Systeme der Heimautomation	23
4.1	Konfiguration von Aktivitäten der ersten Iteration	75
4.2	TeilnehmerInnen der Benutzertests der ersten Iteration	81