

Institut für Visualisierung und Interaktive Systeme
Mensch-Computer-Interaktion
Universität Stuttgart
Pfaffenwaldring 5a
D-70569 Stuttgart

Bachelorarbeit Nr. 23

Visualisierung zur Selbstreflexion der Anwendungsbenutzung auf Mobiltelefonen

Kevin Wenz

Studiengang:	Softwaretechnik
Prüfer:	Prof. Dr. Albrecht Schmidt
Betreuer:	Dipl.-Inf. Niels Henze Alireza Sahami, M. Sc.
begonnen am:	21. Juni 2012
beendet am:	21. Dezember 2012
CR-Klassifikation:	H.5.2, H.1.2

Kurzfassung

Seit der Einführung des Mobiltelefons gewann dieses immer mehr an Bedeutung und entwickelte sich über die Jahre zum ständigen Begleiter für Jugendliche und Erwachsene. Inzwischen bestimmt und organisiert das Mobiltelefon den Alltag junger Menschen. Diese verbringen viel Zeit mit Spielen, Surfen und sonstigen Aktivitäten. In der vorliegenden Arbeit wurde eine graphische Visualisierung von protokollierten Nutzerdaten auf Mobiltelefonen entwickelt. Sie soll dazu dienen, das eigene Nutzungsverhalten auf Mobiltelefonen zu überblicken und zu hinterfragen. Dazu wurde in einem iterativen Prozess eine auf Balkendiagrammen basierende Darstellung entwickelt, die es erlaubt, Ausführungsdaten von Anwendungen in bestimmte Zeiträume eingeteilt darzustellen. Zur Identifikation einzelner Anwendungen wurden Anwendungssicons verwendet. Implementiert wurde die Visualisierung in den Anwendungen *AppTicker* und *HistApp*. Zur Implementierung kam das für Android modifizierte Java zum Einsatz. Dort konnte mit Hilfe des *Canvas*-Elements der Klasse *View* aus einfachen geometrischen Elementen das Diagramm gebaut werden. Schließlich wurden Nutzungsverhaltensänderungen analysiert. Bei einigen Nutzern wurde festgestellt, dass diese bestimmte Anwendungen tendenziell weniger nutzten. Aufgrund ausgeprägter Unregelmäßigkeiten in Mobiltelefonnutzungen, die Ausführungszeiten einzelner Anwendungen schwanken von Tag zu Tag stark, sollten Daten, welche analysiert werden, über einen längeren Zeitraum gesammelt werden.

Inhaltsverzeichnis

1	Einleitung	7
2	Verwandte Arbeiten und Grundlagen	9
2.1	Analyse von Mobiltelefonnutzungen	9
2.2	Life-Logging und Persuasive Design	10
2.3	Konzepte der Visualisierung	11
2.4	Fazit der Verwandten Arbeiten	12
2.5	Grundlagen	12
3	Visualisierungskonzept	15
3.1	Relevante Daten	15
3.2	Art der Darstellung	16
3.3	Diagrammtypen	17
3.4	Beschriftung	22
3.5	Darstellung der Anwendungssicons	22
3.6	Zusammenfassung	24
4	Anwendung	25
4.1	<i>AppTicker</i>	25
4.2	Implementierungsart	26
4.3	Implementierungsvorgehen	28
4.4	Performanz	29
4.5	<i>HistApp</i>	30
4.6	Zusammenfassung	31
5	Benutzerstudie	33
5.1	Teilnehmer	33
5.2	Vorgehen	33
5.3	Ergebnis	34
6	Analyse der Nutzung und Verhaltensänderung	37
6.1	Nutzerfeld	37
6.2	Nutzung der Visualisierung	37
6.3	Nutzungsverhalten und -änderung	39
6.4	Zusammenfassung	40
7	Fazit und Ausblick	41
	Literaturverzeichnis	45

1 Einleitung

Als das Mobiltelefon anfangs an Bedeutung zu gewinnen, diente es einzig und allein dem Zweck, unterwegs mit anderen Menschen telefonisch kommunizieren zu können. Auch die SMS und die später eingeführte MMS sind hauptsächlich auf Kommunikation ausgelegt. Im Laufe der Zeit entwickelte sich das Mobiltelefon zum Smartphone und so zur Multimediazentrale. Neben einem Zugang zum Internet und als Spielekonsole dient ein modernes Smartphone häufig auch für Erinnerungen und als Kommunikationszentrale. Ständige Verfügbarkeit und andauernde soziale Vernetzung sind zum Standard geworden. Das Smartphone ist ständiger Begleiter und ersetzt viele persönliche Gegenstände wie Uhren und Kalender. Durch Spiele und andere multimedialen Funktionen dient es aber auch dem Zeitvertreib und zur Freizeitgestaltung. Eltern warnen ihre Kinder regelmäßig vor zu häufigem Medienkonsum. Im Speziellen Jugendliche und Kinder, welche digitale Medien schon seit frühen Kindertagen kennen und mit ihr aufgewachsen sind, sind sich oft nicht bewusst, wie viel Zeit mit diesen verbracht wird. Für sie ist es normal, sich in ihrer Freizeit „berieseln“ zu lassen. Dadurch wird das selbstständige Denken und die Fähigkeit sich mit sich selbst Beschäftigen nicht gefördert. Das digitale „Ich“ wird gestärkt und reale soziale Beziehungen werden vernachlässigt oder erst gar nicht aufgebaut. Nicht nur junge Menschen verfallen diesen technischen „Spielereien“. Immer mehr Erwachsene verbringen jede freie Minuten mit Surfen oder Spielen am Mobiltelefon oder anderen sozialen Medien. So erobern digitale Unterhaltungsmedien Menschen, welche oftmals bereits Anzeichen von Suchtkranken aufweisen und die solchen Verhaltensmustern nur noch mit Therapie entgegenkommen können.

Mit dieser Arbeit soll das Bewusstsein für digitale Mediennutzung, im Spezialfall Mobiltelefon, gestärkt werden. Dabei soll ein Nutzer erkennen können, wie viel Zeit er mit seinem Mobiltelefon interagiert, für welche Zwecke er es nutzt und was die Motivation dahinter ist. Um diesen Effekt zu erreichen soll eine Visualisierung entworfen werden, die für einen durchschnittlichen Mobiltelefonnutzer verständlich ist. Untersucht werden soll, ob der Nutzer aufgrund der Visualisierung der Anwendungsnutzungsdaten sein Verhalten ändert. Die Visualisierung soll über den offiziellen Anwendungsdistributionskanal des Mobiltelefonbetriebssystem Herstellers verbreitet werden. Dadurch besteht die Möglichkeit einen weiten und vielfältigen Nutzerkreis zu erreichen.

Diese Arbeit beginnt in Kapitel 2 mit einem Überblick über verwandte Themen. Im 2.5. Kapitel werden anschließend einige zum Verständnis nötige Grundlagen erklärt. Fragestellungen und Ideen zur Visualisierung von Anwendungsdaten finden ihre Diskussion in Kapitel 3. Die darauf aufbauende Android-Anwendung wird im 4. Kapitel beschrieben. In den beiden darauf folgenden Kapiteln werden eine Vorstudie zur Benutzbarkeit der Anwendung

(Kapitel 5), sowie die Analyse der von ihr protokollierten Daten (Kapitel 6) vorgestellt. Das letzte Kapitel (7) fasst die Arbeit zusammen und bietet ein Ausblick, was der Inhalt weiterführender Forschung sein kann.

2 Verwandte Arbeiten und Grundlagen

Dieses Kapitel beschreibt verwandte und bisher durchgeführte Arbeiten. Sie dienen als Grundlage für die in dieser Arbeit behandelten Themen und bieten eine Übersicht, welche Forschung im Themenfeld dieser Arbeit bereits durchgeführt wurde. Die vorliegende Arbeit enthält Aspekte verschiedener Forschungsrichtungen. Bereits veröffentlichte Arbeiten, deren Betrachtung im Bezug auf die vorliegende Arbeit interessant sind lassen sich in drei Kategorien einteilen. Zum einen gibt es Arbeiten zur Nutzung von Mobiltelefonen. Wie auch die vorliegende Arbeit wurden bereits Arbeiten zum Sammeln und Analysieren von Nutzungsdaten veröffentlicht. Einen weiteren Gesichtspunkt behandeln Arbeiten zu den Themen Life-Logging und Persuasive Design. Dabei werden unter Berücksichtigung bestimmter Fragestellungen spezielle Verhaltensmuster analysiert und es wird versucht Teilnehmer von Studien aufmerksam zu machen und zu motivieren. In der vorliegenden Arbeit sollen Nutzer auf die Dauer und Art der Verwendung ihres Mobiltelefons aufmerksam gemacht werden. Die letzte Kategorie ist die Visualisierung von Nutzungsdaten im Allgemeinen. Das wird benötigt um Mobiltelefonnutzungen verständlich darzustellen.

2.1 Analyse von Mobiltelefonnutzungen

Bei der Analyse von Mobiltelefonnutzungen werden Ausführungszeiten und andere relevante Informationen, wie die Umstände unter denen die Ausführung stattgefunden hat, zu Anwendungen protokolliert und untersucht. Zu diesem Thema wurden bereits auf vielfältige Weise untersucht und viele Aspekte und Einflüsse mit einbezogen. Es wurden Erkenntnisse gewonnen, welche in der vorliegenden Arbeit berücksichtigt werden müssen. Im folgenden werden bereits veröffentlichte Arbeiten aufgeführt und ihre Ergebnisse angerissen. Falaki et al. analysieren in ihrer Arbeit das Nutzungsverhalten der Nutzer von Mobiltelefonen im Allgemeinen, wobei die Ergebnisse in die Berechnung und Verbesserung von Akku-Laufzeiten einfließen sollen. Die Arbeit zeigt, dass das Verhalten der Nutzer im Umgang mit Mobiltelefonen sehr stark variiert. So wurde mit dem Mobiltelefon im Schnitt 10 - 200 Mal am Tag interagiert. Zur Analyse einer exakten verbliebenen Akkulaufzeit muss jeder Nutzer separat beobachtet werden. Selbst im Nutzungsverhalten einzelner Nutzer wurden Abweichungen festgestellt [FMK⁺10]. In der Arbeit brachten Rahmati und Zhong diesen Umstand mit kontextabhängigen Informationen, wie dem Standort eines Nutzer, in Verbindung und haben Abhängigkeiten zur Nutzung hergestellt [RZ10]. In der Studie wird von Böhmer et al. ebenfalls das Nutzungsverhalten untersucht, jedoch beziehen diese den Kontext, in dem das Telefon benutzt wurde, mit ein. So können Rückschlüsse über

Ausführungszeiten von Anwendungen über den Tagesablauf gezogen werden. Auch standortabhängige Verwendung und die Reihenfolge in der Anwendungen verwendet wurden berücksichtigt. Das Ergebnis enthält einige interessante Erkenntnisse im Zusammenhang mit Kontextabhängigkeit. Außerdem zeigen diese Ergebnisse, dass einzelne Nutzungen meist von kurzer Dauer sind [BHS⁺11]. Zusätzlich zur Kontextabhängigkeit untersuchen Mulder et al. in ihrer Arbeit subjektive Eindrücke des Nutzers in Zusammenhang mit seiner Mobiltelefonnutzung. Damit lässt sich das Telefonnutzungsverhalten mit kontextabhängigen Informationen und Nutzungserfahrungswerten verknüpfen. Diese Erfahrungswerte sollen in medizinischen und biologischen Bereichen zum Einsatz kommen [MHK05].

2.2 Life-Logging und Persuasive Design

Beim Life-Logging und Persuasiven Design werden alltägliche Abläufe, wie das Ess- oder Bewegungsverhalten, protokolliert. Meist wird versucht ein Bewusstsein für einen bestimmten Aspekt zu vermitteln und Nutzer ein bestimmtes Verhalten anzutrainieren. Eine wichtige Rolle spielen dabei soziale Bande, durch welche oftmals ein gewisser Zwang entsteht. Im fünften Kapitel seines Buches *Persuasive Technology* beschreibt J. B. Fogg [Fog02], welche Arten des persuasiven Designs unter Nutzung von Computern angewandt werden können. Mögliche persuasive Designs seien: Physisch, psychologisch, mittels Sprache und mittels sozialer Dynamik und Rollen. Bei der physikalischen Form werden Menschen beispielsweise durch unterstützende und aufmerksame fiktive Zuschauer, wie Animationen, zum Weitermachen ermutigt. Im Rahmen des psychologischen Ansatzes wird versucht, das Gerät mit Gefühlsregungen auszustatten und diese den Nutzer erfahren zu lassen. Mittels Sprache kann ein Nutzer in geschriebener oder gesprochener Form motiviert werden. Ebenfalls ist es möglich, Effekte durch ein soziales Bewusstsein im Umgang mit und sozialen Druck durch anderen Menschen zu erzielen.

Durch eine Arbeit von Gartland und Piasek werden Nutzer auf die Menge ihres produzierten Abfälle aufmerksam gemacht. Das Gerät *Weight Your Waste* [GP09] unterstützt dabei und zeigt die Masse des Abfalls sowie die durch den Abfall verursachten Kosten an. Es dient außerdem als Informationsportal für grüne Techniken wie Recycling und Kompostieren. Das Umweltbewusstsein der Nutzer soll dabei gestärkt werden. Eine Arbeit von Thieme et al., hat das selbe Anliegen. Eine entwickelte *BinCam* [TCM⁺12] soll zum Hinterfragen des Wegwerfverhaltens der Teilnehmer dienen. Ein am Mülleimer befestigtes Mobiltelefon fotografierte entsorgte Gegenstände. Aufgenommene Bilder wurden im sozialen Netzwerk *Facebook* öffentlich gemacht. Es wurde darauf geachtet, den gewohnten Alltag nicht zu verändern. Diese Studie zeigt ein gesteigertes Interesse und Bewusstsein der Teilnehmer am Entsorgungsprozess und der Essensplanung, neigt sie dazu einzelne Teilnehmer bloßzustellen, da die *BinCam*-Bilder in einem sozialen Netzwerk veröffentlicht werden. Mit der Studie verfolgen Arroyo et al. [ABS05] ein ähnliches Konzept. *Waterbot* untersucht jedoch nicht das Wegwerfverhalten, sondern den Wasserverbrauch von Spülbecken und zeigt diesen auf einem digitalen Display an. Es findet, im Gegensatz zur Arbeit von Thieme et al. keine öffentliche Bloßstellung statt. Der Verbrauch wird durch die Anzeige ausschließlich dem

Nutzer zur Verfügung gestellt.

In ihrer Arbeit wollen Arteaga et al. einer steigenden Fettleibigkeit bei Jugendlichen entgegenwirken. Dies geschieht durch Nutzung eines ständigen Begleiters der Jugendlichen, des Mobiltelefons, und den dafür bereits vorhandener Anwendungen. Jugendliche werden mit Spielen wie Geo-Caching, ein Spiel mit Bewegung ähnlich einer Schatzsuche, oder Beschleunigungsmessungsspiele spielerisch motiviert. Am besten funktioniert dies in Wettkampfsituationen und durch gutes Zureden und Loben [AKWK10]. Auch Toscos et al. beschäftigen sich mit der fehlenden Bewegung unter Jugendlichen. Dabei legen sie ihr Hauptaugenmerk auf jugendliche Mädchen. Es wird versucht, diese über soziale Kontakte zur Bewegung zu motivieren. Mit *Chick Clique* [TFAGo6] sollen Mädchen durch eine Art privates soziales Netzwerk, welches vergangene Schritte protokolliert und den anderen Teilnehmern zugänglich macht, zu gegenseitigen Motivatoren werden. Hierdurch war es möglich das Laufverhalten der Teilnehmerinnen zu verändern und das Bewusstsein für Gesundheit zu stärken. Ein ähnliche Studie wurde von Consolvo et al. durchgeführt. Mit *Houston* wurde den Teilnehmern (ebenfalls meist Frauen) einer Studie ein Journal für das Mobiltelefon bereitgestellt, mit dessen Hilfe die eigene körperliche Bewegung gesteigert werden kann. Dafür werden körperliche Aktivitäten protokolliert und Freunden zugänglich gemacht. Generell müssen vier Grundregeln beachtet werden: Nutzer müssen belohnt werden, das eigene Aktivitätslevel muss ihnen bewusst sein, soziale Verknüpfungen sind nötig und *Houston* muss in den Alltag der Nutzer eingebettet sein [CESLo6]. Mit *MOPET* [BCo8] wurde versucht ein System zu bauen, welches in Trainingskleidung integriert ist und es schafft beim Sport zu motivieren. Der Nutzer kann eigene Trainingsziele vorgeben. Unterstützt wird er von dreidimensionalen Animationen und einer Sprachausgabe.

2.3 Konzepte der Visualisierung

Ein zentraler Aspekt der vorliegenden Arbeit ist Visualisierung protokollierter Daten. Zu diesem Thema wurden bereits viele Arbeiten geschrieben. Es folgt eine Auswahl. Orso et al. präsentieren die Technologie *Gammatella* [OJHo3]. Sie ist der Versuch, auftretende Fehler nach der Integration einer neuen Software zu verringern. Zur Visualisierung kamen unter anderem TreeMaps zum Einsatz. Ein wichtiger Bestandteil jeder verwendeten Visualisierung war das Nutzen von Farben, um die Unterscheidung und Einordnung einzelner Elemente zu erleichtern.

Daten, welche periodische und serielle Muster aufweisen, sind schwer darzustellen, wenn beide Aspekte herausgearbeitet werden sollen. In einer Arbeit von Carlis et al. [CK98] beschreiben die Autoren einen Ansatz, der auf der Verwendung von spiralförmigen Diagrammen beruht. Hierbei werden auf den Spiralachsen serielle und auf den Radien periodische Informationen dargestellt. Diese Studie wurde hauptsächlich unter Biologen durchgeführt. Dabei wurde herausgefunden, dass diese Art von Diagrammen von Nutzern schnell verstanden wird und zum Nachdenken anregt.

2.4 Fazit der Verwandten Arbeiten

Aus den aufgeführten Veröffentlichungen konnten Erkenntnisse gewonnen werden, welche im Rahmen dieser Arbeit Anwendung fanden. Das Nutzungsverhalten an Mobiltelefonen variiert sehr stark ([FMK⁺10]). So muss eine sehr flexible Darstellung der Daten gefunden werden, um jedem Nutzer gerecht zu werden. Einzelne Mobiltelefonnutzungen erstrecken sich oft nur über einen recht kurzen Zeitraum. Das Darstellen jeder einzelnen Ausführung ist daher schwierig und meist nur schwer zu interpretieren. Dies führt dazu, dass Ausführungen gruppiert werden sollten. Diese Arbeit möchte neben der Erhebung von rein objektiven Nutzungsdaten diese dem Nutzer zur Selbstreflexion darstellen und prüfen, ob er sich beeinflussen lässt.

Aus den aufgeführten Arbeiten wurde klar, dass Menschen sich in ihrem Verhalten beeinflussen lassen. Sie brauchen jedoch spezielle Motivatoren, welche sie in gewisser Weise zwingen oder ihnen Tatsachen vor Augen führen. Oftmals ist es auch nötig, explizite Belohnungen einzuführen. Neben bereits bekannten Faktoren wie der spielerischen Motivation [AKWK10] wurden in den Arbeiten von Toscos et al. [TFAGo6] und Thieme et al. [TCM⁺12] persuasive Methoden in Form der sozialen Netzwerken erfolgreich eingesetzt. In der vorliegenden Arbeit wird versucht, Nutzer durch die Schaffung eines Bewusstseins bezüglich der Häufigkeit der Mobiltelefonnutzung zu sensibilisieren. Die Veröffentlichung von persönlichen Informationen soll der Nutzer auf Wunsch selbst vornehmen können.

Farbige und Interessante Darstellungen führen nicht nur zu mehr Spaß beim Benutzen, sondern machen es auch einfacher verschiedene Elemente zu unterscheiden. Das soll in dieser Arbeit zur Unterscheidung verschiedener Anwendungen genutzt werden.

2.5 Grundlagen

Im Folgenden werden einige Grundlagen und Begrifflichkeiten erklärt, welche im Verlauf der Arbeit genutzt werden. Technologien werden dabei nicht komplett erläutert, sondern nur Besonderheiten, welche sich auf diese Arbeit beziehen, erklärt.

Android

Das Mobiltelefon-Betriebssystem *Android* zählt neben *Symbian* und *Apples iOS* zu den Systemen mit dem größten Marktanteil und den höchsten Verkaufszahlen [BCNB12, Gar11]. Anwendungen, welche für eine der beiden Plattformen geschrieben werden, haben somit die Möglichkeit weit verbreitet zu werden. Außerdem bietet Android neben freien Entwicklertools auch die Möglichkeit, entwickelte Anwendungen kostenlos über den offiziellen Anwendungsdistributionskanal *GooglePlay* zu vertreiben. Diese Gründe qualifizieren Android als optimale Grundlage um in einem breiten Nutzerfeld Studien durchzuführen.

PHP und MySQL

Die serverseitig ausführbare Skriptsprache PHP¹ eignet sich in besonderer Weise für Zugriffe von beliebigen Anwendungen und Geräten auf einen zentralen Server mit Datenbank. Aufgrund ihrer Konzeption für Webseiten können PHP-Skripte über ein Netzwerk oder das Internet angesprochen werden. Einem Aufruf können Parameter angehängt werden, deren Inhalte weiterverarbeitet und per MySQL in eine Datenbank geschrieben werden. MySQL² ist ein relationales Datenbanksystem, das einen verbreiteten Einsatz im Bereich der dynamischen Webauftritte findet. Es kann mithilfe von PHP direkt angesprochen werden. Zugangs- und Anmeldedaten müssen hierfür im PHP-Skript hinterlegt sein. PHP bietet ebenfalls sogenannte Prepared-Statements, welche übergebene Variablen vor dem Schreiben in eine Datenbank auf schädlichen Datenbank-Code prüfen.

Protokollieren von Nutzungsdaten

Für diese Bachelorarbeit werden protokollierte Daten der Ausführungszeiten einzelner Anwendungen benötigt. Dafür wird für jede Ausführung einer Anwendung der Start- und Endzeitpunkt der Ausführungen zusammen mit einem eindeutigen Identifikator der Anwendung sowie des Mobiltelefons abgespeichert. Dabei ist es wichtig, dass nur Zeiten berücksichtigt werden, in denen der Bildschirm aktiv, das Mobiltelefon also in Benutzung war. Anwendungen deren Nutzung nicht das Ziel der Mobiltelefonnutzung ist werden, zwar protokolliert, jedoch speziell markiert und in dieser Arbeit nicht verwendet. Ein Beispiel hierfür ist die Launcher-Anwendung, aus welcher andere Anwendungen gestartet werden. Extra protokolliert wird außerdem die Nutzung der im Rahmen dieser Arbeit entwickelten Anwendung. Dabei wird notiert, in welchen Zeiträumen welche Ansicht auf welchem Mobiltelefon angezeigt wird. Diese Information wird benötigt, um einsehen zu können, inwiefern die Anwendung verwendet wird.

Zeiteinheit

Einzelne Anwendungsnutzungen auf Mobiltelefonen sind im Allgemeinen eher von kurzer Dauer [BHS⁺11]. Um die Visualisierung solcher Daten einfacher und übersichtlicher zu gestalten, werden Programmausführungen gruppiert. Als Zeiteinheit wird der Zeitraum genannt, über welchen hinweg Anwendungsnutzungen zusammengefasst werden. Im Rahmen dieser Arbeit werden die Zeiteinheiten *Tag*, *Woche* und *Monat* verwendet. Jede Gruppe der Zeiteinheit wird als Zeitraum oder als Zeitraum einer Zeiteinheit bezeichnet. Ein Beispiel für die Zeiteinheit *Tag* ist „27.09.2012“, für *Monat* „September“. Eine weitere Möglichkeit ist die Gruppierung nach einem periodischen Muster, beispielsweise das zusammenfassen aller Ausführungen, die morgens zwischen 8 und 10 Uhr stattgefunden haben.

¹PHP-Webseite: <http://php.net/> zuletzt geprüft am 18.12.2012

²MySQL-Webseite: <http://www.mysql.com/> zuletzt geprüft am 18.12.2012

3 Visualisierungskonzept

Das Visualisierungskonzept, das in diesem Kapitel entwickelt wird, soll Nutzer zur Reflexion ihres eigenen Anwendungsnutzungsverhalten motivieren. Hierfür muss die Visualisierung eine Übersicht über die durch den Nutzer genutzte Anwendungen bieten. Die Visualisierung soll bereits gesammelte Anwendungsnutzungsdaten darstellen und den Nutzer dadurch befähigen, sein vergangenes Nutzungsverhalten zu überblicken und zu hinterfragen. Das Vorgehen zur Entwicklung des Konzepts und dessen zentrale Aspekte werden in diesem Kapitel erläutert.

Dargestellt werden sollen Informationen, mit deren Hilfe Aussagen über die Häufigkeit und die Dauer der Mobiltelefonnutzung getroffen werden können. Dabei ist es wichtig, dass auch die Ausführungszeiten einzelner Anwendungen, sowie der Zeitraum der Ausführung erkennbar sind. Diese Informationen deuten darauf hin, aus welchen Gründen eine Anwendung benutzt wurde und ob zum Beispiel eine Übersetzungsanwendung zum Bearbeiten der Hausaufgaben oder ein Spiel zum Überbrücken der Abendstunden der Nutzungsgrund war.

Die dargestellten Informationen müssen einfach und schnell erfassbar sein, dies soll Nutzer zu einer aktiven Verwendung der Visualisierung bewegen. Aus der Darstellung muss ohne Erklärungen hervorgehen, was die dargestellten Daten bedeuten. Diese grundlegenden Fragestellungen müssen bei der Visualisierung beachtet werden: Wie kann man sich schnell und intuitiv einen Überblick über gesammelte Daten verschaffen? Welche Darstellungsarten eignen sich in besonderer Weise? Was lässt sich auf Mobiltelefonen mit eher kleinen Ausgabebildschirmen noch gut erkennen?

3.1 Relevante Daten

Für die Visualisierung steht jede einzelne Ausführung einer Anwendung mit Namen und eindeutiger Anwendungs-ID, sowie Start- und Endzeitpunkt der Ausführung zur Verfügung. Anwendungen, deren Ausführung nicht der eigentliche Sinn der Mobiltelefonnutzung (zum Beispiel den Launcher - eine Anwendung, die eine Übersicht über ausführbare Anwendungen bietet) ist, werden nicht berücksichtigt. Die in dieser Arbeit verwendeten Daten stammen aus der Protokollierungsdatenbank der Anwendung *AppTicker* (beschrieben in 4.1). Grundsätzlich verwenden alle Darstellungen nur die innerhalb einer Zeiteinheit (beispielsweise ein Tag oder eine Woche) einer Anwendung aufaddierte Dauer. Der Referenzzeitpunkt ist immer der Ausführungsbeginn.

3.2 Art der Darstellung

Prinzipiell gibt es verschiedene Möglichkeiten Informationen darzustellen. Jede Art hat ihre vom Kontext abhängigen Vor- und Nachteile. In dieser Arbeit wurden drei Ansätze analysiert und verglichen.

Prosaform Die Darstellung der Anwendungsnutzungen in Prosaform bedeutet, dass in einem Fließtext alle Anwendungsausführungen in einem bestimmten Zeitraum mit Nennung von Uhrzeit und Dauer aufgeführt werden. Zahlen und Messdaten in einer reinen Prosaform lassen sich angenehm lesen und dienen dem einfachen Verständnis. Fließtext lässt sich generieren, solange er gewissen Regeln und Mustern folgt. Um einen Text zu lesen und zu verstehen wird eine gewisse Zeit benötigt. So ist es nicht möglich, sich einen schnellen Überblick über gesammelte Daten oder einen bestimmten Datenpunkt zu verschaffen. Auch Datereihen lassen sich ohne weiteres nicht direkt vergleichen.

Tabellarisch Bei der tabellarischen Form werden beispielsweise alle Anwendungen gegen eine bestimmte Zeiteinheit aufgetragen und die gesamte Dauer der Anwendungsausführung für eine Zeiteinheit eingetragen. Solch eine Tabelle lässt sich mit wenig Aufwand aufbauen. Die dargestellten Informationen lassen sich schnell erfassen. Es ist möglich, längere Datenreihen in einer Tabelle schnell zu vergleichen und Schlüsse zu ziehen. Tabellen neigen dazu, nicht sofort und intuitiv Auskunft über Daten geben zu können und sind aufgrund fehlender Abhebung einzelner Information oft anstrengend zu lesen. Bei ansteigender Größe eines Datensatzes wird dieser unübersichtlicher und das schnelle Orientieren in der Tabelle immer schwerer.

Diagramm Ähnlich der tabellarischen Darstellung werden Nutzungsdaten auch in Diagrammen zusammengefasst. Sie werden nicht als Zahlenwert, sondern in Formen, Größen und Farben visualisiert. Daten lassen sich graphisch mit Hilfe von Diagrammen gut darstellen. Ähnlich wie bei Tabellen sind Datenwerte in Grafiken für Menschen schnell erfassbar. Bei Diagrammen werden wichtige Aspekte durch die farbliche Gestaltung, unterschiedliche Formen und das Skalieren graphischer Elemente besser herausgearbeitet. Die graphische Darstellung ist sehr flexibel und kann auf vielerlei Art und Weise eingesetzt werden. Es muss eine geeignete Darstellung gefunden werden, die ein sofortiges Verständnis des Betrachters gewährleistet. Das Erstellen von Diagrammen kann einen hohen Programmieraufwand haben, aber auch die spätere Berechnung durch ein Mobiltelefon ist im Vergleich zur Berechnung von Tabellen und Texten aufwendig.

Fazit: Darstellungsart Die Darstellung geschieht in Form eines Diagramms. Die Vorteile der visuellen Hervorhebung durch Farben und Formen sowie die Nutzung von bekannten visuellen Merkmalen, wie die eines Anwendungssymbols, überwiegen den benötigten Aufwand zur Berechnung eines Diagramms. Dem erhöhten Aufwand kann durch eine effiziente Implementierung entgegenwirkt werden.

3.3 Diagrammtypen

Die Art und Weise, wie Daten in einem Diagramm dargestellt werden, entscheidet über die Aussage und Aussagekraft eines Diagramms. Es soll für jede Anwendung die gesamte Ausführungszeit in einem Zeitraum ersichtlich sein. Um die geeignetste Darstellungsart zu ermitteln, wurden verschiedene Ideen erstellt und dann iterativ weiterentwickelt.

Iteration 1: Einfache Diagrammtypen

Im ersten Anlauf lag das Hauptaugenmerk auf einfachen zweidimensionalen Diagrammen (2D-Diagrammen).

In einem Kreisdiagramm steht jeder Kreis für einen Zeitabschnitt. Die in solch einem Abschnitt verwendeten Daten können in Form der Kreisausschnitte miteinander verglichen werden. Es ist gut abschätzbar, welche Anwendung die meiste Zeit genutzt wurde und wie die Nutzungsdauer einzelner Anwendungen im Verhältnis stehen.

Ein Liniendiagramm trägt die aufaddierte Ausführungsdauer gegen deren Zeiteinheit auf. Die verschiedenen Anwendungen werden durch Linien in unterschiedlichen Farben repräsentiert. Anwendungen lassen sich im Liniendiagramm sehr gut vergleichen. Bei einer großen Zahl von Anwendungen wird dieses Diagramm schnell unübersichtlich, da das Diagramm auf einem kleinen Display zu voll wird und zu wenige gut unterscheidbare Farben zur Verfügung stehen.

Eine weitere Idee ist das Häufigkeitsdiagramm (Abbildung 3.1). Es wird dargestellt, wie häufig eine Anwendung genutzt wurde. Jede Zeiteinheit bekommt einen Balken. Je länger eine Anwendung ausgeführt wurde, umso weniger transparent ist der Balken. Diese Darstellung dient vor allem dazu, zyklische Ereignisse (zum Beispiel: Montagmorgen) zu untersuchen. Es setzt somit nicht den gewünschten Fokus.

Da davon auszugehen ist, dass Anwender eine größere Zahl an Anwendungen (>10) nutzen, fällt das Liniendiagramm aufgrund mangelnder Übersichtlichkeit weg und wird nicht weiterentwickelt. Das Häufigkeitsdiagramm fällt ebenso weg, da es nicht die gewünschten Aspekte darstellt.

Iteration 2: Erweiterung in die Dreidimensionalität (3D)

In der zweiten Iteration wurde, neben der Verbesserung des Kreisdiagramm-Konzepts, eine auf Säulendiagramme basierende Idee durchdacht. Es wurden dreidimensionale Ansätze ausprobiert, um alle Informationen darstellen zu können.

In der Verbesserung des Kreisdiagramms wurde das Diagramm aus der ersten Iteration 1 übernommen. Je Zeitraum wurde ein Kreis erstellt, damit mehrere Zeiträume dargestellt

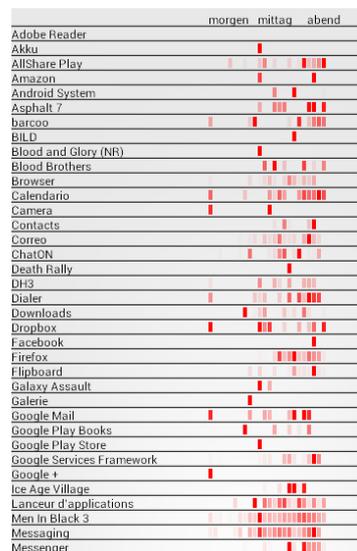


Abbildung 3.1: Das Diagramm stellt die Häufigkeit einer Anwendungsnutzung über den Tagesverlauf dar. Dabei werden Nutzungen aus mehreren Tagen aufaddiert. Je intensiver die Farbe des roten Strichs, desto länger war die Benutzung zu diesem Zeitraum.

werden können. Diese Kreise werden übereinander gestapelt, sodass diese einen Zylinder bilden. Die aufgeschichteten Kuchendiagramme (Abbildung 3.2) können nacheinander durchgescrollt werden. Diese Darstellung impliziert eine 3D-Ansicht.

Einen anderen Diagrammtypen stellen Säulendiagramme dar. In dieser Ansicht (Abbildung 3.3) gibt es drei Achsen. Auf der y-Achse wird die Zeiteinheit (Tag, Woche etc) aufgetragen, auf der x-Achse die verschiedenen Anwendungen und auf der z-Achse die Dauer der Ausführung. Aus dieser Aufteilung erhält man eine Anordnung mit einzelnen Türmen (ähnlich der Vogelperspektive einer Großstadt), wobei jeder Turm eine Anwendung darstellt. Die Stärke dieser Ansicht ist eine gute grobe Übersicht. Man erkennt auf den ersten Blick, was eher häufig benutzte Anwendungen sind und welche nahezu nie verwendet werden. Auch Tendenzen, wann welche Anwendung bevorzugt benutzt wird, lassen sich aus dieser Ansicht ableiten. Ausführungszeiten können hier innerhalb einer Anwendung, aber auch innerhalb einer Zeiteinheit gut anhand der größer der Türme verglichen werden. Die Ausführungen von Anwendungen die nur kurz benutzt werden, kommen nicht zur Geltung und sind nur in der Einzelansicht erkennbar, bei der ausschließlich eine Anwendung oder eine Zeiteinheit dargestellt werden. Eine mögliche Verbesserung könnte durch Vergrößern und passende Skalierung für bestimmte Anwendungen oder Zeiträume erreicht werden.

Um ein Diagramm im dreidimensionalen Raum darstellen zu können, sind aufwändigere und kompliziertere Berechnungen nötig. Dies macht die Programmierung und die Darstellungsberechnung aufwändiger. Ein weiteres Problem stellt hier das manchmal eher kleine Display eines Mobiltelefons dar. 3D-Darstellungen wirken wenn sie auf kleine Flächen ska-

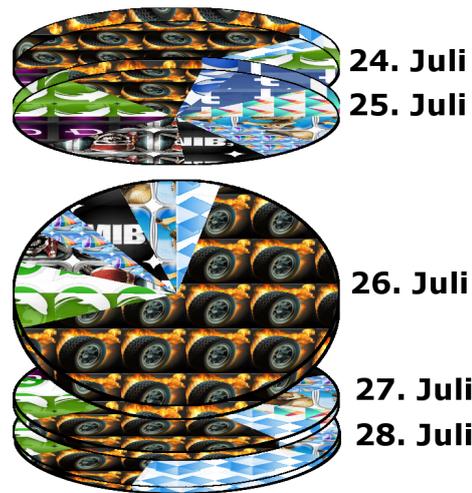


Abbildung 3.2: Dreidimensionales Kreisdiagramm. Jeder Kreis entspricht einem Tag. Die Größe der Kreisausschnitte stellt die Ausführungsdauer einer Anwendung dar.

liert sind schnell unübersichtlich. Aus diesen Gründen fallen Diagramme in 3D-Darstellung weg. Das Säulendiagramm wird in der dritten Iteration weiterentwickelt.

Iteration 3: Vereinfachungen ins Zweidimensionale

Aufgrund der in Iteration 2 beschriebenen Nachteile wurden ab dieser Iteration nur noch 2D-Diagramme verfolgt. Dabei ist zu beachten, dass trotzdem drei Dimensionen (Anwendung, Ausführungszeitraum und Ausführungsdauer) dargestellt werden müssen.

Die in dieser Iteration entwickelte Idee basiert erneut auf Säulendiagrammen, jedoch in 2D-Darstellung. Das Diagramm besteht aus zwei Achsen. Auf der x-Achse werden die Zeiträume der gewählten Zeiteinheit aufgetragen. Die Ausführungszeit einer Anwendung wird als Säule entlang der y-Achse dargestellt. Bei mehreren Anwendungen je Zeitraum werden die Anwendungen übereinander gestapelt. Innerhalb eines Zeitraums lassen sich Anwendungen nun vergleichen. Ein Großteil der Ausführungen ist gut sichtbar. Die skalierten Säulen führen jedoch zu einem falschen Eindruck. Da jede Säule separat auf die gesamte Höhe skaliert wird, können die Größenverhältnisse auf zwei unterschiedlichen Säulen nicht verglichen werden, somit ist ein Vergleich zwischen zwei Zeiträumen nicht möglich. Auch der 3D-Effekt der Säule wurde entfernt, da dieser das Diagramm unnötig überlädt und unübersichtlicher macht (Abbildung 3.4).

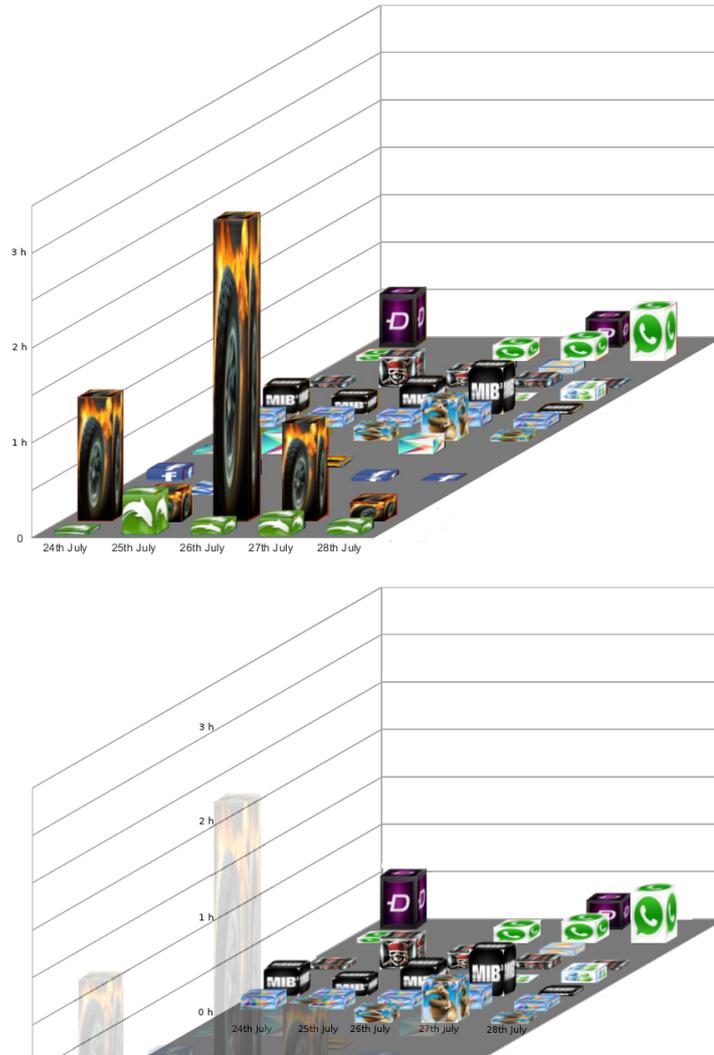


Abbildung 3.3: Dieses dreidimensionale Diagramm stellt die Ausführungsdaten aus einer vogelperspektiven-ähnlichen Visualisierung dar. Die horizontale Achse beschreibt den Tag der angezeigten Ausführungen, die vertikale die aufaddierte Dauer der Ausführungen pro Anwendung an einem Tag und die Achse in die Tiefe um welche Anwendung es sich handelt. Oben: Initialer Zustand. Unten: Nach hineinscrollen.

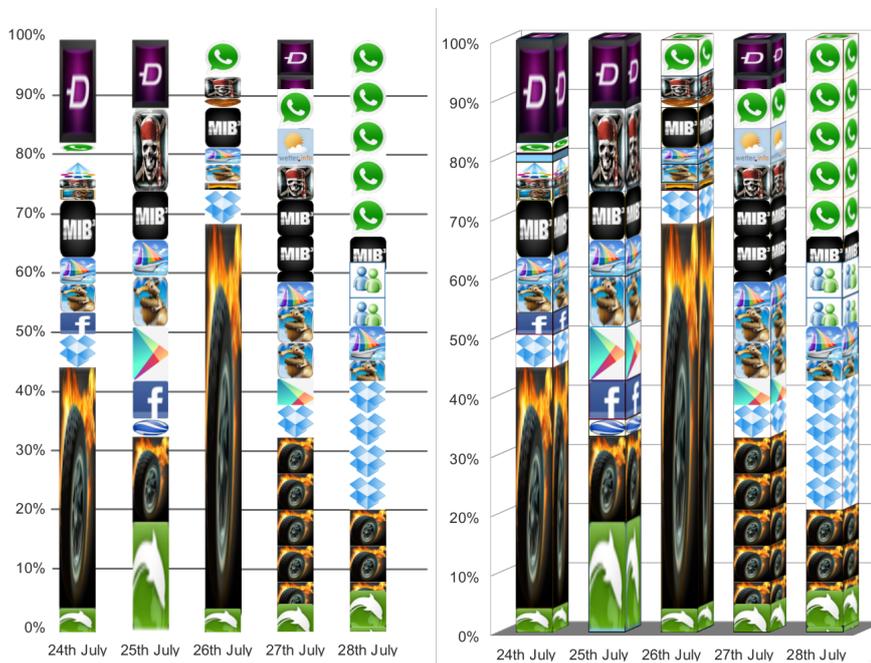


Abbildung 3.4: Vergleich verschiedener Darstellung eines Säulendiagramms. Links: Einfache (zweidimensionale) Darstellung eines Säulendiagramms. Rechts: Dreidimensionale Darstellung eines Säulendiagramms. Die einfache Darstellung wirkt aufgeräumter und übersichtlicher. Insbesondere ist sie auf einem kleineren Display besser erkennbar.

Iteration 4: Wiederherstellen der Vergleichbarkeit

In der letzten Iteration wurde das Säulendiagramm aus der dritten Iteration erneut überarbeitet.

Die Achsen des Diagramms tragen Daten wie in Iteration 3 beschrieben ebenfalls gestapelt auf. Die erste Verbesserung stellt die Tatsache dar, dass die Daten der einzelnen Zeiträume nicht länger auf jeweils 100% skaliert werden. Der Zeitraum mit der längsten Ausführungszeit wird als Referenz für alle anderen Zeiträume verwendet. Durch teilweise starke Abweichungen der Größe einzelner Säulen und die entfernte Skalierung, kommen die Anwendungen, welche lediglich über einen kurzen Zeitraum genutzt wurden, nicht zur Geltung. Auf Mobiltelefonen kann diesem Effekt mithilfe eines durch Multitouch unterstützten Skalierens und Verschiebens des Bildschirminhalts entgegengewirkt werden. Des Weiteren wird das Diagramm um 90 Grad gedreht, sodass die Ausführungszeiten von rechts nach links laufen und einem Balkendiagramm entsprechen.

Zusammenfassung

Das Darstellen von Nutzungsdaten beinhaltet drei Größen. Zum einen muss ersichtlich sein, welche Anwendung beschrieben wird. Des Weiteren müssen zwei Zeiteinheiten dargestellt werden können: Die gesamte Dauer, die eine Anwendung ausgeführt wurde, und die Zeiteinheit, in der die Ausführungen stattfanden. Daraus ergibt sich im Allgemeinen ein Diagramm mit drei Achsen. Auf eine 3D-Darstellung wurde verzichtet, da deren Implementierung rechenintensiver ist und auf einem kleinen Bildschirm schnell zur Unübersichtlichkeit führt. Durch Stapeln der Anwendungen lässt sich ein solches Diagramm in 2D überführen. Als Diagrammtyp wird das 2D-Balkendiagramm verwendet, ohne eine Skalierung auf 100% durchzuführen. Die Achse mit der Ausführungszeit kann skaliert werden, wodurch kurze Ausführungszeiten besser sichtbar werden. Jede Anwendung erhält durch Berühren eine Ansicht, in der dargestellt wird, wann diese Anwendung in der gewählten Zeiteinheit ausgeführt wurde. Vorgesehene Zeiteinheiten sind Tag, Woche und Monat.

Durch nahezu alle Diagrammarten, welche dem Vergleich von Nutzungsdaten dienen, sind starke Unregelmäßigkeiten in den Ausführungszeiten nicht zu kompensieren und machen es schwierig eine geeignete Darstellung zu finden. So kann es vorkommen, dass Daten einer längeren Ausführung alle anderen Daten stark übertreffen und diese kaum sichtbar oder zumindest schwer einschätzbar sind. Der Versuch, die Daten durch beispielsweise eine logarithmische Darstellung anzupassen, scheitert daran, dass diese dadurch nicht mehr einfach zu lesen sind und die schnelle Auffassung ohne weiteres nicht möglich ist.

3.4 Beschriftung

Innerhalb einer Zeiteinheit werden einzelne Ausführungen aufaddiert. Dies verhindert nicht, dass die Ausführungszeit einzelner Anwendungen im Vergleich zu anderen Anwendungen sehr kurz ist. Anwendungen mit kurzer Ausführungszeit nehmen daher in einem Diagramm nur wenig Platz ein. Aufgrund dessen führt die Beschriftung jeder einzelnen Einheit schnell zur Unübersichtlichkeit und verwirren den Betrachter. Anwendungen werden deshalb nicht beschriftet, sondern durch das Icon der Anwendung repräsentiert. Im Allgemeinen führen Repräsentationen durch Icons nicht unbedingt zum besseren und schnelleren Erkennen [Mac92], jedoch wirken sie unterstützend. Anwendungen und deren Icons müssten beim Anwender generell bekannt sein, da der diese regelmäßig benutzt.

3.5 Darstellung der Anwendungssicons

Icons haben meist ein quadratisches Format. Die Darstellung der Ausführungsdauer auf beispielsweise einem Balkendiagramm ergibt in der Regel keine quadratische Fläche. Icons müssen somit skaliert oder abgeschnitten werden. Beim Skalieren wird das Icon es auf die maximalen Ausmaße der Fläche skaliert und daher unter Umständen stark, möglicherweise

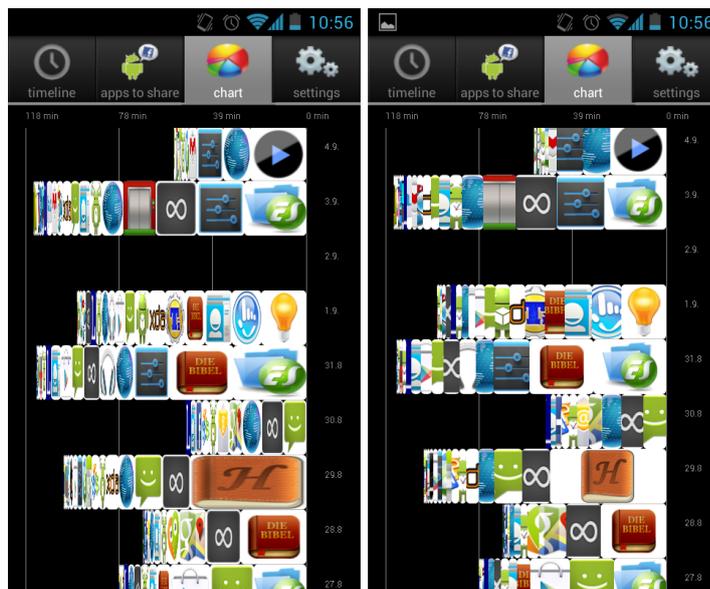


Abbildung 3.5: Vergleich gestreckter mit abgeschnittener Icons. Links: Icons gestreckt. Rechts: Icons abgeschnitten und zentriert. Bei gestreckten Icons können diese bis zur Unkenntlichkeit verzerrt werden, wohingegen bei abgeschnittenen Icons wichtige Erkennungsmerkmale verloren gehen können. Die verwendeten Icons müssen dem Nutzer in beiden Fällen bekannt sein.

bis zur Unkenntlichkeit, verzerrt (Abbildung 3.5). Hingegen werden Icons beim Abschneiden in die feste Höhe eines Querbalken im Diagramm skaliert und in der dynamischen Breite zentriert, dabei werden zu breite Icons abgeschnitten. Es kann passieren, dass das gezeigte Icon nur noch aus einem schmalen Ausschnitt des ursprünglichen Icons besteht und schwer zu erkennen ist (Abbildung 3.5).

Icon-Umfrage

Zur Identifikation der geeigneteren Darstellung der Icons wurde eine Expertenbefragung durchgeführt. Dabei wurde den Teilnehmern abwechselnd Visualisierungen mit abgeschnittenen und gestreckten Icons auf echten Nutzungsdaten gezeigt. Bei der Auswahl der Darstellungsart wurden die Visualisierungsexperten gebeten, einerseits auf das Aussehen, aber vor allem darauf zu achten, wie ein Anwendungsicon besser erkannt werden kann.

Personenkreis Es wurden sieben Mitarbeiter des Instituts für Visualisierung und Interaktive Systeme der Universität Stuttgart und nicht-fachfremde Studenten anhand der Visualisierung echter Anwendungsausführungsdaten befragt. Dabei konnte die Bekannt- und Vertrautheit der Icons nicht berücksichtigt werden.

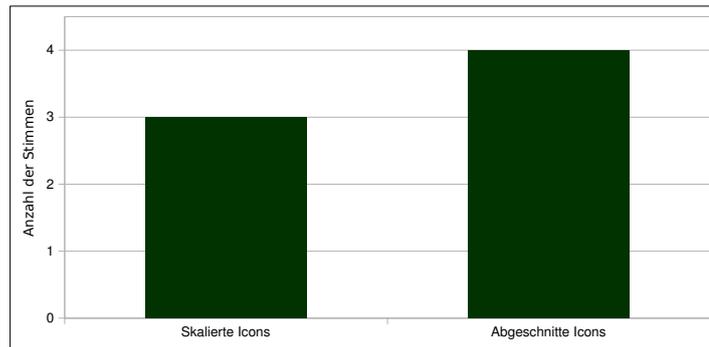


Abbildung 3.6: Ergebnis der Studie bezüglich der Behandlung überlappender Icons. Drei (43%) der sieben Teilnehmer sprachen sich für die Skalierung und Verzerrung der Icons aus, vier (57%) für das Abschneiden der Icons.

Ergebnis Es sprachen sich 57% für abgeschnittene und zentrierte Icons aus. Dieses äußerst knappe Ergebnis zeigt, dass es schwierig ist die Frage nach der Darstellungsart der Icons eindeutig zu beantworten. Bei dieser Entscheidung spielen Geschmack, sowie die Bekanntheit der Icons eine Rolle. Ebenfalls ist zu beachten, dass diese Umfrage mit sieben Teilnehmern nicht repräsentativ ist und daher nur eine Tendenz angibt.

3.6 Zusammenfassung

In der Visualisierung sollen nach Anwendung und Zeitraum sortierte Daten dargestellt werden. Zur Darstellung werden Diagramme verwendet, welche durch Darstellung in Form und Farbe die beste Übersicht bieten. Die Wahl des Diagramms fällt auf ein Balkendiagramm. Bei der Darstellung des Diagramms wird auf spielerische und graphisch aufwändige Elemente (wie eine 3D-Darstellung) verzichtet. Um trotzdem alle drei Dimensionen (Anwendung, Zeitraum und Dauer) darstellen zu können, werden die Anwendungen gestapelt. Um eine bessere Übersicht zu erhalten, werden die Balken des Diagramms nicht beschriftet, sondern mit Anwendungsicons markiert.

Im nun folgenden Kapitel werden die in dieser Arbeit vorgenommene Implementierung und Fragen zum Implementierungsdesign erläutert.

4 Anwendung

Im vorangegangenen Kapitel 3 wurde ein Visualisierungskonzept entwickelt, das dazu dient, Anwendungsnutzungsdaten zu überblicken. Im Folgenden wird beschrieben, welche Möglichkeiten bestehen, eine solche Visualisierung auf Android-Mobiltelefonen zu realisieren und welche Probleme dabei auftreten können. Um Änderungen im Nutzungsverhalten feststellen zu können, wird die Nutzung der Visualisierung protokolliert. Es ist außerdem möglich, angezeigte Diagramme zur späteren Ansicht oder zum Veröffentlichen auf *Facebook* zu exportieren. Die Visualisierung und Protokollierung findet in zwei Anwendungen Verwendung. Zum einen ist sie Bestandteil der Anwendung *AppTicker*, entwickelt vom Institut für Visualisierung und Interaktive Systeme der Universität Stuttgart, und der - in dieser Arbeit entwickelten - Anwendung *HistApp*.

Android-Anwendungen werden grundsätzlich in Java implementiert. Hinzu kommt XML für das Anordnen statischer Oberflächen. Für die Schnittstelle zwischen der Anwendung und dem zentralen MySQL-Server wurde PHP benötigt.

4.1 *AppTicker*

Als Grundlage für die in dieser Bachelorarbeit entwickelte Anwendung dient die bereits bestehende Anwendung *AppTicker*¹. Sie wurde vom Institut für Visualisierung und Interaktive Systeme der Universität Stuttgart entwickelt. Die Idee dieser Anwendung ist es, andere Menschen über Anwendungsinstallationen, -updates, -deinstallationen und -nutzungen zu informieren. Dies geschieht über das soziale Netzwerk *Facebook*. Dort werden die erwähnten Ereignisse als Meldungen veröffentlicht und Freunde des Nutzers können diese einsehen. Wenn ein Nutzer Informationen über eine bestimmte Anwendung oder Installationsänderung nicht veröffentlichen möchte, kann er dies in den Optionen der Anwendung *AppTicker* einstellen. Informationen zu einzelnen Ausführungen werden nicht nur veröffentlicht, sondern auch lokal und zentral auf einem Server der Universität Stuttgart gespeichert. Zur zentralen Speicherung werden bei jeder Ausführung relevante Daten mittels Internet und einer PHP-Schnittstelle an den Server übertragen. Des Weiteren ermöglicht es *AppTicker*, den Verlauf der verwendeten Anwendungen als Live-Hintergrund zu verwenden. Die Anwendung besteht aus drei Registerkarten. Die erste bietet für Nutzer die Möglichkeit, den Verlauf der genutzten Anwendungen als einfache, chronologisch geordnete Liste mit Zeitpunkt und

¹AppTicker: <http://projects.hcilab.org/AppTicker/> zuletzt geprüft am 19.12.2012

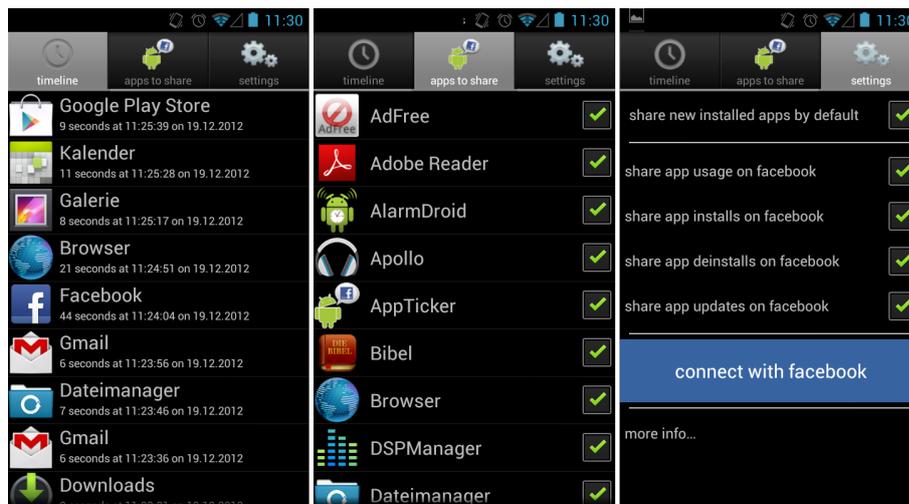


Abbildung 4.1: Bildschirmfotos der Anwendung *AppTicker*. Links: Liste des Anwendungsausführungsverlaufs. Mitte: Einstellungsmöglichkeit, ob einzelne Anwendungen veröffentlicht werden sollen. Rechts: Allgemeine Einstellungen.



Abbildung 4.2: Bildschirmfoto der Veröffentlichung einer Anwendungsnutzung auf *Facebook*.

Nutzungsdauer einzusehen. Im nächsten Register kann ausgewählt werden, welche Anwendungen auf *Facebook* veröffentlicht werden sollen. Allgemeine Veröffentlichungseinstellungen können im dritten Register getätigt werden. Ein Bildschirmfoto der Anwendungen findet sich in *Abbildung 4.2*. *AppTicker* gibt dem Nutzer die Möglichkeit, Informationen zu Änderungen und Ausführungen der durch ihn genutzten Anwendungen zu veröffentlichen.

4.2 Implementierungsart

Grundsätzlich bestehen drei Möglichkeiten, die Darstellung von Diagrammen auf Android-Mobiltelefonen zu implementieren. Es können bereits existierende Android-Bibliotheken genutzt werden, welche speziell auf Diagramme in Android ausgerichtet sind. Androids Darstellungsklasse *WebView* bietet die Möglichkeit, in den Sprachen HTML, CSS und Javascript implementierte Oberflächen darzustellen und so auch Javascript-Bibliotheken zu benutzen. Die dritte Variante ist eine Eigenimplementierung in Java mit einfachen geometrischen Elementen.

Bibliothek-Name	Diagramm Arten	3D	Lizenz	sonstiges
achartengine [4Vio9]	Säulen, Linien, Punkt	nein	Apache 2.0	- - -
aichart [Art]	Säulen, Blasen, Kreis, Linien, etc	nein	kommerziell	reines Java
androidplot [And12]	Säulen, Linien, Scatter, Stufen	nein	frei	- - -
ChartDroid [Kos10]	Säulen, Donut, Kreis, Linien, etc	nein	Apache 2.0	intent-basiert
GraphView [Geh11]	Säulen, Linien	nein	GNU LGPL	- - -
rapidandroid [IU09]	- - -	- - -	Apache 2.0	kaum Informationen
rchart [CJ99]	Säulen Kreis, Linien, etc	ja	freie Testversion	- - -
teechart [SLo6]	Säulen, Kreis, Linien, etc	ja	kommerziell	Java / HTML5

Tabelle 4.1: Übersicht der Android-Diagramm-Bibliotheken. Einige Diagramm-Bibliotheken bieten ein breites und für Mobiltelefone optimiertes Feld verschiedener Diagrammtypen, jedoch sind diese in der Regel nicht ausreichend individualisierbar

Android-Bibliotheken Spezielle Android-Bibliotheken sind oftmals in reinem Java implementiert und für Android und kleine Bildschirme optimiert. Die meisten unterstützen keine 3D-Ansicht, sind jedoch unter freien Lizenzen veröffentlicht. Beispiele hierfür sind: *ACHartEngine* und *ChartDroid*. Die Bibliotheken *RChart* und *TeeChart* bietet 3D-Ansichten, sind jedoch auch nur kommerziell verfügbar. Eine Veränderung des Aussehens der erzeugten Diagramme ist meist nur eingeschränkt möglich. Vor allem Vorgänge wie das Einfügen individueller Icons auf der Diagrammfläche ist normalerweise nicht vorgesehen und nur mit viel Aufwand und unter unschöner Modifikationen möglich. Benutzerinteraktionen sind meist implementiert, jedoch nicht immer ohne weiteres um eigene Optionen erweiterbar. Diese Bibliotheken benötigen keine Internetverbindung und greifen nur auf lokale Daten zurück. Tabelle 4.1 listet die wichtigsten Bibliotheken auf.

Android-Webview Die Klasse *Webview* aus Androids *Webkit* bietet viel Möglichkeiten, da mit dessen Hilfe vollständige Webseiten dargestellt werden können. Die Darstellen von speziellen Attributen wie *SVG* oder die vollständige Unterstützung von *CSS* ist bei nicht aktuellen Android-Versionen (< 3.x) nicht gewährleistet. Neben einer Vielzahl an meist auf Javascript und *CSS* basierten Bibliotheken gibt es vor allem die Möglichkeit, ein Diagramm lokal zu erzeugen. Mithilfe des *HTML5*-Elements *Canvas* lassen sich einfach sowohl 2D- als auch 3D-Diagramme einfach erzeugen. Jedoch ist das erzeugte *Canvas*-Element statisch, sodass es nachträglich nicht mehr verändert werden kann. Dadurch ist beispielsweise das benutzerdefinierte Anzeigen von Icons und Skalieren des Diagramms nicht möglich. Eine weitere Möglichkeit ist das Darstellen von Diagrammen im *PNG*- oder *SVG*-Format, welche zuvor auf einem Server berechnet wurden². Bei *SVG* bestünde, im Gegensatz zu *PNG*, die Möglichkeit, nachträglich Änderungen am Diagramm vorzunehmen. Fehlende *SVG* Unterstützung seitens Android und die unter Umständen umfangreiche Datenübertragung über das Internet machen *Webview* für den Einsatzzweck dieser Arbeit ungeeignet.

²Googles Image Charts: <https://developers.google.com/chart/image/> zuletzt geprüft am 19.12.2012

Eigenimplementierung Eine vollständige Eigenimplementierung ist die flexibelste Variante. Dies beinhaltet jedoch auch einen hohen Programmieraufwand, da neben der Funktionalität auch auf Effizienz in der Darstellungsberechnung geachtet werden muss. Android bietet die Möglichkeit, die Klasse *View*, aus der sämtliche Darstellungsklassen abgeleitet werden, für die Bedürfnisse des Programmierers anzupassen. In ihr können auf einem *Canvas*-Element verschiedene einfache geometrische Elemente, Bilder sowie Text gezeichnet und angezeigt werden. Dabei muss der Programmierer manuell auf Nutzeraktionen reagieren.

Fazit: Implementierung Ihre nur eingeschränkte Modifizierbarkeit macht Android-Bibliotheken für die gegebenen Anforderungen ungeeignet. Auch Webview kommt aufgrund fehlender Unterstützung bei älteren Geräten und der fehlenden Nativität als Implementierungsart nicht in Frage. Jedoch bietet Androids View-Klasse für die meisten Probleme einfache Lösungen. Das Zeichnen von Bildern, Texten und Rechtecken sind Standardfähigkeiten der View-Klasse. Da keine 3D-Eigenschaften benötigt wird, weist die Berechnung der Visualisierung eine niedrige Komplexität auf. Auch für die Skalierung und einfache Bildschirmberührungen bietet sie Schnittstellen.

4.3 Implementierungsvorgehen

Der Zusammenbau und die Darstellung eigener graphischer Elementen wird eine abgewandelte Form der Klasse *View* benötigt. Bei jeder Änderung des Bildschirms (neue Daten, Skalierung, Transition, ...) muss die gesamte Oberfläche erneut gezeichnet werden. Dies geschieht durch das Überschreiben der Funktion *onDraw*. Es muss darauf geachtet werden, dass alle Elemente in der richtigen Reihenfolge gezeichnet werden. Aufgrund des häufigen Zeichenwiederholungen muss die Implementierung dieser Funktion sehr effizient sein. Bei jedem Zeichen-Schritt muss über jeden benötigten Zeitpunkt und jede Anwendung iteriert werden und mit *canvas.draw{Rect,Bitmap,Line,Text,...}* das benötigte Icon in richtiger Größe an die richtige Stelle platziert werden.

Um auf Eingaben wie Transition und Berührungen reagieren zu können, muss die Funktion *onTouchEvent(MotionEvent)* überschrieben werden. Das *MotionEvent* enthält Information über die Bildschirmberührung, wie zum Beispiel die Art der Berührung. *event.getAction()* liefert *ACTION_DOWN*, *ACTION_MOVE*, ... als Ergebnis. Es gibt keine Möglichkeit, einen *ActionListener* auf Bitmaps oder andere graphische Elemente zu binden. Daher muss bei einer Berührung für jedes Icon mit *Rect.contains(X,Y)* geprüft werden, ob es ausgewählt wurde. Diese Funktion ist nur auf einfachen von *Canvas* bereitgestellten Rechtecken möglich. Daher muss für jedes Bitmap ein Rechteck im Hintergrund gezeichnet werden. Während das Verschieben des Bildschirms mit Hilfe von *onTouchEvent* ohne viel Aufwand implementiert werden kann, ist es ratsam, für die Skalierung auf eine Eigenimplementierung der Funktion *onScale* der Klasse *ScaleGestureDetector.SimpleOnScaleGestureListener* zurückzugreifen. Listing 4.1 veranschaulicht den Aufbau einer Klasse, welche die Oberklasse *View* für den Zweck dieser Arbeit implementiert.

Listing 4.1 Beispielimplementierung der Klasse View

```

public class Diagramm extends View {
    // Variablen- und Konstruktordeklaration

    @Override
    public void onDraw(Canvas canvas) {
        // zeichne alle Elemente die sichtbar sein sollen
        // mit canvas.drawBitmap oder canvas.drawRect(new Rect(0,0,100,100))
    }

    @Override
    public boolean onTouchEvent(MotionEvent ev) {
        switch (ev.getAction) {
            case MotionEvent.ACTION_DOWN: ...
            case MotionEvent.ACTION_MOVE: ...
            ...
        }
        return true;
    }

    private class ScaleListener extends ScaleGestureDetector.SimpleOnScaleGestureListener
    {
        @Override
        public boolean onScale(ScaleGestureDetector detector) {
            mScaleFactor *= detector.getScaleFactor();

            return true;
        }
    }
}

```

4.4 Performanz

Sowohl das Laden und Aufbereiten der Daten, als auch das Zeichnen rufen deutlich spürbare Wartezeiten hervor. Die Lade- und Aufbereitungszeit der Daten kann verringert werden, indem nur Daten, die benötigt werden - die anzuzeigenden Icons -, zwischengespeichert werden. Dadurch wird unnötiges Umherschieben großer Objekte verhindert.

Ausführungszeiten und Icons für bestimmte Anwendungen werden in Hashtabellen mit der Anwendungs-ID als Schlüssel gespeichert. Diese IDs bestehen ausschließlich aus Integerwerten. Für diesen Spezialfall bietet Android die effizientere Variante `SparseArray` an. Dies verringert den Speicherbedarf und ermöglicht eine flüssige Bedienbarkeit. Auch wenn Daten nun schneller aus dem Datenspeicher geladen werden, ist die wichtigere Verbesserung das Zeichnen. Mit jeder Berührung des Bildschirms muss der gesamte Inhalt neu gezeichnet werden. `onDraw` muss daher sehr effizient implementiert sein. Während Schleifendurchläufe und das Zeichnen selbst nahezu nicht auszumachen sind, benötigt das Allokieren von Bitmaps und Rechtecken viel Zeit. Bitmaps sollten daher beim Start der Visualisierung einmalig erstellt und nur im äußersten Notfall während der Zeichenphase beschnitten oder skaliert werden.

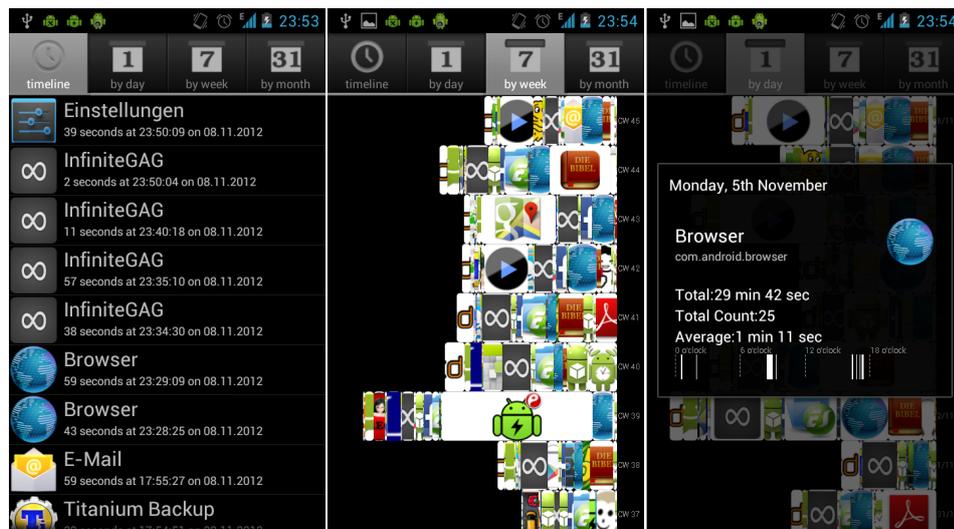


Abbildung 4.3: Bildschirmfotos der Anwendung *HistApp*. Links: Liste des Anwendungsausführungsverlaufs. Mitte: Histogramm-Ansicht im Wochenmodus. Rechts: Ansicht mit Informationen über einzelne Anwendung.

Eine weitere Möglichkeit zur Performanzverbesserung wäre eine Implementierung mittels OpenGL und Android Native Development Kit (NDK). Diese bieten graphische Darstellungen in Echtzeit.

4.5 *HistApp*

Zur Verwendung der in diesem Kapitel beschriebenen Implementierung wurde die Anwendung *HistApp*³ entwickelt. Der Name „*HistApp*“ steht für „Histogramm-Application“ beziehungsweise „History-Application“, das Ziel der Anwendung, den Nutzer über sein Nutzungsverhalten zu informieren. *HistApp* enthält den Protokollierungskomponente der Anwendung *AppTicker*, somit stehen Daten zur Verfügung die in *AppTicker* für die Visualisierung verwendet werden. Eine direkte Verknüpfung zu *Facebook* ist nicht vorgesehen. Nutzer können jedoch ihre erzeugten Diagramme exportieren und dadurch manuell auf *Facebook* oder anderen sozialen Netzwerken veröffentlichen. Die visuelle Struktur (Abbildung 4.3) der Anwendung entspricht der von *AppTicker*. Um die Übersicht und die Handhabung zu vereinfachen, werden die möglichen Histogramm-Ansichten auf einzelne Registerkarten verteilt. Das erste Register enthält, wie bei *AppTicker*, eine textuelle Liste, mit den vergangenen Anwendungsausführungen. Die restlichen drei Register enthalten eine Tages-, Wochen- und Monatsansicht.

³HistApp: <http://play.google.com/store/apps/details?id=org.hcilab.projects.histoapp> geprüft am 19.12.2012

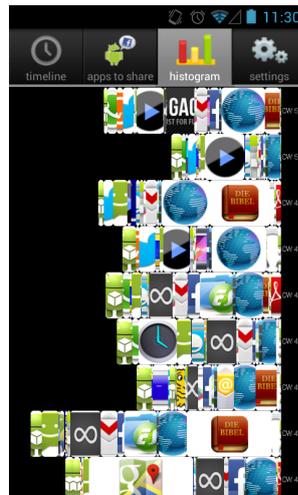


Abbildung 4.4: Bildschirmfoto der Visualisierung in *AppTicker*

4.6 Zusammenfassung

In diesem Kapitel wurde eine Anwendung mit dem Namen *HistApp* entwickelt, welche die in Kapitel 3 beschriebene Visualisierung implementiert. Als Grundlage hierfür diente die Anwendung *AppTicker*. *HistApp* ist auf dem Mobiltelefonbetriebssystem *Android* lauffähig, daher wurde zur Implementierung die Sprache Java verwendet. Bei der Implementierung wurde darauf verzichtet, auf bereits bestehende Bibliotheken zurückzugreifen oder mithilfe der Klasse *WebView* HTML und CSS einzubinden. Eine Eigenimplementierung war aufgrund der einfachen geometrischen Elemente der Visualisierung möglich. Anschließend wurde die erstellte Visualisierung als viertes Register in die Anwendung *AppTicker* integriert und den Nutzern zur Verfügung gestellt (Abbildung 4.4).

Das folgende Kapitel beschreibt ein Studie, in der die Visualisierung in einer ersten Version durch einen kleinen Nutzerkreis vorab getestet wurde, sowie deren Ergebnis.

5 Benutzerstudie

Vor der Verbreitung der Anwendung über GooglePlay wurde zur Evaluation der Anwendung eine Studie zu ihrer Nutzbarkeit durchgeführt. Teilnehmer sollten die Bedienbarkeit der Visualisierung auf verschiedenen Geräten testen und prüfen, ob diese ohne weitere Erklärungen verständlich ist.

Die Fragestellung, welche mit dieser Studie geklärt werden sollte, lässt sich in zwei Unterpunkte aufteilen. Zum einen ist dies die Bedienbarkeit der Visualisierung. So diente die Studie dazu, Darstellungsfehler in der Anwendung zu finden. Da bei einer Visualisierung die Erkennbarkeit der dargestellten Elemente, in diesem Fall die Ausführungszeiten, das Wichtigste ist, sollte auch geprüft werden, wie die Visualisierung auf Oberflächen mit geringer Auflösung und Displaygröße dargestellt wird, und ob sie zu verstehen ist. Des Weiteren sollte auch die Qualität der Visualisierung getestet werden. Geprüft wurde, ob die Visualisierung ohne große Erklärungen verständlich ist und ob die Aussage für einen durchschnittlichen Mobiltelefon-Nutzer ohne weiteres ersichtlich ist.

5.1 Teilnehmer

Als Teilnehmer dieser Studie wurden Personen ausgewählt, welche bereits ein Mobiltelefon besitzen. Somit bestand die Möglichkeit die Anwendung auf deren Telefon zu installieren. Dies hatte die Vorteile, dass die Teilnehmer im Umgang mit Mobiltelefonen bereits Vorerfahrung besaßen und mit Bedienkonzepten vertraut waren. Zudem konnten authentische Daten von verschiedenen Personen gesammelt werden. Gleichzeitig wurde ein breites Spektrum verschiedener Mobiltelefontypen und dadurch auch Displaygrößen und -auflösungen analysiert. Es wurde darauf geachtet, dass unter den Studienteilnehmern auch Personen, deren Mobiltelefon eine kleine Auflösung (kleiner als 800x400) aufweist, vertreten waren. Die Studie hatte acht Teilnehmer, wovon sechs Studenten der Informatik oder eines verwandten Studiengangs und zwei fachfremde Arbeitnehmer waren. Der jüngste Teilnehmer war 21, der älteste 53 Jahre alt. Alle Teilnehmer waren männlich.

5.2 Vorgehen

Zu Beginn der Studie wurde jeder Nutzer belehrt und über die Idee und den Rahmen der Studie, sowie die erhobenen Daten informiert. Zur Durchführungen wurde das „Android Application Package File“ (APK) an die Nutzer verteilt und diese gebeten, es zu installieren,

wobei die Teilnehmer bei Bedarf unterstützt wurden. Daraufhin wurde der Nutzer gebeten, sein Mobiltelefon wie gewohnt zu nutzen und bei Interesse die neue Anwendung zu testen. Spätestens nach fünf bis sieben Tagen sollte jeder Nutzer die Anwendung anschauen und verwenden, anschließend den Eindruck verbal oder schriftlich mitteilen. Dabei wurde er gebeten, auf die Bedienbarkeit und das Verständnis der Visualisierung zu achten.

5.3 Ergebnis

Nach der Testphase durch die Test-Nutzer mussten einige Nachbesserungen geleistet werden.

Verständlichkeit Um die Darstellung möglichst übersichtlich zu halten, wurde wenn möglich, auf Text verzichtet. Dies führte an einigen Stellen zu Unklarheiten über die Bedeutung der visualisierten Daten. Um im Hinblick darauf eine Verbesserung zu erreichen, werden zu Beginn einer Visualisierung angezeigt, was auf dem Display zu sehen ist und wie interagiert werden kann.

Bedienbarkeit Interaktionen (skalieren und verschieben) in der Visualisierung waren intuitiv klar. Die auf Mobiltelefonen gängige Möglichkeit, Elemente anzuklicken, wurde nicht immer sofort verstanden und musste teilweise extra erläutert werden. Besondere Probleme rief das Konzept des *langen Klicks*, bei dem ein Element für eine längere Zeit berührt wird, um in ein Optionen-Menü zu gelangen, hervor. Nach einer anfänglichen Klick-Dauer von mindestens 1500 ms wurde dieser Wert allmählich auf 500 ms (Androidstandard) verringert und schließlich komplett entfernt, da diese Funktion für zu viel Verwirrung sorgte. Stattdessen werden nur noch kurze Klicks (kürzer als 500 ms) berücksichtigt. Beim Klick auf ein Anwendungssicon werden Informationen über die Anwendung angezeigt, beim Berühren einer freien Stelle auf dem Bildschirm werden verfügbare Optionen dargestellt.

Korrektur der Darstellung Die automatische Korrektur der Darstellung, welche dafür sorgt, dass immer der gesamte Bildschirm in Benutzung ist und kein Rand entsteht wurde verbessert. Diese Korrektur war fehlerhaft implementiert, sodass Teile der Visualisierung verschwanden und ein falscher Eindruck entstand.

Die Verbesserungen wurden durch den Teilnehmer, der etwas angemerkt hat, erneut validiert. Anschließend wurden sowohl *AppTicker* als auch *HistApp* auf dem Android Distributionskanal *GooglePlay* veröffentlicht. Abbildung 5.1 zeigt die veröffentlichte Anwendung *AppTicker*. Für die Veröffentlichung der Anwendung *HistApp* wurde ein Beschreibungstext, so wie das Anwendungssicon und ein Anwendungsbanner (Abbildung 5.2) benötigt. Als bereits verbreitete Anwendung war dies für *AppTicker* bereits vorhanden.

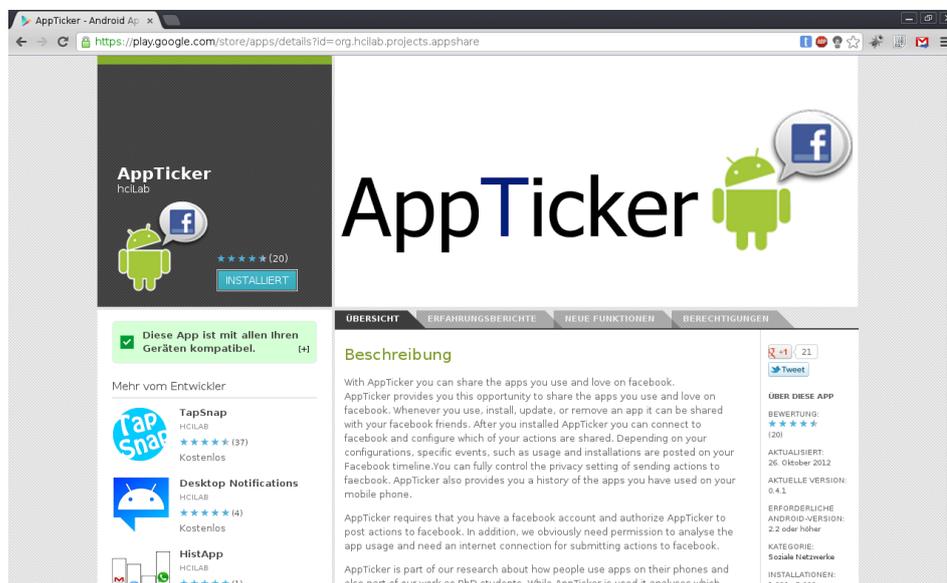


Abbildung 5.1: Bildschirmfoto der Anwendung *AppTicker* auf *GooglePlay*



Abbildung 5.2: Die Abbildung zeigt den „FeatureGraph“ der Anwendung *HistApp*. Das Säulendiagramm rechts entspricht den Anwendungsicon.

6 Analyse der Nutzung und Verhaltensänderung

In diesem Kapitel werden die mithilfe der Visualisierung ermittelten Nutzungsdaten analysiert und es wird untersucht, inwiefern die Visualisierung das Nutzungsverhalten beeinflusst. Die dieser Analyse zugrunde liegenden Daten wurden ausschließlich durch die Anwendung AppTicker (beschrieben in 4.1) gesammelt, da HistApp zum Zeitpunkt der Auswertung einen zu kleinen Nutzerkreis hatte.

6.1 Nutzerfeld

Nutzer dieser Studie sind automatisch alle Nutzer der Anwendung AppTicker. AppTicker wird über die offizielle Verbreitungsplattform für Android-Anwendungen (GooglePlay¹) vertrieben. Dies bietet ein weites Nutzerfeld und die Chance auf eine gemischte Teilnehmerschaft, welche bei oftmals an Universitäten durchgeführten Studien nicht möglich ist.

Die Anwendung *AppTicker* wurde durch mehr als 2000 Nutzer installiert. Hiervon sind 482 Nutzer aktive Nutzer. Als „Aktive Nutzer“ werden in dieser Analyse Nutzer genannt, die mindestens fünf Tage aktiv sind. Ein Großteil der Nutzer kommt dabei aus Deutschland und anderen europäischen Ländern (83%), des Weiteren sind Nutzer aus Asien (11%), den USA (5%) und sonstigen Ländern (2%). Von den zum Einsatz kommenden Geräten sind 62% der Marke Samsung, gefolgt von HTC mit 13% und Sony Ericsson mit 10%. Am häufigsten kommt Gingerbread (2.3; mit 44%) und Ice Cream Sandwich (4.0; mit 37%) von Android zum Einsatz.

6.2 Nutzung der Visualisierung

Auf die Nutzung der Histogramm-Funktion ließen sich, über den Zeitraum vom 26. Oktober bis 30. November 2012, rund 260 Nutzer ein, 29 davon nutzten diese aktiv (Abbildung 6.2). Dabei teilten sich die Nutzungen auf die verschiedenen Zeiteinheitenansichten folgendermaßen auf. Die Tagesansicht wurde 87 Mal aufgerufen, die Wochenansicht 351 und die Monatsansicht 54 Mal (Abbildung 6.1). Es fällt auf, dass die Wochenansicht deutlich

¹GooglePlay: <http://play.google.com/> zuletzt geprüft am 19.12.2012

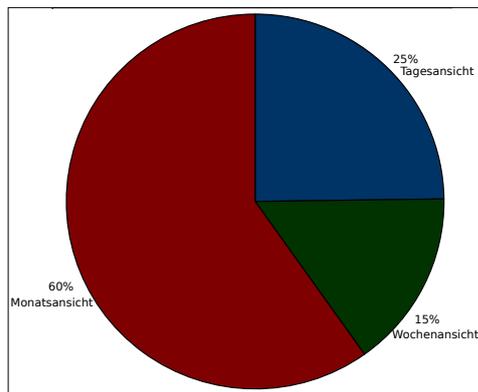


Abbildung 6.1: Das Diagramm beschreibt das Verhältnis der verschiedenen genutzten Zeiteinheit-Ansichten: Monatsansicht (Rot), Wochenansicht (Grün) und Tagesansicht (Blau)

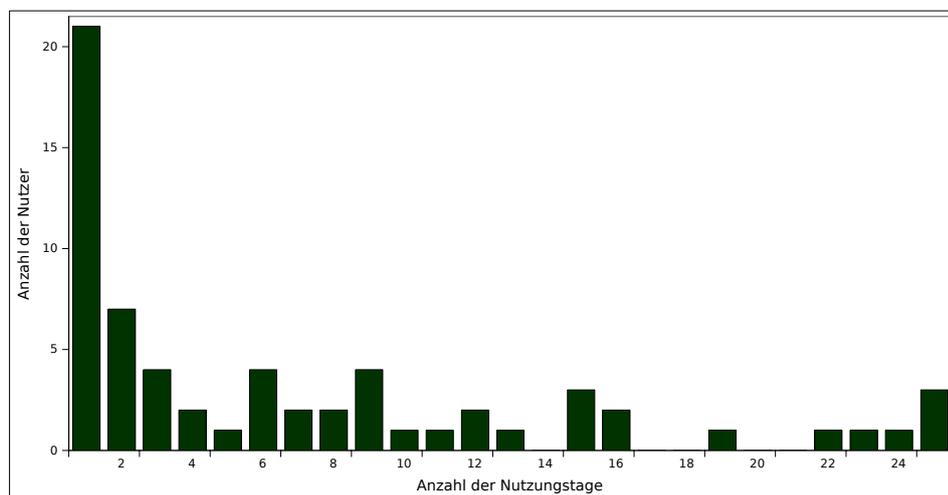


Abbildung 6.2: Das Schaubild beschreibt die Anzahl der Nutzer, die die Visualisierung über eine bestimmte Anzahl an Tage genutzt haben.

häufiger genutzt zu werden scheint als. Diese Annahme ist irreführend, da die Wochenansicht automatisch als erstes angezeigt wird und somit als Einstiegspunkt dient. Nach einbeziehen dieser Information bleiben für die Wochenansicht noch 210 Aufrufe, was die übrigen Ansichtsoptionen dennoch übertrifft. Die Visualisierung wurden somit knapp 500 Mal von insgesamt rund 260 Nutzern, von denen wiederum 30 Personen als aktiv eingestuft wurden, angeschaut. In Visualisierung interagiert wurde durch insgesamt 70 Nutzer, davon waren sechs aktive Nutzer. Die zusätzliche anwendungsspezifische Nutzungsansicht hatte lediglich 20 Nutzer, davon vier aktive (Abbildung 6.3). Insgesamt fällt auf, dass viele Nutzer die Anwendung zwar installiert, jedoch nicht gestartet haben. Außerdem fand nach einem erfolgten Start in vielen Fällen (rund 88,5%) keine weitere Verwendung der Anwendung statt.

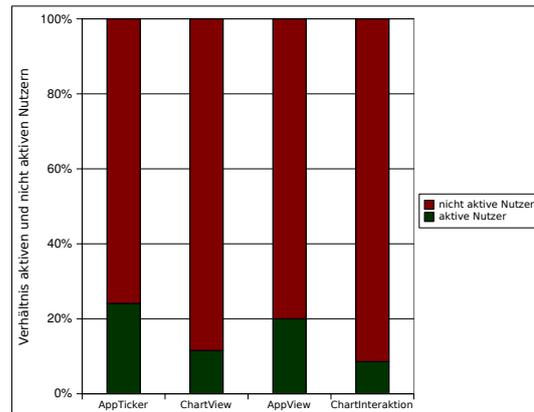


Abbildung 6.3: Darstellung die nach einmaligem Ausprobieren (Rot) einer Anwendungseigenschaft diese über einen längeren Zeitraum (Grün) genutzt haben. AppTicker ist Nutzung der Anwendung. ChartView das Aufrufen der Visualisierung. AppView das Aufrufen der anwendungsspezifischen Ansicht. ChartInteraction sind Interaktionen wie Skalieren

6.3 Nutzungsverhalten und -änderung

Die Analyse der Daten zeigt, dass die Ausführungszeiten einer Anwendung selten gleichmäßig verlaufen. Die Nutzungsdauer schwankt bei einem Nutzer und innerhalb einer Anwendung von Tag zu Tag sehr stark. Somit ist es schwierig, generelle Aussagen über eine Verhaltensänderung eines Nutzers zu treffen. Im gesamten Nutzungsverhalten stechen einige Anwendungsarten heraus, welche eine recht lange Gesamtausführungszeit aufweisen. Dazu gehören Informations- und Nachrichten-Anwendungen wie die der *Tagesschau* oder des *Kickers*, sowie Dienstprogramme wie Straßenkartenfunktionen, Einstellungsoptionen und der Browser. Ebenso stark vertreten sind unterhaltende Anwendungen wie die Videoplattform *YouTube*, der digitale Bücherdienst *Kindle* und vor allem Spiele. Neben diesen Anwendungen, die erst durch das „Smartphone“ auf dem Mobiltelefon Verbreitung fanden, sind aber auch klassische Kommunikationsanwendungen wie E-Mail-Dienste und Chatanwendungen unter den häufig und lange ausgeführten Anwendungen vertreten. Die ursprüngliche Hauptanwendung, das Telefonieren, ist ebenfalls vertreten.

Klare Verhaltensänderungen im Bezug auf einzelne Anwendungen konnte lediglich bei zehn Nutzern festgestellt werden, was einem Anteil von 30% der aktiven Nutzer entspricht. Unter den von der Änderung betroffenen Anwendungen waren neben Spiel-Anwendungen vor allem Dienstprogramme. Bei Betrachtung der gesamten Mobiltelefonnutzungszeit ist bei 70% der Nutzer eine tendenzielle Abnahme erkennbar. 30% hingegen nutzten das Mobiltelefon vermehrt. Wenn man die Ausführungszeit der Anwendungen unabhängig vom Nutzer betrachtet, hat sich bei 53% der Anwendungen die Ausführungszeit eher verlängert und bei 47% verringert.

6.4 Zusammenfassung

Die Analyse der Nutzungsdaten fand rund einen Monat nach der Veröffentlichung der Visualisierung statt. In diesem Zeitraum wurde die Visualisierung von 29 Nutzern aktiv genutzt. Von diesen Nutzern ließen sich nur wenig dazu bewegen, Interaktionen durchzuführen oder die Einzelansichten der Anwendungen zu öffnen. Insgesamt war es schwierig Tendenzen in der Mobiltelefonnutzungsdauer zu erkennen, da Nutzungen von Tag zu Tag stark variieren. Verhaltensänderungen im Bezug auf einzelne Anwendungen wurde bei 30% der Nutzer festgestellt. Aussagekräftigere Ergebnisse könnten durch Verlängern des Analysezeitraums, was im Rahmen dieser Bachelorarbeit aufgrund der begrenzten Zeit nicht möglich war, erreicht werden.

7 Fazit und Ausblick

Im diesem Kapitel werden der Inhalt und die Ergebnisse der vorliegenden Arbeit zusammengefasst. Am Ende folgt ein Ausblick, welche Ansätze weiter verfolgt werden können.

Das Mobiltelefon bestimmt den Alltag vieler Menschen, speziell Jugendlichen und jungen Erwachsenen. Oftmals ist dient das Mobiltelefon als Gedächtnisstütze für Termine und bietet eine ständige Möglichkeit zur Kommunikation. Es dient nicht nur als Erinnerung und zur ständigen Verbindung mit anderen Menschen. Durch Spiele und soziale Medien nimmt es auch viel Zeit in Anspruch. In dieser Arbeit wurde eine Darstellung entwickelt, welche veranschaulicht, wie häufig und wozu ein Mobiltelefon genutzt wird. Zur Entwicklung wurde ein iterativer Prozess genutzt, bei dem in jeder Iteration die Vor- und Nachteile einer Darstellung reflektiert und anschließend verbessert wurden. Die entwickelte Darstellung visualisiert graphisch mithilfe von Diagrammen Nutzungen, die in einem vorgegebenen Zeitraum (Tag, Woche oder Monat) stattfinden. Jeder Zeitraum wird mittels eines horizontalen Balkens dargestellt. Auf einem Balken die Anwendungen - je nach Ausführungszeit - in unterschiedlicher Breite dargestellt. Dadurch können einzelne Anwendungen direkt miteinander verglichen werden. Um das Diagramm möglichst übersichtlich zu halten, wurden zur Identifikation der Anwendungen deren Icons benutzt. Es ist außerdem möglich, durch Berühren des Icons einer Anwendung weitere Informationen über diese zu erhalten. Die Darstellung wurde in Java implementiert und in die Anwendungen *AppTicker* und *HistApp* für Android-Mobiltelefone integriert. Über den Zeitraum der Protokollierung konnten Änderungen im Nutzungsverhalten festgestellt werden. Die Nutzungszeit einzelner Anwendungen hat sich bei 30% der Nutzer deutlich gesenkt. Die gesamte Nutzungszeiten des Mobiltelefons je Nutzer ist in 70% der Fälle gefallen.

Ausblick

Im Folgenden werden Ideen zur Weiterführung der Arbeit aufgeführt, welche im Rahmen dieser Arbeit nicht möglich gewesen wären beziehungsweise zu viel Zeit gekostet hätten, jedoch aufschlussreichere Ergebnisse liefern könnten. Damit die Aussagekraft der Daten verbessert wird und deutlichere Tendenzen in der Nutzung zu erkennen sind, sollte der Zeitraum, über den hinweg Daten gesammelt werden, verlängert werden. Dies könnte zu einer besseren Datengrundlage führen. Dabei ist es ebenfalls notwendig, dass Teilnehmer bereit sind, die Anwendung über einen längeren Zeitraum hinweg zu nutzen. Die Bereitschaft zur längeren Teilnahme kann durch ein Belohnungssystem erhöht werden [CESLo6]. In der Visualisierung gibt es, trotz der Gruppierung der Anwendungen Bereiche, in denen nur wenig zu erkennen

ist, da deren Ausführungszeit im Vergleich zu anderen Anwendungen gering ist. Deshalb könnte eine weitere Gruppierung nach der Kategorie der Anwendung erfolgen. Dabei ist noch nicht klar, wie diese Kategorisierung erfolgen kann. Es müsste geprüft werden, ob es möglich ist, eine Kategorisierung anhand der Einteilung auf *GooglePlay* zu erstellen oder ob dies manuell durch den Nutzer geschehen müsste. Der Schritt einer weiteren Kategorisierung würde außerdem die Analyse der Daten vereinfachen und allgemeinere Tendenzen aufzeigen. So würden beispielsweise nicht nur die Chat-Anwendung *WhatsApp* separat, sondern alle Chat-Anwendungen gemeinsam behandelt, was zu allgemeineren Aussagen führen könnte. Damit die Ergebnisse nicht nur auf das Mobiltelefon beschränkt sind, könnte an einer Studie mit weiteren Medien gearbeitet werden. Neben anderen Mobiltelefonbetriebsystemen wäre es möglich, das Angebot auf den Fernsehkonsum zu erweitern. Naheliegender wäre ebenso das Miteinbeziehen der Zeit, die am Computer verbracht wird. In beiden Fällen könnten ebenfalls eine Kategorisierung der Inhalte vorgenommen werden. Um noch mehr über die Umstände und den Grund einer Mediennutzung herauszufinden, wäre es möglich, Informationen zum Kontext zu protokollieren. Geschehen könnte dies durch das Hinterlegen von Tagesabläufen eines bestimmten Nutzers oder zusätzliche Protokollierung des Nutzerstandortes mittels GPS. Je mehr Daten protokolliert werden, desto kritischer ist die Erhebung der Daten. Der Datenschutz und ein umfangreiches Informieren der Nutzer darf dabei nicht fehlen.

Abbildungsverzeichnis

3.1	Beispiel eines Häufigkeitsdiagramms	18
3.2	Dreidimensionales Kreisdiagramm	19
3.3	Dreidimensionales Säulendiagramm	20
3.4	Vergleich verschiedener Darstellung eines Säulendiagramms	21
3.5	Vergleich Icons gestreckt und Icons abgeschnitten	23
3.6	Ergebnis Icon-Studie	24
4.1	Bildschirmfotos der Anwendung AppTicker	26
4.2	Bildschirmfoto der Veröffentlichung einer Anwendungsnutzung auf <i>Facebook</i> .	26
4.3	Bildschirmfotos der Anwendung <i>HistApp</i>	30
4.4	Bildschirmfoto der Visualisierung in <i>AppTicker</i>	31
5.1	Bildschirmfoto der Anwendung <i>AppTicker</i> auf <i>GooglePlay</i>	35
5.2	„FeatureGraph“ der Anwendung <i>HistApp</i>	35
6.1	Verhältnis der Nutzung der Zeiteinheiten	38
6.2	Aktive Visualisierungsnutzer	38
6.3	Vergleich aktive und nicht aktive Nutzer	39

Tabellenverzeichnis

4.1	Übersicht der Android-Diagramm-Bibliotheken	27
-----	---	----

Verzeichnis der Listings

4.1	Beispielimplementierung der Klasse View	29
-----	---	----

Literaturverzeichnis

- [4Vio9] 4ViewSoft. AChartEngine, 2009. URL <http://www.achartengine.org>. [Zuletzt geprüft am 18.12.2012].
- [ABS05] E. Arroyo, L. Bonanni, T. Selker. Waterbot: exploring feedback and persuasive techniques at the sink. In *ACM Conference on Human Factors in Computing Systems*, S. 631–639. 2005.
- [AKWK10] S. M. Arteaga, M. Kudeki, A. Woodworth, S. Kurniawan. Mobile system to motivate teenagers' physical activity. In *International Conference on Interaction Design and Children*, S. 1–10. 2010.
- [And12] AndroidPlot.com. AndroidPlot, 2012. URL <http://www.achartengine.org>. [Zuletzt geprüft am 18.12.2012].
- [Art] ArtfullBits. aiChart. URL <http://www.artfulbits.com/products/android/aicharts.aspx>. [Zuletzt geprüft am 18.12.2012].
- [BCo8] F. Buttussi, L. Chittaro. MOPET: A context-aware and user-adaptive wearable system for fitness training. *Artificial Intelligence in Medicine*, S. 153–163, 2008.
- [BCNB12] U. Banerjee, J. Cowan, L. Nauges, K. Benedict. Android Expected to Reach Its Peak This Year as Mobile Phone Shipments Slow, According to IDC, 2012. URL <http://www.sys-con.com/node/2291309>. [Zuletzt geprüft am 18.12.2012].
- [BHS⁺11] M. Böhmer, B. Hecht, J. Schöning, A. Krüger, G. Bauer. Falling asleep with Angry Birds, Facebook and Kindle: a large scale study on mobile application usage. In *International Conference on Human-Computer Interaction with Mobile Devices and Services.*, S. 47–56. 2011.
- [CESL06] S. Consolvo, K. Everitt, I. E. Smith, J. A. Landay. Design requirements for technologies that encourage physical activity. In *International Conference on Mobile Systems, Applications, and Services*, S. 457–466. 2006.
- [CJ99] ConSerT.net, Java4Less.com. RChart, 1999. URL <http://www.java4less.com/android/charts.php>. [Zuletzt geprüft am 18.12.2012].
- [CK98] J. V. Carlis, J. A. Konstan. Interactive Visualization of Serial Periodic Data. In *ACM Symposium on User Interface Software and Technology.*, S. 29–38. 1998.
- [FMK⁺10] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, D. Estrin. Diversity in smartphone usage. In *International Conference on Mobile Systems, Applications, and Services*, S. 179–194. 2010.

- [Fog02] B. J. Fogg. *Persuasive Technology: Using Computers to Change What We Think and Do (Interactive Technologies)*, Kapitel Computers as Persuasive Social Actors. Morgan Kaufmann, 2002.
- [Gar11] Gartners. Gartner Says Android to Command Nearly Half of Worldwide Smartphone Operating System Market by Year-End 2012, 2011. URL <http://www.gartner.com/it/page.jsp?id=1622614>. [Zuletzt geprüft am 18.12.2012].
- [Geh11] J. Gehring. GraphView, 2011. URL <http://www.jjoe64.com/p/graphview-library.html>. [Zuletzt geprüft am 18.12.2012].
- [GP09] A. A. Gartland, P. Piasek. Weigh your waste: a sustainable way to reduce waste. In *ACM Conference on Human Factors in Computing Systems. Extended Abstracts.*, S. 2853–2858. 2009.
- [IU09] D. Inc., UNICEF. Rapidandroid, 2009. URL <http://www.rapidandroid.org>. [Zuletzt geprüft am 18.12.2012].
- [Kos10] Kostmo. ChartDroid, 2010. URL <http://code.google.com/p/chartdroid/>. [Zuletzt geprüft am 18.12.2012].
- [Mac92] J. N. MacGregor. A comparison of the effects of icons and descriptors in videotex menu retrieval. *International Journal of Man-Machine Studies*, 37(6):767 – 777, 1992.
- [MHK05] I. Mulder, G. ter Hofte, J. Kort. SocioXensor: Measuring user behaviour and user eXperience in conteXt with mobile devices. In *User Centered Innovation, TNO Information and Communication Technology*, S. 179–194. 2005.
- [OJHo3] A. Orso, J. Jones, M. J. Harrold. Visualization of program-execution data for deployed software. In *ACM symposium on Software visualization*, S. 67–ff. 2003.
- [RZ10] A. Rahmati, L. Zhong. A Longitudinal Study of Non-Voice Mobile Phone Usage by Teens from an Underserved Urban Community. *Clinical Orthopaedics and Related Research*, abs/1012.2832, 2010.
- [SL06] S. S. SL. TeeChart, 2006. URL <http://www.steema.com/teechart/mobile>. [Zuletzt geprüft am 18.12.2012].
- [TCM⁺12] A. Thieme, R. Comber, J. Miebach, J. Weeden, N. Kraemer, S. Lawson, P. Olivier. "We've bin watching you": designing for reflection and social persuasion to promote sustainable lifestyles. In *ACM Human Factors in Computing Systems, CHI '12*, S. 2337–2346. 2012.
- [TFAG06] T. Toscos, A. Faber, S. An, M. P. Gandhi. Chick clique: persuasive technology to motivate teenage girls to exercise. In *ACM Conference on Human Factors in Computing Systems. Extended Abstracts.*, S. 1873–1878. 2006.

Erklärung

Hiermit versichere ich, diese Arbeit selbständig verfasst und nur die angegebenen Quellen benutzt zu haben.

(Kevin Wenz)