Evaluation von Passworteingaben auf Multitouch-Smartphones in Bezug auf Usability und Security

Ruben Deyhle

Bachelorarbeit

Gutachter
Prof. Dr.-Ing. Michael Weber

Betreuer
Florian Schaub

Institut für Medieninformatik
Fakultät für Ingenieurwissenschaften
und Informatik
Universität Ulm

2012

ulm university universität
uulm

Universität Ulm | 89069 Ulm | Germany

Fakultät für
Ingenieurwissenschaften
und Informatik
Institut für Medieninformatik

Evaluation von Passworteingaben auf Multitouch-Smartphones in Bezug auf Usability und Security

Bachelorarbeit an der Universität Ulm

Vorgelegt von:
Ruben Deyhle
ruben.deyhle@uni-ulm.de

Gutachter:
Prof. Dr.-Ing. Michael Weber

Betreuer:
Florian Schaub

2012

Fassung vom 21. Februar 2012

© 2012 Ruben Deyhle

Diese Arbeit ist lizensiert unter der Creative Commons **Namensnennung-Keine kommerzielle Nutzung-Weitergabe unter gleichen Bedingungen 3.0 Deutschland** Lizenz. Nähere Informationen finden Sie unter http://creativecommons.org/licenses/by-nc-sa/3.0/de/.

Satz: PDF-LaTeX 2$_\varepsilon$

Abstract

Multitouch-Smartphones revolutionieren das jahrzehntelang bestehende Eingabeparadigma „Tastatur“ durch dynamische Anzeige einer virtuellen Tastatur. Die bislang stets als gegeben vorausgesetzte Hardwaretastatur verschwindet. Dennoch bleiben auf ihr begründete Paradigmen, wie die Authentifizierung durch Eingabe eines Passworts, bestehen. Wie gut lassen sich Passwörter, im täglichen Gebrauch auch bei Smartphones unabdingbar, auf Multitouch-Tastaturen eingeben?
Der Ort der Passworteingabe hat sich mit der zunehmenden Verbreitung von Smartphones außerdem vom sicheren Schreibtisch in die Öffentlichkeit verlagert. Shoulder-Surfing bei der Passworteingabe ist eine reale Gefahr bei Smartphones.
Hier wurde eine Studie zur Untersuchung von Usability und Shoulder-Surfing-Sicherheit bei aktuellen Multitouch-Smartphones durchgeführt, die Unterschiede zwischen den verschiedenen Systemen aufzeigt.

Inhaltsverzeichnis

1. Einleitung **1**
1.1. Problemanalyse & Fragestellung 1
1.2. Methodik und Aufbau 2

2. Verwandte Arbeiten **3**

3. Tastaturanalyse **7**
3.1. Plattformen 7
3.1.1. Android 7
3.1.2. iOS 10
3.1.3. Windows Phone 10
3.1.4. Symbian^3 12
3.1.5. MeeGo 12
3.1.6. Andere Systeme 14
3.1.7. Festgelegte Einstellungen für das weitere Vorgehen 14
3.1.8. Zusammenfassung 15
3.2. Tastaturanalyse 15
3.2.1. Hauptansicht 16
3.2.2. Sonderzeichen-Ansicht 16
3.2.3. PopUp-Tasten 18

4. Passwörter und Sicherheit **19**
4.1. Grundlegendes 19
4.1.1. Analyse der LulzSec-Passwörter 20
4.2. Passwortgenerierung 20
4.2.1. Klassifikation von Zeichen in Passwörtern 22
4.2.2. Passwortschemata 23

5. Studienentwicklung und -aufbau **25**
5.1. Hypothesenbildung 25

5.2. Einführender Fragebogen 26
5.3. Usability-Test 27
5.4. Shoulder-Surfing-Test 29
5.5. Testsoftware 33
5.6. Ort der Studie und technische Begebenheiten 35
5.7. Probandenanwerbung 35

6. Auswertung **37**
6.1. Probanden 37
6.2. Usability 39
6.2.1. Eingabegeschwindigkeit (H_1) 39
6.2.2. Eingabefehlerzahl (H_2) 42
6.2.3. Auswertung des Usability-Fragebogens (H_3) 43
6.3. Shoulder-Surfing 46
6.3.1. Erfolg beim Shoulder-Surfing (H_4) 46
6.3.2. Bewertung des Shoulder-Surfing (H_5) 48
6.3.3. Shoulder-Surfing-Strategien (H_6) 50
6.3.4. Passwortanalysen 52
6.3.5. Gesamtbewertung 54
6.4. Zusammenfassung 55
6.5. Grenzen der Studie 56

7. Diskussion **59**

A. Fragebögen **61**

B. Auswertung **69**

1. Einleitung

1.1. Problemanalyse & Fragestellung

Smartphones sind mittlerweile allgegenwärtig. Für viele Menschen sind sie ein Alltagsgegenstand geworden, mit dem sie jederzeit und überall die Möglichkeit haben, auf das Internet zuzugreifen.
Die meisten Internetdienste sind aber nach wie vor primär auf die Nutzung von zuhause ausgelegt; die Nutzerauthentifikation über eine Kombination von Benutzername und Passwort hat sich als Quasi-Standard durchgesetzt. Diese Kombination hat aber auf mobilen Geräten vermutlich einige Nachteile, insbesondere, wenn es sich um Geräte ohne Hardware-Tastatur handelt: Die Eingabe von langen, komplizierten Passwörtern auf Multitouch-Display-Tastaturen dauert vermutlich lange und solche Smartphones sind gegen sogenanntes „Shoulder Surfing“, also Mitlesen der Eingabe durch einen Dritten, vermutlich besonders anfällig:
Um dem Nutzer zu signalisieren, dass er die richtige Taste getroffen hat, zeigen einige Geräte den eingegebenen Buchstaben nochmals groß an, auch für einen „Shoulder Surfer“ gut sichtbar.
Sonderzeichen – für ein sicheres Passwort unerlässlich – lassen sich meist nicht direkt eingeben, sondern erfordern das Umschalten auf eine oder mehrere „Sonderzeichen-Tastaturen“. Dadurch wird die Eingabe komplizierter und dürfte länger dauern.
Diese beiden Kernprobleme der Passworteingabe auf Multitouch-Smartphones sollen in der Arbeit untersucht werden:

- Wie komfortabel können Passwörter eingegeben werden?
- Wie groß ist die Gefahr des Shoulder-Surfing?

Dabei wird überprüft, ob es Unterschiede in diesen Bereichen bei verschiedenen Software-Tastaturen von Multitouch-Smartphones gibt und welchen Einfluss diese Unterschiede auf Usability und Shoulder-Surfing-Sicherheit haben.

1.2. Methodik und Aufbau

Um oben genannte Fragen zu klären, wurde eine Benutzerstudie durchgeführt. Diese bestand hauptsächlich aus einem Usability-Test, der explizit auf die Passworteingabe an Multitouch-Smartphones abgestimmt war, sowie einem Shoulder-Surfing-Experiment, bei dem die Probanden versuchten, Passwörter per Shoulder Surfing mitzulesen. Der Aufbau dieser Studie wird in Kapitel 5 ausführlich erklärt. Zuvor werden die zur Verfügung stehenden Software-Tastaturen sowie eventuell vorhandene Varianten genau analysiert (Kap. 3) und erörtert, wie Passwörter aufgebaut sind und wie sie für die Studie generiert werden können (Kap. 4)
Es wird untersucht, wie schnell verschiedene Passwörter bei den einzelnen Systemen eingegeben werden können, wie fehleranfällig die Systeme sind und wie leicht eine Passworteingabe per Shoulder-Surfing mitzubekommen ist (Kap. 6).
Des Weiteren wird dargestellt, welche Designentscheidungen bei Softwaretastaturen sinnvoll sind und welche zu eher problematischen Ergebnissen führen. Schließlich wird erörtert, worauf man bei der Wahl von Passwörtern, die mobil verwendet werden, achten sollte (Kap. 7).

2. Verwandte Arbeiten

Passworteingabe auf mobilen Geräten ist ein beliebtes Forschungsthema – allerdings geht es meist lediglich um die Nutzerauthentifizierung am Gerät. Diese findet bei Smartphones standardmäßig entweder gar nicht statt[1], oder besteht aus einer einfachen PIN oder einem Passwort. Hierzu werden alternative Ansätze erforscht: beispielsweise fanden Buchoux und Clarke [4] heraus, dass Keystroke-Analyse einen Nutzer über die Art, wie er ein Passwort eingibt, zusätzlich identifizieren kann, wenn das Passwort lang genug ist.

Smartphones mit Touchscreen bieten darüber hinaus weitere interessante Methoden zur Nutzerauthentifizierung: beispielsweise grafische Passwörter, bei denen der Nutzer auf bestimmte Bilder oder Bildteile tippen muss. Nicholson [19] untersuchte einige solcher Systeme, auch in Bezug auf Shoulder Surfing. Dabei führte er auch ein Shoulder-Surfing-Experiment durch: die Probanden mussten sich auf eine von drei festgelegten Positionen stellen und eine Passworteingabe beobachten, die stets von der gleichen Person durchgeführt wurde. Die Probanden mussten sich die Passwörter dabei merken und im Anschluss einige Aufgaben lösen, um das Arbeitsgedächtnis zu leeren. Nicholson kommt zu dem Ergebnis, dass eine einfache PIN-Eingabe sehr anfällig gegen Shoulder Surfing ist, während bei grafischen Passwortsystemen teilweise gar keine Passworteingabe korrekt mitgelesen werden konnte.

Auch Cai und Chen [6] beschäftigten sich hauptsächlich mit der Nutzerauthentifizierung am Gerät: sie stellten im August 2011 eine App für moderne Android-Smartphones vor, die mit Hilfe der Bewegungssensoren versucht, eingegebene Passwörter zu erkennen. Dabei hatten sie bei numerischen Passwörtern auf einer Zifferntastatur bereits eine Erfolgsrate von 70%. Es ist zu vermuten, dass mit noch genaueren Bewegungssensoren in zukünftigen Geräten so auch alphanumerische Passwörter erkannt werden können. Die Website „Is This An Earthquake?“[2] zeigt eindrucksvoll, dass die Daten der

[1] Bei den meisten Systemen ist standardmäßig zum Entsperren keine PIN-Eingabe o.ä. erforderlich.

[2] `http://isthisanearthquake.com`

Bewegungssensoren auch im Webbrowser ausgewertet werden können – eine weitere Bedrohung der Passwortsicherheit in modernen Smartphones, da so theoretisch eine Auswertung der Bewegungssensoren (z.B. während einer Passworteingabe) auch möglich sein könnte, ohne dass der Nutzer dafür aktiv eine Software installieren muss.

Doch es geht hier stets um das Entsperren des Smartphones, um die Authentifizierung des Nutzers gegenüber dem Smartphone selbst. Ein Feature, das bei den meisten Systemen standardmäßig ohnehin deaktiviert ist. Wenn eine Authentifizierung plattformübergreifend möglich sein soll – im Internet – führt nicht viel an der klassischen Benutzername-Passwort-Kombination vorbei. Auch wenn ein Smartphone eine innovative graphische Authentifizierungsmethode verwendet: bei Diensten wie Facebook wird der Nutzer dennoch ein klassisches Passwort eingeben müssen.
Daher müssen andere Wege gefunden werden, um die Sicherheit zu erhöhen. So kann es beispielsweise Sinn ergeben, den Nutzern Tipps für gute Passwörter zu geben oder gar genau vorzuschreiben, wie Passwörter aussehen sollten. Yan, Blackwell, Anderson und Grant [29] fanden heraus, dass mnemonische Passwörter[3] genauso „stark“ sind wie zufallsgenerierte Passwörter und daher einen guten Kompromiss aus Sicherheit und Merkbarkeit bieten.
Tatsächlich verwenden die meisten Nutzer aber schlechte Passwörter, die sie häufig vergessen oder beispielsweise durch Phishing preisgeben [9]. Dazu kommt, dass die Implementierung von Sicherheitsmechanismen selbst häufig unsicher ist, da Webentwickler nicht zwangsläufig Security-Spezialisten sind oder auch ganz bewusst andere Schwerpunkte setzen. [3].

Tari, Ozok und Holden untersuchten 2006 Shoulder-Surfing am PC. Dabei fanden sie heraus, dass vermeintlich „sichere“ Passwörter, die nicht im Wörterbuch stehen, sehr anfällig gegenüber Shoulder Surfing sind. Dagegen erwiesen sich einfache Wörter als relativ resistent gegen Shoulder-Surfing, mutmaßlich aufgrund der höheren Eingabegeschwindigkeit. [25]

Es fällt auf, dass die in den letzten Jahren aufgekommenen Software-Tastaturen bisher in der Forschung kaum Beachtung finden. Untersuchungen zu Passwörtern gehen meist immer noch primär von einer Eingabe an einer klassichen PC-Tastatur

[3] Mnemonische Passwörter sind so aufgebaut, dass sie über Eselsbrücken merkbar sind. Beispielsweise überlegt man sich einen Satz und nimmt von jedem Wort den ersten Buchstaben für das Passwort. Der Satz ist genauso gut zu merken wie ein „einfaches“ Passwort aus nur einem Wort, das resultierende Passwort ist aber schwieriger zu erraten [29].

aus. Hier soll nun untersucht werden, ob die bisherigen Erkenntnisse teilweise auch für Multitouch-Smartphone-Software-Tastaturen bestätigt werden können oder ob aufgrund der anderen Eingabemethode neue Schlüsse gezogen werden müssen.

3. Tastaturanalyse

3.1. Plattformen

In diesem Kapitel sollen die im weiteren Verlauf verwendeten Smartphone-Betriebssysteme und ihre verschiedenen Tastaturvarianten zunächst analysiert werden. Von Interesse sind vor allem die drei „großen“ Plattformen: Android, iOS und Windows Phone. Zusätzlich bestand die Möglichkeit, Nokia-Geräte zu testen – einerseits mit Symbian^3 „Anna“, andererseits mit MeeGo auf dem Nokia N9. Beide Systeme werden jedoch vermutlich nicht mehr in neuen Geräten eingesetzt, zukünftig will Nokia auf Windows Phone als Betriebssystem setzen.[1]

Alle verwendeten Smartphones lassen sich im Hoch- und im Querformat bedienen. Im Querformat wird die Tastatur vergrößert, ändert sich sonst aber nicht wesentlich. Um den Test um die Variable „Ausrichtung“ zu reduzieren, wird im Folgenden grundsätzlich die standardmäßig verwendete Ausrichtung[2] verwendet. Diese ist bei allen Testgeräten das Hochformat.

3.1.1. Android

Android ist die derzeit größte Smartphone-Plattform [1] und wird von der Open Handset Alliance entwickelt, deren Hauptakteur Google ist. Die Software basiert auf dem Linux-Kernel und steht unter einer Open Source Lizenz (Apache 2.0).[22] Es können sowohl Hardwarehersteller als auch Provider eigene Modifikationen vornehmen, was in einer großen Fragmentierung und einer Vielzahl an Softwaretastaturvarianten resultiert. In dieser Arbeit wird vor allem die „offizielle“, unmodifizierte Version direkt von Google berücksichtigt werden. Zusätzlich wird die Software-Tastatur der Sense-Oberfläche untersucht, die auf HTC-Geräten mit Android vorinstalliert ist.

[1] In den USA will Nokia keine Symbian-Geräte mehr auf den Markt bringen [10], zu MeeGo erklärte Nokia-CEO Elop, dass das Unternehmen sich auf Windows-Phone-Geräte konzentrieren werde, selbst wenn das Nokia N9 ein Hit würde.[13]

[2] basierend auf initialem Zustand, Beschriftungen des Geräts und Herstellerfotos

Abbildung 3.1.: Die native Android-Tastatur

Als Ergänzung gibt es mit Swype [24] eine weitere Tastatur, die auf manchen Android-Geräten vorinstalliert ist[3] oder nachträglich installiert werden kann. Der Vorteil von Swype ist, dass zwischen einzelnen Eingaben der Finger nicht abgehoben werden muss: stattdessen kann ein „Pfad“ auf der Tastatur „gezeichnet“ werden, an den Knicken des Pfades wird dann ein Tastendruck erkannt. Die Eingabe von Passwörtern unterstützt Swype dabei nicht, hier muss „normal“ getippt werden.
Swype ist auch auf Symbian-Geräten verfügbar; dort kann es von Nokias Ovi-Store heruntergeladen werden.[21] Da sich das Tastaturlayout und die Bedienung jedoch nicht unterscheiden, wird im Test nur Swype auf Android-Geräten verwendet.

Android - Vanilla

Als Testgerät für das unmodifizierte Android-Betriebssystem dient ein Google Nexus One mit Android-Version 2.3.6.[4] Die Tastatur besteht auf der ersten Seite neben den Buchstaben aus Tasten für die Eingabe von Punkt und Komma (Abb. 3.1). Auch Ziffern können hier direkt eingegeben werden, indem auf die entsprechenden Tasten in der oberen Buchstabenreihe länger gedrückt wird. Außerdem gibt es eine extra Taste, die die Tastatureinstellungen aufruft. Dem Benutzer steht eine Vielzahl an Einstellmöglichkeiten zur Verfügung: neben Tastentönen sind auch Tasten-Vibration, Anzeige eines PopUps des eingegebenen Zeichens und die Anzeige der Einstellungs-Taste aktivier- und deaktivierbar. Sind mehrere Tastaturlayouts aktiviert, so kann über horizontales Wischen auf der Leertaste zwischen diesen gewechselt werden. Zusätzlich gibt es auf Android die Möglichkeit der Spracheingabe. Der Button dafür wird jedoch bei Passwortfeldern automatisch ausgeblendet und hat daher keine Relevanz für diese Arbeit.

[3] u.A. Samsung Galaxy S, Galaxy S II, Galaxy 3 [18]
[4] Im Folgenden ist mit „Android“ jeweils die native Android-Tastatur (Vanilla) gemeint.

Abbildung 3.2.: Android mit Sense-Tastatur.

Android - Sense

HTC, der größte Hersteller von Smartphones mit Android [1], verwendet auf den meisten Geräten eine modifizierte grafische Oberfläche: HTC Sense. Diese beinhaltet auch eine eigene Tastatur, die vom Android-Standard abweicht (Abb. 3.2). Sie bietet eine Vielzahl an Sonderzeichen schon auf der ersten Tastatur-Seite durch längeres Gedrückthalten an.
Alternativ kann auch eine Telefontastatur (Zahlentastatur) oder eine „kompakte QWERTZ"-Tastatur[5] verwendet werden, der Standard ist aber eine „normale" QWERTZ-Tastatur. Einstellbar sind Tastentöne und Tastenvibration. Die Tastatur bietet eine zusätzliche Taste für die Umschaltung der Tastaturlayouts (für verschiedene Sprachen) an, außerdem auch eine Taste, die direkt die Tastatureinstellungen aufruft. Der Gravis (`) ist auf der Tastatur überhaupt nicht vorhanden.

Android - Swype

Swype[6] ist auf dem Nexus One nicht vorinstalliert, jedoch als Beta-Version 3.26 nachinstallierbar. Ähnlich wie bei der Sense-Tastatur sind einige Sonderzeichen über längeres Gedrückthalten schon auf der ersten Tastatur-Seite eingebbar (Abb. 3.3). Es gibt eine Taste für die Swype-Einstellungen. Die Einstellmöglichkeiten decken wiederum Töne und Vibration ab, außerdem kann eingestellt werden, ob der komplette Swype-Pfad dargestellt wird oder nur der letzte Teil des Pfads. Das Verhältnis zwischen Geschwindigkeit und Genauigkeit kann stufenlos eingestellt werden. Für den Test werden die

[5] In den „kompaktes QWERTZ"-Einstellungen werden immer zwei Buchstaben über eine Taste eingegeben: für den zweiten Buchstaben sind dann zwei kurze Taps vonnöten.
[6] im Folgenden: „Android/Swype"

Standardeinstellungen verwendet: es wird nicht der komplette Swype-Pfad dargestellt, und das Verhältnis Geschwindigkeit/Genauigkeit ist ausgeglichen.

3.1.2. iOS

Apple ist der größte Smartphone-Hersteller und iOS nach Android das am häufigsten genutzte Betriebssystem. [1] Es ist die einzige Plattform (neben Blackberry), bei der Software und Hardware aus einer Hand kommen. Als Testgerät für iOS liegt ein iPhone 4S mit Softwareversion 5.0.1 vor.
Auffällig ist, dass die Buchstabentasten im Gegensatz zu allen anderen Tastaturen in dieser Studie stets einen Großbuchstaben anzeigen. Ob tatsächlich ein Großbuchstabe eingegeben wird, ist nur an einer leuchtenden Shift-Taste erkennbar – bei den anderen Tastaturen werden bei aktivierter Kleinschreibung automatisch Kleinbuchstaben auf den Tasten angezeigt.
Einstellbar sind lediglich Tastentöne; Vibration bei Tastendruck bietet das iPhone nicht. Sind mehrere Tastaturlayouts für verschiedene Sprachen aktiviert, wird eine zusätzliche Taste für die Umschaltung zwischen den Layouts angeboten. Das iPhone 4S hat zudem eine Sprachsteuerung, für die es ebenfalls eine extra Taste gibt. Wie auch bei Android ist die Sprachsteuerung aber in Passwortfeldern nicht verfügbar und die Taste wird automatisch ausgeblendet.

3.1.3. Windows Phone

Windows Phone 7 ist das jüngste der drei „großen“ Smartphone-Betriebssysteme[7] und eine Neuentwicklung als Nachfolger von Windows Mobile. Als Testgerät dient das HTC 7 Trophy mit Softwareversion 7.5. Die Tastatur wirkt auf den ersten Blick wie eine Mischung aus nativer Android-Tastatur und iOS-Tastatur (Abb. 3.5). Vibration bei Tastendruck bietet das Testgerät nicht an, einstellbar sind lediglich Tastentöne. Sind Tastaturlayouts für mehrere Sprachen aktiviert, wird ebenfalls eine zusätzliche Taste für den Wechsel eingeblendet.

[7] Tatsächlich lag der Marktanteil von Windows Phone im Quartal 3 in den USA mit nur 1.3% auf dem 7. Platz, wird allerdings in den Medien häufig als die Nummer drei gehandelt, da die tatsächlichen Plätze 3 bis 6 (Blackberry, Windows Mobile, Symbian, WebOS) entweder gar nicht oder nur noch schleppend weiterentwickelt werden und dadurch an Bedeutung verlieren.

Abbildung 3.3.: Swype auf Android

Abbildung 3.4.: Die iOS-Tastatur

Abbildung 3.5.: Die Windows-Phone-7-Tastatur

3.1.4. Symbian^3

Symbian ist das älteste getestete Betriebssystem, jedoch erst seit Version S60 v5 Touchscreen-kompatibel.[2] Die aktuelle Version ist Symbian^3 und wurde von Nokia entwickelt. Zum Test liegen zwei Nokia-Geräte vor, das C7 und das N8.

Symbian^3 - QWERTZ

Erst seit Version Symbian^3 „Anna“ [8] wird dabei im Hochformat eine QWERTY-/QWERTZ-Tastatur angeboten, davor gab es diese nur im Querformat. Zuvor war im Hochformat ausschließlich die alphanumerische T9-Tastatur verfügbar, welche auch immer noch die Standardeinstellung ist. Daher wird im Test zwischen T9-Tastatur[9] (Abb. 3.7) und QWERTZ-Tastatur[10] (Abb. 3.6) unterschieden. Als Testgerät für die QWERTZ-Tastatur wird das Nokia N8 verwendet (aufgrund des etwas größeren Displays).
Konfigurierbar sind lediglich Tastentöne und Vibration.

Symbian^3 - T9

Die T9-Automatik, also automatisches „Raten“ des richtigen Buchstabens der gedrückten Taste, ist in Passwortfeldern sinnvollerweise automatisch deaktiviert, muss also hier nicht weiter berücksichtigt werden. Infolgedessen sind beispielsweise für die Eingabe des Buchstaben „c“ in einem Passwortfeld grundsätzlich drei Taps auf die 2 notwendig. Als Testgerät für die T9-Tastatur wird das Nokia C7 verwendet. Der Gravis (`) ist auf der Tastatur überhaupt nicht vorhanden.

3.1.5. MeeGo

MeeGo ist ein weiteres Smartphone-Betriebssystem, das wie Android auf Linux basiert und hauptsächlich von Nokia und Intel entwickelt wurde. Bislang gibt es nur ein Smartphone, das MeeGo einsetzt: das Nokia N9 – welches laut Nokia auch das einzige bleiben wird [13] und das in Deutschland, den USA und Großbritannien nicht verkauft wird. Die Tastatur bietet ähnlich wie Swype eigene Tasten für deutsche Umlaute an (Abb. 3.8). Einstellbar sind Tastentöne und -vibration. Die geschweiften Klammern ({ und }) sind auf der Tastatur nicht vorhanden.

[8] C7 und N8 wurden mit der Vorgängerversion ausgeliefert; Releasedatum von „Anna“: 18.8.11.[20] Die hier verwendeten Geräte laufen beide unter der neuesten Version.
[9] im Folgenden: „Symbian/T9“
[10] im Folgenden: „Symbian/QWERTZ“

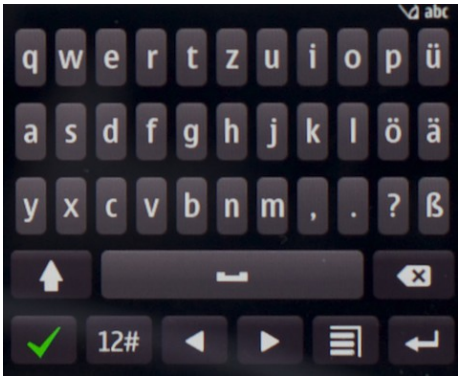


Abbildung 3.6.: Die alternative QWERTZ-Tastatur von Symbian

Abbildung 3.7.: Die standardmäßig aktive T9-Tastatur von Symbian

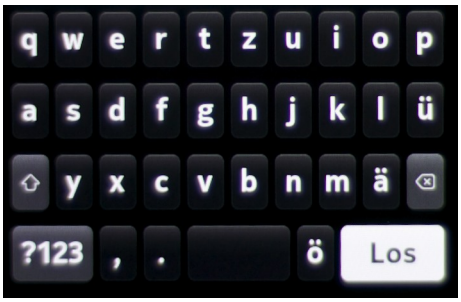


Abbildung 3.8.: Die Tastatur von MeeGo auf dem Nokia N9.

3.1.6. Andere Systeme

Weitere Smartphone-Betriebssysteme sind BlackBerry von RIM und HP webOS. Das BlackBerry-Betriebssystem ist mittlerweile auch in einer Variante für Multitouch-Smartphones ohne physikalisches Keyboard verfügbar. Leider wurde von BlackBerry aber kein Testgerät für diese Studie zur Verfügung gestellt.
HP webOS ist der Nachfolger des Palm OS und ein Smartphone- und Tablet-Betriebssystem, das beispielsweise auf dem Palm Pre und dem HP TouchPad zum Einsatz kam. HP kündigte im August 2011 an, keine weiteren Geräte mit webOS mehr zu entwickeln. [23] Im Dezember 2011 kündigte HP an, webOS als Open Source zur Verfügung zu stellen.[14] Mangels Testgerät kann auch webOS in dieser Studie leider nicht berücksichtigt werden.

3.1.7. Festgelegte Einstellungen für das weitere Vorgehen

Es zeigt sich also, dass eine Vielzahl an Einstellmöglichkeiten vorhanden ist. Für den weiteren Verlauf der Studie werden die Geräte folgendermaßen eingerichtet, um eine Vergleichbarkeit der Ergebnisse zu gewährleisten:

- Es wird grundsätzlich nur das Hochformat verwendet. Der Unterschied der Tastaturen im Hoch- und Querformat ist bei den meisten Plattformen marginal, abgesehen von der Größe der Bildschirmtasten. Außerdem werden alle Testgeräte standardmäßig im Hochformat bedient.

- Tastentöne werden deaktiviert. Gerade beim Shoulder-Surfing-Test liegt das Augenmerk auf dem visuellen Aspekt. Aktive Tastentöne würden das Mitlesen möglicherweise unnötig erleichtern, da die Anzahl der eingegebenen Zeichen gehört werden könnte.

- Tastenvibration wird deaktiviert, um die Vergleichbarkeit zu gewährleisten. Die Funktion wird nicht auf allen Geräten unterstützt; außerdem könnte sie aufgrund des Vibrationsgeräuschs ähnliche unerwünschte Nebeneffekte wie Tastentöne haben.

- Es wird das deutsche Tastaturlayout verwendet. Zusätzlich ist allerdings das US-Englische Layout aktiviert, um eine eventuelle Layout-Umschalttaste anzuzeigen.

System	Testgerät	OS-Version	Auflösung	Bildschirm	Besonderheiten
Android/Sense	HTC Desire	2.2	480x800px	3,7 Zoll	Dolphin Browser
Android/Swype	Nexus One	2.3.6	480x800px	3,7 Zoll	Swype 3.26
Android/Vanilla	Nexus One	2.3.6	480x800px	3,7 Zoll	
iOS	iPhone 4S	5.0.1	640x960px	3,5 Zoll	
MeeGo	Nokia N9	1.2	480x854px	3,9 Zoll	
Symbian/QWERTZ	Nokia N8	025.007	360x640px	3,5 Zoll	QWERTZ aktiviert
Symbian/T9	Nokia C7	022.014	360x640px	3,5 Zoll	Alphanum. Tastatur
Windows Phone	HTC 7 Trophy	7.5	480x800px	3,7 Zoll	

Tabelle 3.1.: Übersicht über die verwendeten Systeme und Geräte.

Dies dürfte der Konfiguration bei einem Großteil der Nutzerschaft entsprechen, spätestens seit es für einige Geräte Emoji-Tastaturlayouts[11] gibt.

- Weitere Einstellmöglichkeiten, wie beispielsweise Eingabehilfen und Pop-up-Anzeige bei Tastendruck, werden auf den Standardeinstellungen der jeweiligen Plattform belassen.

3.1.8. Zusammenfassung

Es stehen also 8 verschiedene Systeme für die Studie zur Verfügung. Dabei werden zwei Systeme, Android/Sense und Android/Swype, auf dem selben Gerät getestet. Die Umschaltung zwischen den Tastaturen ist einfach möglich und stellt daher kein Problem dar.
Die Displaygrößen sind alle ähnlich (siehe Tabelle 3.1), die Unterschiede folglich vernachlässigbar.

3.2. Tastaturanalyse

Nun sollen Gemeinsamkeiten und Unterschiede der einzelnen Tastaturen gefunden und zusammengefasst werden, um im weiteren Verlauf auf die speziellen Eigenschaften von Multitouch-Tastaturen eingehen zu können.
Zunächst wird die Hauptansicht der Tastaturen betrachtet, der initiale Zustand, der den Nutzer erwartet, wenn dieser etwas eingeben möchte.

[11] Japanische Emoticons, seit Version 6.0 Bestandteil von Unicode [26, Seite 500ff.] und z.B. seit Version 5 in iOS verfügbar.

Abbildung 3.9.: Eine übliche Sonderzeichentastatur, hier Windows Phone 7, beide Ansichten

3.2.1. Hauptansicht

Die 26 Haupt-Buchstaben sind stets mit einer eigenen Taste auf der „ersten Seite“ der Bildschirmtastaturen zu finden (Ausnahme: Symbian/T9). Zusätzlich sind Shift, Space und Backspace auf allen Tastaturen zu finden sowie eine Taste, um auf die Sonderzeichen-Tastatur umzuschalten.
Eigene Buttons für Zahlen fehlen bei allen Systemen auf der ersten Seite (Ausnahme ist natürlich Symbian/T9) – bei den drei Android-Varianten können Zahlen durch längeres Drücken der Buttons in der oberen Buchstabenreihe der ersten Tastaturseite eingegeben werden, sie werden dort auch klein angezeigt. Android, Windows Phone und Symbian/QWERTZ bieten Punkt (.) und Komma (,) als eigene Buttons an, Symbian/QWERTZ und Android/Swype auch die deutschen Umlaute (bei den anderen Systemen werden die Umlaute über längeres Gedrückthalten der Tasten der jeweiligen Vokale eingegeben, siehe unten).
Bis auf iOS zeigen alle Tastaturen direkt auf den jeweiligen Buchstabentasten an, ob gerade groß oder klein geschrieben wird. Bei iOS leuchtet bei aktivierter Großschreibung nur die Shift-Taste auf, die Buchstaben auf den Tasten sind stets groß geschrieben.

3.2.2. Sonderzeichen-Ansicht

Auf allen getesteten Tastaturen gibt es eine alternative „Sonderzeichen-Seite“, die die standardmäßig angezeigten Buchstabentasten ersetzt (Abb. 3.9). Über einen Sonderzeichen-Button wird diese Ansicht aufgerufen, über einen Button „ABC“ o.ä. kommt man wieder zurück auf die Buchstabentastaturansicht. Bei iOS wird nach Eingabe eines Sonderzeichens automatisch wieder auf die Buchstabentastatur umgeschalten. Dabei finden sich bei allen Systemen mindestens die Zeichen @, (,), /, :, ;, ?, ! sowie

Abbildung 3.10.: Die Sonderzeichen-Tabelle von Symbian mit T9-Tastatur (Nokia C7)

Punkt und Komma auf der Sonderzeichentastatur – diese beiden Zeichen werden also teilweise redundant angezeigt. Bei den meisten Tastaturen finden sich auch noch &, %, # und Dollar- oder Euro-Zeichen (oder beides).

Auf allen Tastaturen gibt es noch eine zweite Sonderzeichen-Seite, zu der über eine Taste mit Aufschrift „ALT“, „1/2“, einen Pfeil o.ä. oder bei iOS durch die Taste „#+=“ (welche gleich verrät, welche Zeichen dort u.a. zu finden sind) umgeschaltet werden kann. Hier lassen sich bei den meisten Plattformen mathematische Symbole, weitere Klammertypen u.ä. finden. Die genaue Aufteilung der Sonderzeichen auf die zwei Seiten ist von System zu System verschieden.
Bei Symbian/T9 ist die „Sonderzeichen-Tastatur“ (auch optisch) eher eine Zeichentabelle als eine Tastatur – die Zeichen sind nacheinander aufgelistet, auf der zweiten Seite finden sich nur noch zwei Zeilen, die nicht mehr auf die erste Seite passten (Abb. 3.10).

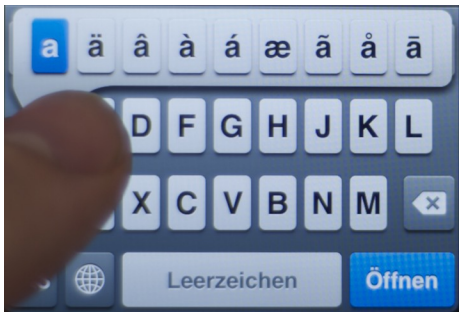


Abbildung 3.11.: PopUp des Buchstaben A bei iOS.

3.2.3. PopUp-Tasten

Bei iOS, Android, Android/Swype und bei Windows Phone sind einige Tasten mehrfach belegt: durch längeres Antippen öffnet sich ein PopUp (Abb. 3.11), in dem z.B. Varianten des Buchstabens mit diakritischen Zeichen zur Eingabe angeboten werden (ä, å, á, æ etc.), bei Android auch Zahlen in der obersten Reihe, bei Android/Swype gar etliche Sonderzeichen. Dadurch ergeben sich oft alternative Wege, um Sonderzeichen einzugeben.
Diese zusätzlichen Funktionen sind lediglich bei Android/Swype immer sichtbar, bei den anderen Tastaturen muss stets geraten werden.

4. Passwörter und Sicherheit

Nach der Analyse der verschiedenen Softwaretastaturen werden nun Passwörter analysiert, um repräsentative, für die Studie verwendbare Passwörter zu erhalten.

4.1. Grundlegendes

Im Internet hat sich zur Authentifizierung von Nutzern eine Kombination aus einem eindeutigen Nutzernamen (oft eine eMail-Adresse) und einem Passwort durchgesetzt. Das Passwort kann dabei im Allgemeinen vom Nutzer frei gewählt werden, was darin resultiert, dass oft einfach zu merkende und einfach einzugebende Trivialpasswörter verwendet werden. Es gibt verschiedene Ansätze, die Nutzer zu sichereren Passwörtern zu bewegen, indem beispielsweise Mindestlänge oder zu verwendende Zeichen vorgegeben werden. Allerdings gibt es teilweise auch wieder Einschränkungen, z.B. dass aus technischen Gründen nur ausgewählte Sonderzeichen verwendet werden dürfen oder gar eine Höchstlänge von Passwörtern.[1]
Viele Dienste lassen weitestgehend eine freie Wahl der Passwörter zu, zeigen aber direkt eine Bewertung der Sicherheit des Passworts durch Ampelfarben an.[2] So wird die Bewertung bei Verwendung von Sonderzeichen und mit zunehmender Länge besser. Dies ist sicher ein Ansatz, der hilft, die Nutzer tatsächlich zu besserer Passwortqualität zu erziehen. Florencio und Herley [9] erhielten in einer großangelegten Web-Passwort-Studie allerdings ernüchternde Ergebnisse – es werden immer noch überwiegend Trivialpasswörter verwendet. Die nun folgende Analyse einiger geleakter Passwörter bestätigt dies.

[1] Beispiel: der Instantmessenger ICQ. Zwar müssen die Passwörter der Nutzer mindestens 6 Zeichen lang sein, aus Groß- und Kleinbuchstaben sowie Ziffern gebildet sein und dürfen nicht den Namen des Nutzers beinhalten; gleichzeitig gilt aber eine Höchstlänge von 8 Zeichen und es dürfen keine Sonderzeichen verwendet werden.[15]
[2] Beispiel: Google. Direkt bei der Wahl des Passworts wird eine Bewertung angezeigt.[12]

4.1.1. Analyse der LulzSec-Passwörter

Am 16. Juni 2011 veröffentlichte die Hackergruppe „LulzSec“ rund 62.000 eMail-Passwort-Kombinationen.[17] Rund 12.000 davon stammen von der Website writerspace.com, über die Herkunft der restlichen Passwörter ist nichts bekannt.[7] Als (vermutlich) einigermaßen aktuelle Ansammlung von im Web verwendeten Passwörtern bietet sich der Leak an, einige statistische Auskünfte über solche Passwörter zu gewinnen – für diese Studie ist vor allem die Häufigkeit von Sonderzeichen interessant.
Dafür wurden zuerst die eMail-Adressen und Trenner mithilfe von regulären Ausdrücken entfernt, außerdem der von LulzSec hinzugefügte Beschreibungstext und die Trennmarkierungen zwischen den Datensätzen. Die dann nur noch Passwörter enthaltende Datei wurde dann auf Häufigkeit von (Sonder-)Zeichen überprüft. Dabei sind einige Unschärfen zu beachten: mindestens eine der gehackten Websites prüfte offenbar nicht, ob die eMail-Adressen gültig sind. Das automatische Entfernen der Adressen war daher nicht unbedingt zu 100% möglich, jedoch wurden die Daten vor der Auswertung nochmals manuell überprüft. Wenn unsicher war, welches Datum Passwort und welches (vermeintliche) eMail-Adresse ist, wurde der Datensatz komplett entfernt – andernfalls wäre zu befürchten gewesen, dass unrealistisch hohe Werte für Punkt und @-Zeichen auftreten. Verbleibende 62.120 Passwörter wurden dann analysiert. Die Häufigkeitsverteilung von alphanumerischen Zeichen enthält dabei keine großen Überraschungen und entspricht ungefähr der Häufigkeitsverteilung von Buchstaben in der englischen Sprache. Die Häufigkeitsverteilung der Sonderzeichen ist in Abbildung 4.1 aufgezeigt (Achtung: logarithmische Skala). Es ist zu beachten, dass die analysierten Passwörter nur aus ASCII-Zeichen bestanden.

4.2. Passwortgenerierung

Für die Durchführung der Studie müssen Passwörter erstellt werden, die die Tester eingeben sollen. Diese Passwörter sollten zufällig generiert sein, aber dennoch untereinander vergleichbar – andernfalls würden Sie die Testergebnisse negativ beeinflussen (schließlich ist es ein Unterschied, ob drei Sonderzeichen hintereinander oder drei Buchstaben hintereinander eingegeben werden müssen). Die Passwörter sollten auf allen Systemen ungefähr gleich schwer eingebbar sein.
Außerdem ist ein ansteigender Schwierigkeitsgrad wünschenswert, um dem Tester schrittweise die zu testende Tastatur näher zu bringen.

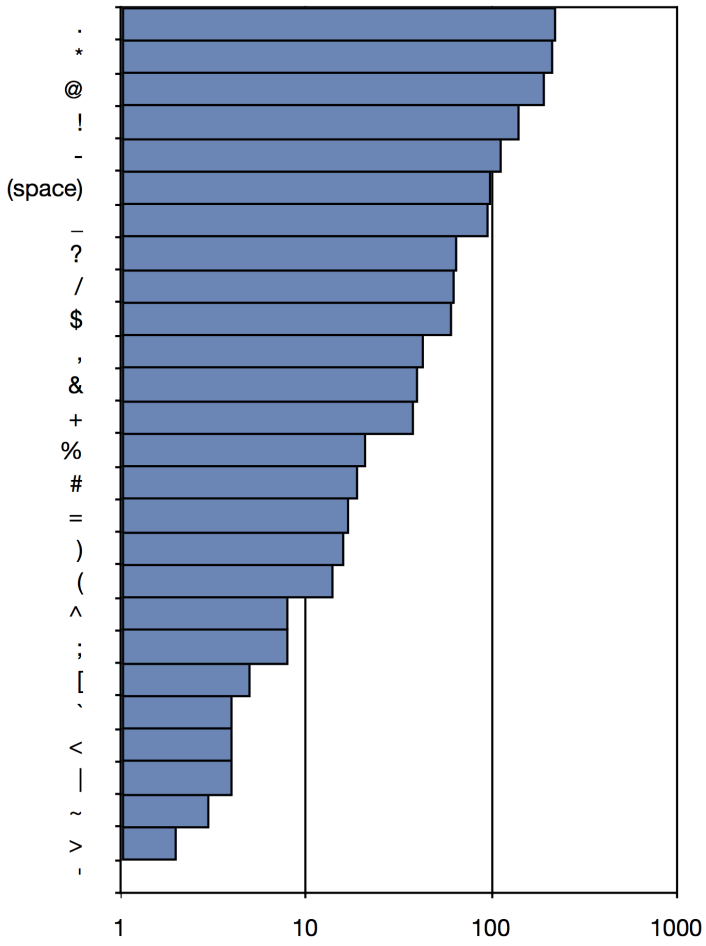


Abbildung 4.1.: Häufigkeit von Sonderzeichen in Web-Passwörtern

Zeichengruppe	Symbol
Großbuchstaben	Z
Kleinbuchstaben	z
Ziffern	0
Sonderzeichen Tier 1 (einfach)	1
Sonderzeichen Tier 2	2
Sonderzeichen Tier 3	3
Sonderzeichen Tier 4 (schwer)	4

Tabelle 4.1.: Zuordnung von Symbolen zu Zeichengruppen

4.2.1. Klassifikation von Zeichen in Passwörtern

Als kleinster gemeinsamer Nenner auf allen Systemen wurden die Zeichen des ASCII-Zeichensatzes gewählt; einzige Ausnahme stellt das Euro-Zeichen dar, das ebenfalls auf allen getesteten Systemen eingebbar ist. Gravis (`) und geschweifte Klammern wurden nicht berücksichtigt, da sie – wie oben erwähnt – nicht auf allen Systemen eingebbar sind.
Zunächst ergeben sich vier Gruppen von Zeichen: Ziffern, Großbuchstaben, Kleinbuchstaben und Sonderzeichen. Die Zeichen jeder Gruppe sind dabei stets ungefähr gleich „schwierig“ einzugeben – außer Sonderzeichen. Hier gibt es erhebliche Unterschiede: so ist das Leerzeichen z.B. stets sehr einfach einzugeben, wohingegen exotischere Zeichen wie die Tilde (~) mehrere Taps benötigen. Um eine Klassifikation für die Sonderzeichen zu erhalten, wurde zunächst für jedes Zeichen auf jedem System ermittelt, wieviele Taps für die Eingabe nötig sind (Tabelle 4.2). Dabei wurde ein einfacher Tap mit dem Wert 1 gezählt, „länger gedrückt halten“ und „Tap mit Swipe“ mit dem Wert 1.5. Sind mehrere Möglichkeiten vorhanden, wurde das Minimum verwendet.
Anschließend wurde für jedes Zeichen ein Mittelwert ermittelt. Diese Werte lagen zwischen 1.0 und 3.0625. So konnten 4 Gruppen von Sonderzeichen festgelegt werden: Zwischen 1.0 und 1.5, zwischen 1.5 und 2.3, zwischen 2.3 und 2.75 sowie zwischen 2.75 und 3.1. Die Zeichen in einer Gruppe sind nun bei allen Systemen ungefähr gleich schwierig eingebbar.
Alle Zeichen konnten nun also einer von 7 Gruppen (Ziffern, Groß- und Kleinbuchstaben sowie die vier Sonderzeichengruppen) zugeordnet werden. Jede Gruppe erhielt ein Symbol (Tabelle 4.1). Damit konnten nun Passwortschemata festgelegt werden.

Zeichen-gruppe	Zeichen	Wkt. in d. Gruppe	Android/ Vanilla	Android/ Sense	Android/ Swype	iOS	MeeGo	Symbian/ QWERTZ	Symbian/ T9	Windows Phone 7	ø
Tier 1	(Leer)	0.27	1	1	1	1	1	1	1	1	1
Tier 1	.	0.61	1	1	1	2	1	1	1	1	1.125
Tier 1	,	0.12	1	1	1	2	1	1	2	1	1.25
Tier 2	?	0.06	2	1.5	1.5	2	2	1	2	2	1.75
Tier 2	@	0.19	2	1.5	1.5	2	2	2	2	2	1.875
Tier 2	!	0.14	2	1.5	1.5	2	2	2	2	2	1.875
Tier 2	-	0.11	2	1.5	1.5	2	2	2	2	2	1.875
Tier 2	/	0.06	2	1.5	1.5	2	2	2	2	2	1.875
Tier 2	)	0.02	2	1.5	1.5	2	2	2	2	2	1.875
Tier 2	(	0.01	2	1.5	1.5	2	2	2	2	2	1.875
Tier 2	;	0.01	2	1.5	1.5	2	2	2	2	2	1.875
Tier 2	&	0.04	2	1.5	1.5	2	2	3	2	2	2
Tier 2	#	0.02	2	1.5	1.5	3	2	2	3	2	2.125
Tier 2	*	0.21	2	1.5	1.5	3	2	2	2	3	2.125
Tier 2	+	0.04	2	1.5	1.5	3	2	2	2	3	2.125
Tier 2	_	0.09	3	1.5	1.5	3	2	2	2	3	2.25
Tier 3	%	0.13	2	1.5	2.5	3	3	3	2	2	2.375
Tier 3	$	0.13	2	1.5	1.5	3	3	3	2	3	2.375
Tier 3	=	0.13	3	3	1.5	3	2	2	2	3	2.438
Tier 3	€	0.13	3	3	2.5	2	3	3	2	2	2.563
Tier 3	\	0.13	3	3	2.5	3	2	2	3	2	2.563
Tier 4	~	0.13	3	3	2.5	3	2	3	3	3	2.813
Tier 4	[	0.15	3	3	2.5	3	3	3	2	3	2.813
Tier 4	]	0.15	3	3	2.5	3	3	3	2	3	2.813
Tier 4	<	0.12	3	3	2.5	3	3	3	2	3	2.813
Tier 4	>	0.09	3	3	2.5	3	3	3	2	3	2.813
Tier 4	^	0.24	3	3	2.5	3	3	3	3	3	2.938
Tier 4	|	0.12	3	3	2.5	3	3	4	3	3	3.063

Tabelle 4.2.: Eingabeschwierigkeiten und Wahrscheinlichkeiten von Sonderzeichen

4.2.2. Passwortschemata

Schließlich wurde ein Generator geschrieben, der als Eingabe ein Passwortschema annimmt und daraus ein Passwort zufällig generiert.
Dabei wurde innerhalb der Sonderzeichengruppen eine Wahrscheinlichkeitsverteilung festgelegt, die den Wahrscheinlichkeiten der analysierten LulzSec-Passwörter entspricht (Tabelle 4.2).

Die Passwortschemata für den Usability-Test (siehe 5.3), mit ansteigendem Schwierigkeitsgrad, wurden wie folgt gewählt:

- Passwort 1: **z00Z2z3** – Zunächst ein Kleinbuchstabe, zwei Ziffern und ein Großbuchstabe, was alles einfach und intuitiv eingebbar sein dürfte. Zum Abschluss noch zwei mittelschwere Sonderzeichen und ein Kleinbuchstabe dazwischen, damit auf jeden Fall auf die erste Sonderzeichen-Seite und zurück gewechselt werden muss. Beispiele: `d78M-f%`, `c11N@q$` und `z59O*h=`.
- Passwort 2: **Z3z00z4Z** – Nun zuerst ein Großbuchstabe und ein Sonderzeichen, dann eine einfachere Folge von Kleinbuchstaben und Zahlen. Am Ende schließlich ein Sonderzeichen der schwersten Stufe und ein Großbuchstabe, wodurch ein Wechsel auf die zweite Sonderzeichen-Seite und ein anschließender Wechsel auf die Buchstabentastatur im Großbuchstaben-Modus nötig wird. Insgesamt ein Zeichen mehr als Passwort 1. Beispiele: `J$h07r]H`, `Q€u92u~W` und `I€w57q<S`.

- Passwort 3: **z4z302Z0z** – Eine Folge von 9 Zeichen: Buchstaben, Ziffern und Sonderzeichen der drei schwereren Kategorien. Es wurde darauf geachtet, dass keine Zeichen der gleichen Gruppe nebeneinander stehen. Beispiele: `r^h€8@A4j`, `c~t\7!U2y` und `h|p$1@C5k`.
- Passwort 4: **0z4Z20zZ2** – Eine Variation von Passwort 3, erstmals mit einer Ziffer am Anfang anstatt einem Buchstaben. Beispiele: `9m|C*7iV+`, `6v[S@6yR/` und `4l>R(1mX-`.
- Passwort 5: **22z1z34Zz** – Nun wurde mit zwei aufeinanderfolgenden Sonderzeichen 2. Ordnung begonnen. Nach einem Kleinbuchstaben wurde ein „einfaches“ Sonderzeichen verwendet, das teilweise direkt auf der Buchstabentastatur eingebbar ist. Schließlich wurden noch zwei aufeinanderfolgende Sonderzeichen höherer Ordnungen verwendet. Beispiele: `*-b b€[Ug`, `;@o.e$]Wc` und `#@v,v=^Yn`.

Auf ein reines alphanumerisches Passwort wurde hier verzichtet. Der Erkenntnisgewinn dürfte minimal sein, da Buchstaben bei allen Systemen (Ausnahme: Symbian/T9) gleich eingegeben werden. Zum „Erlernen“ der Tastatur wurde den Testern vorher extra Zeit gegeben (dazu mehr im folgenden Kapitel 5.3).

Für das Shoulder-Surfing-Experiment (siehe 5.4) wurden drei Passwörter benötigt, die der Testleiter eingibt und die die Probanden mitzulesen versuchen müssen. Diese wurden wie folgt festgelegt:

- Passwort 1: **sunshine** – Zunächst sollte ein Trivialpasswort verwendet werden, das im Wörterbuch steht. „sunshine“ ist eines der am häufigsten verwendeten Passwörter des LulzSec-Leaks (siehe 4.1.1), das diese Anforderung erfüllt.
- Passwort 2: **uMo.37 x** – Das zweite Passwort verwendet das relativ einfach erscheinende Schema zZz1001z und wurde zufällig generiert. Es wurde gewählt, da die beiden Sonderzeichen auf allen Tastaturen relativ schnell eingegeben werden können, was interessante Ergebnisse erwarten lässt.
- Passwort 3: **g<G,9o-l** – Zuletzt sollte ein möglichst schweres Passwort verwendet werden. Es verwendet das Schema z4Z10z2z, wurde jedoch nicht zufällig generiert sondern bewusst gewählt. Es sollten Zeichen verwendet werden, die möglichst leicht verwechselt werden können (g und 9, Komma und Punkt, o und 0, l und I) und keine Buchstabenfolgen beinhalten. Die direkt aufeinanderfolgenden Zeichen 9 und o können auf manchen Tastaturen (z.B. Android/Sense) über die selbe Taste eingegeben werden.

5. Studienentwicklung und -aufbau

Abbildung 5.1.: Ablauf der Studie.

Die beiden Hauptthemen der Arbeit, Usability und Shoulder-Surfing-Security, bilden zugleich die beiden Hauptteile der Studie. In einem Usabilitytest sollten die Probanden vorgegebene Passwörter eingeben, in einem Shoulder-Surfing-Experiment sollten sie versuchen, Passwörter per Shoulder Surfing mitzulesen. Am Anfang, am Ende und zwischen den beiden praktischen Teilen waren Fragebögen auszufüllen, die die praktischen Teile einführen und abschließen (Abb. 5.1).

Es wurde ein Inter-Group-Test durchgeführt, d.h. jedem Probanden wurde ein System zur Bewertung zugewiesen. Jeden Probanden jedes System testen zu lassen hätte zwar in einer niedrigeren Zahl der benötigten Probanden resultiert, allerdings hätte die erforderliche Zeit der Probanden massiv zugenommen und es hätte die Gefahr bestanden, dass die Probanden die teilweise recht ähnlichen Systeme verwechseln. Für das Shoulder-Surfing-Experiment war außerdem wichtig, dass die Probanden die zu erspähenden Passwörter noch nicht kannten, es hätten also für jedes System unterschiedliche Passwörter verwendet werden müssen.
Bei einem Inter-Group-Test können sich die Probanden voll auf ein System konzentrieren und müssen nicht mehrere Tastaturen erlernen. Außerdem kann möglichst ein System zugeteilt werden, mit dem der Proband bereits etwas Erfahrung hat, wodurch sich die Einarbeitungszeit verkürzt.

5.1. Hypothesenbildung

Um die Usability der verschiedenen Systeme bewerten zu können, sind zum einen die Messergebnisse der Passworteingabe und zum anderen die Auswertung eines

Usability-Fragebogens interessant. Gemessen werden kann bei einer Passworteingabe die Eingabedauer der einzelnen Zeichen eines Passworts, daraus resultierend die Gesamteingabedauer, außerdem die Anzahl der gemachten Eingabefehler (auch wenn sie korrigiert wurden). Hier ist interessant, ob sich signifikante Unterschiede bei den Systemen erkennen lassen. Daraus ergeben sich folgende Hypothesen:

- **H_1:** Es gibt signifikante Unterschiede in der Eingabegeschwindigkeit von Passwörtern zwischen den Systemen.
- **H_2:** Es gibt signifikante Unterschiede bei der Anzahl der Eingabefehler zwischen den Systemen.
- **H_3:** Es gibt signifikante Unterschiede bei der Usability-Bewertung von den Probanden zwischen den Systemen.

Beim Shoulder-Surfing lässt sich prüfen, welche Teile der Passwörter korrekt mitgelesen wurden. In einem Fragebogen vor und nach dem Experiment können die Probanden einschätzen, wie sicher das System gegen Shoulder-Surfing ist. Außerdem werden sie befragt, welche Shoulder-Surfing-Strategie sie angewendet haben. Daraus ergeben sich folgene Hypothesen:

- **H_4:** Shoulder-Surfing ist bei den verschiedenen Systemen signifikant unterschiedlich leicht.
- **H_5:** Es gibt signifikante Unterschiede bei der Bewertung der Sicherheit gegen Shoulder-Surfing von den Probanden zwischen den verschiedenen Systemen.
- **H_6:** Es gibt signifikante Unterschiede des Erfolgs bei unterschiedlichen Shoulder-Surfing-Strategien.

Es sollen also 6 Hypothesen überprüft werden. Die entsprechenden Nullhypothesen werden durch die Negierung der jeweiligen Aussagen gebildet, also dass es jeweils keine signifikanten Unterschiede gibt.

5.2. Einführender Fragebogen

Jeder Proband wurde zunächst darum gebeten, einen einführenden Fragebogen an einem bereitgestellten PC einzugeben. Dort wurde er zuerst über das Thema der Studie und die Verwendung seiner Daten aufgeklärt (Abb. A.1). Dann folgte ein Fragebogen, der allgemeine demographische Daten – Alter, Geschlecht und Beruf/Studienfach –

abfragte. Außerdem sollten die Probanden angeben, wie gut sie sich mit den verschiedenen Systemen und Technologien auskennen, sowie welches Mobiltelefon sie selbst hauptsächlich verwenden. Dafür wurden einfache 7-Punkt-Likert-Skalen ohne „keine Angabe“-Item verwendet (Abb. A.2) – kennt der Proband ein System oder eine Technologie überhaupt nicht, wäre das Item ganz links das richtige („gar keine Erfahrung“), ein zusätzliches „keine Angabe“-Item wird dadurch überflüssig. Anschließend wurde dem Probanden vom System eine eindeutige Tester-Nummer zugewiesen und der nachfolgende Usabilitytest erläutert. Außerdem wurde er aufgefordert, eine eventuell vorhandene Brille aufzusetzen, damit im Folgenden davon ausgegangen werden kann, dass der Proband ausreichend gut sieht.

Währenddessen wurden die Antworten sofort ausgewertet, um dem Probanden ein System zuzuweisen. Dabei wurde bevorzugt ein System ausgegeben, das der Proband kennt und selbst verwendet. Leider war dies nicht bei allen Systemen gleichermaßen möglich, da beispielsweise kein Proband ein Windows Phone besaß. Dieses System wurde dann vornehmlich an Probanden ausgegeben, die allgemein angaben, sich mit vielen Systemen auszukennen, da diese sich vermutlich schnell einarbeiten könnten. Den Probanden, die angaben, gar kein Smartphone zu besitzen, wurde, wenn sie zumindest etwas Erfahrung hatten, Symbian/T9 (aufgrund der Ähnlichkeit der Tastatur mit einer klassischen Hardware-Telefontastatur), andernfalls zufällig ein System zugewiesen.

5.3. Usability-Test

Nun sollte jeder Proband sich zunächst mit der Tastatur des ihm zugewiesenen Geräts vertraut machen. Dafür stand ein Freitextfeld zur Verfügung. Er konnte auch einen vorgegebenen Blindtext, der einige Sonderzeichen enthielt, abtippen.[1] Dabei wurde explizit darauf hingewiesen, dass er gleich Passwörter eingeben werden muss und er sich daher auch mit der Eingabe von Sonderzeichen vertraut machen soll. Wenn der Proband selbst der Meinung war, mit der Tastatur zurecht zu kommen, konnte er selbstständig fortfahren. Ihm wurde nun ein erstes Passwort angezeigt, das gemäß Passwort-Schema 1 (siehe 4.2.2) generiert wurde. Dieses Passwort musste er in ein unterhalb angezeigtes Passwort-Eingabefeld eingeben und anschließend auf den But-

[1] Der Blindtext lautete: „Hallo & willkommen! Ich bin ein Blindtext - das bedeutet, dass ich keinen Sinn habe/ergebe. Bananen kosten < 40,- €(das sind $51.04 inkl. 19% MwSt.)“

Abbildung 5.2.: Passworteingabemaske (hier auf iOS). Analog für alle 5 Passwörter im Usabilitytest.

ton „Senden“ drücken (Abb. 5.2). Diese Prozedur wurde für alle fünf Passwortschemata wiederholt.
Zuvor wurde dem Probanden erklärt, dass Geschwindigkeit (H_1) und Genauigkeit (H_2) im Hintergrund gemessen werden und dass er Tippfehler korrigieren darf. Es fand keine Überprüfung statt, ob das Passwort korrekt eingegeben wurde, da den Probanden keine Gelegenheit gegeben werden sollte, ein Passwort mehrfach einzugeben – dies hätte möglicherweise zu einem Bias bei der Messung der Eingabezeit geführt, da die Eingabe beim zweiten Mal vermutlich schneller vonstatten gegangen wäre.
Über dem Eingabefeld war vermerkt, dass zwischen Groß- und Kleinschreibung unterschieden wird, und dass das Passwort auch Sonder- und Leerzeichen enthalten kann.

Anschließend kehrte der Proband wieder zurück an einen PC, um den zweiten Fragebogen auszufüllen. Dieser basierte auf dem PSSUQ (Post-Study System Usability Questionnaire), einem allgemeinen Usability-Fragebogen, der bei IBM entwickelt wurde. Er besteht aus 19 Aussagen, zu denen der Proband jeweils angeben muss, wie stark er der Aussage zustimmt. Die Fragen sind vier Gruppen zugeordnet, die die Komponenten „Nützlichkeit“, „Informationsqualität“, „Interfacequalität“ und „Gesamt“ bilden. Dadurch ergibt sich nicht nur eine Bewertung der Usability insgesamt, sondern es lassen sich

auch Aussagen darüber treffen, welche Bereiche der Usability wie gut bewertet werden. Der Fragebogen ist ein bewährtes Instrument mit exzellenter Reliabilität (Cronbachs $\alpha > 0.9$ für alle Komponenten), der darauf ausgelegt ist, nach einem Usabilitytest ausgefüllt zu werden. [16]
Daher ist der PSSUQ perfekt für diese Studie geeignet (Bewertung von H_3). Die Fragen wurden lediglich ins Deutsche übersetzt und an den Test angepasst (so wurde “the system” mit „die Tastatur“ übersetzt). Jede Frage wurde mithilfe einer 7-Punkt-Likert-Skala beantwortet; es bestand stets auch die Möglichkeit, „keine Angabe“ zu machen, außerdem konnte zu jeder Frage ein Kommentar geschrieben werden, wie das für den PSSUQ vorgesehen ist. Die angefertigte deutsche Übersetzung des PSSUQ ist in Abb. A.3 und Abb. A.4 dargestellt.

5.4. Shoulder-Surfing-Test

Direkt im Anschluss wurde dem Probanden erklärt, was man unter „Shoulder Surfing“ versteht („das Mitlesen eines Passworts, während es an einem öffentlichen Ort eingegeben wird“) und sie wurden um eine Einschätzung der Sicherheit der Tastatur in Bezug auf Shoulder Surfing gebeten (H_5, Teil 1).
Dafür wurden vier Aussagen vorgegeben (in Abb. A.5 dargestellt), denen der Proband auf einer 7-Punkt-Likert-Skala zustimmen musste. Wie auch beim PSSUQ bestand die Möglichkeit, „keine Angabe“ zu machen, sowie einen Kommentar zu schreiben. Anschließend wurde dem Probanden das nachfolgende Shoulder-Surfing-Experiment erläutert:

Zunächst sollte das Shoulder-Surfing-Experiment ähnlich wie bei Nicholson [19] durchgeführt werden – die Shoulder-Surfer sollten sich das Passwort merken, einige Aufgaben lösen, um das Arbeitsgedächtnis zu leeren, und es anschließend eingeben. In einem prototypischen Vorab-Test für diese Studie wurde dabei jedoch von keinem Probanden erfolgreiches Shoulder Surfing durchgeführt. Problematisch bei diesem Versuchsaufbau ist auch, dass die Merkfähigkeit des Probanden eine starke Störvariable ist. Daher wurde für diese Studie ein etwas anderer Versuchsaufbau verwendet:

Dabei gab der Testleiter nacheinander drei Passwörter ein (siehe 4.2.2). Der Proband konnte wählen, ob er sich hinter ihn links, in die Mitte oder rechts stellen möchte (Abb. 5.4) und bekam eine Notizmöglichkeit (Abb. 5.5). Das Ziel des Probanden: alle

Abbildung 5.3.: Eingabemaske für den Testleiter im Shoulder-Surfing-Experiment.

Passwörter mitzulesen. Zwischen jedem Passwort hatte er kurz Zeit, um sich Notizen zu machen. Der Testleiter saß stets in der gleichen Position und hatte die drei Passwörter im Vorfeld auf allen Systemen eingeübt, um sie möglichst flüssig eingeben zu können (Abb. 5.3). Die Passwörter wurden stets mit dem Daumen der rechten Hand eingegeben, die Hand und das Gerät wurden auf einem Tisch abgestützt. Die Voraussetzungen waren also für alle Probanden gleich, und der Erfolg des Shoulder-Surfing dürfte nur vom System und der Shoulder-Surfing-Strategie des Probanden abhängen.
Anschließend sollte der Proband die mitgelesenen Passwörter am gleichen Gerät eingeben (Abb. 5.6). Dabei hatte er maximal 3 Versuche pro Passwort – war ein Passwort schon beim 1. oder 2. Versuch korrekt, wurde direkt zum nächsten Passwort übergegangen. Im Hintergrund wurden die Eingabeversuche aufgezeichnet, um später genau nachvollziehen zu können, wieviele Versuche der Proband für jedes Passwort benötigte und welche Zeichen korrekt eingegeben wurden (H_4). Da die Eingabegeschwindigkeit hier nicht von Interesse ist, wurde nicht wie beim Usabilitytest schon während des Eingabeversuchs die Eingabe gespeichert, sondern wie bei einem „regulären“ Passworteingabefeld erst nach dem Absenden des Passworts.

Danach wurde der Proband nochmals um seine Einschätzung der Shoulder-Surfing-Sicherheit gebeten (H_5, Teil 2). Dafür wurde ein ähnlicher Fragebogen wie vor dem

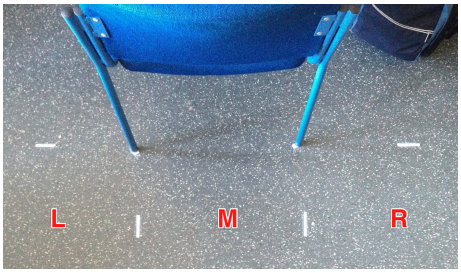


Abbildung 5.4.: Die drei möglichen Shoulder-Surfing-Zonen.

Tester-Nr.:

Bitte versuchen Sie, die drei Passwörter mitzulesen. Sie können sich hier Notizen machen.
Die Anzahl der Platzhalter hat nichts mit der Länge der Passwörter zu tun, sondern dienen lediglich als Hilfestellung.

Passwort 1:

_ _ _ _ _ _ _ _ _ _

Passwort 2:

_ _ _ _ _ _ _ _ _ _

Passwort 3:

_ _ _ _ _ _ _ _ _ _

Abbildung 5.5.: Notizzettel für das Shoulder-Surfing

Abbildung 5.6.: Eingabemaske für den Probanden im Shoulder-Surfing-Experiment.

Experiment verwendet (siehe Abb. A.6 – zwei Fragen wurden direkt übernommen, eine wurde vom Konjunktiv in den Indikativ umformuliert, die Frage nach dem Einsatz in öffentlichen Verkehrsmitteln wurde zu einer Frage nach dem persönlichen Sicherheitsempfinden geändert). Außerdem wurde der Proband nach seiner Shoulder-Surfing-Strategie gefragt (H_6). Dafür konnte er für die drei Komponenten „Fingerposition“, „Tastaturanzeige“ und „Eingabefeld“ jeweils auf einer 7-Punkt-Likert-Skala ohne „keine Angabe“-Item und ohne Kommentarfeld angeben, wie stark er beim Shoulder Surfing auf die jeweilige Komponente geachtet hatte. Es stand außerdem noch ein Freitextfeld zur Verfügung, in dem der Proband weitere Kommentare zum Shoulder-Surfing abgeben konnte.

Abschließend wurde der Proband noch um Angaben zu seinem Gesamteindruck der Tastatur in Bezug auf alle Aspekte gebeten (Abb. A.7). Dafür standen wiederum zwei 7-Punkt-Likert-Skalen inklusive Kommentarfeld, aber ohne „keine Angabe“-Item zur Verfügung, mit denen der Proband seine Zustimmung zu den Aussagen „Insgesamt ist die Tastatur sehr gut zur Passworteingabe geeignet.“ sowie „Ich würde ein Gerät mit dieser Tastatur weiterempfehlen.“ angeben konnte. Hier bestand also für den Probanden die Möglichkeit, eine eventuell im Usability-Teil zum Ausdruck gebrachte Meinung nach Durchführung des Shoulder-Surfing-Experiments zu revidieren.
Zuletzt konnte der Proband noch bei Interesse an den Ergebnissen der Studie eine

eMail-Adresse angeben. Diese eMail-Adressen wurden in einer separaten Datenbanktabelle ohne Bezug zu den restlichen Testergebnissen gespeichert.
Damit war die Teilnahme an der Studie für den jeweiligen Probanden abgeschlossen. Als Aufwandsentschädigung erhielt er eine Tafel Schokolade.

5.5. Testsoftware

Die drei Fragebogenteile sollten digital an bereitgestellten PCs ausgefüllt werden, um die Daten direkt digital zu speichern.
Dafür wurde auf HTML-5-Formularelemente zurückgegriffen: dies hatte den Vorteil, dass vom Browser (es kam Google Chrome 18 zum Einsatz) ohne zusätzliches JavaScript sichergestellt wurde, dass Pflichtfelder ausgefüllt werden oder dass z.B. das Alter als Zahl angegeben wurde. Außerdem konnten auf einfache Art Platzhalterwerte angezeigt werden (dies kam bei den Fragen zu Beruf/Studienfach und zum eigenen Mobiltelefon zum Einsatz, siehe Abb. A.2). Zusätzlich konnte per HTML-Attribut die standardmäßig aktive Autovervollständigung des Browsers deaktiviert werden, um z.B. bei der Altersangabe nichts über andere Probanden, die die Fragen zuvor am selben PC ausgefüllt hatten, zu verraten. Bei den Eingabefeldern für Studienfach/Beruf und eigenem Mobiltelefon wurde die Funktion explizit aktiviert gelassen, um Tippfehler zu vermeiden und die Daten möglichst bereits zu vereinheitlichen.

Als Backendtechnologie wurde PHP 5 verwendet; die Daten wurden in einer SQLite-3-Datenbank abgelegt. Die Software lief auf einem Webserver der Universität Ulm.
Auch für die Messsoftware auf den Geräten bot sich eine Webanwendung aufgrund der Plattformunabhängigkeit an. Zwar hätten mit der Alternative, nativen Testanwendungen, möglicherweise bei manchen Systemen mehr Daten gesammelt werden können, beispielsweise wie ein Zeichen bei redundanten Eingabemöglichkeiten auf einer Tastatur tatsächlich eingegeben wurde; allerdings wären die Ergebnisse möglicherweise aufgrund der unterschiedlichen Implementierungen nicht vergleichbar gewesen oder nur bedingt auf die Eingabe von Web-Passwörtern übertragbar, die nunmal im Browser eingegeben werden. Mit einer reinen Webanwendung können theoretisch Messungen an beliebigen internetfähigen Geräten vorgenommen werden. Aufgrund der stets gleichen Messmethode sind die Ergebnisse vergleichbar.
Die Messsoftware lief auf dem selben Server wie die Fragebögen, dadurch konnten die gesammelten Daten direkt verknüpft werden.

Als Schriftart für die Passwortanzeige im Usability-Test wurde die Standard-Monospace-Schriftart des jeweiligen Systems verwendet, um eine gute Lesbarkeit zu gewährleisten. Für das Passwort-Eingabefeld wurde eine Größe gewählt, die auf allen Systemen gut lesbar ist.

Für den Usabilitytest wurde per JavaScript mit dem JQuery-Framework ermittelt, was der Proband in das Passwortfeld eingibt. Hier kam ein JQuery-Plugin [8] zum Einsatz, das Eingaben sehr zuverlässig erkannte (es verwendet das HTML-5-JavaScript-Event „oninput“ [28] und einige Fallback-Lösungen für Browser, die dies noch nicht unterstützen). Jede Eingabe wurde zusammen mit einem Timestamp (in Millisekunden) an die Datenbank übertragen und gespeichert. Die Eingabegeschwindigkeit konnte dann durch subtrahieren des früheren vom späteren Timestamp für jedes Zeichen ermittelt werden. Die Summe ergibt dann die Gesamteingabezeit des Passworts.

Ein Problem gab es hier bei Android/Sense. Der Sense-Browser behandelt HTML-Input-Felder offenbar anders als die restlichen Systeme, was zur Folge hat, dass der Inhalt per JavaScript nicht zuverlässig auslesbar ist. Lediglich das erste Zeichen in einem Feld ist zugreifbar, selbst der Fallback-Code, der per Timer in regelmäßigen Abständen den Inhalt des Eingabefeldes ausliest, führte zu keinem Ergebnis.
Als Workaround wurde der alternative Browser „Dolphin“ installiert, bei dem die HTML-Input-Felder normal behandelt werden und die Test-Webanwendung funktionierte. Auf das Testergebnis hat dies keinen Einfluss, da natürlich dennoch die Sense-Tastatur zum Einsatz kommt.

Für das Shoulder-Surfing-Experiment wurden die Passworteingabe-Versuche des Probanden sowie die Anzahl der benötigten Versuche gespeichert.
Die Position, die der Proband für das Shoulder Surfing wählte (Links, Mitte oder Rechts) wurde auf seinem Notizzettel notiert und nach Abschluss der Studie manuell in die Datenbank eingetragen.

Die Identifikation der einzelnen Probanden und Zuordnung der gesammelten Daten erfolgte über die dem Probanden zu Beginn zugeordneten „Tester-Nummer“. Diese wurde beim zugewiesenen Testgerät, bei den PCs, an denen der Proband Fragebogen II und III ausfüllte, sowie auf dem Notizzettel für das Shoulder-Surfing-Experiment jeweils vom Testleiter oder einem der Testbetreuer angegeben. Hier wurde auf Einfachheit und schnelle Reaktionsfähigkeit des Studienaufbaus auf die Probanden Wert gelegt, so

wurde vom System der jeweils nächste Fragebogen nach Eingabe der Testernummer automatisch aufgerufen.

5.6. Ort der Studie und technische Begebenheiten

Die gesamte Studie wurde in einem dafür reservierten Raum am Institut für Medieninformatik der Universität Ulm Ende Januar 2012 durchgeführt. Für das Ausfüllen der Fragebögen stand ein Desktop-PC mit Windows 7 zur Verfügung, der per Ethernet-Kabel mit dem Internet verbunden war. Außerdem wurde ein mobiler Macintosh-Computer mit OS X 10.7 mit externem Bildschirm, Tastatur und Maus versehen und als zweite Fragebogen-Station angeboten. Dieser war per Wireless LAN mit dem Internet verbunden.
Es wurde auf beiden Computern der Browser Google Chrome 18 (zum Zeitpunkt der Studie noch in einer Developer-Preview-Version) installiert und verwendet, da dieser alle benötigten HTML-5-Features bot. Die Fragebogen-Websites wurden im Vollbildmodus angezeigt, um die Probanden nicht unnötig abzulenken.
Auf der anderen Seite des Raumes war der Aufbau für das Shoulder-Surfing-Experiment, bestehend aus einem Stuhl, einem Tisch und Bodenmarkierungen, die die Zonen, in denen die Probanden stehen durften, markierten, sowie die Position des Stuhls (Abb. 5.4). Von den Studienbetreuern wurde darauf geachtet, dass eventuell schon anwesende andere Probanden das Shoulder-Surfing-Experiment nicht beobachten, um diesen keinen Vorteil zu verschaffen - sie konnten währenddessen auf der anderen Seite des Raumes Fragebögen ausfüllen oder vor der Tür warten.
Die Testgeräte wurden ohne Mobilfunk betrieben und über Wireless LAN mit dem Internet verbunden. Dafür wurde das öffentliche Wireless LAN der Universität Ulm verwendet, in dem der Universitäts-Webserver ohne weitere Authentifizierung erreichbar ist.

5.7. Probandenanwerbung

Die Studie wurde zeitgleich mit einer ähnlichen Studie durchgeführt, wodurch Probanden gemeinsam angeworben werden konnten. Es wurden Plakate aufgehängt und Flyer verteilt. Außerdem wurden eMails über diverse Mailinglisten der Universität versen-

det. Auf einer dafür eingerichteten Website[2], die auf dem Werbematerial angegeben war, wurden Details zu den Studien erklärt und Links auf Doodle-Umfragen gesetzt, bei denen potentielle Probanden sich einen Termin reservieren konnten. Während der Studiendurchführung wurden auch Studierende und Universitäts-Mitarbeiter direkt angesprochen.

[2] `http://www.uni-ulm.de/?multitouch`, Abruf: 31.1.2012

6. Auswertung

6.1. Probanden

Es konnten insgesamt 80 Probanden gewonnen werden. Dadurch ergaben sich pro getestetem System 10 Probanden. Die 80 Personen teilen sich auf 21 Frauen und 59 Männer auf. Sie waren ungefähr gleichmäßig auf die Gruppen aufgeteilt (Abb. B.1). Es handelte sich überwiegend um Studierende und wissenschaftliche Mitarbeiter aus den Fächern Informatik und Medieninformatik (Abb. B.2). Dies erklärt auch die Altersverteilung der Probanden (Abb. B.3).

2 Probanden (2,5%) gaben an, gar kein Mobiltelefon zu besitzen. 23 Probanden (28,75%) gaben an, kein Smartphone, aber ein herkömmliches Mobiltelefon zu besitzen. Die restlichen 55 Personen, die angaben, ein Smartphone zu besitzen (68,75%), waren rund zur Hälfte Android-Nutzer (27 Personen, 33,75% aller Probanden). 10 Probanden gaben an, iOS zu verwenden (12,5% aller Probanden), der Rest (18 Personen, 22,5% aller Probanden) verwendete ein anderes Smartphone (Abb. 6.1).

Das Ziel bei der Zuweisung des zu testenden Systems war, möglichst jedem Probanden ein System zu geben, mit dem er etwas Erfahrung hat. Dies würde die notwendige Einarbeitungszeit verkürzen und realistischere Messwerte ergeben, da die Probanden schon Übung im Umgang mit der Tastatur hätten. Es sollte möglichst gemessen werden, wie gut sich Passwörter im Alltagsgebrauch eingeben lassen, und nicht, wie gut mit einem neuen, unbekannten Gerät.
Probanden, die angaben, bislang keine oder nur wenig Erfahrung mit Multitouch-Smartphones zu haben, wurde bevorzugt Symbian/T9 zugewiesen, da die Tastatur vom Aufbau her ähnlich wie herkömmliche Mobiltelefontastaturen ist. Die exotischeren Systeme, wie MeeGo und Windows Phone, mit denen kaum Probanden Erfahrung hatten, wurden zufällig zugewiesen.

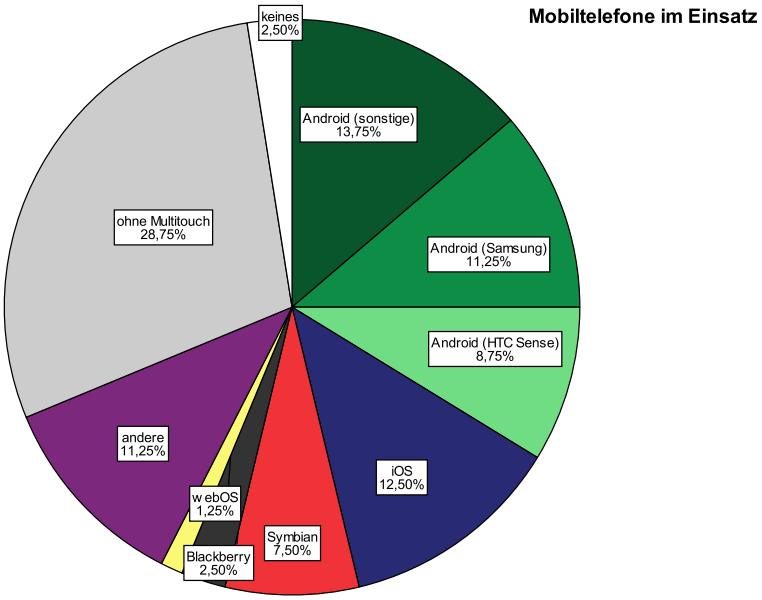


Abbildung 6.1.: Von den Probanden verwendete Systeme.

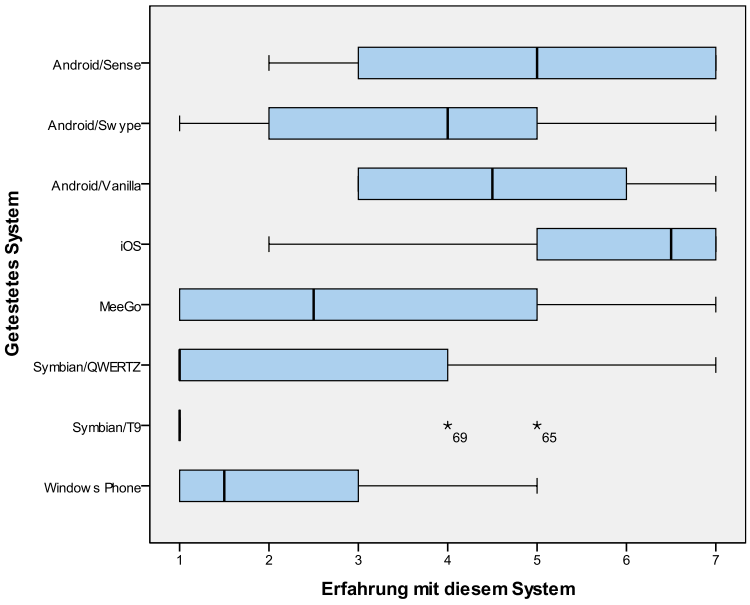


Abbildung 6.2.: Auswertung der Angaben zur Erfahrung mit dem getesteten System.

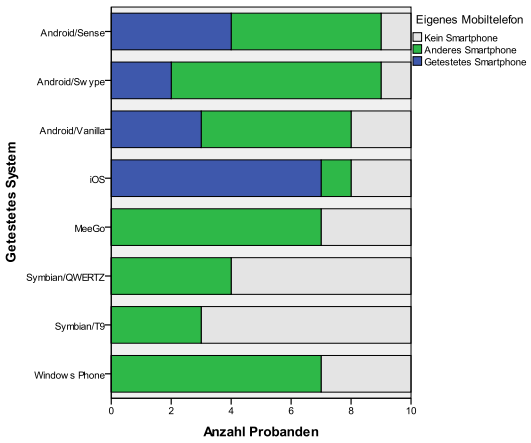


Abbildung 6.3.: Besitzverteilung in den Gruppen.

Abb. 6.2 zeigt die Auswertung zur Erfahrung mit dem jeweils getesteten System. Dies deckt sich mit Abb. 6.1 – die Probanden hatten mit Android- und iOS-Systemen mehr Erfahrung. Die Konzentration bei Symbian/T9 um den Wert 1 („keine Erfahrung“) ist mit oben beschriebener Direktive zu erklären, dass Nicht-Multitouch-Smartphone-Besitzer bevorzugt Symbian/T9 zugewiesen bekamen.

In Abb. 6.3 ist abgetragen, wieviele Probanden pro Testsystem selbst ein Gerät mit diesem System besitzen, wieviele ein anderes Smartphone besitzen und wieviele gar kein Smartphone besitzen. Demzufolge ist ein Bias bei Android- und vor allem iOS-Systemen möglich, da davon auszugehen ist, dass Probanden ihr eigenes System tendenziell besser bewerten.

6.2. Usability

Nun folgt die Auswertung des Usability-Teils der Studie: Eingabegeschwindigkeit, Eingabefehlerzahl und Bewertung im Usability-Fragebogen.

6.2.1. Eingabegeschwindigkeit (H_1)

Zunächst wurde hier ermittelt, wie sich die fünf verwendeten Passwortschemata insgesamt unterscheiden.

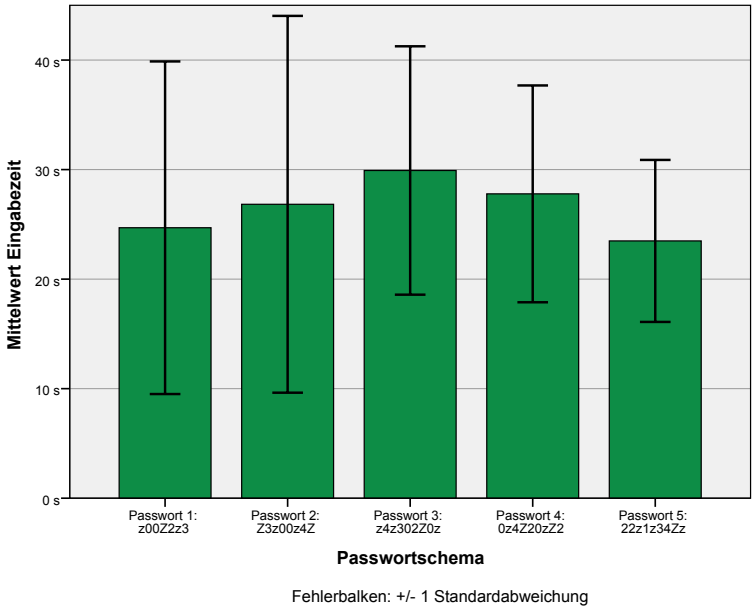


Abbildung 6.4.: Eingabezeit der Passwortschemata über alle Systeme.

Es ist zu beachten, dass nur Messwerte berücksichtigt wurden, bei denen das Passwort am Ende korrekt abgesendet wurde. Dadurch werden zu kurze Messwerte herausgefiltert, die entstanden sind, wenn Probanden ein Passwort nicht komplett eingeben wollten oder konnten. Außerdem ist die ermittelte Gesamtzeit die Zeit, die zwischen der Eingabe des ersten und des letzten Zeichens verstrichen ist. Die Zeit zwischen Anzeige des Passworts und Eingabe des ersten Zeichens wurde nicht berücksichtigt, da sie sehr stark variiert – offenbar haben einige Tester sich das Passwort zunächst eingeprägt, um es dann schneller eingeben zu können, und daher länger gewartet, bis sie mit der Eingabe begannen.
Betrachtet man die Mittelwerte über alle Systeme hinweg, ist erkennbar (Abb. 6.4), dass die Zeit zunächst ansteigt – die Passwörter werden also, wie gewünscht, schrittweise schwerer. Die Abnahme des Werts bei den beiden letzten Passwörtern lässt sich mit dem Lerneffekt erklären – nach dem dritten Passwort waren die Probanden offenbar soweit mit der Tastatur vertraut, dass sie wieder schneller wurden, obwohl die Passwörter eher noch komplexer wurden (die Zahlenwerte sind in Abb. B.4 zusammengefasst.)

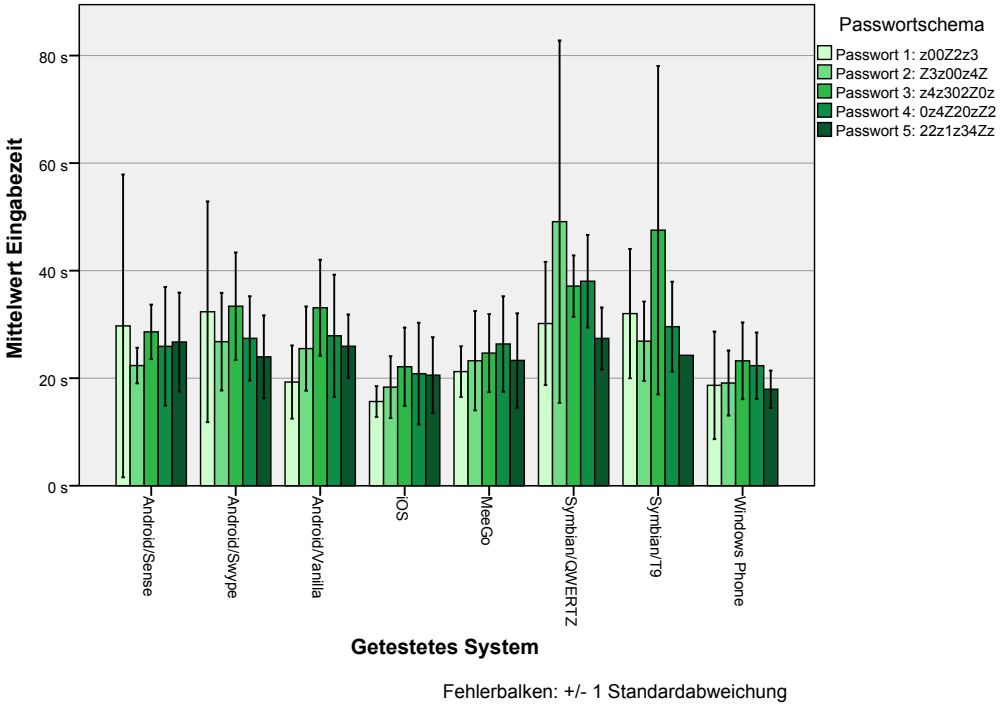


Abbildung 6.5.: Eingabezeit der Passwortschemata pro System.

Betrachten wir nun die Eingabegeschwindigkeit nach Systemen aufgeteilt. (Abb. 6.5) Die Werte der Variable „Eingabezeit“ sind nicht normalverteilt.[1] Daher bietet sich als Signifikanztest der nichtparametrische Kruskal-Wallis-Test an. Dieser ergibt, dass es einen statistisch signifikanten Unterschied[2] zwischen den Systemen gibt ($H(7) = 74,403, P = 0,000$, Mittlere Ränge in B.5)[3]. Damit ist H_1 bestätigt und die Nullhypothese H_1^0 („Es gibt keine signifikanten Unterschiede in der Eingabegeschwindigkeit von Passwörtern zwischen den Systemen“) kann verworfen werden.
Der Levene-Test [5] auf Gleichheit der Fehlervarianzen ergibt mit $P = 0,088$, dass Vari-

[1] lt. Shapiro-Wilk-Test

[2] Ein Unterschied ist statistisch signifikant, wenn es sehr unwahrscheinlich ist, dass er durch Zufall zustandegekommen ist.

[3] Mit $H(Df)$ wird die Prüfgröße des Kruskal-Wallis-Tests bezeichnet. Df ist der Freiheitsgrad, also die Anzahl der frei wählbaren Parameter (stets 1 weniger als die Anzahl der zu vergleichenden Gruppen). $H(Df)$ wird mit einem Referenzwert aus der Chi-Quadrat-Verteilung verglichen, wodurch sich ein Signifikanzwert p ergibt. Ist $p < 0.05$, dem Signifikanzniveau, so kann davon ausgegangen werden, dass sich mindestens zwei Gruppen signifikant unterscheiden. [27]

anzhomogenität vorliegt. Folglich sollte der Bonferroni-Test[4] zuverlässig klären können, zwischen welchen Systemen signifikante Unterschiede vorliegen. Der Post-hoc-Test ergibt, dass vor allem die Werte von Symbian/QWERTZ signifikant schlechter sind als die von Android/Sense ($p = 0,018$), Android/Vanilla ($p = 0,001$), iOS ($p = 0,000$), MeeGo ($p = 0,000$) und Windows Phone ($p = 0,000$). Die Werte von Symbian/T9 und Android/Swype sind beide signifikant schlechter als die von iOS (Symbian/T9: $p = 0,002$, Android/Swype: $0,012$) und Windows Phone (Symbian/T9: $p = 0,005$, Android/Swype: $0,028$). Die vollständigen Ergebnisse sind in Abb. B.6 zusammengefasst.

Als Ergebnis lässt sich also festhalten, dass die beiden Symbian-Varianten (vor allem Symbian/QWERTZ) eher mehr Zeit benötigen, und bei iOS und Windows Phone die Passwörter am schnellsten eingegeben werden konnten. Von den drei Android-Varianten schneidet Android/Swype am schlechtesten ab.

6.2.2. Eingabefehlerzahl (H_2)

Gute Usability bedeutet auch, dass Fehler vermieden werden. Daher soll geprüft werden, wie viele Fehler bei der Eingabe gemacht wurden und korrigiert werden mussten. Wie auch bei der Eingabezeit ist es sinnvoll, nur die Fälle zu betrachten, in denen das Passwort am Ende korrekt abgesendet wurde. Unvollständig abgesendete Passwörter würden die Statistik negativ verfälschen, da diese sehr hohe Fehlerzahlen produzieren würden, die aber nicht durch Tippfehler, sondern durch nur teilweise Eingabe entstanden.
Bei dieser Statistik ist zu beachten, dass bei Symbian/T9 und Android/Sense zunächst irrtümlicherweise Fehler auftraten: die bei Android/Sense über das Gedrückthalten der Buchstabentasten eingebbaren Ziffern und Sonderzeichen verursachten stets einen Messfehler, da von der Tastatur zunächst der Buchstabe in das Textfeld eingesetzt wird; bleibt die Taste gedrückt, wird beim Loslassen der Buchstabe durch das jeweilige alternative Zeichen ersetzt. Ganz ähnlich geschieht es bei Symbian/T9: zunächst wird der erste Buchstabe der Taste eingetragen, bei erneutem Tippen, um die restlichen Buchstaben oder die Ziffer der Taste einzugeben, wird der schon im Eingabefeld eingetragene Buchstabe dann ersetzt. Da für die Messung stets der Inhalt des Eingabefelds überprüft wird, wurden hier fälschlicherweise zunächst Fehler erkannt. Diese mussten bei diesen beiden Systemen hinterher wieder herausgerechnet werden – wenn also

[4] Der Bonferroni-Test ist ein Post-Hoc-Test, der Paarvergleiche zwischen allen Gruppen durchführt und die dabei zunehmende Fehlerwahrscheinlichkeit durch eine Anpassung des Signifikanzniveaus kompensiert. [27], [5]

beispielsweise bei Android/Sense anstatt einer 9 ein kleines o erkannt wurde, wird dies in folgender Statistik nicht als Fehler gezählt (ist das kleine o in diesem Beispiel dennoch auch beim Absenden des Passworts noch vorhanden, so fällt der Fall ohnehin durch das Raster, da nur korrekt abgesendete Passwörter berücksichtigt wurden). Der Effekt tritt bei den anderen Systemen, wo gleichermaßen Zeichen eingegeben werden können, nicht auf.
Die nicht-normalverteilten[5] Werte ergeben im Kruskal-Wallis-Signifikanztest, dass es signifikante Unterschiede zwischen den Systemen gibt ($H(7) = 26,854, P = 0,000$, mittlere Ränge in B.8). Damit ist H_2 ebenfalls bestätigt. Laut Levene-Test ($P = 0,000$) liegt keine Varianzhomogenität[6] vor. Um herauszufinden, zwischen welchen Systemen signifikante Unterschiede vorliegen, bietet sich daher der Games-Howell-Post-Hoc-Test an.[7] Dieser zeigt, dass die Werte von Windows Phone signifikant besser sind als die von Android/Swype ($p = 0,047$), Android/Vanilla ($p = 0,029$) und insbesondere Symbian/QWERTZ ($p = 0,002$) (gesamte Ergebnistabelle: B.9, Visualisierung: Abb. 6.6).

Durchschnittlich wurde also bei jeder Passworteingabe an Symbian/QWERTZ ein Fehler gemacht (Mittelwert: $1,07$, siehe B.7). Die besten Werte erzielten Windows Phone, iOS – und Symbian T9.

6.2.3. Auswertung des Usability-Fragebogens (H_3)

Nach diesen quantitativen Ergebnissen wird nun noch der qualitative Teil des Usability-Tests ausgewertet, der PSSUQ.
Zunächst werden den Vorgaben zum PSSUQ von Lewis [16] aus den einzelnen Werten für jeden Probanden die Mittelwerte für die vier PSSUQ-Komponenten gebildet:

- SYSUSE (System Usefulness, Aussagen 1–8)
- INFOQUAL (Information Quality, Aussagen 9–15)
- INTERQUAL (Interface Quality, Aussagen 16–18)
- OVERALL (Overall Satisfaction Score, Aussagen 1–19)

[5] lt. Shapiro-Wilk-Test
[6] Varianzhomogenität liegt vor, wenn sich die Varianzen zwischen den Stichproben nicht signifikant unterscheiden.
[7] Der Games-Howell-Test ist ein Post-Hoc-Test, der keine Varianzhomogenität voraussetzt.

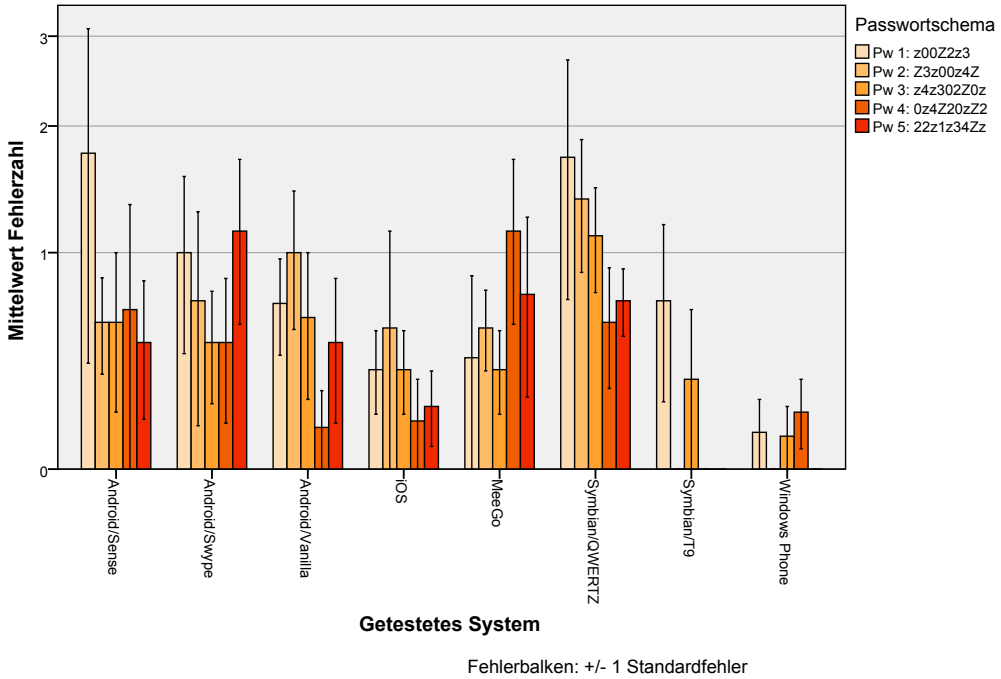


Abbildung 6.6.: Durchschnittliche Fehlerzahl pro Passwort und System.

Die Werte der SYSUSE-, INFOQUAL- und INTERQUAL-Komponente sind in dieser Studie jeweils nicht normalverteilt, die Werte der OVERALL-Komponente sind normalverteilt.[8]

Für die drei Einzelkomponenten kann also wieder der Kruskal-Wallis-Test durchgeführt werden. Dieser ergibt, dass es bei SYSUSE ($H(7) = 18,829, P = 0,009$) und INTERQUAL ($H(7) = 19,628, P = 0,006$) signifikante Unterschiede zwischen den Systemen gibt (Mittlere Ränge siehe B.10). Nicht der Fall ist das dagegen bei INFOQUAL ($H(7) = 10,783, P = 0,148$). Dies genügt aber dennoch bereits, um die Nullhypothese H_3^0 zu verwerfen – denn es liegt auf jeden Fall ein signifikanter Unterschied vor, wenn auch nicht bei allen Komponenten.

Der Levene-Test auf Gleichheit der Fehlervarianzen ergibt für SYSUSE mit $P = 0,735$ und für INTERQUAL mit $P = 0,114$, dass Varianzhomogenität vorliegt. Beide Komponenten können also wieder mit dem Bonferroni-Test weiter untersucht werden. Dabei konnte bei SYSUSE keine Signifikanz zwischen zwei Gruppen festgestellt werden (B.11), jedoch ist bei INTERQUAL der Unterschied zwischen Symbian/QWERTZ und

[8] lt. Shapiro-Wilk-Test

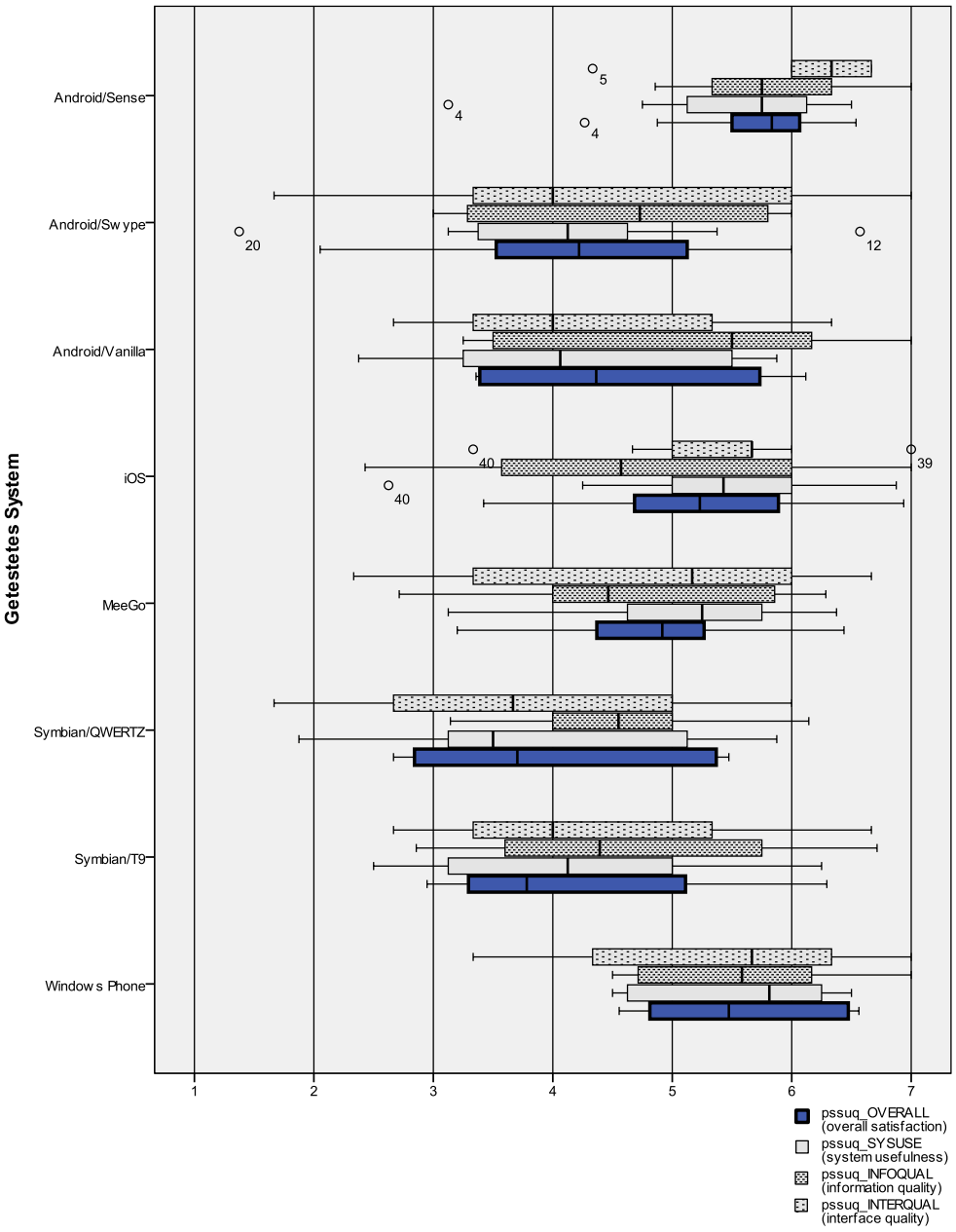


Abbildung 6.7.: Auswertung der Likert-Skalen des PSSUQ, aufgeteilt in die 4 Komponenten.

Android/Sense signifikant ($p = 0,006$, siehe B.12).
Für die normalverteilte OVERALL-Komponente kann die einfaktorielle ANOVA[9] durchgeführt werden. Diese zeigt, dass ein signifikanter Unterschied zwischen den Systemen vorliegt ($F(7,72) = 3,503, p = 0,003$ – siehe B.14). Ein Tukey-post-hoc-Test (B.13) ergibt, dass Android/Sense (mit $p = 0,021$) und Windows Phone (mit $p = 0,040$) signifikant jeweils besser bewertet wurden als Symbian/QWERTZ. Dies erweitert und bestätigt die Auswertung der INTERQUAL-Komponente, die in der OVERALL-Komponente mit enthalten ist.
H_3 kann also bestätigt werden: die Systeme wurden signifikant unterschiedlich bewertet.

6.3. Shoulder-Surfing

Im Folgenden werden die Ergebnisse des Shoulder-Surfing-Experiments ausgewertet – der Erfolg beim Mitlesen, die Bewertung im Fragebogen und schließlich die Auswertung der von den Probanden angegebenen Strategien.

6.3.1. Erfolg beim Shoulder-Surfing (H_4)

Um herauszufinden, wie erfolgreich die Probanden beim Shoulder-Surfing waren, wurde zunächst für jeden Eingabeversuch die Levenshtein-Distanz [11] ermittelt. Diese gibt an, wieviele Einfüge-, Lösch- und Ersetzoperationen auf einer Zeichenkette notwendig wären, um sie in eine vorgegebene andere umzuwandeln. Beispiel: die Levenshtein-Distanz zwischen $sunshine$ und $susanne$ beträgt 3 – das erste n in $sunshine$ wird gelöscht, das h wird zu einem a und das h wird zu einem n ersetzt.[10] Ermittelt man die Levenshtein-Distanz zwischen den erkannten Passwörtern und den Originalpasswörtern, so ist sie 0, wenn das Passwort korrekt eingegeben wurde, jeder Fehler (ausgelassenes Zeichen, falsches Zeichen, hinzugefügtes Zeichen) wird mit dem Wert 1 gezählt. Je näher der Eingabeversuch also am korrekten Passwort ist, desto niedriger ist die Levenshtein-Distanz. Dadurch bietet sie sich als Maß für den Erfolg beim Shoulder-Surfing an.
Anschließend wurde für jeden Probanden für jedes der drei Passwörter der jeweils niedrigste Wert der Levenshtein-Distanz aus den maximal drei Passwortversuchen ausgewählt – also der Fall mit dem größten Erfolg. Diese Fälle – für jeden Probanden

[9] ANOVA = analysis of variance
[10] Die Levenshtein-Distanz ist vergleichbar mit der Hamming-Distanz, mit dem Unterschied, dass die beiden Zeichenketten nicht die gleiche Länge haben müssen, sondern auch Einfüge- und Löschoperationen möglich sind.

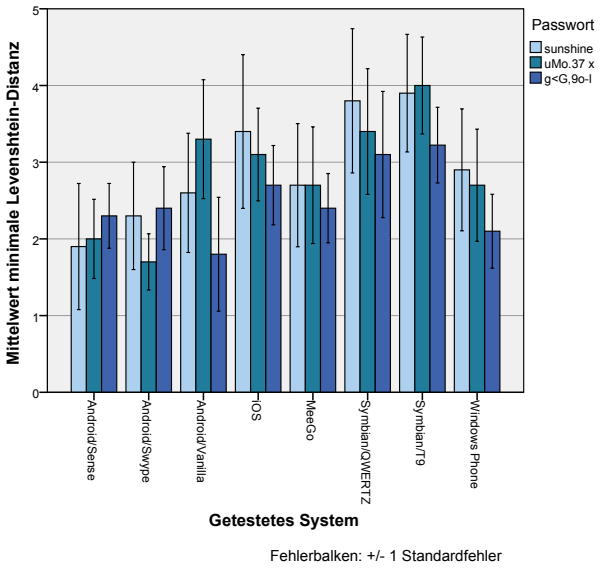


Abbildung 6.8.: Fehlerzahl beim Shoulder-Surfing.

und jedes Passwort der größte Erfolg – werden nun statistisch untersucht.

Die Levenshtein-Distanz ist dabei nicht für jedes System normalverteilt[11], die einfaktorielle ANOVA kann folglich nicht angewendet werden. Stattdessen soll der nichtparametrische Kruskal-Wallis-Test wieder als Signifikanztest dienen. Dieser ergibt, dass es einen signifikanten Unterschied zwischen den Systemen gibt ($H(7) = 14,978, p = 0,036$, mittlere Ränge siehe B.15). Die Nullhypothese H_4^0, dass Shoulder-Surfing bei den verschiedenen Systemen nicht signifikant unterschiedlich leicht ist, kann also verworfen werden.
Der Levene-Test ergibt, dass keine Varianzhomogenität vorliegt ($p = 0,030$). Folglich kann als Post-Hoc-Test der Games-Howell-Test durchgeführt werden. Dieser ergibt, dass Symbian/T9 signifikant besser (mehr Fehler beim Shoulder-Surfing) ist als Android/Sense ($p = 0,032$) und Android/Swype ($p = 0,033$), Details in B.16. Auf dem Diagramm 6.8) lässt sich außerdem erkennen, dass auch iOS und Symbian/QWERTZ einen relativ hohen Wert erreichen – dieses Ergebnis lässt sich aber aufgrund der fehlenden Signifikanz nicht über die Studie hinaus verallgemeinern.
Betrachtet man die Levenshtein-Distanz binär – nur die Fälle, in denen sie 0 ist, also das Passwort komplett korrekt erkannt wurde – so werden deutlichere Unterschiede

[11] lt. Shapiro-Wilk-Test

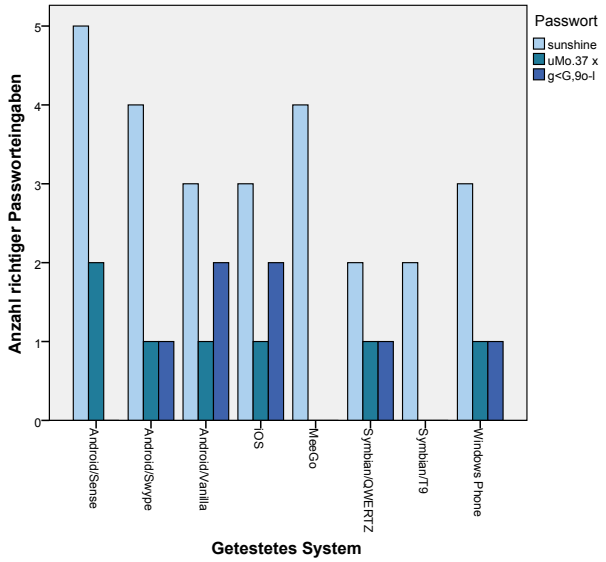


Abbildung 6.9.: Absolute Anzahl korrekter Passworteingaben beim Shoulder-Surfing.

sichtbar (Abb. 6.9). So wurde das alphanumerische Passwort auf allen Systemen häufiger erkannt als die komplizierteren. Zwischen diesen beiden Passwörtern ist kein offensichtlicher Unterschied erkennbar. Bei den Symbian-Varianten und bei MeeGo wurden dabei insgesamt weniger Passwörter erkannt als bei den Android-Varianten und iOS.

6.3.2. Bewertung des Shoulder-Surfing (H_5)

Es wurde sowohl vor (A.5) als auch nach (A.6) dem Shoulder-Surfing-Experiment ein Fragebogen mit je 4 Likert-Fragen gestellt. Die erste Frage des Pre-Fragebogens war dabei negativ formuliert, deren Werte müssen also invertiert werden. Ein Reliabilitätstest ergab für Cronbachs α[12] einen Wert von $0,605$. Dies deutet darauf hin, dass mehr Fragen hätten gestellt werden müssen; allerdings ist das Ergebnis noch brauchbar, da der Wert $> 0,5$ ist. Die Likert-Fragen messen also das gleiche und es kann ein valider Mittelwert gebildet werden.
Beim Post-Fragebogen waren die ersten beiden Fragen negativ formuliert. Ein Realibilitätstest ergab für Cronbachs α einen Wert von $0,751$, was einer akzeptablen Reliabilität

[12] Ein hoher Wert für Cronbachs α deutet auf eine hohe Reliabilität hin. 1.0 wäre der perfekte Wert, Werte über $0,5$ sind akzeptabel.

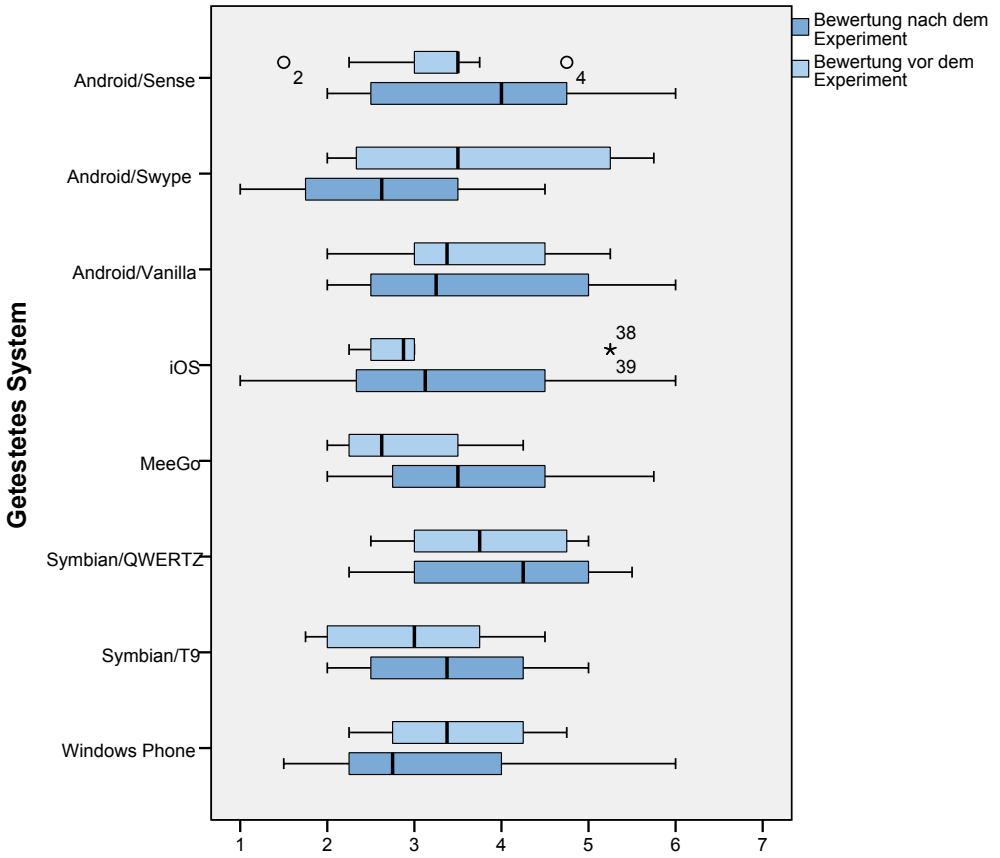


Abbildung 6.10.: Auswertung der Likert-Skalen vor und nach dem Shoulder-Surfing-Experiment.

entspricht. Auch diese Fragen können also miteinander verrechnet werden, um einen einzelnen (Mittel-)Wert pro Tester zu ergeben. Die Reliabilitätstests ergaben, dass ein Weglassen von einzelnen Fragen zu keiner nennenswerten Verbesserung des α führen würden (B.17).

Da die gemittelten Gesamtwerte normalverteilt sind[13], kann ein Signifikanztest mithilfe der einfaktoriellen ANOVA durchgeführt werden. Bei den Fragebögen konnte kein signifikanter Unterschied zwischen den Gruppen festgestellt werden (Pre: $F(7,71) = 1,069, p = 0,392$, Post: $F(7,72) = 0,954, p = 0,471$, siehe B.18). Dies schlägt sich auch im Plot nieder (Abb. 6.10). Folglich kann H_5 nicht bestätigt werden – es gibt bei der Bewertung der Sicherheit gegen Shoulder-Surfing also keine signifikanten Unterschiede

[13] lt. Shapiro-Wilk-Test

zwischen den Systemen. Es lässt sich allerdings sagen, dass die Systeme insgesamt eher schlecht bewertet wurden (Zentralwert Pre: $3,25$, Post: $3,50$ (B.19) – bei den Likert-Skalen liegt die Mitte beim Wert $4,00$.).

Interessant ist noch die Frage, ob es zwischen der Pre- und Post-Bewertung bei einem System einen signifikanten Unterschied gibt. Dafür wurde ein t-Test für abhängige Messungen[5] für jedes System durchgeführt. Dieser ergab jedoch bei keinem System einen signifikanten Unterschied zwischen der Bewertung vor und nach dem Shoulder-Surfing-Experiment.

6.3.3. Shoulder-Surfing-Strategien (H_6)

Um herauszufinden, welche Shoulder-Surfing-Strategien erfolgreich waren, wurde zunächst ein Erfolgswert für jeden Probanden folgendermaßen berechnet: aus den maximal drei Versuchen pro Passwort und Proband wurde zunächst die jeweils geringste Levenshtein-Distanz ausgewählt, also der Versuch mit dem größten Erfolg. Aus diesen Werten der drei Passwörter wurde dann ein Mittelwert gebildet. Ein Reliabilitätstest ergab für Cronbachs α einen Wert von $0,702$ für die jeweils geringste Levenshtein-Distanz der drei Passwörter pro Proband – es kann also daraus ein Mittelwert gebildet werden, der als akzeptables Maß für den Erfolg des Probanden insgesamt dienen kann. Es ist zu beachten, dass wie bei der Levenshtein-Distanz ein niedriger Wert für die Variable „Erfolg“ einen erfolgreicheres Shoulder-Surfing bedeutet.

Zunächst soll überprüft werden, ob die Wahl der Shoulder-Surfing-Zone (rechts, links oder in der Mitte hinter dem Eingebenden) zu signifikanten Unterschieden beim Shoulder-Surfing-Erfolg führte. Die nicht normalverteilten[14] Werte (siehe Abb. 6.11) für den Erfolg können wieder mit dem Kruskal-Wallis-Test auf Signifikanz überprüft werden. Dieser ergibt, dass es keine signifikanten Unterschiede zwischen den drei Zonen gibt ($H(2) = 4,701, p = 0,095$, mittlere Ränge siehe B.20). Die Nullhypothese H_6^0 muss also beibehalten werden. Dies dürfte vor allem an der ungleichen Verteilung der Probanden auf die Zonen liegen – 58 der 80 Probanden wählten die linke Zone, 20 die rechte und nur 2 Probanden die mittlere Zone – dies dürfte vor allem daran gelegen haben, dass die Geräte mit der rechten Hand gehalten und die Passwörter mit dem Daumen der rechten Hand eingegeben wurden. Für signifikante Ergebnisse hätten hier vermutlich die Zonen gleichmäßig zugeteilt werden müssen, anstatt die Probanden frei wählen zu

[14] lt. Shapiro-Wilk-Test

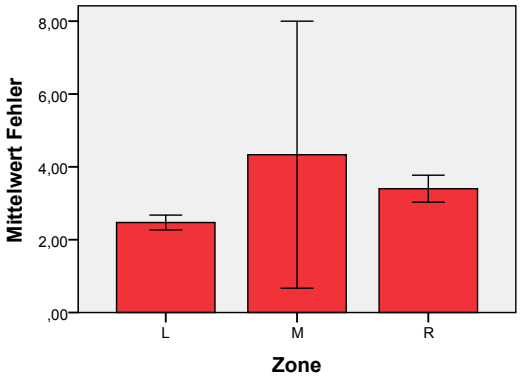


Abbildung 6.11.: Durchschnittliche Fehler beim Shouldersurfing in den verschiedenen Zonen (niedriger Wert ≙ großer Erfolg)

lassen.

Die Probanden konnten auf drei Skalen gewichten, worauf sie beim Shoulder Surfing wie stark geachtet hatten. Nun soll der Einfluss dieser drei Komponenten („Tastaturanzeige“, „Fingerbewegung“ und „Eingabefeld“) auf den Shoulder-Surfing-Erfolg untersucht werden.
Typischerweise würde man hier einen Pearson-Korrelationstest für jede Komponente durchführen; dieser würde allerdings keine zuverlässigen Ergebnisse liefern, da die Werte der Komponenten nicht normalverteilt sind.[15] Es ist also wiederum ein nichtparametrischer Test vonnöten, hier wird der Spearman-Rangkorrelationskoeffizienten-Test durchgeführt.[5]
Dieser sagt aus, dass eine statistisch signifikante, aber nicht besonders starke negative Korrelation zwischen der Fehlerrate und der Komponente „Eingabefeld“ vorliegt ($r_S(78) = -0,303$, $P = 0,006$). Die Probanden, die angaben, mehr auf das Eingabefeld geachtet zu haben, hatten also eine höhere Wahrscheinlichkeit, ein gutes Ergebnis (≙ wenige Fehler) zu erzielen.
Der Spearman-Test ergibt außerdem (B.21), dass negative Korrelationen zwischen den Komponenten „Eingabefeld“ und „Tastaturanzeige“ sowie zwischen „Eingabefeld“ und „Fingerposition“ vorliegen, und folglich eine positive Korrelation zwischen „Tastaturanzeige“ und „Fingerposition“. Dies sagt aus, dass die Probanden offensichtlich entweder

[15] lt. Shapiro-Wilk-Test

stärker auf das Eingabefeld oder stärker auf Tastaturanzeige und Fingerposition achteten, was logisch erscheint – entweder beobachteten sie mehr den oberen Teil des Bildschirms, wo das Eingabefeld liegt, oder mehr den unteren Teil, wo sich Tastatur und Finger befinden.

Zwar liegt eine statistisch signifikante Korrelation zwischen dem Erfolg beim Shoulder-Surfing und der Gewichtung, wie stark auf das Eingabefeld geachtet wurde, vor. Da diese Korrelation aber sehr schwach ist, genügt sie nicht, um die Nullhypothese H_6^0 abzulehnen; bei so einer schwachen Korrelation kann noch nicht auf einen allgemeingültigen Zusammenhang geschlossen werden.

6.3.4. Passwortanalysen

Abschließend sollen nun die Erkennungsraten der Zeichen der drei beim Shoulder-Surfing verwendeten Passwörter noch näher untersucht werden, unabhängig vom jeweiligen System. Es soll untersucht werden, welche Zeichen besser oder schlechter erkannt wurden, um eventuell allgemeingültige Empfehlungen ableiten zu können, wie möglichst Shoulder-Surfing-resistente mobile Passwörter gebildet werden können.
Dafür wurde für jedes Zeichen in jedem Passwort geprüft, ob es korrekt erkannt wurde (d.h., im eingegebenen Passwort vorkommt). Daraus lässt sich eine Erkennungsrate für jedes Zeichen ermitteln (siehe B.22).

Passwort 1: sunshine

Es ist erkennbar (Abb. 6.12), dass die Erkennungsrate bei dem Wörterbuch-Passwort gleich beim 2., sowie beim 5. Zeichen leicht einbricht. Dies könnte damit erklärbar sein, dass der rechte Daumen für die Eingabe des s auf einer QWERTZ-Tastatur weit nach links muss und einen großen Teil der Tastatur bedeckt, die danach folgenden, weiter rechts liegenden Zeichen können dann schnell „im zurückbewegen“ eingegeben werden können.
Insgesamt ist eine leichte Tendenz erkennbar, nach der die Zeichen am Anfang zuverlässiger erkannt wurden als am Ende des Passworts; das letzte Zeichen wurde allerdings relativ häufig erkannt. Dies ist vermutlich darauf zurückzuführen, dass normalerweise die Anzeige des gerade eingegebenen Zeichens im Passwortfeld vorzeitig verschwindet, wenn das nächste Zeichen eingegeben wird. Das letzte Zeichen dagegen ist die volle

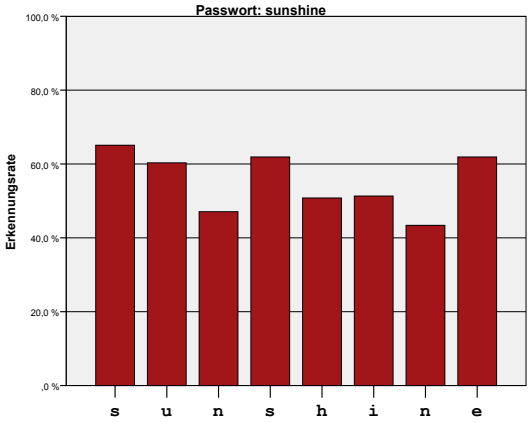


Abbildung 6.12.: Erkennungsrate der Zeichen des ersten Passworts.

Zeit (ca. 1 Sekunde) sichtbar.

Passwort 2: uMo.37 x

Beim zweiten Passwort (Abb. 6.13) wird diese Tendenz bestätigt, vor allem beim Buchstaben-Teil am Anfang. Bei den beiden Sonderzeichen sind die tiefsten Erkennungsraten aufgetreten. Der Wechsel zur Zahleneingabe wurde gut erkannt, allerdings nimmt die Erkennungsrate bei der zweiten Ziffer ebenfalls wieder ab. Das Leerzeichen – auf allen Tastaturen relativ schnell eingebbar – wurde nur sehr selten erkannt (17%). Auch der Punkt als schnell und einfach eingebbares Sonderzeichen erhält gute Werte.

Passwort 3: g<G,9o-l

Das dritte Passwort (Abb. 6.14) beginnt mit dem Kleinbuchstaben g, der im Vergleich zu den ersten Zeichen der anderen beiden Passwörter nicht besonders gut erkannt wurde. Dies könnte möglicherweise mit einer eventuell nachlassenden Konzentration der Tester beim dritten Passwort erklärt werden. Das relativ komplizierte Sonderzeichen, das meist auf der zweiten Sonderzeichen-Tastatur liegt, erfordert durchschnittlich 2,8 Taps, wodurch die Erkennung offensichtlich einfacher wird. Das Komma hat, genau wie der Punkt, einen relativ guten Wert, da es schnell eingebbar ist. Außerdem besteht die Möglichkeit, dass es aufgrund der geringen Größe und oft ähnlichen Platzierung auf

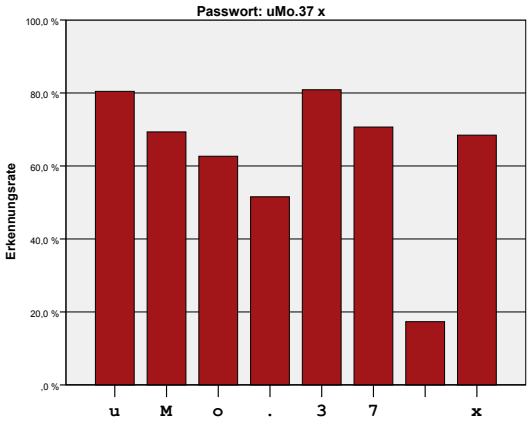


Abbildung 6.13.: Erkennungsrate der Zeichen des zweiten Passworts.

der Tastatur mit dem Punkt verwechselt wird.
Genauso verwechselt werden kann das o – der niedrige Wert ist aber nicht nur mit der optischen Ähnlichkeit zur 0, sondern auch mit der Position auf einer QWERTZ-Tastatur mit Zahlen direkt unterhalb der 0 erklärbar. Teilweise kann die davorliegende 9 sogar mit der selben Taste eingegeben werden wie das o (z.B. bei den drei Android-Varianten), wodurch es beim Shoulder-Surfing offensichtlich leicht „untergeht“. Eine ähnliche Verwechslungsgefahr besteht offensichtlich auch beim kleinen l am Ende, es kann aufgrund optischer und örtlicher Nähe zum großen I leicht verwechselt werden. Einige Probanden waren auch der Meinung, eine Pipe (|) erkannt zu haben.
Wieder ist ein insgesamter Trend zu niedrigeren Erkennungsraten gegen Ende erkennbar.

6.3.5. Gesamtbewertung

Zuletzt wurden die Probanden noch zu ihrem Gesamteindruck des getesteten Systems in Bezug auf alle Aspekte befragt (Abb. A.7). Die Antworten zu diesen beiden Likert-Fragen sind nicht normalverteilt[16]. Der Kruskal-Wallis-Test ergab, dass keine signifikanten Unterschiede zwischen den Systemen vorliegen ($H(7) = 7,225, p = 0,406$, mittlere Ränge siehe B.23). Dennoch lässt sich erkennen (Abb. 6.15), dass die beiden Symbian-Varianten von den Probanden in dieser Studie weniger weiterempfohlen würden als die anderen Systeme. Außerdem wurden iOS und Windows Phone als am

[16] lt. Shapiro-Wilk-Test

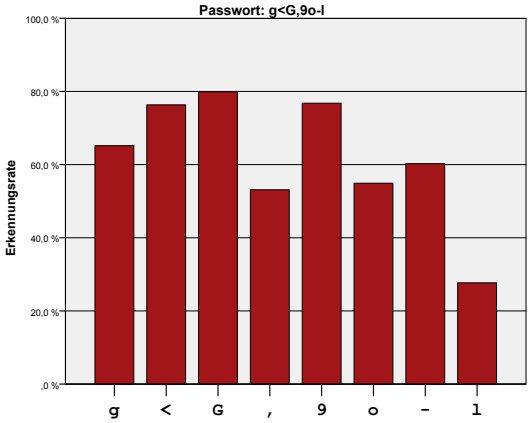


Abbildung 6.14.: Erkennungsrate der Zeichen des dritten Passworts.

besten zur Passworteingabe geeignet bewertet. Diese Ergebnisse sind jedoch aufgrund der fehlenden Signifikanz nicht verallgemeinerbar.

6.4. Zusammenfassung

In Bezug auf die Hypothesen lassen sich also folgende Ergebnisse festhalten:

- **H_1:** Es gibt signifikante Unterschiede in der Eingabegeschwindigkeit von Passwörtern zwischen den Systemen – iOS und Windows Phone schnitten gut ab, die Symbian-Varianten benötigten eher mehr Zeit bei der Eingabe.

- **H_2:** Es gibt signifikante Unterschiede bei der Anzahl der Eingabefehler zwischen den Systemen – bei iOS, Symbian/T9 Windows Phone traten eher weniger Eingabefehler auf, bei Android/Swype, Android/Vanilla und Symbian/QWERTZ eher viele Eingabefehler.

- **H_3:** Es gibt signifikante Unterschiede bei der Usability-Bewertung von den Probanden zwischen den Systemen – Android/Sense und Windows Phone wurden überwiegend gut bewertet, Symbian/QWERTZ überwiegend schlecht.

- **H_4:** Shoulder-Surfing ist bei den verschiedenen Systemen signifikant unterschiedlich leicht – bei Symbian/T9 konnten weniger viele Teile der Passwörter korrekt mitgelesen werden als bei Android/Sense und Android/Swype.

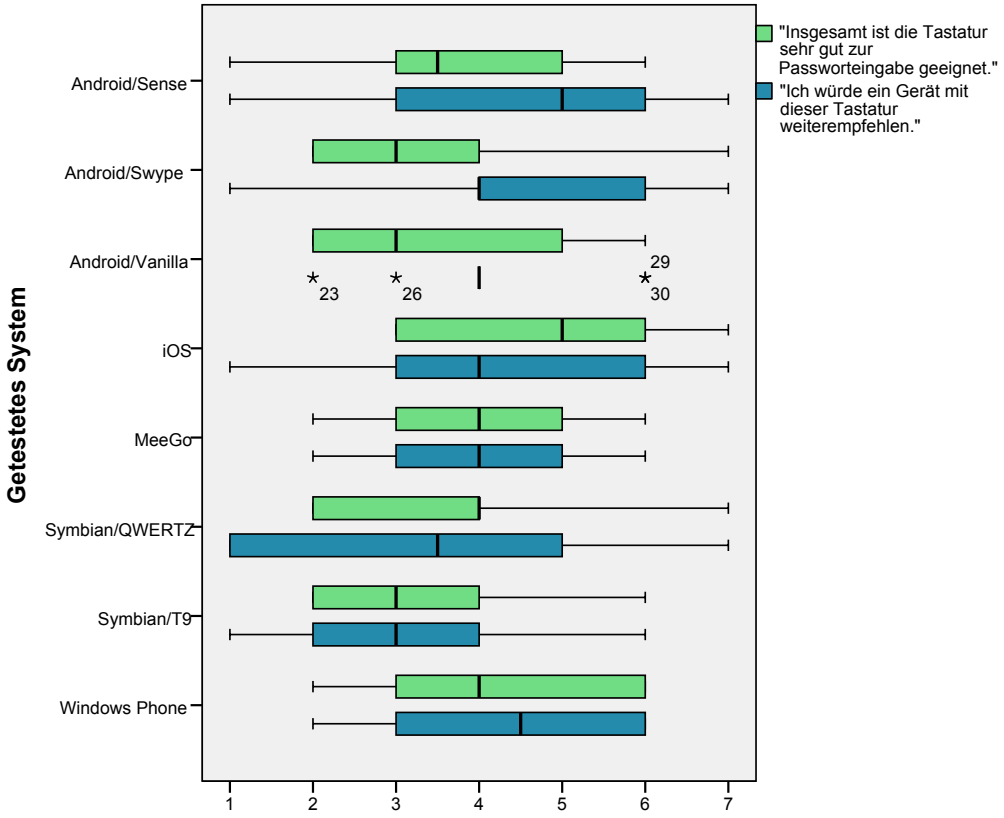


Abbildung 6.15.: Auswertung der Antworten zu den beiden letzten Fragen zum Gesamteindruck des getesteten Systems.

- **$H_5{}^0$:** Es gibt **keine** signifikanten Unterschiede bei der Bewertung der Sicherheit gegen Shoulder-Surfing von den Probanden zwischen den verschiedenen Systemen.
- **$H_6{}^0$:** Es gibt **keine** signifikanten Unterschiede des Erfolgs bei unterschiedlichen Shoulder-Surfing-Strategien.

6.5. Grenzen der Studie

Idealerweise wären bei dieser Studie die einzelnen Systeme nur von Probanden getestet worden, die das jeweilige System selbst einsetzen. Dies war aber praktisch nicht machbar, da beispielsweise kaum Probanden Erfahrung mit Windows Phone und

MeeGo hatten. Außerdem hatten viele der verfügbaren Probanden noch gar keine Smartphone-Erfahrung. Dies ergab das Problem, dass einigen Probanden die Übung fehlte, während andere das getestete System schon sehr gut kannten.
Alternativ hätte den Probanden stets ein System zugewiesen werden können, das sie noch nicht kennen, oder aber die Systeme hätten zufällig zugewiesen werden können.

Ein weiteres Problem entstand durch eventuelle persönliche Präferenzen der Probanden. So wurde von einigen Probanden mündlich und in den Kommentarfeldern insbesondere eine Rivalität zwischen Android- und iOS-Nutzern zum Ausdruck gebracht. Dies kann in übermäßig guten Bewertungen für das „eigene“ und übermäßig schlechten Bewertungen für das „fremde“ System resultieren. Verhindern lässt sich dies kaum. Eine aufwändige Möglichkeit wäre gewesen, die Geräte so zu „verpacken“, dass nur noch das Display sichtbar und für den Probanden nicht direkt ersichtlich ist, um welches System es sich handelt – ein für den Probanden fremdes System wäre dann vermutlich nicht mehr so leicht erkannt worden.

Wie oben ausgeführt, waren etliche Variablen nicht normalverteilt. Dies hätte vermutlich durch eine größere Probandenzahl verbessert werden können.

Interessant für die Deutung der Ergebnisse wäre eine noch tiefergehende Analyse der Systeme gewesen. Zum einen können Unterschiede durch unterschiedliche Sensitivität der Touchscreens der Geräte entstanden sein. Zum anderen wäre eine Analyse der Größe der Flächen, die einen Tastendruck erkennen, interessant gewesen (aber technisch schwer durchführbar). Diese Fläche kann entweder exakt genauso groß sein wie die sichtbare Taste oder größer als diese Taste. Hier ist dann interessant, wo die Grenze zwischen den Tasten ist – diese kann entweder genau in der Mitte zwischen zwei Tasten sein, oder sich dynamisch anpassen. So ist beispielsweise im iOS Simulator (mithilfe der Genauigkeit einer Computermaus) einfach nachvollziehbar, dass iOS mit Wahrscheinlichkeiten arbeitet, welcher Buchstabe als nächstes eingegeben wird, und die sensitive Fläche (nicht aber die visuelle Darstellung der Tasten) von wahrscheinlichen Buchstaben entsprechend vergrößert, um die Eingabe zu erleichtern.

7. Diskussion

Die Studie zeigte teilweise deutliche Unterschiede zwischen Multitouch-Smartphone-Tastaturen auf.

iOS und Windows Phone erwiesen sich dabei als die am schnellsten und mit den wenigsten Tippfehlern zu bedienenden Systeme. Die größten Probleme hat hier Symbian/QWERTZ; von den Probanden wurde vor allem die kleine Tastengröße häufig bemängelt, durch die es oft zu Tippfehlern kam. Bei Symbian/T9 dagegen gab es kaum Tippfehler – vermutlich aufgrund der großen Tasten. Dafür dauert die (Passwort-) Eingabe aufgrund der notwendigen Mehrfach-Taps sehr lange.
Zwischen den drei Android-Varianten wurden im Usability-Test keine signifikanten Unterschiede festgestellt, insgesamt liegen sie meist im Mittelfeld. Im Usability-Fragebogen wurde allerdings Android/Sense insgesamt am besten bewertet. Die bei Android deutlich gewordene hohe Anzahl an Tippfehlern ist möglicherweise auf die alternativen Eingabemöglichkeiten für Ziffern und Sonderzeichen durch längeres Gedrückthalten von Buchstabentasten zurückzuführen.
Die Unterschiede können aber auch in der Größe der sensitiven Fläche, die einen Tastendruck erkennt, begründet liegen – dies könnte in zukünftigen Arbeiten genauer untersucht werden.

Beim Shoulder-Surfing zeigte sich, dass die bei der Usability problematischen Systeme eher Vorteile haben: so wurde bei Symbian/T9 am wenigsten richtig mitgelesen, auch die Werte von Symbian/QWERTZ sind relativ gut. Allerdings ist kein allgemeiner Zusammenhang zwischen großer Shoulder-Surfing-Gefahr und guter Usability erkennbar: Windows Phone und iOS, bei der Usability führend, hatten beim Shoulder-Surfing akzeptable Werte. Am leichtesten fiel den Probanden das Shoulder-Surfing bei den drei Android-Varianten, insbesondere Android/Swype und Android/Sense. Dabei nannten die Probanden als Hauptproblem das vergrößerte Anzeigen der Buchstaben beim Tippen. Offenbar ist Shoulder-Surfing aber bei allen Systemen ein Problem. Es bliebe zu testen, ob die Anzeige der Vergrößerung beim Tippen wie auch die Anzeige des letzten

Zeichens im Eingabefeld bei der Passworteingabe wirklich notwendig ist, oder ob ohne massive Beeinträchtigung der Usability hier darauf verzichtet werden könnte. Dies würde Shoulder-Surfing sehr erschweren.

Als Nutzer kann man zunächst durch lange Passwörter und Passwörter, die keine Wörter bilden, Shoulder-Surfing erschweren. Der Tipp, ein Passwort durch Ziffern sicherer zu machen, hat beim Shoulder-Surfing an mobilen Geräten allerdings keine Gültigkeit. Durch den Wechsel auf die alternativen Tastaturen kann ein Shoulder-Surfer diese Zeichen in der Regel gut mitlesen. Gleiches gilt für komplizierte Sonderzeichen. Stattdessen sollten Sonderzeichen eingebaut werden, die schnell und unauffällig eingegeben werden können – Punkt, Komma und vor allem Leerzeichen. Sinnvoll sind auch Zeichen, die leicht verwechselt werden können und daher rein optisch eine Hürde für Shoulder-Surfer darstellen.
Es ist sicher empfehlenswert, häufig mobil einzugebende Passwörter auf das jeweilige System abzustimmen und Zeichen zu verwenden, die speziell auf diesem System schnell eingebbar sind oder von Mitlesern leicht verwechselt werden können – beispielsweise auf Android-Geräten Folgen von Buchstaben und Zahlen, die über den selben Button eingegeben werden können, oder bei iOS die Folge „Punkt Leerzeichen“, die durch Doppel-Tappen auf die Leertaste sehr schnell und unauffällig eingegeben werden kann.

Alternative Authentifizierungsmöglichkeiten, wie graphische Passwortverfahren – die teilweise proprietär schon zum Entsperren von Smartphones im Einsatz sind – wären auch für allgemeingültige Authentifizierungsvorgänge, die bislang durch Kombinationen von Benutzernamen und Passwörtern geschehen, wünschenswert. Hierfür müsste ein plattformunabhängiger, abwärtskompatibler Standard geschaffen werden.

Abschließend lässt sich feststellen, dass die Softwaretastaturen von iOS und Windows über eine sehr gute Usability verfügen und beim Shoulder-Surfing im Mittelfeld liegen, während Symbian mit schlechter Usability große Sicherheit beim Shoulder-Surfing bewies. Android liegt bei der Usability im Mittelfeld, ist allerdings sehr anfällig gegen Shoulder-Surfing.
Hauptproblem beim Shoulder-Surfing ist, wie zu Beginn vermutet, die Anzeige von getippten Zeichen auf dem Bildschirm.

A. Fragebögen

Im Folgenden sind die relevanten Fragebögen dargestellt.

Passworteingabe bei Multitouch-Smartphones

Fragebogen I/III, Teil 1/4

ulm university universität ulm

Ruben Deyhle, Institut für Medieninformatik

Guten Tag!

Vielen Dank, dass Sie sich für die Teilnahme an dieser Studie bereiterklärt haben.
Ziel der Studie ist es, Usability und Security von Passworteingabe an Multitouch-Smartphones mit Bildschirmtastaturen zu evaluieren.
Die komplette Studie wird für Sie ungefähr **15 Minuten** dauern und besteht aus drei Fragebögen sowie zwei praktischen Tests.

Die dabei verwendeten Passwörter sind grundsätzlich vorgegeben und zufallsgeneriert. Sie müssen in **keinem** Teil der Studie eigene Passwörter verwenden oder Angaben über Ihr persönliches Passwortverhalten machen. Alle erhobenen Daten werden **anonymisiert gespeichert** und ausgewertet und lassen keinen Rückschluss auf Ihre Person zu. Personenbezogene Daten werden lediglich zur demographischen Einordnung erhoben.

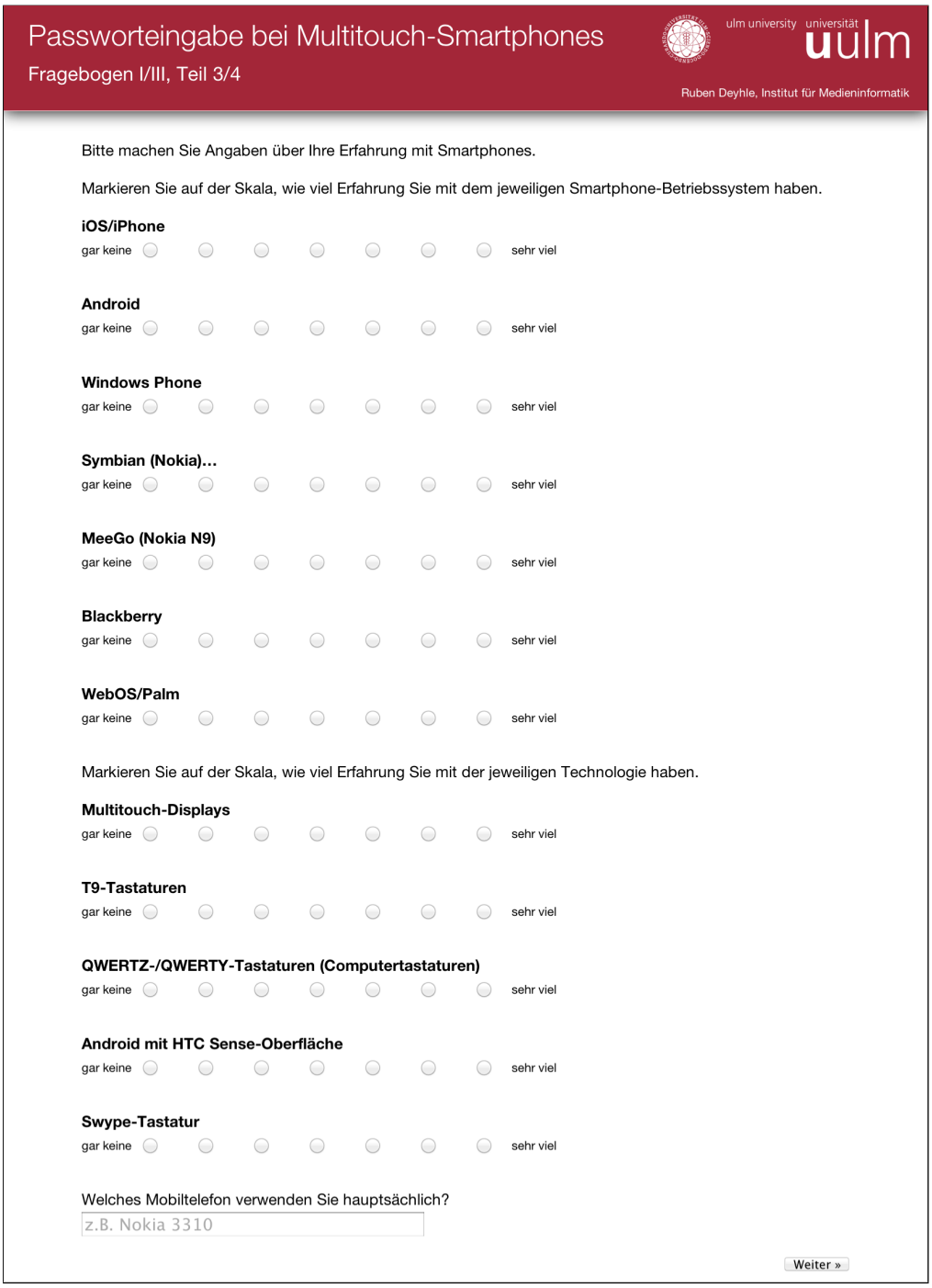
Mit der Teilnahme an dieser Studie erklären Sie sich bereit, dass die erhobenen Daten zum Zwecke der Forschung gespeichert, weiterverarbeitet und zur Ergebnisdarstellung in wissenschaftlichen Vorträgen und Forschungsberichten genutzt werden. Die erhobenen Daten können zum Zwecke der Forschung außerdem an kooperierende Institutionen weitergegeben und veröffentlicht werden.

Alle Daten werden anonym erhoben.

☐ Ich bin mit den beschriebenen Bedingungen einverstanden

Weiter »

Abbildung A.1.: Fragebogen I/III, 1/4



Passworteingabe bei Multitouch-Smartphones

Fragebogen I/III, Teil 3/4

ulm university universität uulm

Ruben Deyhle, Institut für Medieninformatik

Bitte machen Sie Angaben über Ihre Erfahrung mit Smartphones.

Markieren Sie auf der Skala, wie viel Erfahrung Sie mit dem jeweiligen Smartphone-Betriebssystem haben.

iOS/iPhone
gar keine ○ ○ ○ ○ ○ ○ ○ sehr viel

Android
gar keine ○ ○ ○ ○ ○ ○ ○ sehr viel

Windows Phone
gar keine ○ ○ ○ ○ ○ ○ ○ sehr viel

Symbian (Nokia)…
gar keine ○ ○ ○ ○ ○ ○ ○ sehr viel

MeeGo (Nokia N9)
gar keine ○ ○ ○ ○ ○ ○ ○ sehr viel

Blackberry
gar keine ○ ○ ○ ○ ○ ○ ○ sehr viel

WebOS/Palm
gar keine ○ ○ ○ ○ ○ ○ ○ sehr viel

Markieren Sie auf der Skala, wie viel Erfahrung Sie mit der jeweiligen Technologie haben.

Multitouch-Displays
gar keine ○ ○ ○ ○ ○ ○ ○ sehr viel

T9-Tastaturen
gar keine ○ ○ ○ ○ ○ ○ ○ sehr viel

QWERTZ-/QWERTY-Tastaturen (Computertastaturen)
gar keine ○ ○ ○ ○ ○ ○ ○ sehr viel

Android mit HTC Sense-Oberfläche
gar keine ○ ○ ○ ○ ○ ○ ○ sehr viel

Swype-Tastatur
gar keine ○ ○ ○ ○ ○ ○ ○ sehr viel

Welches Mobiltelefon verwenden Sie hauptsächlich?
z.B. Nokia 3310

Weiter »

Abbildung A.2.: Fragebogen I/III, 3/4

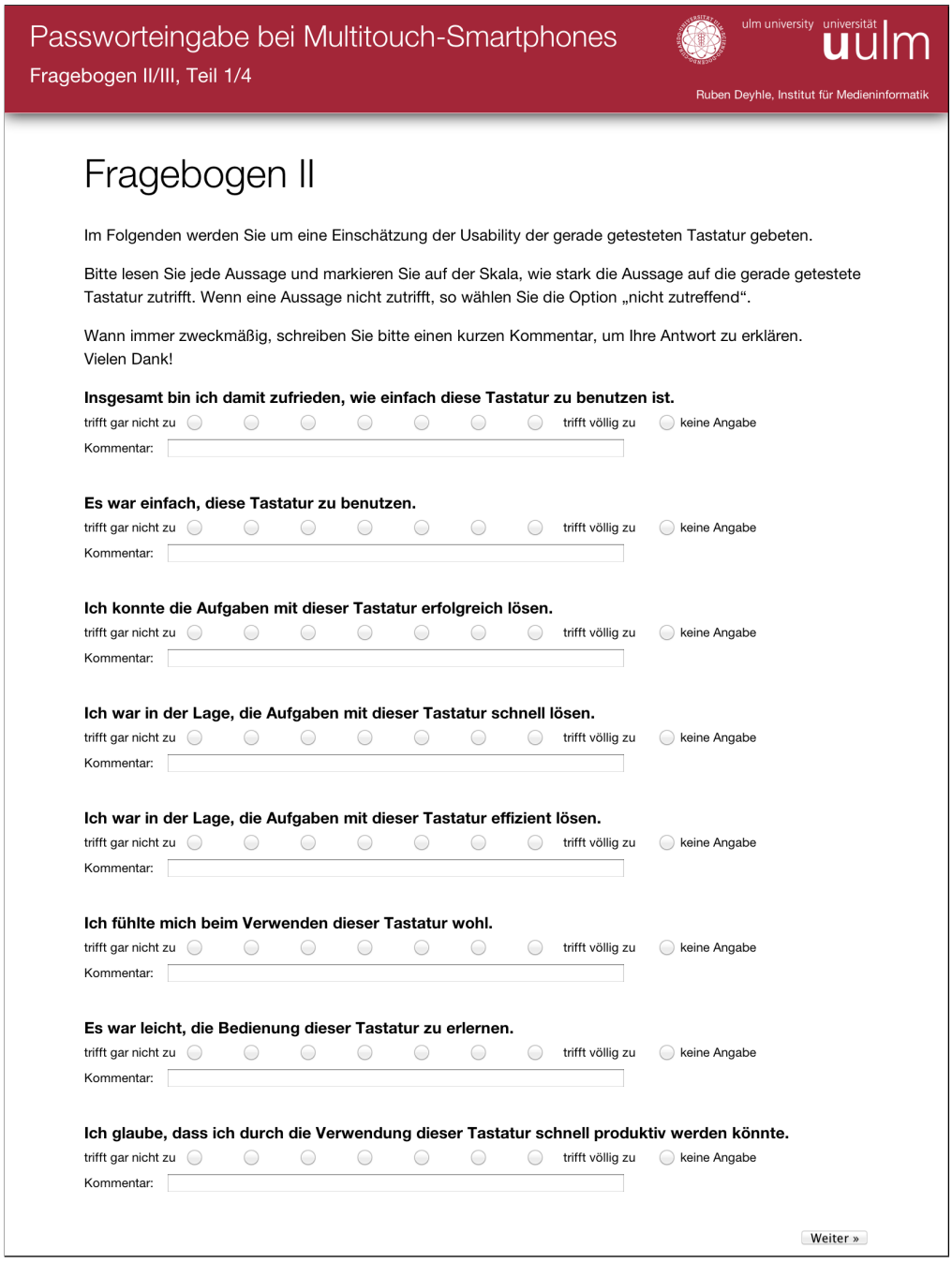


Abbildung A.3.: Fragebogen II/III, 1/4 (deutsche Übersetzung des PSSUQ, Teil 1)

Passworteingabe bei Multitouch-Smartphones

Fragebogen II/III, Teil 2/4

ulm university universität uulm

Ruben Deyhle, Institut für Medieninformatik

Bitte lesen Sie jede Aussage und markieren Sie auf der Skala, wie stark die Aussage auf die gerade getestete Tastatur zutrifft. Wenn eine Aussage nicht zutrifft, so wählen Sie die Option „nicht zutreffend“.

Wann immer zweckmäßig, schreiben Sie bitte einen kurzen Kommentar, um Ihre Antwort zu erklären.
Vielen Dank!

Die Tastatur gab Fehlermeldungen aus, die klar erklärten, wie Probleme gelöst werden können.
trifft gar nicht zu ○ ○ ○ ○ ○ ○ ○ trifft völlig zu ○ keine Angabe
Kommentar:

Fehler konnte ich einfach und schnell korrigieren.
trifft gar nicht zu ○ ○ ○ ○ ○ ○ ○ trifft völlig zu ○ keine Angabe
Kommentar:

Die Informationen (wie Direkthilfe, Bildschirm-Meldungen und andere Dokumentation), die das System zur Verfügung gestellt hat, waren klar verständlich.
trifft gar nicht zu ○ ○ ○ ○ ○ ○ ○ trifft völlig zu ○ keine Angabe
Kommentar:

Es war leicht, benötigte Informationen zu finden.
trifft gar nicht zu ○ ○ ○ ○ ○ ○ ○ trifft völlig zu ○ keine Angabe
Kommentar:

Die zur Verfügung gestellten Informationen waren leicht verständlich.
trifft gar nicht zu ○ ○ ○ ○ ○ ○ ○ trifft völlig zu ○ keine Angabe
Kommentar:

Die Informationen halfen mir erfolgreich dabei, die Aufgaben zu lösen.
trifft gar nicht zu ○ ○ ○ ○ ○ ○ ○ trifft völlig zu ○ keine Angabe
Kommentar:

Die Anordnung der Informationen auf der Tastatur war logisch.
trifft gar nicht zu ○ ○ ○ ○ ○ ○ ○ trifft völlig zu ○ keine Angabe
Kommentar:

Die Darstellung und Gestaltung der Tastatur ist angenehm.
trifft gar nicht zu ○ ○ ○ ○ ○ ○ ○ trifft völlig zu ○ keine Angabe
Kommentar:

Ich benutzte diese Tastatur gerne.
trifft gar nicht zu ○ ○ ○ ○ ○ ○ ○ trifft völlig zu ○ keine Angabe
Kommentar:

Diese Tastatur bietet alle Funktionen und Möglichkeiten, die ich von ihr erwarte.
trifft gar nicht zu ○ ○ ○ ○ ○ ○ ○ trifft völlig zu ○ keine Angabe
Kommentar:

Insgesamt bin ich mit dieser Tastatur zufrieden.
trifft gar nicht zu ○ ○ ○ ○ ○ ○ ○ trifft völlig zu ○ keine Angabe
Kommentar:

Weiter »

Abbildung A.4.: Fragebogen II/III, 2/4 (deutsche Übersetzung des PSSUQ, Teil 2)

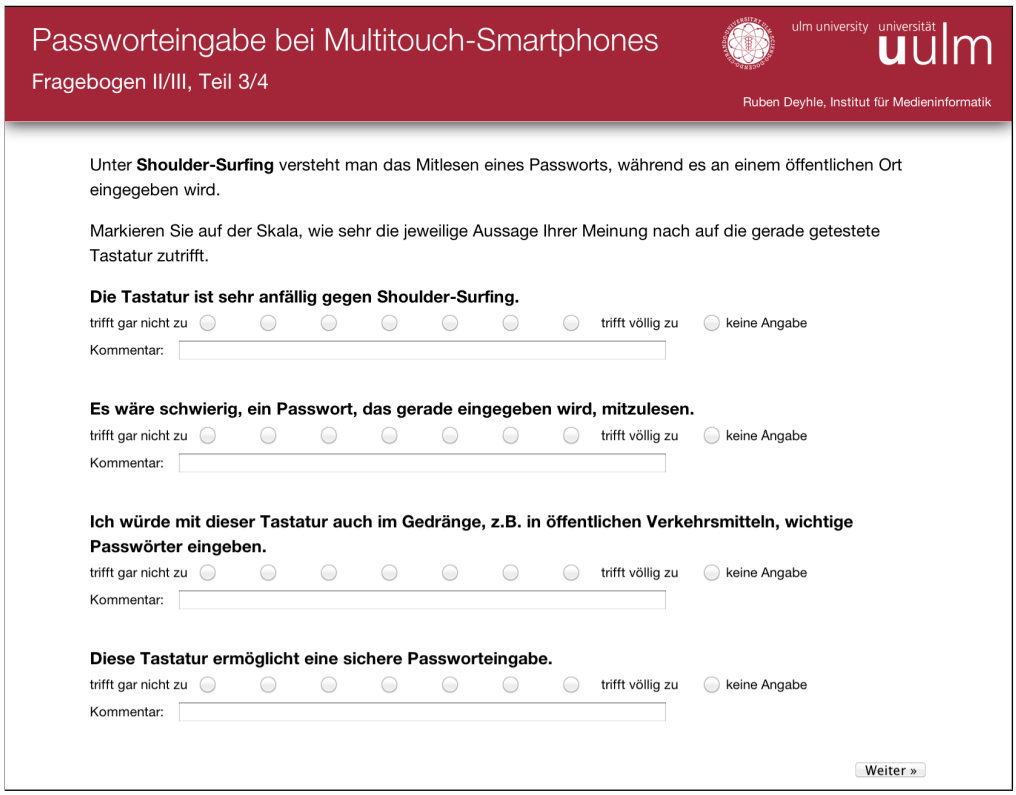
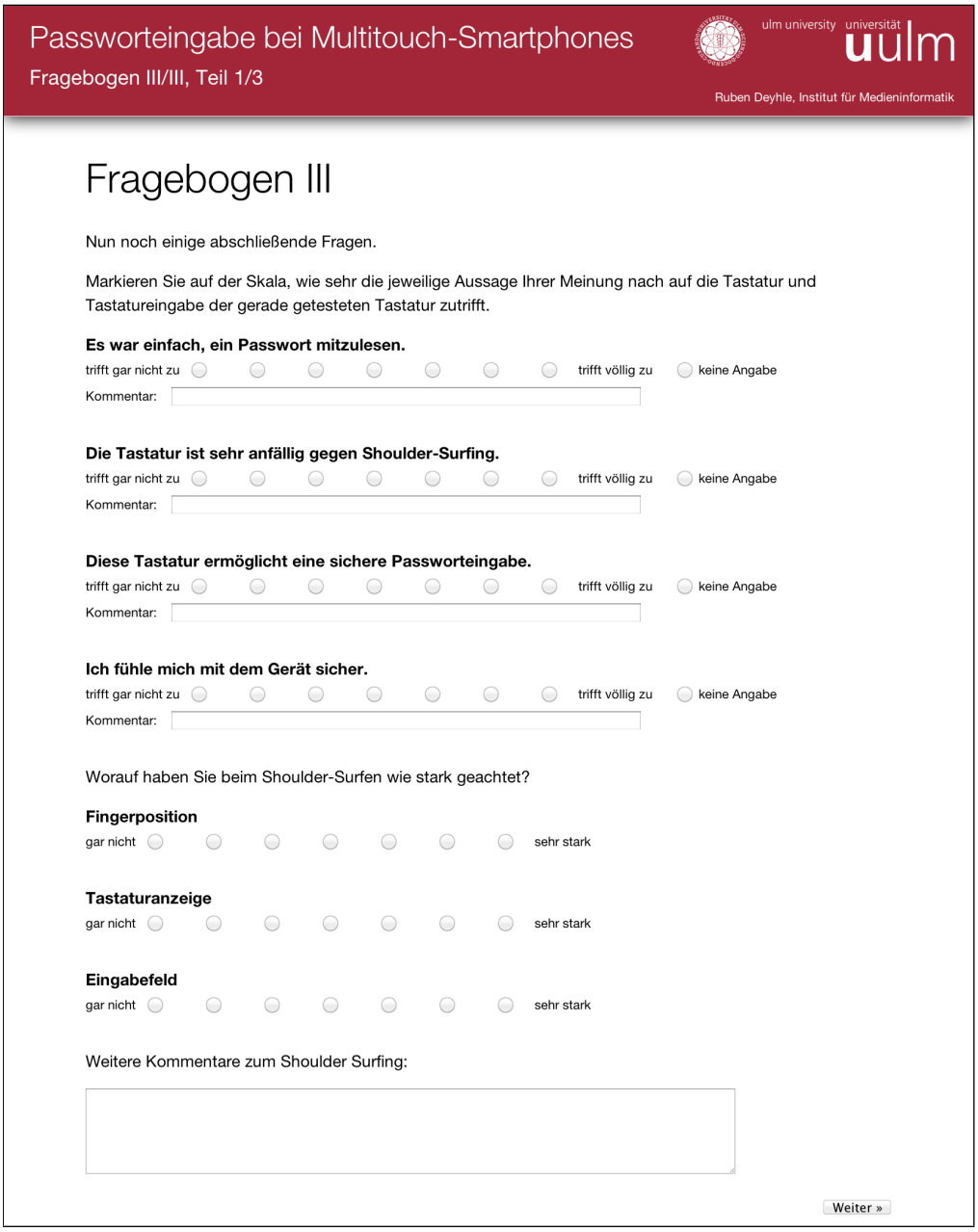


Abbildung A.5.: Fragebogen II/III, 3/4



Passworteingabe bei Multitouch-Smartphones

Fragebogen III/III, Teil 1/3

ulm university universität uulm

Ruben Deyhle, Institut für Medieninformatik

Fragebogen III

Nun noch einige abschließende Fragen.

Markieren Sie auf der Skala, wie sehr die jeweilige Aussage Ihrer Meinung nach auf die Tastatur und Tastatureingabe der gerade getesteten Tastatur zutrifft.

Es war einfach, ein Passwort mitzulesen.

trifft gar nicht zu ○ ○ ○ ○ ○ ○ ○ trifft völlig zu ○ keine Angabe

Kommentar:

Die Tastatur ist sehr anfällig gegen Shoulder-Surfing.

trifft gar nicht zu ○ ○ ○ ○ ○ ○ ○ trifft völlig zu ○ keine Angabe

Kommentar:

Diese Tastatur ermöglicht eine sichere Passworteingabe.

trifft gar nicht zu ○ ○ ○ ○ ○ ○ ○ trifft völlig zu ○ keine Angabe

Kommentar:

Ich fühle mich mit dem Gerät sicher.

trifft gar nicht zu ○ ○ ○ ○ ○ ○ ○ trifft völlig zu ○ keine Angabe

Kommentar:

Worauf haben Sie beim Shoulder-Surfen wie stark geachtet?

Fingerposition

gar nicht ○ ○ ○ ○ ○ ○ ○ sehr stark

Tastaturanzeige

gar nicht ○ ○ ○ ○ ○ ○ ○ sehr stark

Eingabefeld

gar nicht ○ ○ ○ ○ ○ ○ ○ sehr stark

Weitere Kommentare zum Shoulder Surfing:

Weiter »

Abbildung A.6.: Fragebogen III/III, 1/3

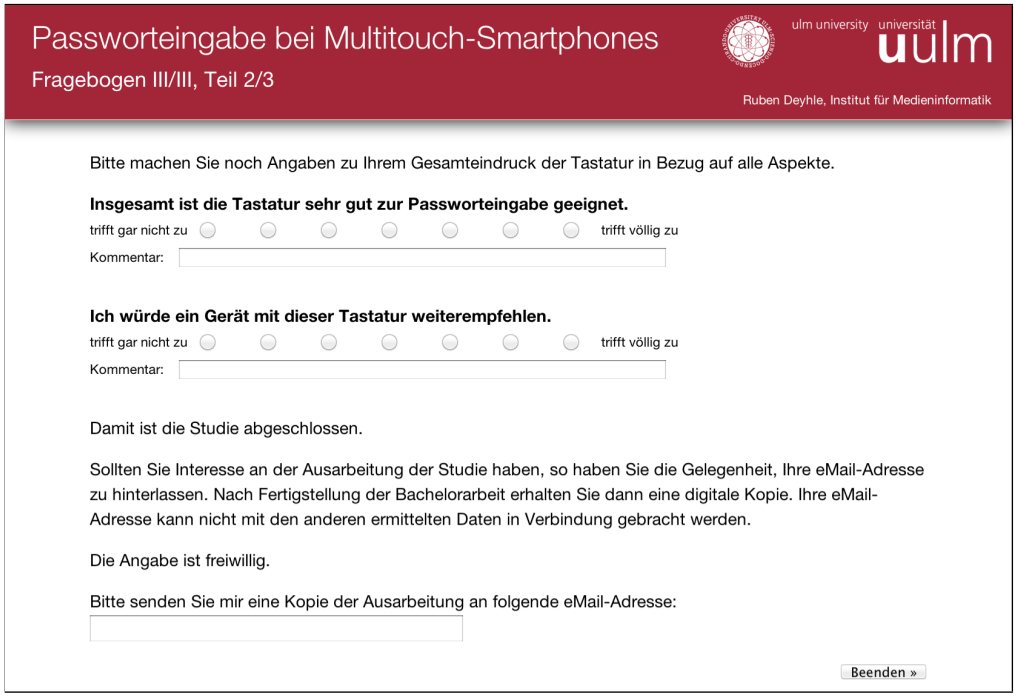


Abbildung A.7.: Fragebogen III/III, 2/3

B. Auswertung

In diesem Anhang sind ergänzende Diagramme und Tabellen aufgeführt, die die statistischen Ergebnisse belegen und nachvollziehbar machen.

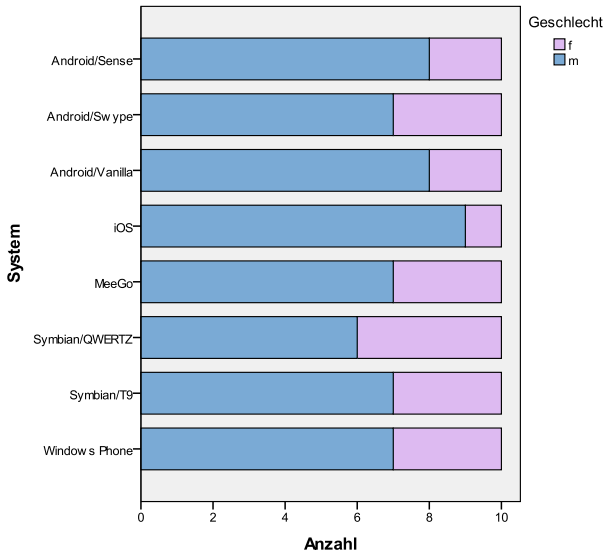


Abbildung B.1.: Geschlechterverteilung der Probanden in den Gruppen.

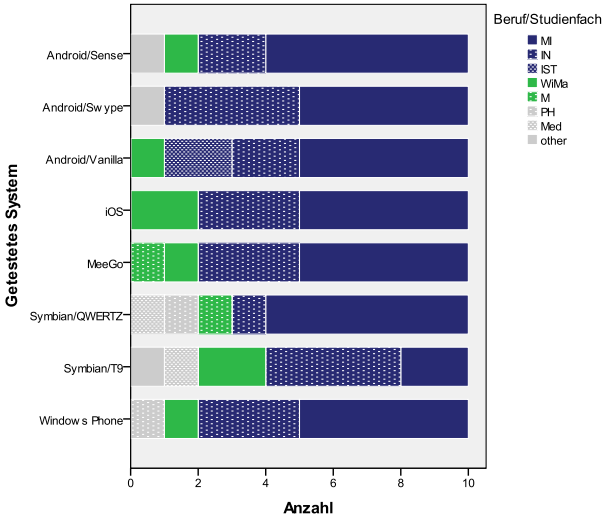


Abbildung B.2.: Beruf/Studienfach der Probanden.

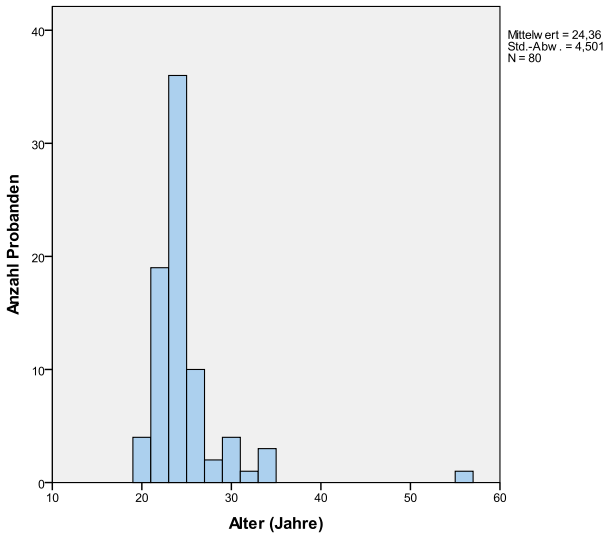


Abbildung B.3.: Altersverteilung der Probanden.

Bericht

zeit

schema	Mittelwert	Median	Standardabweichung
Passwort 4: 0z4Z20zZ2	27,78376	25,45000	9,898242
Passwort 5: 22z1z34Zz	23,48674	22,59700	7,395500
Passwort 1: z00Z2z3	24,69159	19,83750	15,183423
Passwort 2: Z3z00z4Z	26,83167	24,26850	17,200906
Passwort 3: z4z302Z0z	29,91836	28,84350	11,338414
Insgesamt	26,52708	24,25400	12,855447

Abbildung B.4.: Eingabedauer der Usabilitytest-Passwortschemata über alle Systeme.

Kruskal-Wallis-Test

Ränge

	system_nr	N	Mittlerer Rang
zeit	Android/Sense	29	139,74
	Android/Swype	37	162,78
	Android/Vanilla	38	146,36
	iOS	38	83,26
	MeeGo	37	131,24
	Symbian/QWERTZ	41	214,37
	Symbian/T9	20	183,65
	Windows Phone	43	94,58
	Gesamt	283	

Statistik für Test[a,b]

	zeit
Chi-Quadrat	74,403
df	7
Asymptotische Signifikanz	,000

a. Kruskal-Wallis-Test
b. Gruppenvariable: system_nr

Abbildung B.5.: Ergebnisse des Kruskal-Wallis-Tests für die Gesamteingabedauer der Usabilitytest-Passwörter.

Multiple Comparisons

zeit
Bonferroni

(I)system_nr	(J)system_nr	Mittlere Differenz (I-J)	Standardfehler	Sig.	95%-Konfidenzintervall	
					Untergrenze	Obergrenze
Android/Sense	Android/Swype	-2,09194	2,896449	1,000	-11,22896	7,04508
	Android/Vanilla	1,06779	2,879655	1,000	-8,01625	10,15183
	iOS	7,55102	2,879655	,258	-1,53302	16,63506
	MeeGo	3,26971	2,896449	1,000	-5,86731	12,40673
	Symbian/QWERTZ	-9,79371*	2,833687	,018	-18,73274	-,85468
	Symbian/T9	-5,40914	3,394516	1,000	-16,11735	5,29906
	Windows Phone	6,62377	2,806253	,531	-2,22872	15,47626
Android/Swype	Android/Sense	2,09194	2,896449	1,000	-7,04508	11,22896
	Android/Vanilla	3,15973	2,697316	1,000	-5,34912	11,66857
	iOS	9,64296*	2,697316	,012	1,13412	18,15181
	MeeGo	5,36165	2,715238	1,000	-3,20373	13,92703
	Symbian/QWERTZ	-7,70177	2,648185	,110	-16,05563	,65209
	Symbian/T9	-3,31721	3,241271	1,000	-13,54199	6,90758
	Windows Phone	8,71571*	2,618808	,028	,45452	16,97690
Android/Vanilla	Android/Sense	-1,06779	2,879655	1,000	-10,15183	8,01625
	Android/Swype	-3,15973	2,697316	1,000	-11,66857	5,34912
	iOS	6,48324	2,679273	,453	-1,96869	14,93516
	MeeGo	2,20192	2,697316	1,000	-6,30692	10,71076
	Symbian/QWERTZ	-10,86150*	2,629806	,001	-19,15737	-2,56562
	Symbian/T9	-6,47693	3,226272	1,000	-16,65440	3,70054
	Windows Phone	5,55598	2,600221	,938	-2,64657	13,75854
iOS	Android/Sense	-7,55102	2,879655	,258	-16,63506	1,53302
	Android/Swype	-9,64296*	2,697316	,012	-18,15181	-1,13412
	Android/Vanilla	-6,48324	2,679273	,453	-14,93516	1,96869
	MeeGo	-4,28131	2,697316	1,000	-12,79016	4,22753
	Symbian/QWERTZ	-17,34473*	2,629806	,000	-25,64061	-9,04886
	Symbian/T9	-12,96017*	3,226272	,002	-23,13764	-2,78270
	Windows Phone	-,92725	2,600221	1,000	-9,12980	7,27530
MeeGo	Android/Sense	-3,26971	2,896449	1,000	-12,40673	5,86731
	Android/Swype	-5,36165	2,715238	1,000	-13,92703	3,20373
	Android/Vanilla	-2,20192	2,697316	1,000	-10,71076	6,30692
	iOS	4,28131	2,697316	1,000	-4,22753	12,79016
	Symbian/QWERTZ	-13,06342*	2,648185	,000	-21,41728	-4,70956
	Symbian/T9	-8,67885	3,241271	,220	-18,90364	1,54593
	Windows Phone	3,35406	2,618808	1,000	-4,90712	11,61525
Symbian/QWERTZ	Android/Sense	9,79371*	2,833687	,018	,85468	18,73274
	Android/Swype	7,70177	2,648185	,110	-,65209	16,05563
	Android/Vanilla	10,86150*	2,629806	,001	2,56562	19,15737
	iOS	17,34473*	2,629806	,000	9,04886	25,64061
	MeeGo	13,06342*	2,648185	,000	4,70956	21,41728
	Symbian/T9	4,38457	3,185311	1,000	-5,66369	14,43282
	Windows Phone	16,41748*	2,549220	,000	8,37582	24,45915
Symbian/T9	Android/Sense	5,40914	3,394516	1,000	-5,29906	16,11735
	Android/Swype	3,31721	3,241271	1,000	-6,90758	13,54199
	Android/Vanilla	6,47693	3,226272	1,000	-3,70054	16,65440
	iOS	12,96017*	3,226272	,002	2,78270	23,13764
	MeeGo	8,67885	3,241271	,220	-1,54593	18,90364
	Symbian/QWERTZ	-4,38457	3,185311	1,000	-14,43282	5,66369
	Windows Phone	12,03292*	3,160930	,005	2,06158	22,00426
Windows Phone	Android/Sense	-6,62377	2,806253	,531	-15,47626	2,22872
	Android/Swype	-8,71571*	2,618808	,028	-16,97690	-,45452
	Android/Vanilla	-5,55598	2,600221	,938	-13,75854	2,64657
	iOS	,92725	2,600221	1,000	-7,27530	9,12980
	MeeGo	-3,35406	2,618808	1,000	-11,61525	4,90712
	Symbian/QWERTZ	-16,41748*	2,549220	,000	-24,45915	-8,37582
	Symbian/T9	-12,03292*	3,160930	,005	-22,00426	-2,06158

Grundlage: beobachtete Mittelwerte.
Der Fehlerterm ist Mittel der Quadrate(Fehler) = 136,392

*. Die mittlere Differenz ist auf dem ,05-Niveau signifikant.

Abbildung B.6.: Ergebnisse des Bonferroni-Tests für die Gesamteingabedauer der Usabilitytest-Passwörter.

Bericht

fehlerzahl

Getestetes System	Mittelwert	Standardabweichung
Android/Sense	,90	2,093
Android/Swype	,78	1,272
Android/Vanilla	,61	,887
iOS	,34	,745
MeeGo	,65	1,060
Symbian/QWERTZ	1,07	1,421
Symbian/T9	,30	,801
Windows Phone	,09	,294
Insgesamt	,60	1,185

Abbildung B.7.: Mittlere Fehlerzahl pro Passwort.

Kruskal-Wallis-Test

Ränge

	Getestetes System	N	Mittlerer Rang
fehlerzahl	Android/Sense	29	150,28
	Android/Swype	37	153,72
	Android/Vanilla	38	150,14
	iOS	38	128,82
	MeeGo	37	148,30
	Symbian/QWERTZ	41	173,65
	Symbian/T9	20	117,93
	Windows Phone	43	106,40
	Gesamt	283	

Statistik für Test[a,b]

	fehlerzahl
Chi-Quadrat	26,854
df	7
Asymptotische Signifikanz	,000

a. Kruskal-Wallis-Test
b. Gruppenvariable: Getestetes System

Abbildung B.8.: Ergebnisse des Kruskal-Wallis-Tests für die mittlere Fehlerzahl der Usabilitytest-Passwörter.

Mehrfachvergleiche

fehlerzahl
Games-Howell

(I) Getestetes System	(J) Getestetes System	Mittlere Differenz (I-J)	Standardfehler	Signifikanz	95%-Konfidenzintervall	
					Untergrenze	Obergrenze
Android/Sense	Android/Swype	,113	,441	1,000	-1,29	1,52
	Android/Vanilla	,291	,414	,996	-1,04	1,62
	iOS	,554	,407	,868	-,76	1,87
	MeeGo	,248	,426	,999	-1,12	1,61
	Symbian/QWERTZ	-,177	,448	1,000	-1,60	1,24
	Symbian/T9	,597	,428	,855	-,77	1,97
	Windows Phone	,804	,391	,466	-,47	2,08
Android/Swype	Android/Sense	-,113	,441	1,000	-1,52	1,29
	Android/Vanilla	,179	,254	,997	-,62	,97
	iOS	,442	,242	,604	-,32	1,20
	MeeGo	,135	,272	1,000	-,72	,99
	Symbian/QWERTZ	-,289	,305	,980	-1,24	,66
	Symbian/T9	,484	,275	,651	-,38	1,35
	Windows Phone	,691*	,214	,047	,01	1,38
Android/Vanilla	Android/Sense	-,291	,414	,996	-1,62	1,04
	Android/Swype	-,179	,254	,997	-,97	,62
	iOS	,263	,188	,854	-,32	,85
	MeeGo	-,043	,226	1,000	-,75	,66
	Symbian/QWERTZ	-,468	,264	,642	-1,30	,36
	Symbian/T9	,305	,230	,883	-,43	1,04
	Windows Phone	,512*	,151	,029	,03	,99
iOS	Android/Sense	-,554	,407	,868	-1,87	,76
	Android/Swype	-,442	,242	,604	-1,20	,32
	Android/Vanilla	-,263	,188	,854	-,85	,32
	MeeGo	-,307	,212	,832	-,97	,36
	Symbian/QWERTZ	-,731	,253	,092	-1,52	,06
	Symbian/T9	,042	,216	1,000	-,65	,74
	Windows Phone	,249	,129	,537	-,16	,66
MeeGo	Android/Sense	-,248	,426	,999	-1,61	1,12
	Android/Swype	-,135	,272	1,000	-,99	,72
	Android/Vanilla	,043	,226	1,000	-,66	,75
	iOS	,307	,212	,832	-,36	,97
	Symbian/QWERTZ	-,425	,282	,803	-1,30	,46
	Symbian/T9	,349	,250	,855	-,44	1,14
	Windows Phone	,556	,180	,064	-,02	1,13
Symbian/QWERTZ	Android/Sense	,177	,448	1,000	-1,24	1,60
	Android/Swype	,289	,305	,980	-,66	1,24
	Android/Vanilla	,468	,264	,642	-,36	1,30
	iOS	,731	,253	,092	-,06	1,52
	MeeGo	,425	,282	,803	-,46	1,30
	Symbian/T9	,773	,285	,140	-,12	1,67
	Windows Phone	,980*	,226	,002	,26	1,70
Symbian/T9	Android/Sense	-,597	,428	,855	-1,97	,77
	Android/Swype	-,484	,275	,651	-1,35	,38
	Android/Vanilla	-,305	,230	,883	-1,04	,43
	iOS	-,042	,216	1,000	-,74	,65
	MeeGo	-,349	,250	,855	-1,14	,44
	Symbian/QWERTZ	-,773	,285	,140	-1,67	,12
	Windows Phone	,207	,185	,945	-,41	,83
Windows Phone	Android/Sense	-,804	,391	,466	-2,08	,47
	Android/Swype	-,691*	,214	,047	-1,38	-,01
	Android/Vanilla	-,512*	,151	,029	-,99	-,03
	iOS	-,249	,129	,537	-,66	,16
	MeeGo	-,556	,180	,064	-1,13	,02
	Symbian/QWERTZ	-,980*	,226	,002	-1,70	-,26
	Symbian/T9	-,207	,185	,945	-,83	,41

*. Die Differenz der Mittelwerte ist auf dem Niveau 0.05 signifikant.

Abbildung B.9.: Ergebnisse des Games-Howell-Tests für die Gesamteingabedauer der Usabilitytest-Passwörter.

Kruskal-Wallis-Test

Ränge

	system_nr	N	Mittlerer Rang
pssuq_SYSUSE	Android/Sense	10	52,80
	Android/Swype	10	29,35
	Android/Vanilla	10	33,15
	iOS	10	52,50
	MeeGo	10	44,65
	Symbian/QWERTZ	10	26,75
	Symbian/T9	10	29,60
	Windows Phone	10	55,20
	Gesamt	80	
pssuq_INFOQUAL	Android/Sense	9	54,89
	Android/Swype	10	31,35
	Android/Vanilla	10	43,35
	iOS	9	38,22
	MeeGo	10	33,95
	Symbian/QWERTZ	10	31,45
	Symbian/T9	10	33,50
	Windows Phone	10	50,70
	Gesamt	78	
pssuq_INTERQUAL	Android/Sense	10	62,20
	Android/Swype	10	34,60
	Android/Vanilla	10	32,30
	iOS	10	49,80
	MeeGo	10	38,85
	Symbian/QWERTZ	10	24,60
	Symbian/T9	10	32,60
	Windows Phone	10	49,05
	Gesamt	80	

Statistik für Test[a,b]

	pssuq_ SYSUSE	pssuq_ INFOQUAL	pssuq_ INTERQUAL
Chi-Quadrat	18,829	10,783	19,628
df	7	7	7
Asymptotische Signifikanz	,009	,148	,006

a. Kruskal-Wallis-Test
b. Gruppenvariable: system_nr

Abbildung B.10.: Ergebnisse des Kruskal-Wallis-Tests für die Einzelkomponenten des PSSUQ.

Multiple Comparisons

pssuq_SYSUSE
Bonferroni

(I)system_nr	(J)system_nr	Mittlere Differenz (I-J)	Standardfehler	Sig.	95%-Konfidenzintervall	
					Untergrenze	Obergrenze
Android/Sense	Android/Swype	1,2929	,52372	,446	-,4064	2,9921
	Android/Vanilla	1,1125	,52372	1,000	-,5867	2,8117
	iOS	,0321	,52372	1,000	-1,6671	1,7314
	MeeGo	,4000	,52372	1,000	-1,2992	2,0992
	Symbian/QWERTZ	1,5125	,52372	,143	-,1867	3,2117
	Symbian/T9	1,2500	,52372	,550	-,4492	2,9492
	Windows Phone	-,1500	,52372	1,000	-1,8492	1,5492
Android/Swype	Android/Sense	-1,2929	,52372	,446	-2,9921	,4064
	Android/Vanilla	-,1804	,52372	1,000	-1,8796	1,5189
	iOS	-1,2607	,52372	,522	-2,9599	,4385
	MeeGo	-,8929	,52372	1,000	-2,5921	,8064
	Symbian/QWERTZ	,2196	,52372	1,000	-1,4796	1,9189
	Symbian/T9	-,0429	,52372	1,000	-1,7421	1,6564
	Windows Phone	-1,4429	,52372	,208	-3,1421	,2564
Android/Vanilla	Android/Sense	-1,1125	,52372	1,000	-2,8117	,5867
	Android/Swype	,1804	,52372	1,000	-1,5189	1,8796
	iOS	-1,0804	,52372	1,000	-2,7796	,6189
	MeeGo	-,7125	,52372	1,000	-2,4117	,9867
	Symbian/QWERTZ	,4000	,52372	1,000	-1,2992	2,0992
	Symbian/T9	,1375	,52372	1,000	-1,5617	1,8367
	Windows Phone	-1,2625	,52372	,517	-2,9617	,4367
iOS	Android/Sense	-,0321	,52372	1,000	-1,7314	1,6671
	Android/Swype	1,2607	,52372	,522	-,4385	2,9599
	Android/Vanilla	1,0804	,52372	1,000	-,6189	2,7796
	MeeGo	,3679	,52372	1,000	-1,3314	2,0671
	Symbian/QWERTZ	1,4804	,52372	,170	-,2189	3,1796
	Symbian/T9	1,2179	,52372	,640	-,4814	2,9171
	Windows Phone	-,1821	,52372	1,000	-1,8814	1,5171
MeeGo	Android/Sense	-,4000	,52372	1,000	-2,0992	1,2992
	Android/Swype	,8929	,52372	1,000	-,8064	2,5921
	Android/Vanilla	,7125	,52372	1,000	-,9867	2,4117
	iOS	-,3679	,52372	1,000	-2,0671	1,3314
	Symbian/QWERTZ	1,1125	,52372	1,000	-,5867	2,8117
	Symbian/T9	,8500	,52372	1,000	-,8492	2,5492
	Windows Phone	-,5500	,52372	1,000	-2,2492	1,1492
Symbian/QWERTZ	Android/Sense	-1,5125	,52372	,143	-3,2117	,1867
	Android/Swype	-,2196	,52372	1,000	-1,9189	1,4796
	Android/Vanilla	-,4000	,52372	1,000	-2,0992	1,2992
	iOS	-1,4804	,52372	,170	-3,1796	,2189
	MeeGo	-1,1125	,52372	1,000	-2,8117	,5867
	Symbian/T9	-,2625	,52372	1,000	-1,9617	1,4367
	Windows Phone	-1,6625	,52372	,062	-3,3617	,0367
Symbian/T9	Android/Sense	-1,2500	,52372	,550	-2,9492	,4492
	Android/Swype	,0429	,52372	1,000	-1,6564	1,7421
	Android/Vanilla	-,1375	,52372	1,000	-1,8367	1,5617
	iOS	-1,2179	,52372	,640	-2,9171	,4814
	MeeGo	-,8500	,52372	1,000	-2,5492	,8492
	Symbian/QWERTZ	,2625	,52372	1,000	-1,4367	1,9617
	Windows Phone	-1,4000	,52372	,260	-3,0992	,2992
Windows Phone	Android/Sense	,1500	,52372	1,000	-1,5492	1,8492
	Android/Swype	1,4429	,52372	,208	-,2564	3,1421
	Android/Vanilla	1,2625	,52372	,517	-,4367	2,9617
	iOS	,1821	,52372	1,000	-1,5171	1,8814
	MeeGo	,5500	,52372	1,000	-1,1492	2,2492
	Symbian/QWERTZ	1,6625	,52372	,062	-,0367	3,3617
	Symbian/T9	1,4000	,52372	,260	-,2992	3,0992

Grundlage: beobachtete Mittelwerte.
Der Fehlerterm ist Mittel der Quadrate(Fehler) = 1,371

Abbildung B.11.: Ergebnisse des Bonferroni-Tests für die SYSUSE-Komponente des PSSUQ.

Multiple Comparisons

pssuq_INTERQUAL
Bonferroni

(I)system_nr	(J)system_nr	Mittlere Differenz (I-J)	Standardfehler	Sig.	95%-Konfidenzintervall	
					Untergrenze	Obergrenze
Android/Sense	Android/Swype	1,7167	,58487	,125	-,1810	3,6143
	Android/Vanilla	1,7167	,58487	,125	-,1810	3,6143
	iOS	,5833	,58487	1,000	-1,3143	2,4810
	MeeGo	1,3167	,58487	,768	-,5810	3,2143
	Symbian/QWERTZ	2,2833*	,58487	,006	,3857	4,1810
	Symbian/T9	1,7500	,58487	,106	-,1476	3,6476
	Windows Phone	,7167	,58487	1,000	-1,1810	2,6143
Android/Swype	Android/Sense	-1,7167	,58487	,125	-3,6143	,1810
	Android/Vanilla	,0000	,58487	1,000	-1,8976	1,8976
	iOS	-1,1333	,58487	1,000	-3,0310	,7643
	MeeGo	-,4000	,58487	1,000	-2,2976	1,4976
	Symbian/QWERTZ	,5667	,58487	1,000	-1,3310	2,4643
	Symbian/T9	,0333	,58487	1,000	-1,8643	1,9310
	Windows Phone	-1,0000	,58487	1,000	-2,8976	,8976
Android/Vanilla	Android/Sense	-1,7167	,58487	,125	-3,6143	,1810
	Android/Swype	,0000	,58487	1,000	-1,8976	1,8976
	iOS	-1,1333	,58487	1,000	-3,0310	,7643
	MeeGo	-,4000	,58487	1,000	-2,2976	1,4976
	Symbian/QWERTZ	,5667	,58487	1,000	-1,3310	2,4643
	Symbian/T9	,0333	,58487	1,000	-1,8643	1,9310
	Windows Phone	-1,0000	,58487	1,000	-2,8976	,8976
iOS	Android/Sense	-,5833	,58487	1,000	-2,4810	1,3143
	Android/Swype	1,1333	,58487	1,000	-,7643	3,0310
	Android/Vanilla	1,1333	,58487	1,000	-,7643	3,0310
	MeeGo	,7333	,58487	1,000	-1,1643	2,6310
	Symbian/QWERTZ	1,7000	,58487	,136	-,1976	3,5976
	Symbian/T9	1,1667	,58487	1,000	-,7310	3,0643
	Windows Phone	,1333	,58487	1,000	-1,7643	2,0310
MeeGo	Android/Sense	-1,3167	,58487	,768	-3,2143	,5810
	Android/Swype	,4000	,58487	1,000	-1,4976	2,2976
	Android/Vanilla	,4000	,58487	1,000	-1,4976	2,2976
	iOS	-,7333	,58487	1,000	-2,6310	1,1643
	Symbian/QWERTZ	,9667	,58487	1,000	-,9310	2,8643
	Symbian/T9	,4333	,58487	1,000	-1,4643	2,3310
	Windows Phone	-,6000	,58487	1,000	-2,4976	1,2976
Symbian/QWERTZ	Android/Sense	-2,2833*	,58487	,006	-4,1810	-,3857
	Android/Swype	-,5667	,58487	1,000	-2,4643	1,3310
	Android/Vanilla	-,5667	,58487	1,000	-2,4643	1,3310
	iOS	-1,7000	,58487	,136	-3,5976	,1976
	MeeGo	-,9667	,58487	1,000	-2,8643	,9310
	Symbian/T9	-,5333	,58487	1,000	-2,4310	1,3643
	Windows Phone	-1,5667	,58487	,256	-3,4643	,3310
Symbian/T9	Android/Sense	-1,7500	,58487	,106	-3,6476	,1476
	Android/Swype	-,0333	,58487	1,000	-1,9310	1,8643
	Android/Vanilla	-,0333	,58487	1,000	-1,9310	1,8643
	iOS	-1,1667	,58487	1,000	-3,0643	,7310
	MeeGo	-,4333	,58487	1,000	-2,3310	1,4643
	Symbian/QWERTZ	,5333	,58487	1,000	-1,3643	2,4310
	Windows Phone	-1,0333	,58487	1,000	-2,9310	,8643
Windows Phone	Android/Sense	-,7167	,58487	1,000	-2,6143	1,1810
	Android/Swype	1,0000	,58487	1,000	-,8976	2,8976
	Android/Vanilla	1,0000	,58487	1,000	-,8976	2,8976
	iOS	-,1333	,58487	1,000	-2,0310	1,7643
	MeeGo	,6000	,58487	1,000	-1,2976	2,4976
	Symbian/QWERTZ	1,5667	,58487	,256	-,3310	3,4643
	Symbian/T9	1,0333	,58487	1,000	-,8643	2,9310

Grundlage: beobachtete Mittelwerte.
Der Fehlerterm ist Mittel der Quadrate(Fehler) = 1,710

*. Die mittlere Differenz ist auf dem ,05-Niveau signifikant.

Abbildung B.12.: Ergebnisse des Bonferroni-Tests für die INTERQUAL-Komponente des PSSUQ.

Mehrfachvergleiche

pssuq_OVERALL
Tukey-HSD

(I) system_nr	(J) system_nr	Mittlere Differenz (I-J)	Standardfehler	Signifikanz	95%-Konfidenzintervall	
					Untergrenze	Obergrenze
Android/Sense	Android/Swype	1,32866	,46149	,092	-,1120	2,7693
	Android/Vanilla	1,07530	,46149	,292	-,3654	2,5160
	iOS	,32440	,46149	,997	-1,1163	1,7651
	MeeGo	,74335	,46149	,742	-,6973	2,1840
	Symbian/QWERTZ	1,58778*	,46149	,021	,1471	3,0285
	Symbian/T9	1,33545	,46149	,089	-,1052	2,7761
	Windows Phone	,10703	,46149	1,000	-1,3337	1,5477
Android/Swype	Android/Sense	-1,32866	,46149	,092	-2,7693	,1120
	Android/Vanilla	-,25335	,46149	,999	-1,6940	1,1873
	iOS	-1,00425	,46149	,378	-2,4449	,4364
	MeeGo	-,58531	,46149	,907	-2,0260	,8554
	Symbian/QWERTZ	,25912	,46149	,999	-1,1816	1,6998
	Symbian/T9	,00679	,46149	1,000	-1,4339	1,4475
	Windows Phone	-1,22163	,46149	,156	-2,6623	,2191
Android/Vanilla	Android/Sense	-1,07530	,46149	,292	-2,5160	,3654
	Android/Swype	,25335	,46149	,999	-1,1873	1,6940
	iOS	-,75090	,46149	,733	-2,1916	,6898
	MeeGo	-,33196	,46149	,996	-1,7726	1,1087
	Symbian/QWERTZ	,51247	,46149	,952	-,9282	1,9532
	Symbian/T9	,26014	,46149	,999	-1,1805	1,7008
	Windows Phone	-,96828	,46149	,426	-2,4090	,4724
iOS	Android/Sense	-,32440	,46149	,997	-1,7651	1,1163
	Android/Swype	1,00425	,46149	,378	-,4364	2,4449
	Android/Vanilla	,75090	,46149	,733	-,6898	2,1916
	MeeGo	,41895	,46149	,984	-1,0217	1,8596
	Symbian/QWERTZ	1,26338	,46149	,128	-,1773	2,7041
	Symbian/T9	1,01104	,46149	,370	-,4296	2,4517
	Windows Phone	-,21738	,46149	1,000	-1,6581	1,2233
MeeGo	Android/Sense	-,74335	,46149	,742	-2,1840	,6973
	Android/Swype	,58531	,46149	,907	-,8554	2,0260
	Android/Vanilla	,33196	,46149	,996	-1,1087	1,7726
	iOS	-,41895	,46149	,984	-1,8596	1,0217
	Symbian/QWERTZ	,84443	,46149	,602	-,5963	2,2851
	Symbian/T9	,59210	,46149	,902	-,8486	2,0328
	Windows Phone	-,63632	,46149	,864	-2,0770	,8044
Symbian/QWERTZ	Android/Sense	-1,58778*	,46149	,021	-3,0285	-,1471
	Android/Swype	-,25912	,46149	,999	-1,6998	1,1816
	Android/Vanilla	-,51247	,46149	,952	-1,9532	,9282
	iOS	-1,26338	,46149	,128	-2,7041	,1773
	MeeGo	-,84443	,46149	,602	-2,2851	,5963
	Symbian/T9	-,25233	,46149	,999	-1,6930	1,1884
	Windows Phone	-1,48075*	,46149	,040	-2,9214	-,0401
Symbian/T9	Android/Sense	-1,33545	,46149	,089	-2,7761	,1052
	Android/Swype	-,00679	,46149	1,000	-1,4475	1,4339
	Android/Vanilla	-,26014	,46149	,999	-1,7008	1,1805
	iOS	-1,01104	,46149	,370	-2,4517	,4296
	MeeGo	-,59210	,46149	,902	-2,0328	,8486
	Symbian/QWERTZ	,25233	,46149	,999	-1,1884	1,6930
	Windows Phone	-1,22842	,46149	,152	-2,6691	,2123
Windows Phone	Android/Sense	-,10703	,46149	1,000	-1,5477	1,3337
	Android/Swype	1,22163	,46149	,156	-,2191	2,6623
	Android/Vanilla	,96828	,46149	,426	-,4724	2,4090
	iOS	,21738	,46149	1,000	-1,2233	1,6581
	MeeGo	,63632	,46149	,864	-,8044	2,0770
	Symbian/QWERTZ	1,48075*	,46149	,040	,0401	2,9214
	Symbian/T9	1,22842	,46149	,152	-,2123	2,6691

*. Die Differenz der Mittelwerte ist auf dem Niveau 0.05 signifikant.

Abbildung B.13.: Ergebnisse des Tukey-Tests für die OVERALL-Komponente des PSSUQ.

ONEWAY ANOVA

pssuq_OVERALL

	Quadratsumme	df	Mittel der Quadrate	F	Signifikanz
Zwischen den Gruppen	26,109	7	3,730	3,503	,003
Innerhalb der Gruppen	76,671	72	1,065		
Gesamt	102,780	79			

Abbildung B.14.: Ergebnisse der einfaktoriellen ANOVA für die OVERALL-Komponente des PSSUQ.

Kruskal-Wallis-Test

Ränge

	Getestetes System	N	Mittlerer Rang
levenshtein_min	Android/Sense	30	99,23
	Android/Swype	30	103,02
	Android/Vanilla	30	111,07
	iOS	30	131,02
	MeeGo	30	115,40
	Symbian/QWERTZ	30	135,48
	Symbian/T9	30	153,88
	Windows Phone	30	114,90
	Gesamt	240	

Statistik für Test[a,b]

	levenshtein_min
Chi-Quadrat	14,978
df	7
Asymptotische Signifikanz	,036

a. Kruskal-Wallis-Test
b. Gruppenvariable: Getestetes System

Abbildung B.15.: Ergebnisse des Kruskal-Wallis-Tests für die minimale Levenshtein-Distanz beim Shoulder Surfing.

Mehrfachvergleiche

fehlerzahl
Games-Howell

(I) Getestetes System	(J) Getestetes System	Mittlere Differenz (I-J)	Standardfehler	Signifikanz	95%-Konfidenzintervall	
					Untergrenze	Obergrenze
Android/Sense	Android/Swype	,113	,441	1,000	-1,29	1,52
	Android/Vanilla	,291	,414	,996	-1,04	1,62
	iOS	,554	,407	,868	-,76	1,87
	MeeGo	,248	,426	,999	-1,12	1,61
	Symbian/QWERTZ	-,177	,448	1,000	-1,60	1,24
	Symbian/T9	,597	,428	,855	-,77	1,97
	Windows Phone	,804	,391	,466	-,47	2,08
Android/Swype	Android/Sense	-,113	,441	1,000	-1,52	1,29
	Android/Vanilla	,179	,254	,997	-,62	,97
	iOS	,442	,242	,604	-,32	1,20
	MeeGo	,135	,272	1,000	-,72	,99
	Symbian/QWERTZ	-,289	,305	,980	-1,24	,66
	Symbian/T9	,484	,275	,651	-,38	1,35
	Windows Phone	,691*	,214	,047	,01	1,38
Android/Vanilla	Android/Sense	-,291	,414	,996	-1,62	1,04
	Android/Swype	-,179	,254	,997	-,97	,62
	iOS	,263	,188	,854	-,32	,85
	MeeGo	-,043	,226	1,000	-,75	,66
	Symbian/QWERTZ	-,468	,264	,642	-1,30	,36
	Symbian/T9	,305	,230	,883	-,43	1,04
	Windows Phone	,512*	,151	,029	,03	,99
iOS	Android/Sense	-,554	,407	,868	-1,87	,76
	Android/Swype	-,442	,242	,604	-1,20	,32
	Android/Vanilla	-,263	,188	,854	-,85	,32
	MeeGo	-,307	,212	,832	-,97	,36
	Symbian/QWERTZ	-,731	,253	,092	-1,52	,06
	Symbian/T9	,042	,216	1,000	-,65	,74
	Windows Phone	,249	,129	,537	-,16	,66
MeeGo	Android/Sense	-,248	,426	,999	-1,61	1,12
	Android/Swype	-,135	,272	1,000	-,99	,72
	Android/Vanilla	,043	,226	1,000	-,66	,75
	iOS	,307	,212	,832	-,36	,97
	Symbian/QWERTZ	-,425	,282	,803	-1,30	,46
	Symbian/T9	,349	,250	,855	-,44	1,14
	Windows Phone	,556	,180	,064	-,02	1,13
Symbian/QWERTZ	Android/Sense	,177	,448	1,000	-1,24	1,60
	Android/Swype	,289	,305	,980	-,66	1,24
	Android/Vanilla	,468	,264	,642	-,36	1,30
	iOS	,731	,253	,092	-,06	1,52
	MeeGo	,425	,282	,803	-,46	1,30
	Symbian/T9	,773	,285	,140	-,12	1,67
	Windows Phone	,980*	,226	,002	,26	1,70
Symbian/T9	Android/Sense	-,597	,428	,855	-1,97	,77
	Android/Swype	-,484	,275	,651	-1,35	,38
	Android/Vanilla	-,305	,230	,883	-1,04	,43
	iOS	-,042	,216	1,000	-,74	,65
	MeeGo	-,349	,250	,855	-1,14	,44
	Symbian/QWERTZ	-,773	,285	,140	-1,67	,12
	Windows Phone	,207	,185	,945	-,41	,83
Windows Phone	Android/Sense	-,804	,391	,466	-2,08	,47
	Android/Swype	-,691*	,214	,047	-1,38	-,01
	Android/Vanilla	-,512*	,151	,029	-,99	-,03
	iOS	-,249	,129	,537	-,66	,16
	MeeGo	-,556	,180	,064	-1,13	,02
	Symbian/QWERTZ	-,980*	,226	,002	-1,70	-,26
	Symbian/T9	-,207	,185	,945	-,83	,41

*. Die Differenz der Mittelwerte ist auf dem Niveau 0.05 signifikant.

Abbildung B.16.: Ergebnisse des Games-Howell-Tests für die minimale Levenshtein-Distanz beim Shoulder Surfing.

Vor dem Shoulder-Surfing

Reliabilitätsstatistiken

Cronbachs Alpha	Anzahl der Items
,605	4

Item-Skala-Statistiken

	Skalenmittelwert, wenn Item weggelassen	Skalenvarianz, wenn Item weggelassen	Korrigierte Item-Skala-Korrelation	Cronbachs Alpha, wenn Item weggelassen
Die Tastatur ist NICHT sehr anfällig gegen Shoulder-Surfing.	7,1757	12,667	,250	,624
Es wäre schwierig, ein Passwort, das gerade eingegeben wird, mitzulesen.	6,9189	10,213	,423	,505
Ich würde mit dieser Tastatur auch im Gedränge, z.B. in öffentlichen Verkehrsmitteln, wichtige Passwörter eingeben.	7,6081	9,940	,402	,523
Diese Tastatur ermöglicht eine sichere Passworteingabe.	7,0000	10,575	,479	,468

Nach dem Shoulder-Surfing

Reliabilitätsstatistiken

Cronbachs Alpha	Anzahl der Items
,751	4

Item-Skala-Statistiken

	Skalenmittelwert, wenn Item weggelassen	Skalenvarianz, wenn Item weggelassen	Korrigierte Item-Skala-Korrelation	Cronbachs Alpha, wenn Item weggelassen
Es war NICHT einfach, ein Passwort mitzulesen.	7,0513	15,790	,577	,677
Die Tastatur ist NICHT sehr anfällig gegen Shoulder-Surfing	7,5128	17,084	,561	,685
Diese Tastatur ermöglicht eine sichere Passworteingabe.	7,5641	17,210	,634	,649
Ich fühle mich mit dem Gerät sicher.	7,5641	19,132	,428	,755

Abbildung B.17.: Reliabilitätsanalyse der Fragebogen-Items vor und nach dem Shoulder-Surfing.

ONEWAY ANOVA

		Quadratsumme	df	Mittel der Quadrate	F	Signifikanz
shoulderpre_OVERALL	Zwischen den Gruppen	7,619	7	1,088	1,069	,392
	Innerhalb der Gruppen	72,281	71	1,018		
	Gesamt	79,900	78			
shoulderafter_OVERALL	Zwischen den Gruppen	11,910	7	1,701	,954	,471
	Innerhalb der Gruppen	128,417	72	1,784		
	Gesamt	140,326	79			

Abbildung B.18.: Ergebnisse der einfaktoriellen ANOVA für die Fragebögen vor und nach dem Shoulder-Experiment.

Bericht

system_nr		shoulderpre_OVERALL	shoulderafter_OVERALL
Android/Sense	Mittelwert	3,2500	3,8750
	Median	3,5000	4,3750
	Standardabweichung	,92702	1,44458
Android/Swype	Mittelwert	3,7333	2,6750
	Median	3,5000	2,6250
	Standardabweichung	1,45551	1,18468
Android/Vanilla	Mittelwert	3,6500	3,6250
	Median	3,3750	3,2500
	Standardabweichung	,97325	1,50578
iOS	Mittelwert	3,2000	3,3583
	Median	2,8750	3,1250
	Standardabweichung	1,11679	1,54912
MeeGo	Mittelwert	2,8833	3,6750
	Median	2,6250	3,5000
	Standardabweichung	,77996	1,24192
Symbian/QWERTZ	Mittelwert	3,7500	3,9750
	Median	3,7500	4,2500
	Standardabweichung	,84984	1,16934
Symbian/T9	Mittelwert	3,0250	3,4250
	Median	3,0000	3,3750
	Standardabweichung	,93132	1,00727
Windows Phone	Mittelwert	3,4667	3,2250
	Median	3,3750	2,7500
	Standardabweichung	,86976	1,47879
Insgesamt	Mittelwert	3,3713	3,4792
	Median	3,2500	3,5000
	Standardabweichung	1,01211	1,33277

Abbildung B.19.: Mittelwerte, Zentralwerte und Standardabweichungen für die Fragebögen vor und nach dem Shoulder-Experiment.

Kruskal-Wallis-Test

Ränge

	Zone	N	Mittlerer Rang
Fehler (Ø)	L	58	37,10
	M	2	43,00
	R	20	50,10
	Gesamt	80	

Statistik für Test[a,b]

	Erfolg
Chi-Quadrat	4,701
df	2
Asymptotische Signifikanz	,095

a. Kruskal-Wallis-Test
b. Gruppenvariable: Zone

Abbildung B.20.: Ergebnisse des Kruskal-Wallis-Tests für die Shoulder-Surfing-Zonen.

Korrelationen

			Fehler	Fingerposition	Tastaturanzeige	Eingabefeld
Spearman-Rho	Fehler	Korrelationskoeffizient	1,000	,064	,207	-,303**
		Sig. (2-seitig)	.	,573	,066	,006
		N	80	80	80	80
	Fingerposition	Korrelationskoeffizient	,064	1,000	,309**	-,335**
		Sig. (2-seitig)	,573	.	,005	,002
		N	80	80	80	80
	Tastaturanzeige	Korrelationskoeffizient	,207	,309**	1,000	-,561**
		Sig. (2-seitig)	,066	,005	.	,000
		N	80	80	80	80
	Eingabefeld	Korrelationskoeffizient	-,303**	-,335**	-,561**	1,000
		Sig. (2-seitig)	,006	,002	,000	.
		N	80	80	80	80

**. Die Korrelation ist auf dem 0,01 Niveau signifikant (zweiseitig).

Abbildung B.21.: Spearman-Korrelationskoeffizienten zur Shoulder-Surfing-Strategie.

Bericht

original		char1	char2	char3	char4	char5	char6	char7	char8
g<G,9o-l	Mittelwert	,65	,76	,80	,53	,77	,55	,60	,28
	Standardabweichung	,477	,426	,402	,500	,423	,499	,490	,448
sunshine	Mittelwert	,75	,61	,65	,75	,51	,52	,65	,62
	Standardabweichung	,436	,490	,477	,436	,501	,501	,477	,488
uMo.37 x	Mittelwert	,80	,69	,63	,52	,81	,71	,17	,68
	Standardabweichung	,409	,462	,485	,501	,394	,456	,379	,466
Insgesamt	Mittelwert	,73	,69	,69	,59	,70	,59	,47	,52
	Standardabweichung	,446	,462	,461	,492	,457	,491	,499	,500

Abbildung B.22.: Erkennungsraten und Standardabweichungen der Shoulder-Surfing-Passwörter.

Kruskal-Wallis-Test

Ränge

	Getestetes System	N	Mittlerer Rang
"Insgesamt ist die Tastatur sehr gut zur Passworteingabe geeignet."	Android/Sense	10	38,75
	Android/Swype	10	34,00
	Android/Vanilla	10	34,30
	iOS	9	51,50
	MeeGo	9	42,00
	Symbian/QWERTZ	10	39,40
	Symbian/T9	10	32,90
	Windows Phone	10	44,60
	Gesamt	78	
"Ich würde ein Gerät mit dieser Tastatur weiterempfehlen."	Android/Sense	10	47,10
	Android/Swype	10	44,55
	Android/Vanilla	10	40,05
	iOS	9	40,78
	MeeGo	9	40,22
	Symbian/QWERTZ	10	32,40
	Symbian/T9	10	25,75
	Windows Phone	10	45,35
	Gesamt	78	

Statistik für Test[a,b]

	"Insgesamt ist die Tastatur sehr gut zur Passworteingabe geeignet."	"Ich würde ein Gerät mit dieser Tastatur weiterempfehlen."
Chi-Quadrat	5,349	7,225
df	7	7
Asymptotische Signifikanz	,617	,406

a. Kruskal-Wallis-Test
b. Gruppenvariable: Getestetes System

Abbildung B.23.: Ergebnisse des Kruskal-Wallis-Tests für die beiden abschließenden Likert-Fragen.

Literaturverzeichnis

[1] ACNIELSEN: *The Mobile Media Report Q3 2011*. http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2011-Reports/state-of-mobile-Q3-2011.pdf. Version: Dezember 2011

[2] BLANDFORD, Rafe: *S60 5th Edition - touch and sensor enablers, enriched Internet and multimedia*. AllAboutSymbian. http://www.allaboutsymbian.com/news/item/8203_S60_5th_Edition-touch_and_sens.php. Version: 2.10.08, Abruf: 26.11.11

[3] BONNEAU, Joseph: *Passwords in the wild, part I*. Light Blue Touchpaper. http://www.lightbluetouchpaper.org/2010/07/27/passwords-in-the-wild-part-i-the-gap-between-theory-and-implementation/. Version: 27.7.10, Abruf: 26.11.11

[4] BUCHOUX, A. ; CLARKE, N.L.: Deployment of Keystroke Analysis on a Smartphone. In: *Proceedings of the 6th Australian Information Security Management Conference*. Perth, Western Australia : SECAU - Security Research Centre, 2008, 40–47

[5] BÜHNER, Markus ; ZIEGLER, Matthias: *Statistik für Psychologen und Sozialwissenschaftler*. 1. Aufl. München : Pearson Deutschland GmbH, 2009. – ISBN 3827372747

[6] CAI, Liang ; CHEN, Hao: TouchLogger: inferring keystrokes on touch screen from smartphone motion. In: *Proceedings of the 6th USENIX conference on Hot topics in security*. Berkeley, CA, USA : USENIX Association, 2011 (HotSec'11), 9–9

[7] DOTSON, Kit: *LulzSec Dumps Cargo of 62k Logins, 12k of them Belong to WriterSpace.com Users*. Silicon Angle. http://siliconangle.com/blog/2011/06/17/lulzsec-dumps-cargo-of-62k-logins-12k-of-them-belong-to-writerspace-com-users/. Version: 17.6.11, Abruf: 4.2.12

[8] EARNSHAW, Andy: *Cross browser oninput event for jQuery*. http://whattheheadsaid.com/projects/input-special-event. Version: 18.10.10, Abruf: 4.2.12

[9] FLORENCIO, Dinei ; HERLEY, Cormac: A large-scale study of web password habits. In: *Proceedings of the 16th international conference on World Wide Web*. New York, NY, USA : ACM, 2007 (WWW '07). – ISBN 978–1–59593–654–7, 657–666

[10] FRIED, Ina: *Exclusive: Nokia to Exit Symbian, Low-End Phone Businesses in North America*. AllThingsD. http://allthingsd.com/20110809/exclusive-nokia-to-exit-symbian-low-end-phone-businesses-in-north-america/. Version: 9.8.11, Abruf: 23.11.11

[11] GILLELAND, Michael: *Levenshtein Distance, in Three Flavors*. Merriam Park. http://www.merriampark.com/ld.htm, Abruf: 17.2.12

[12] GOOGLE INC.: *Neues Google-Konto erstellen*. https://accounts.google.com/SignUp, Abruf: 13.2.12

[13] HEATER, Brian: *Nokia's Stephen Elop is still over MeeGo, even if the N9 is a hit*. Engadget. http://www.engadget.com/2011/06/25/nokias-stephen-elop-is-still-over-meego-even-if-the-n9-is-a-hi/. Version: 25.6.11, Abruf: 23.11.11

[14] HEWLETT-PACKARD COMPANY: *HP to Contribute webOS to Open Source*. HP Newsroom. http://www.hp.com/hpinfo/newsroom/press/2011/111209xa.html. Version: 9.12.2011, Abruf: 6.2.12

[15] ICQ: *ICQ beitreten*. http://www.icq.com/join/de, Abruf: 13.2.12

[16] LEWIS, James R.: IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. In: *International Journal of Human-Computer Interaction* 7 (1995), Nr. 1, 57-78. http://dx.doi.org/10.1080/10447319509526110. – DOI 10.1080/10447319509526110

[17] @LULZSEC: *Tweet*. Twitter. https://twitter.com/#!/LulzSec/status/81327464156119040. Version: 16.6.11, Abruf: 4.2.12

[18] MAI, Florian: *Mit Swype schneller schreiben auf Smartphones*. Gadgetspy. http://gadgetspy.de/2011/08/mit-swype-schneller-schreiben-auf-smartphones/, Abruf: 16.11.11

[19] NICHOLSON, John: *Design of a Multi-Touch Shoulder Surfing Resilient Graphical Password*, Newcastle University, Diss., 2009. http://john-nicholson.co.uk/papers/John_Nicholson_Dissertation.pdf

[20] NOKIA: *Symbian Anna now available for download*. Nokia Press Releases. http://press.nokia.com/2011/08/18/symbian-anna-now-available-for-download/. Version: 18.8.11, Abruf: 26.11.11

[21] NOKIA STORE: *Swype brought to you by Nokia*. http://store.ovi.com/content/58438, Abruf: 21.11.11

[22] OPEN HANDSET ALLIANCE: *Android*. http://www.openhandsetalliance.com/android_overview.html, Abruf: 4.2.12

[23] PAKALSKI, Ingo: *Keine Zukunft: HP gibt WebOS-Geräte auf*. Golem. http://www.golem.de/1108/85841.html. Version: 18.8.11, Abruf: 6.2.12

[24] http://www.swype.com/

[25] TARI, Furkan ; OZOK, A. A. ; HOLDEN, Stephen H.: A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords. In: *Proceedings of the second symposium on Usable privacy and security*. New York, NY, USA : ACM, 2006 (SOUPS '06). – ISBN 1–59593–448–0, 56–66

[26] UNICODE CONSORTIUM: *The Unicode Standard Version 6.0*. http://www.unicode.org/versions/Unicode6.0.0/UnicodeStandard-6.0.pdf. Version: Feb. 2011

[27] UNIVERSITÄT ZÜRICH: *Kruskal-Wallis-Test*. http://www.methodenberatung.uzh.ch/datenanalyse/unterschiede/zentral/kruskal.html, Abruf: 13.2.12

[28] WHATWG: *HTML Living Standard – Web application APIs*. http://www.whatwg.org/specs/web-apps/current-work/multipage/webappapis.html#handler-oninput, Abruf: 14.2.12

[29] YAN, J. ; BLACKWELL, A. ; ANDERSON, R. ; GRANT, A.: Password Memorability and Security: Empirical Results. In: *IEEE Security & Privacy* 2 (2004), Nr. 5

Erklärung

Ich, Ruben Deyhle, Matrikelnummer [redacted], erkläre, dass ich die Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den .....

Ruben Deyhle