

A hand holding a smartphone, with a green overlay and white text. The text is arranged in three lines: 'K M U', 'G O E S', and 'M O B I L E'. A horizontal line is positioned to the right of the 'U' in the first line.

K M U —
G O E S
M O B I L E

INSTITUT
Internet-Technologien & -Anwendungen

Johannes Feiner (Hrsg.) / Petra Kletzenbauer (Hrsg.)

KMU goes mobile Sammelband¹

**Internet-Technologien und -Anwendungen
FH JOANNEUM**

März 2015

¹Das Projekt KMU goes mobile wurde gefördert von der Österreichischen Forschungsförderungsgesellschaft FFG im Rahmen der COIN - (COoperation und INovation) Förderschiene.

© 2015 Verlag der FH JOANNEUM Gesellschaft mbH

Herausgeber: Johannes Feiner (Hsg) und Petra Kletzenbauer (Hsg)
Michael Brickmann, Sonja Gögele, Mathias Knoll
Elmar Krajnc, Franz Niederl, Manfred Pamsl
Marvin Puchmüller, Michael Ulm, Kaja Unger und
Wilhelm Zugaj

Verlag der FH JOANNEUM Gesellschaft mbH
Alte Poststraße 149
A-8020 Graz
www.fh-joanneum.at

ISBN print: 978-3-902103-54-3
ISBN eBook: 978-3-902103-55-0

Das Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt.
Die Verwertung ist nur im Rahmen der Creative Commons Lizenz
Namensnennung - Nicht-kommerziell - Keine Bearbeitung (CC
BY-NC-ND 3.0 Österreich¹) gestattet.



¹<http://creativecommons.org/licenses/by-nc-nd/3.0/at/>

Inhaltsverzeichnis

1	Vorwort	9
2	Einleitung	11
	<i>Michael Ulm, Johannes Feiner</i>	
3	Die Private Cloud Alternative - Daten und Services in der eigenen Hand	21
	<i>Manfred Pamsl</i>	
3.1	Einführung	21
3.2	Eigenschaften von Cloud Computing	22
3.3	Servicemodelle	24
3.4	Bereitstellungsmodelle	24
3.5	Potentielle Risiken einer Public Cloud Lösung	26
3.6	Private Cloud Alternative ownCloud	27
3.7	Private Cloud Alternative OpenStack	29
3.8	Zusammenfassung	33
4	Rechtliche Rahmenbedingungen bei der Nutzung einer Cloud im Unternehmensumfeld	35
	<i>Michael Brickmann</i>	
4.1	Datenarten	35
4.2	Vertrag	36
4.2.1	Ort der Speicherung	36
4.2.2	Art der Speicherung	36
4.2.3	Beendigung des Vertrags	37
4.2.4	Insolvenz / Veräußerung des Anbieters	37
4.2.5	Verfügbarkeit	37
4.3	Sicherheitsaudits	37

4.4	Zertifizierung	37
5	Understanding the social web – Soziale Medien in KMU	39
	<i>Franz Niederl, Elmar Krajnc und Michael Ulm</i>	
5.1	Einleitung	39
5.2	Definition von Social Media	39
5.3	Mobile Nutzung von Social Media	43
5.4	Modelle der Plattformen	46
5.4.1	Relationship Modelle	46
5.4.2	Honeycombmodell	48
5.5	Einsatz von Social Media	51
5.5.1	Posting-Strategie	52
5.5.2	Unternehmen vs. Mensch	52
5.5.3	Beschwerdemanagement	53
5.5.4	Interaktion	53
5.5.5	Monitoring	54
5.5.6	Verantwortlichkeiten	54
5.5.7	Zeitaufwand	55
5.6	Zusammenfassung	55
6	Bring your own device – BOYD	59
	<i>Kaja Unger</i>	
6.1	BYOD – Chancen nutzen – Risiken vermeiden	59
6.2	Technische Gefahren	60
6.3	Arbeitsrechtliche Gefahren	62
6.4	Support	63
6.5	Rechtsrahmen	64
7	Der Weg zur eigenen App – Herausforderungen im Mobile App Development	67
	<i>Johannes Feiner, Elmar Krajnc, Franz Niederl</i>	
7.1	Herausforderungen bei der Entwicklung mobiler Apps	67
7.2	Was schätzen die BenutzerInnen?	69
7.3	Wie entsteht eine mobile App?	70
7.4	Warum im Team entwickeln?	72
7.5	Wann ist die Software fertig?	73
7.5.1	Deployment und Marketing	74

7.5.2	Mobile-Usability-Testing	75
7.6	Wer übernimmt Pflege und Wartung?	76
7.7	Native App oder Web-App?	76
7.8	Was ist an mobiler Entwicklung so schwierig?	76
7.9	Was bringt die Zukunft?	78
8	Cross-Platform-App-Development	83
	<i>Marvin Puchmüller</i>	
8.1	Was ist Cross-Platform-App-Development?	83
8.2	Welche Plattformen müssen berücksichtigt werden?	83
8.3	Möglichkeiten Cross-Platform-Apps zu entwickeln	84
8.4	Funktionsweise hybrider Apps	86
8.5	Herausforderungen bei der Entwicklung und beim Testen hybrider Apps	86
8.6	Aufbau einer App – HTML, CSS und JavaScript	90
8.7	Die Zukunft liegt im Browser	95
8.8	Sind Cross-Platform Apps die Zukunft?	99
9	Modernes Web Design	103
	<i>Mathias Knoll, Franz Niederl</i>	
9.1	Einleitung	103
9.1.1	Responsive Web Design (RWD)	106
9.1.2	Adaptives Web Design (AWD)	106
9.1.3	Responsive Design und Server Side Components (RESS)	107
9.1.4	Progressive Verbesserung	107
9.2	Strategien zur Umsetzung	107
9.2.1	Herkömmliche Entwicklung	108
9.2.2	Moderne Entwicklung	109
9.3	Kosten	110
9.3.1	Fortbildung	110
9.3.2	Qualitätssicherung	111
9.4	Faktoren	111
9.5	Werkzeuge	112
9.5.1	Prototypen	112
9.5.2	Bibliotheken	113
9.5.3	Entwicklungsumgebungen	113

9.6	Layout	113
9.6.1	Historie	114
9.6.2	Statisches Layout	114
9.6.3	Fluid Layout	114
9.6.4	Dynamisches (modernes) Layout	114
9.7	Feststellen des Mediums („Media Queries“)	115
9.7.1	Media Features	115
9.7.2	Browserspezifische Features	116
9.8	Typographie	117
9.9	Bilder	118
9.10	Komponenten	119
9.10.1	Navigation	119
9.10.2	Werbung	119
9.10.3	Modale Dialoge	120
9.10.4	Tabellen	120
9.10.5	CMS und Unterstützung	120
9.11	Testen	121
9.12	RWD in der Praxis	123
9.13	Beispielprojekt einer RWD – Adaptierung	126
9.14	Zusammenfassung und Ausblick	127
10	KMU IT-Security-Scan	131
	<i>Klaus Gebeshuber</i>	
10.1	KMU-IT-Landschaft	132
10.2	Bedrohungsszenarien	133
10.3	Systemanforderungen	135
10.4	Systemarchitektur	136
10.5	Funktionsumfang	138
10.6	Bedienung	139
10.7	Ergebnis nach durchgeführten Sicherheitsmaßnahmen	142
10.8	Ergebnisse	143
11	Mobile Usability Testing	149
	<i>Johannes Feiner, Elmar Krajnc</i>	
11.1	Intro	150
11.2	User-Experience und Benutzbarkeit	150
11.3	Beispiele aus dem täglichen Leben	154

11.4 Usability verbessern! Aber wie?	156
11.5 Zukünftige Entwicklungen	158

12 Datenqualität - KMU - Mobile Anwendung: Ein auflösbarer Widerspruch? 161

Wilhelm Zugaj

12.1 Ein offensichtlicher Widerspruch	161
12.2 Datenqualität – eine Definition	162
12.3 Datenqualität auf dem Prüfstand	165
12.4 Lösungsszenarien für Klein- und Mittelbetriebe	167
12.5 Conclusio – ein auflösbarer Widerspruch	170

1 Vorwort

KMU bilden das Rückgrat der österreichischen Wirtschaft – rund 99,6% aller Unternehmen in Österreich sind so genannte Klein- und Mittelunternehmen. Sie beschäftigen rund 1,7 Millionen Österreicher und Österreicherinnen¹ und bewältigen damit den Löwenanteil der österreichischen Privatwirtschaft. Unterstützung bei dieser überaus wichtigen Aufgabe erhalten sie durch das von der österreichischen Forschungs- und Förderungsgesellschaft geförderte Projekt der FH JOANNEUM *KMU-goes-mobile*. Ziel des Projektes ist es, KMU an den Vorteilen einer mobilen IT-Infrastruktur teilhaben zu lassen, damit österreichische KMU auch in Zukunft im internationalen Wettbewerb bestehen können. Denn eines ist klar: Mobile Computing ist allgegenwärtig. PCs werden durch iPhone, Android Smartphones und Tablets in vielen – oft neuen - Bereichen ergänzt oder auch komplett abgelöst. Was für die Gesellschaft schon längst eine Selbstverständlichkeit ist, haben viele KMU aber noch nicht geschafft: das Leben und Arbeiten mit mobilen personalisierten Geräten. Nicht selten sehen Firmen den aktuellen Wettbewerbsnachteil durch fehlende mobile Angebote noch gar nicht, verlieren dadurch den Kontakt zu ihrer mobilen Kundschaft und laufen letztlich Gefahr, den Technologiewandel zu verpassen und massiv an Wettbewerbsfähigkeit zu verlieren.

Damit KMU zukünftig die strategischen sowie wirtschaftlichen Vorteile vom mobilen Applikationen für sich nutzen können, wird im Rahmen des *KMU-goes-mobile* Projektes an der FH JOANNEUM ein **Kompetenzzentrum** für KMU fokussiertes Mobile Computing etabliert. An diesem Kompetenzzentrum feilen die Spezialisten und Spezialistinnen der FH JOANNEUM an neuen Apps, vermitteln **technologisches Know-How** an die Unternehmen und helfen moderne **eBusiness-Modelle** in die Tat umzusetzen. Die FH JOANNEUM ist durch die Studiengänge Advanced Se-

¹Quelle: WKO, Wirtschaftskraft KMU - Vorfahrt für Österreichs KMU, S 5f.

curity Engineering/IT & Mobile Security sowie IT-Recht & Management auf Masterniveau und Internettechnik/Software Design als Bachelorstudiengänge in der einzigartigen Position ihr fundiertes Wissen und ihre umfangreiche Erfahrung für die Sicherheits- und Softwareentwicklungsaspekte von mobilen Anwendungen an die speziellen Bedürfnisse von KMU anzupassen. Die Unternehmen werden hier von der Konzeptphase über die Implementierung bis hin zum Roll-out und Test von mobilen Apps unterstützt.

An der Schnittstelle zwischen Forschung und Praxis liegt der Fokus einerseits auf innovativen und zeitgemäßen Technologieeinsatz, andererseits auf der bestmöglichen Handhabung. Diese Usability wird durch ein eigenes **Usability Lab** mit umfangreicher Auswahl an mobilen Geräten gesichert. Zur Bewusstseins-schaffung für mobile Chancen von KMU wurde ein spezielles „**Mobile KMU Portal**“ eingerichtet. Die Unternehmen profitieren von frei nutzbaren Prototypen und finden hier Infos und Tools zu mobilen Themen (Design-Guides, Leitfäden, Checklisten, Best-Practice). Um Integration in bestehende Firmeninfrastruktur durch Cloud- und Webservices zu gewährleisten, wird **Beratung** in Rahmen von **Workshops** und **Konzepterstellung** bei Konfiguration und Erweiterung der Serversysteme angeboten. **Security Checks** in vielfacher Form geben den Firmen Vertrauen in die Applikationen, die auf unternehmenskritische Daten zugreifen. Mit Hilfe von Apps vergrößern die Firmen ihre Kundenbasis und binden diese durch personalisierte Applikationen auch langfristig näher an ihr Unternehmen.

Der vorliegende Sammelband gibt eine Auswahl von Themengebieten wieder, mit denen sich die Experten und Expertinnen der FH JOANNEUM im Zuge des Projektes *KMU-goes-mobile* befasst haben. Von der Cloud über Social Media Marketing bis hin zu Sicherheit und Web-Design finden Sie darin alles, was mobile KMU in Zukunft bewegt.

FH-Prof. Mag. Dr. Sonja Gögele, MBA MSc

Institutsleiterin
Internet- Technologien & -Anwendungen
Department für Angewandte Informatik

2 Einleitung

Michael Ulm, Johannes Feiner

KMU goes mobile – Erfolgreich die Welt der mobilen Geräte nutzen

In der Informationstechnologie zog in den letzten Jahren ein Trend der mobilen Endgeräte¹ ein, der nun bereits Stand der Technik wurde bzw. sich weiterhin durch fortlaufende Neuerungen und Weiterentwicklungen verändert. Unternehmen müssen sich der Herausforderung einer neuen mobilen Welt stellen, wo sich deren Kunden weiterhin zwischen zwei Welten befinden. Einerseits bewährte Systeme, die quasi traditionell auf PC oder Notebook verwendet werden, und andererseits eben mobile Entwicklungen, wie zum Beispiel mobile Websites oder Apps auf dem Smartphone.

Unternehmen müssen daher für sich entscheiden, ob sie mit diesem Trend mithalten möchten oder, wenn nicht, potentielle Kunden möglicherweise nicht mehr erreichen können. Warum ist diese mobile Innovation überhaupt notwendig wenn „*eh alles funktioniert*“? Eben darum! Fortschritt bringt neue Ideen und neue Ideen bzw. deren Umsetzungen bringen Fortschritt. Dieser Prozess geht Hand in Hand und ist für Unternehmen von entscheidender Bedeutung. Kleine und mittlere Unternehmen sollten daher den Schritt wagen, diese neuen Technologien der mobilen Welt einzusetzen, die für auch weniger IT affine Benutzer bereits gang und gäbe sind. Dieser scheinbar kleine Schritt ist doch ein wichtiger großer

¹Worldwide Smartphone Sales to End Users by Vendor in 2Q13: 225 Mio (Gartner 2013 <http://www.gartner.com/newsroom/id/2573415>)

Schritt in Richtung Zukunft und Konkurrenzfähigkeit im heutigen Konsumentenverhalten.

Es gibt viele Wege, die ein Unternehmen gehen kann um sich auch mobil auf jedem Endgerät dem Endkunden zu zeigen. Ein Responsive Web Design, sprich eine Website die sich je Endgerät optimal auf die Displaygröße anpasst, wäre eine Möglichkeit sich auf schnellem Wege dem Kunden im Internet auch auf dem Smartphone zu präsentieren. Bereits die Information über die Öffnungszeiten kann einem Kunden die erste Entscheidung erleichtern, ob überhaupt der Weg zum Geschäft angetreten werden soll, oder ob doch bequem am Sofa das Geschenk für die Oma bestellt werden soll.

Shoppen im Internet ist ein gutes Stichwort, das die nächste Möglichkeit zur mobilen Präsentation anhand eines kurzen Beispiels näherbringen kann. Viele Unternehmen besitzen bereits Webshops, wo es dem Kunden bequem von zuhause aus möglich ist Produkte zu bestellen. Doch nur wenige besitzen aktuell mobile, geschweige denn responsive Webshops, welche für Handys geeignet sind. Haben Sie bereits mobil eingekauft oder nutzen Sie Apps am Smartphone zur Bestellung einer Pizza? Viele Unternehmen stehen vor der Entscheidung „machen wir einen Webshop als Website oder lassen wir eine App programmieren?“ Die Entscheidung fällt oft zugunsten einer Website bzw. eines Webshops, da der Aufwand einer nativen Entwicklung für viele verschiedene Smartphone Plattformen, wie z.B. Android, iOS und Windows Phone, sich nicht lohnen würde.

Cross-Plattform-Development, sprich die Entwicklung einer App für alle Plattformen, bringt eine Lösung für dieses Problem. Es ist keine einfache Aufgabe alle Kunden über eine Plattform zu erreichen, doch über die drei am weitest verbreitetsten Plattformen können bereits über 80%² der Kunden auch über eine App am Smartphone erreicht werden. Apps bieten mehr Möglichkeiten als eine responsive Website. Es dürfen Dienste

² Laut *IDC Global Marketshare* von dritten Quartal 2014, siehe <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>, haben bereits Android mit 84,4% und iOS mit 11.7% in Summe bereits über 96% Marktanteil nach rund 330 Millionen verkauften Geräten in diesem Zeitraum.

im Hintergrund weiterlaufen, aber auch die GPS-, Gyro-, Helligkeit-, Abstandssensoren von Handys können angesprochen werden. So könnte zum Beispiel eine native App Ihres Pizzazustellers Sie - ähnlich einer SMS - mittels „Notification“ informieren, dass die Pizza in 10 Minuten angeliefert wird, auch wenn Sie sich aktuell nicht in der App befinden.

So sehr sich ein Unternehmen eine enge Bindung auch über das Internet zu den Kunden wünscht, so treten auch dementsprechend Gefahren aus dem Internet auf. Je mehr Systeme, wie Webshop, Mail, Website, mobile Apps ein Unternehmen anbietet, desto mehr Möglichkeiten tun sich für potentielle Angreifer aus dem Internet auf. Diese Angreifer nutzen Lücken der Systeme aus, um auf Firmendaten zuzugreifen. Gegenmaßnahmen, achtsames Handeln und Sicherungen der Unternehmenssysteme müssen beachtet und umgesetzt werden.

Durch diese wenigen Beispiele ist es bereits möglich zu erkennen, was einem Unternehmen abverlangt wird, um mit der IT „up-to-date“ zu bleiben und im Sinne der Kundenbindung ein zeitgerechtes Web-Auftreten zu bieten, sowie moderne Zusammenarbeiten zu ermöglichen. Viele Unternehmen besitzen jedoch nicht die Zeit und die finanziellen Ressourcen sich mit diesen Themen auseinander zu setzen oder neue Technologien in ihrem Unternehmen zu etablieren. An dieser Stelle kommt das Forschungsprojekt KMU goes mobile ins Spiel. Es ist ein von öffentlicher Hand³ gefördertes Projekt in der Förderschiene COIN, COoperation und INnovation⁴ welche generell die Innovationsleistung in Österreich zu verbessern versucht. Konkret bietet das Projekt KMU goes mobile verschiedene Ansätze, um Klein- und Mittelbetriebe in der veränderten, mobilen Welt mit Know-How für Innovationen in Bereichen wie zum Beispiel Private Cloud Infrastrukturen, Social Media Marketing, mobile Workflows und Appentwicklung auszustatten und damit einen Marktvorteil zu verschaffen. Die Eckdaten für das KMU goes mobile Projekt sind ein Umsetzungszeitraum von 2013 bis 2017, wobei in zwei Phasen 18 Arbeitspakete mit einem Gesamtvolumen von 620t € und rund 9470 h geplant sind.

Um einen Überblick zu bekommen, welche unterschiedlichen Ergeb-

³Österreichische Forschungsförderungsgesellschaft FFG <https://www.ffg.at/>

⁴Siehe <https://www.ffg.at/coin-cooperation-innovation/>

nisse im Rahmen von KMU goes mobile erarbeitet werden, folgen nun exemplarische Auszüge aus den Teilbereichen des Projektes. Zuerst wurden und werden die Wünsche und Bedürfnisse der KMUs betrachtet (a), dann folgen Möglichkeiten des Wissenstransfers auf persönlicher Ebene in Workshops (b), sowie online über digitale Medien (c). Weiters wird Know-How online frei such- und abrufbar über ein Portal bereitgestellt (d). Besonders für Themen wie für mobile und sichere Softwareentwicklung (e) wird dieses speziell und zielgruppenorientiert aufbereitet. Implementierte Software wird bereits von der Prototypen-Phase (f) angefangen Richtung Usability und User Experience (g) getestet. Das Querschnittsthema IT - Security (h) reicht von Aspekten der Softwarequalität bis hin zu Themen wie Hacking. Begleitende Evaluierungen (i) runden das Projekt ab und liefern zeitgerecht Feedback für Reflexion und Optimierung.

(a) Was brauchen KMUs? Anhand einer Umfrage durch lange Einzelinterviews wurden die *Bedarfe der Firmen* erhoben. Überraschenderweise meinten viele Firmen zuerst aktuell überhaupt keinen speziellen Bedarf an mobilen Geräten, Software oder an Infrastruktur zu haben.

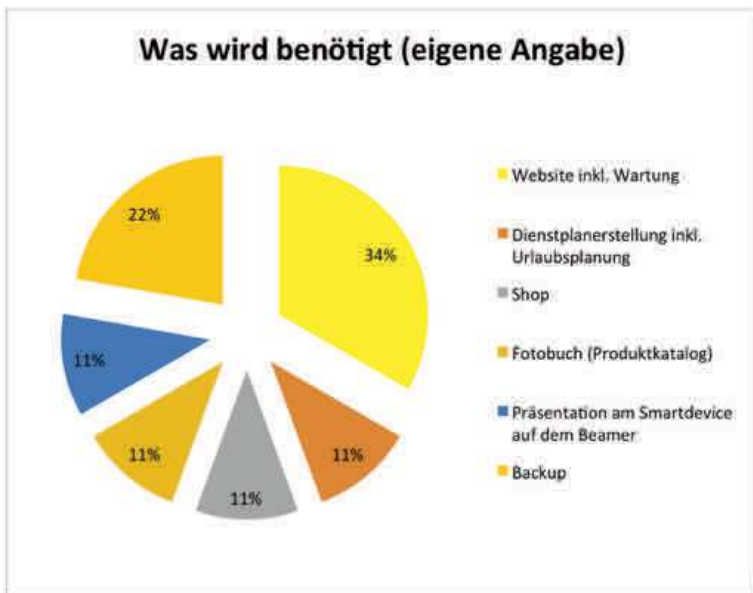


Abbildung 2.1: Umfrage in KMUs

Erst auf Nachfragen hin, wurden einzelne mögliche Aspekte wie zum

Beispiel der Einsatz von einzelnen Handy Apps benannt. Tendenziell wurden dabei jene mobilen Apps aufgezählt, welche bereits in den Stores existieren und somit auch den Firmen bekannt sind. Völlig neue Bedarfe oder auch kreative, innovative Ideen für zukünftige mobile Workflows oder ähnliche Anwendungsfälle kamen auf diese Weise leider nicht zu Tage. Die Ideen der Firmen selbst beschränkten sich wie in Abb 2.1 ersichtlich, also auf Website und Terminverwaltung, Shop oder Fotobuch. Anderes, wie Unterstützung bei Social Media Marketing, oder der Wunsch nach Entwicklung von neuen Handy Apps, sowie Security Aspekte von Datenschutz bis Firewalls oder auch diesbezügliche Beratungsleistungen wurden beinahe nie genannt.

(b) Große Wertigkeit hat im Projekt KMU goes mobile auch der *Wissenstransfer* hinein in die Unternehmen. Durch mehrere Workshops wurden bereits verschiedenste Thematiken behandelt. Zum Beispiel wurden die Security Aspekte (siehe auch Kapitel 10) mit dem Fokus auf Kleinbetriebe unter der Leitung von Security Spezialist Klaus Gebeshuber analysiert und diskutiert.



Abbildung 2.2: Workshops

Erste Bewusstseinsbildung (Awareness) Richtung Security wurde mit einem speziell entwickeltem One-Click IT Security Check ermöglicht. Weitere Themen für KMUs waren Social-Media-Marketing, aber auch rechtliche Problematiken wie Bring-Your-Own-Device (siehe Kapitel 6). Im Bereich der Applikationsentwicklung findet der Wissenstransfer

teilweise vor Ort in den Firmen statt und User-Centred-Design, User-Experience Planung, beziehungsweise Mobile-Usability Evaluierungstechniken wurden und werden vermittelt. Mehr über die nötige Expertise in App-Development (Eco-App Portierung, Land Steiermark Entwicklung) findet man in den Kapiteln 7 und 8. Zu guter Letzt ist Präsenz in den relevanten Communities auch von vitalem Interesse um sich mit den ExpertInnen des jeweiligen Feldes zu vernetzen. Als Beispiel seien hier Usability Konferenzen wie UX-Day genannt, welche von der FH JOANNEUM mitorganisiert werden und Forschungsergebnisse auch von KMU goes mobile präsentiert und diskutiert werden.

(c) Den Firmen – und speziell den EntwicklerInnen in den Firmen – werden weiters auch verschiedenste *Anleitungen, Checklists und Guidelines* bereitgestellt. So helfen zum Beispiel Best Practice Richtlinien



Abbildung 2.3: Guidelines

von klassischem Webaustritt zu „Responsive Web Design“ (RWD) bei der produktiven und effizienten Umsetzung einer redesigneten Homepage. Bereitgestellte Dokumente können vom Portal downgeloaded und frei verwendet werden. Auch können bei Bedarf die ExpertInnen im Haus für Beratungsdienstleistungen kontaktiert werden.

(d) Die Datenhaltung in der Cloud und/oder der Datenaustausch über verschiedenste Clouddienste sind privat äußerst beliebt und werden nun auch in den Firmen mehr und mehr gefordert. Der Bedarf wurde hier zusätzlich verstärkt durch die Bewusstseinsbildung nach der Veröffentlichung von Dokumenten durch Edward Snowden über die flächendeckende Massenüberwachung durch die NSA und andere Geheimdienste. Im Wissen um diese Problematik bei der Speicherung auf



Abbildung 2.4: Cloud

Fremdsystemen, sind *Private Cloud Alternativen* gefragt. Es wurde exemplarisch die Software ownCloud (siehe auch Private Cloud Workshop Manfred Pamsl) ausgewählt um am Beispiel eines der vielen Private Cloud Angebote den Einsatz für Firmen darzustellen. So kann entweder direkt in Zusammenarbeit mit den Betrieben eine vernünftige Inhouse Lösung, oder eine Umsetzung bei Providern gefunden und implementiert werden. Wie schon unter Punkt (c) behandelt, wurden auch hier zusätzliche Dokumente, Anleitungen, Tutorials wie zum Beispiel die Konfiguration von Server und der Clients entwickelt und sind online bereitgestellt.

(e) In Bezug auf Softwareentwicklung, im speziellen User-Centred-Design und Development werden in verschiedensten Technologien Softwarepakete konzipiert und implementiert, sowie unter freier Lizenz bereitgestellt. Ein Beispiel ist die *Cross-Platform Entwicklung* für Handy Applikationen, wo Know How durch das Projekt KMU goes mobile aufgebaut und weitergegeben werden kann. So wurden hier bereits Nachfolgeaufträge für das Land Steiermark (*Zwei und Mehr App*) oder für die Firma Adenso (*EcoApp*) aquiriert.

(f) Der Aspekt des benutzer-bezogenen Entwickelns ist ein Anliegen gerade im mobilen Bereich. Gerade auf den Smartphones wird von Applikationen erwartet, dass diese den speziellen Bedingungen und Anforderungen entsprechen. So sollte zum Beispiel keine Anleitungen nötig sein



Abbildung 2.5: Paper Prototyping

und intuitive Bedienung mit Multitouchgesten und Sensorunterstützung wird erwartet. Mit Hilfe von *User Centred Design* UCD und intensivem *Paper Prototyping* wird dieses bereits in der Konzeptphase beachtet und an Endanwendern getestet. Es lernen SoftwareentwicklerInnen (zum Beispiel jene der Firma Era für deren iPad Softwareprodukte) in Workshops diese Techniken anzuwenden und für ihr Unternehmen erfolgreich umzusetzen.

(g) *User Experience* wird jene Erfahrung bezeichnet, welche die Endbenutzer mit den Produkten erleben, erfüllen und spüren. Im Bereich der mobilen Applikationen soll dieses – hoffentlich spannende oder zumindest effiziente – Erlebnis während der Nutzung von Apps optimiert werden. Durch Usability-Tests mit Endusern wird die Software evaluiert, um sicherzustellen, dass die mobile Benutzung (Stichwort Accessibility) auch wirklich problemlos vonstatten geht. Diese Tests können nicht am Desktop simuliert werden, sondern sie finden im Gegensatz zu klassischen Usability Testing in real life Situationen unterwegs mit dem Handy statt. Das Human Centred Design und Usability Know-How wird, wie bereits oben erwähnt, bei Konferenzen wie zum Beispiel beim UXDay in Graz auch in Kooperationen mit weiteren tertiären Bildungseinrichtungen (TU Graz, Keith Andrews) in der Usability Community geteilt und reflektiert.

(h) Im Bereich der Security und speziell der *IT Security* sind vielerlei Aktivitäten nötig. Einerseits ist das Bewusstsein für Datenschutz leider generell noch nicht stark ausgeprägt und Firmen reagieren oft erst nach Schadensfällen auf existierende Probleme. Mit einem eigens entwickel-



Abbildung 2.6: Mobile-Usability

tem One-Click IT Security Scan von Klaus Gebeshuber (siehe Kapitel 10) können nun Firmen relativ einfach eine Erstanalyse ihres Systems durch einen simulierten Angriff von außen checken. Daraufhin können gezielt



Abbildung 2.7: One Click IT Security Scan

Abwehrmaßnahmen entwickelt und umgesetzt werden. In weiterer Folge bedeutet Security natürlich noch einiges mehr: Von Backups bis hin zu Privacy-Aspekten. Im Rahmen von Softwareentwicklung wird Security auch zu einem großen Teil durch Software Qualität unterstützt.

(i) Um die einzelnen Ergebnisse (Erfolge, aber natürlich gegebenenfalls auch Misserfolge) zu bewerten und daraus für Folgeaktivitäten

zu lernen, sind begleitende Evaluierungsmaßnahmen vorgesehen. Diese ermöglichen Rückschluss und Feedback für die Beteiligten, um im Sinne der ständigen Optimierung die konstruktive Weiterentwicklung zu gewährleisten.

Ein spezieller Aspekt im Zusammenhang mit Förderprojekte im tertiären Bildungsbereich, hier konkret der Fachhochschule FH JOANNEUM, ist der *Transfer der Ergebnisse auf die Lehre*. So werden im Sinne der Nachhaltigkeit die Bachelorstudierenden von Internettechnik (ITM), Softwaredesign (SWD) sowie die Masterstudierenden von IT und Mobile Security (IMS) und IT Recht und Management (IRM) quasi zu Multiplikatoren, welche ihr Wissen bereits im Praxissemester oder später als MitarbeiterInnen in die Firmen tragen.

3 Die Private Cloud Alternative - Daten und Services in der eigenen Hand

Manfred Pamsl

3.1 Einführung

„The best thing about cloud computing is that word: cloud. Telling consumers their data is in the cloud is like telling a kid his dog has gone to doggie heaven. There is no doggie heaven, and your data isn't in a cloud. It's in a windowless, fortress-like data center somewhere in the rural U.S.“

Mit diesen Einführungsworten kommentierte Lev Grossman¹ in einem durchaus auch kritischen Artikel am Portal „time.com“ die öffentliche Ankündigung des „iCloud“ Dienstes durch die Firma „Apple Inc.“ am 6.6.2011. Dieses Datum kann sicher nicht als die Geburtsstunde des „Cloud Computing“ bezeichnet werden, aber spätestens seit diesem Tag war dieser neue Trend in der IT-Branche unübersehbar.

Historisch betrachtet gehen die Ideen für das heutige Cloud Computing in die 1960er Jahre zurück. Schon damals gab es die Ideen, IT-Ressourcen einer breiten Masse von Benutzern gegen eine Gebühr zur Verfügung zu stellen². Allerdings fehlten zu dieser Zeit noch die technischen Voraussetzungen, die dann aber mit der Entstehung, Verbreitung und Weiterentwicklung des Internets sukzessive geschaffen wurden.

¹Siehe <http://content.time.com/time/magazine/article/0,9171,2076711,00.html>

²Artur P. Schmidt, 2014, <http://www.heise.de/tp/artikel/42/42261/1.html>

Im Jahr 1999 begann das neu gegründete Unternehmen „Salesforce.com“ Enterprise-Anwendungen als Web-Applikationen über das Internet zugänglich zu machen. 2002 stellte der Online Händler Amazon.com seinen Benutzern über die „Amazon Web Services“ die Möglichkeit zur Datenspeicherung zur Verfügung, im Jahre 2006 wurde dann das Angebot um die Elastic-Compute-Cloud (EC2) und den Simple Storage Service (S3) erweitert³. Im selben Jahr begann Google seinen Benutzern die Dienste für die Online Erstellung von Office Dokumenten zur Verfügung zu stellen und hat seither sein Cloud Angebot um eine Vielzahl von weiteren Diensten ergänzt⁴. Andere Firmen wie beispielsweise Rackspace, Microsoft, Apple und IBM folgten in den darauf folgenden Jahren mit ihren eigenen Angeboten von Cloud Diensten.

Unter dem Begriff „Cloud Computing“ wurde also im Laufe der Jahre eine Vielzahl von unterschiedlichen Dienstleistungen angeboten, von der Speicherung von Dateien über virtuelle Maschinen bis zu vollständigen Enterprise Applikationen. Was ist allen diesen Diensten gemeinsam, was macht eigentlich einen Dienst zu einem „Cloud Service“?

3.2 Eigenschaften von Cloud Computing

Cloud Computing stellt keine grundsätzlich neue Technologie dar, vielmehr wurden nur die vorhandenen Technologien zur Virtualisierung von Computer Ressourcen konsequent weiterentwickelt und über neue Bereitstellungsmodelle zugänglich gemacht. Als „Computer Ressourcen“ werden dabei nicht nur „klassische“ Betriebsmittel wie Prozessoren, Haupt- und Festplattenspeicher, sowie Netzwerk Bandbreite gesehen, sondern auch „höhere“ Abstraktionseinheiten wie etwa Datenbank-Kapazitäten oder die Bereitstellung ganzer Anwendungssysteme. Die beteiligten Partner in diesem Ökosystem sind der Konsument der Ressourcen (Kunde) sowie der Anbieter der Ressourcen (Dienstleister, Provider).

Das National Institute of Standards und Technologie (NIST) hat im September 2011 in der „Special Publication 800-145“ die wesentlichen

³Salesforce, 2014, <http://www.salesforce.com/de/socialsuccess/cloud-computing/die-geschichte-von-cloud-computing.jsp>

⁴Sourya Biswas, 2010, <http://cloudtweaks.com/2010/12/cloud-computing-and-google-docs/>

Eigenschaften, Service- und Bereitstellungsmodelle von Cloud Computing definiert. In diesem Dokument wird Cloud Computing als ein Modell für den allgegenwärtigen, bequemen sowie bedarfsorientierten Zugang zu einem gemeinsam genutzten Pool an konfigurierbaren Computer Ressourcen bezeichnet.

Das Modell weist folgende essentielle Charakteristika auf:

- „On-demand self service“: Der Konsument erhält Zugriff auf Computer Ressourcen, wie beispielsweise Rechenzeit oder Speicher, ohne einen manuellen, menschlichen Eingriff seitens des Dienstleisters. Die Zuordnung der Ressourcen kann vom Konsumenten typischerweise über ein entsprechendes elektronisches Interface selbst durchgeführt werden
- „Broad network access“: Die Ressourcen sind über ein privates Netzwerk oder das Internet mittels Standard-Mechanismen von unterschiedlichen Plattformen (Mobiltelefone, Tablets, Laptops, Workstations, ...) aus nutzbar.
- „Resource Pooling“: Die Ressourcen des Dienstleisters werden in Pools zusammengefasst und unterschiedlichen Nutzern nach Bedarf zugeordnet. Die einzelnen Kunden haben dabei keinerlei Kontrolle oder Wissen darüber, wo sich die ihnen zugeordneten Ressourcen tatsächlich im physikalischen Sinne befinden.
- „Rapid elasticity“: Die Vergrößerung oder Reduzierung der zugeordneten Mittel erfolgt dynamisch und in manchen Fällen auch automatisch. Für den einzelnen Kunden entsteht der Eindruck, dass die Ressourcen bei Bedarf praktisch in einem unlimitierten Ausmaß zur Verfügung stehen.
- „Measured service“: Die Zuordnung der Betriebsmittel wird gemessen, kontrolliert und aufgezeichnet, sodass sowohl für den Bereitsteller als auch den Kunden die notwendige Transparenz über die tatsächliche Nutzung der Ressourcen zur Verfügung steht.

3.3 Servicemodelle

Je nach der Art der zur Verfügung gestellten Ressourcen werden beim Cloud Computing folgende Servicemodelle unterschieden:

- „Software as a Service (SaaS)“: Dem Kunden wird die Möglichkeit zur Verfügung gestellt, eine fertige Anwendung auf der Cloud-Plattform des Providers zu nutzen. Der Verwaltung der Hardware, des Betriebssystems, der Anwendungssoftware und des Netzwerkes liegt außerhalb des Verantwortungsbereiches und der Kontrolle des Kunden und wird vom Dienstleister autonom durchgeführt.
- „Platform as a Service (PaaS)“: Der Kunde erhält die Möglichkeit, eigene Anwendungen auf der Plattform des Providers einzusetzen, die der Spezifikation des Anbieters in Hinblick auf die Programmiersprache, Bibliotheken, zur Verfügung stehenden Diensten und Werkzeugen entsprechen. Der Kunde ist also noch selbst für die Verwaltung und Konfiguration der Anwendung zuständig, während Server Hardware, Betriebssystem und Netzwerk aber außerhalb seines Zuständigkeitsbereiches liegen.
- „Infrastructure as a Service (IaaS)“: Bei IaaS werden grundlegende Infrastruktur-Ressourcen wie Rechenleistung, Speicherplatz oder Netzwerk-Bandbreite zur Verfügung gestellt auf der der Kunde seine Software eventuell inklusive des Betriebssystems installiert.

Zusätzlich zu diesen drei ursprünglich definierten Servicemodellen finden sich in manchen Beschreibungen noch weitere Modelle, die aber in der Regel nur spezialisierte Fälle der drei Basismodelle darstellen: Network as a Service, Firewall as a Service, Business Process as a Service, Unified Communication as a Service, ...

3.4 Bereitstellungsmodelle

Abhängig davon, in welcher „Beziehung“ die Konsumenten und Provider der Ressourcen zueinander stehen, unterscheidet man zwischen folgenden vier Bereitstellungsmodellen:

- „Privat Cloud“: Die Cloud Infrastruktur wird nur von einer einzigen Organisation genutzt. Sie kann durch die Organisation selbst oder

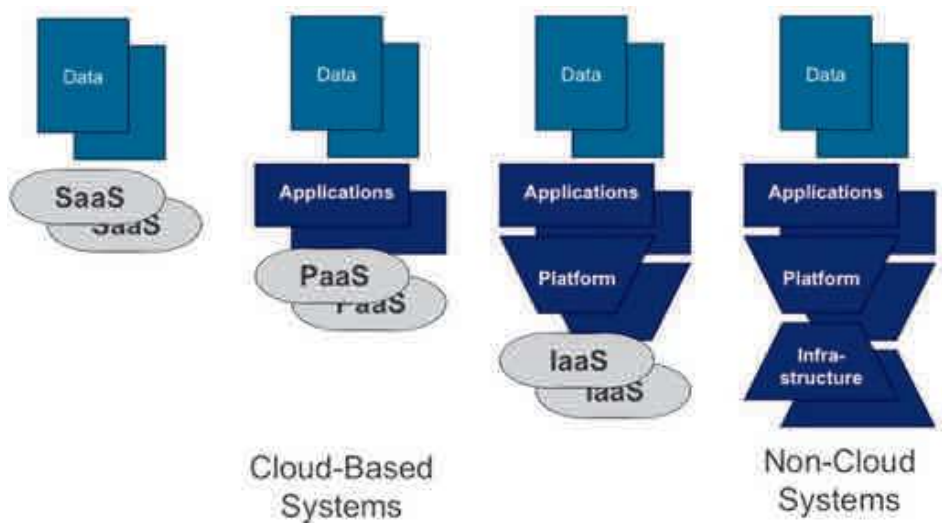


Abbildung 3.1: Opengroup⁵

von Dritten zur Verfügung gestellt und verwaltet werden. Der Aufstellungsort der Infrastruktur kann ebenfalls innerhalb oder außerhalb der Organisation liegen.

- „Community Cloud“: Die Infrastruktur wird für eine dedizierte Gemeinschaft von Organisationen zur Verfügung gestellt, die spezielle Bedürfnisse oder Anforderungen teilen, zum Beispiel in Hinblick auf Sicherheit oder Compliance. Sie wird von einem Partner, mehreren Partnern gemeinsam oder einem Dritten zur Verfügung gestellt oder betrieben und kann innerhalb oder außerhalb der Organisationen liegen.
- „Public Cloud“: Die Cloud Infrastruktur wird der breiten Öffentlichkeit zur Verfügung gestellt und von einer privaten, akademischen oder öffentlichen Organisation betrieben und zur Verfügung gestellt.
- „Hybrid Cloud“: Die Cloud Infrastruktur stellt eine Kombination von zwei oder mehr der drei vorigen Bereitstellungsmodelle dar. Jeder Infrastrukturteil bleibt für sich gesehen eigenständig, die Teile werden aber durch eine übergeordnete Technologie miteinander

⁵Opengroup.com/G135.pdf/18.07.2013
<https://www2.opengroup.org/ogsys/catalog/g135>

verbunden. Ein Beispiel wäre etwa eine private Cloud Infrastruktur, in die zur Lastabdeckung in Spitzenzeiten über einen Lastverteilungsmechanismus auch die Ressourcen eines Public-Cloud-Providers eingebunden werden.

3.5 Potentielle Risiken einer Public Cloud Lösung

Speziell die Nutzung von Public-Cloud-Diensten stellt eigentlich eine spezielle Form der Anwendung eines in der IT Branche altbekannten Prinzips dar, nämlich des Outsourcings – mit allen bekannten Vor- und Nachteilen.

Benötigte Dienste, die aber meistens nicht zu den Kernaufgaben und -kompetenzen des jeweiligen Unternehmens gehören, werden nicht selbst erbracht, sondern einem Leistungspartner übergeben, der diese Aufgaben „besser“, daher mit einer signifikant höheren Qualität und Stabilität, erbringen kann. Da der Dienstleister in der Regel auf die Erbringung der angebotenen Leistungen spezialisiert ist, kann durch die höhere Effizienz bei der Bereitstellung der Leistung je nach Anwendungsfall auch noch ein Kostenvorteil entstehen.

Auf der anderen Seite begibt sich das Unternehmen in eine möglicherweise vitale Abhängigkeit von einem Dienstleister, bei der trotz aller abgeschlossenen Service-Level-Agreements (SLAs) und Verträge im Problemfall keine direkte Einflussnahme auf die Maßnahmen zur Problembehebung möglich ist, und im Extremfall eventuell nur mehr eine gerichtliche Aufarbeitung des entstanden Schadens bleibt.

Bei den Public Cloud Providern kommt noch hinzu, dass diese auf den Massenmarkt ausgerichtet sind und in der Regel global agieren. Durch den Massenmarkt sind individuelle Abkommen oder Anpassungen an die Wünsche des Konsumenten nur in einem sehr beschränkten Ausmaß machbar und konkrete Ansprechpartner beim Provider oft nur schwer zu erreichen. Die globale Ausrichtung der Provider hingegen führt dazu, dass die Daten und Dienste an einem vom Kunden nicht beeinflussbaren Ort außerhalb des Landes lagern können – eine Situation die zu regulatorischen und spätestens seit der „NSA-Affäre“ zu vertrauens-technischen Problemen führen kann.

In einem Artikel in der Zeit Online vom 18. Februar 2011 wird die Situation des Internet-Startups „Radio.de“ beschrieben, das 48 Stunden ohne Zugriff auf die Daten und Büro Software auskommen musste. Der Grund: Durch einen Fehler im Bezahlssystem von Google konnte ein Rechnungsbetrag von wenigen hundert Euro nicht abgebucht werden. Als Konsequenz wurde Radio.de der Zugang zu ihren Daten von Google kurzerhand gesperrt⁶. Im privaten Bereich bearbeiten viele ihre Fotos, E-Mails und andere Dokumente nicht mehr lokal sondern auf der Cloud Plattform eines großen Providers. Manche dieser Dienste sind sogar gratis nutzbar, die dafür zu akzeptierenden Geschäftsbedingungen sind aber umfangreich und gelegentlichen Änderungen unterworfen. Nur wenige Nutzer werden den Überblick darüber behalten können, welche Rechte er an seinen Daten dem Cloud Anbieter überlässt.

Kommt also eine Public-Cloud-Lösung nicht infrage, kann immer noch eine Private-Cloud-Infrastruktur unter eigener Kontrolle in Erwägung gezogen werden. Selbst wenn die Infrastruktur von der eigenen Organisation zur Verfügung gestellt und betrieben wird, können sich jenseits des potentiell fehlenden Outsourcings Vorteile noch immer genügend positive Effekte durch die eingesetzten Technologien ergeben: Nahezu unbeschränkter Netzwerk Zugriff auf die Applikationen und Daten sowie die leichte Skalierung und Verwaltung der angebotenen Ressourcen.

3.6 Private Cloud Alternative ownCloud

Mit der Software „ownCloud“ kann eine private Cloud Infrastruktur zur Speicherung, Synchronisierung und Verteilung von Dateien realisiert werden. Das Projekt wurde 2010 vom damaligen KDE Entwickler Frank Karlitschek ins Leben gerufen, um eine Alternative zum kommerziellen Public-Cloud-Dienst „Dropbox“ bieten zu können, da dieser Dienst seiner Meinung nach eine Bedrohung für die Privatsphäre seiner Kunden darstellte. Im Jahre 2011 gründete er zusammen mit zwei Partnern die Firma „ownCloud“ und ist seither für alle Aspekte der Produktentwicklung der Software verantwortlich. Das Projekt ist derzeit so erfolgreich, dass es im Dezember 2014 von den Lesern des „Linux Journals“ bei der Wahl der

⁶Dirk Asendorpf, 2011, <http://www.zeit.de/2011/08/Cloud-Computing>

besten neuen Linux und Open-Source Produkte und Projekte an der vierter Position gereiht wurde⁷.

Die Hauptfunktionalität der Software liegt in der Synchronisierung und Verteilung von Dateien über das Netzwerk. Ein direkter Zugriff auf die in der Infrastruktur gespeicherten Daten ist über einen normalen Web-Browser möglich. Um die Dateien auch über Betriebssystemgrenzen hinweg synchron halten zu können existieren aber auch eigene native Synchronisierungs-Clients für die gängigen Desktop-Systeme und mobilen Endgeräte Windows, MacOS, Linux, Android und iOS.

Die Software wird beständig weiterentwickelt und um neue Features erweitert. Die Synchronisierung von Kalender- und Kontaktdaten ist über die Standardprotokolle CalDAV und CardDAV möglich, OpenDocument Dokumente können online über das Web Portal im Stile von „Google Docs“ erstellt und verändert werden.

Optional ist es auch möglich, die Dateien im Filesystem auf der Infrastruktur verschlüsselt abzulegen. Der für die Verschlüsselung und Entschlüsselung benötigte Key kann dabei nur über das Passwort des Besitzers der jeweiligen Datei erlangt werden, wodurch die Vertraulichkeit der Daten auch bei unautorisierten Zugriffen auf die Dateien gewährleistet bleibt.

Die Plattform ist zur Authentifizierung auch an einen LDAP basierenden Verzeichnisdienst wie etwa Microsoft Active Directory anbindbar. Damit kann in einem Firmennetzwerk eine mehrfache Account-Verwaltung vermieden und ein Single-Sign-On mit anderen Diensten realisiert werden. Ein weiterer Bestandteil für die Enterprise-Integration ist die Einbindung anderer Storage-Backends, womit beispielsweise einem Benutzer die Daten auf seinem Home-Share eines Windows-Fileservers des Unternehmens über den Cloud Dienst zugänglich gemacht werden können.

Die Software stellt außerdem ein Framework zur Entwicklung und Einbindung eigener „Apps“ zur Verfügung um die zentrale Applikation an die eigenen Bedürfnisse anpassen zu können. Die Liste der offiziell registrierten Apps ist unter <https://apps.owncloud.com> zugänglich. Die

⁷Readers' Choice Awards, 2014, <http://www.linuxjournal.com/rc2014?page=21>

Serverkomponente von ownCloud ist eine PHP Applikation und läuft daher praktisch auf allen Web-Servern mit PHP Unterstützung. Eine Installation und Standard-Konfiguration auf einem Linux Apache2 System sollte auch von einem nur mäßig erfahrenen Systemadministrator innerhalb von wenigen Minuten durchführbar sein. Die Einstiegshürde für die Nutzung des Systems ist also abseits der üblichen Infrastrukturprobleme wie Netzwerkverknüpfung und Datensicherung nicht sehr hoch.

Die Software steht als Community-Version oder als Enterprise-Version zur Verfügung. Die Liste der Funktionsunterschiede zwischen den beiden Versionen ist überschaubar (siehe <https://owncloud.com/community-enterprise/>), die kostenpflichtige-Enterprise Version wird aber von der Firma „ownCloud“ offiziell unterstützt und gewartet.

Die Einsatzgebiete für diese Private-Cloud-Lösung sind aufgrund der implementierten Features im privaten wie im Unternehmensbereich vielfältig, beispielsweise Sicherung von Adressbüchern, Terminen und Fotos von Mobilgeräten, Synchronisierung von Terminen zwischen verschiedenen Endgeräten und/oder Personen, oder gemeinsame Bearbeitung und Verteilung von Dokumenten.

Durch die Implementierung als privater Cloud-Dienst können alle diese Funktionalitäten genutzt werden, ohne dass sensible Daten wie etwa Adressbücher und private Fotos einem fremden Unternehmen zur vertrauensvollen Verwaltung übergeben werden müssen.

3.7 Private Cloud Alternative OpenStack

Das erst vier Jahre alte Projekt „OpenStack“ ist der derzeitige Rockstar der freien Cloud Projekte. In einer von der „Linux Foundation“ durchgeführten Online-Befragung wurde das Projekt 2014 von den Teilnehmern zum besten und populärsten Open Source Cloud Projekt gewählt⁸.

Das Projekt ging im Jahre 2010 aus einer Kooperation zwischen der Firma „Rackspace Hosting“ und der NASA hervor, wird aber jetzt von

⁸Alexander Williams, 2014, <https://www.linux.com/news/enterprise/cloud-computing/784573-the-top-open-source-cloud-projects-of-2014>

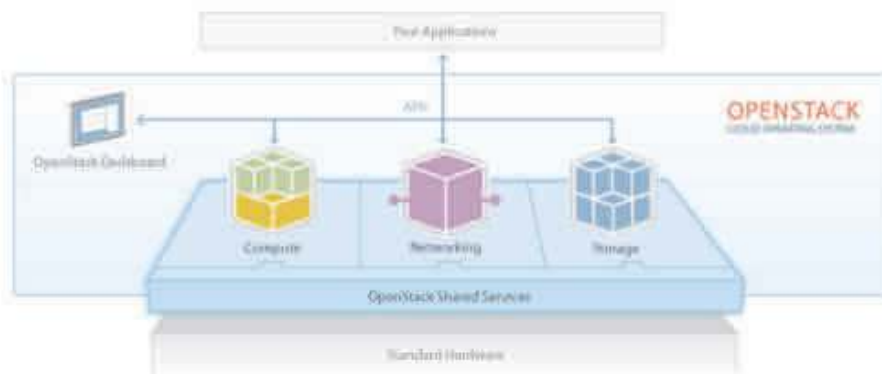


Abbildung 3.2: Openstack-Software-Diagramm⁹

der „OpenStack Foundation“ verwaltet und von vielen weiteren namhaften Firmen wie beispielsweise AT&T, Hewlett Packard, IBM, Intel und Redhat unterstützt und eingesetzt.

Im Gegensatz zu OwnCloud ist es im Wesentlichen auf IaaS Dienste beschränkt, deckt dort aber die ganze Bandbreite der Rechen-, Speicher und Netzwerk-Ressourcen ab, die über ein zentrales Dashboard oder die OpenStack API konfiguriert und verwaltet werden.

OpenStack bezeichnet wird zwar auf der eigenen Projektseite als „Cloud Operation System“ bezeichnet¹⁰, läuft aber als eine hauptsächlich in der Script-Sprache Python entwickelte Applikation auf dem „General Purpose Operating System“ Linux und sollte daher eher als Cloud-Orchestrierungssoftware gesehen werden. Alle von OpenStack zur Verfügung gestellten Ressourcen kommen letztendlich aus den unterliegenden Betriebssystemen. Auch der für ganze virtuelle Maschinen benötigte Hypervisor ist keine eigene Implementierung, es werden hier nur die Schnittstellen zu bekannten Technologien wie KVM, Xen, Hyper-V oder VMware genutzt. OpenStack stellt hier nur die Mittel zur Konfiguration und Verwaltung zur Verfügung, abstrahiert aus vorhandenen Ressourcen neue und ermöglicht die Nutzung über das Netzwerk.

⁹<http://www.openstack.org/themes/openstack/images/openstack-software-diagram.png>

¹⁰About Owncloud, 2014, <https://owncloud.com/about/>

Konzeptionell setzt sich OpenStack aus den folgenden Komponenten zusammen¹¹:

- Das Web-Portal „Horizon“ stellt einen Selbstbedienungszugang zu den elementaren Diensten von OpenStack zur Verfügung. Damit können Benutzer etwa virtuelle Maschinen starten, IP Adressen zuordnen und Zugriffskontrollen konfigurieren.
- Mit dem „Nova“ Computing-Service wird der Life-Cycle Management Prozess der Computing Instanzen implementiert. Virtuelle Maschinen können damit eingerichtet und wieder gelöscht werden.
- „Neutron“ stellt einen „Networking-Connectivity as a Service“ Dienst zur Verfügung, der nicht nur von Benutzern direkt sondern auch vom Nova-Computing-Dienst genutzt wird. Über ein API können Netzwerke per Software definiert und mit Komponenten wie virtuelle Maschinen oder virtuelle Router verbunden werden. Die Software unterstützt über eine erweiterbare Architektur die Ein- und Anbindung der Netzwerk-Hardware unterschiedlicher Hersteller.
- Der „Swift“ Object Store erlaubt eine skalierbare und redundante Speicherung von beliebigen Datenobjekten in der Cloud Infrastruktur. Der Zugriff erfolgt nicht über ein herkömmliches Netzwerk-Filesystem Protokoll sondern eine RESTful API auf http Basis.
- „Cinder“ stellt virtuellen Blockspeicher zur Verfügung. Über diesen Dienst können beispielsweise Nova-Computing-Instanzen virtuelle Disken zur Verfügung gestellt werden.
- Der „Keystone“ Identity Service stellt die notwendigen Authentisierungsfunktionen für alle anderen OpenStack Dienste und deren Benutzer zur Verfügung.
- „Glance“ verwaltet virtuelle Disk Images und wird von Nova während der Bereitstellung neuer virtueller Maschinen benötigt. Die verschiedenen Images werden dabei von Glance als Swift Objekte gespeichert.

¹¹Get started with OpenStack, 2015, <http://docs.openstack.org/admin-guide-cloud/content/ch.getting-started-with-openstack.html>

- „Ceilometer“ überwacht und misst die Verwendung von OpenStack Ressourcen für Verrechnung und statistische Auswertungen.
- „Heat“ implementiert eine übergeordnete Orchestrierung von zusammengesetzten Cloud-Applikationen, deren benötigte Bestandteile über eine Template-Datei beschrieben wird.
- Der „Trove Database as a Service“ Dienst stellt die Funktionalitäten von relationalen und nicht-relativen Datenbank Systemen zur Verfügung.

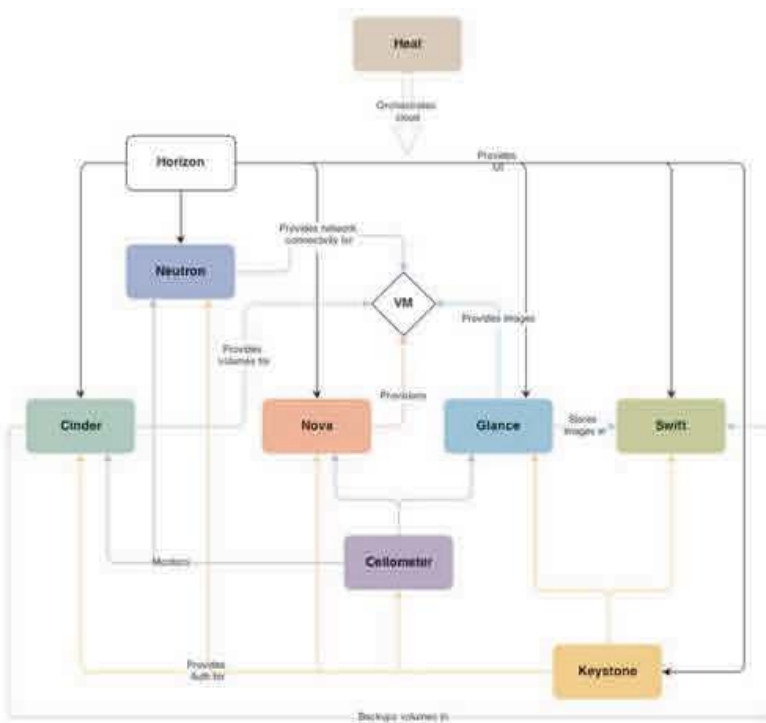


Abbildung 3.3: OpenStack Architektur¹²

OpenStack ist mit diesen Diensten in der Lage die Basis IT Infrastruktur ganzer Firmen abzubilden. Da das System außerdem voll mandantenfähig ist, kann es zur Implementierung von Private und Public Clouds

¹²http://docs.openstack.org/juno/install-guide/install/yum/content/figures/1/a/common/figures/openstack_havana_conceptual_arch.png

eingesetzt werden. Der Leistungsfähigkeit der Plattform sind aufgrund der guten Skalierbarkeit der einzelnen Dienste kaum Grenzen gesetzt. Allerdings werden diese Vorteile über ein Maß an Komplexität erkaufte, das weit über das einer standard OwnCloud-Installation liegt. OpenStack Installationen werden daher im privaten oder kleinem Unternehmensumfeld nur selten zu finden sein und eher den erfahrenen IT Abteilungen mittlerer und größere Unternehmen vorbehalten bleiben

3.8 Zusammenfassung

Auch wenn Cloud Dienste keine grundsätzlich neuen Technologien darstellen, können sie in der Regel aufgrund der neuartigen Art der Verwaltung und Bereitstellung doch viele interessante Vorteile bieten: Zugang und Nutzung von Ressourcen von überall und fast grenzenlos erscheinende Skalierbarkeit. Als Kunde hat man über die vielfältigen Angebote der diversen Public-Cloud-Provider Zugang zu einer Dienst-Qualität, wie sie früher nur durch die IT Abteilungen großer Firmen geboten werden konnte. Wer aber die Kontrolle über die Daten und Dienste nicht abgeben kann oder will, hat hinreichende Möglichkeiten, seine eigene Private Cloud zu schaffen und zu betreiben.

4 Rechtliche Rahmenbedingungen bei der Nutzung einer Cloud im Unternehmensumfeld

Michael Brickmann

Um Cloud-Dienste im unternehmerischen Umfeld nutzen zu können, ist je nach Art der Daten ein erhöhter technischer Sicherungsaufwand von Nöten. Der Grad der Sicherheit ist je nach Art der Daten verschieden. Wird nur ein Cloud-Speicherdienst genutzt um Bilder zu speichern und eventuell mit Angestellten zu teilen, muss weniger beachtet werden als wenn die Personalverwaltung online abgewickelt wird. Je nach Größe des Unternehmens ist die Bestellung eines/einer eigenen Datenschutzbeauftragten sinnvoll.

4.1 Datenarten

Generell ist besonderes Augenmerk auf die Art der Daten, welche nicht im eigenen Unternehmen gespeichert werden, zu legen. Handelt es sich um personenbezogene Daten von Angestellten oder Kunden und Kundinnen im Sinne des § 4 Z 1 Datenschutzgesetz 2000¹ (DSG 2000), besteht besondere Sorgfaltspflicht. Personenbezogene Daten unterliegen gem. § 1 DSG 2000 dem Datenschutz und müssen entsprechend gegen Fremdzugriff geschützt werden. Dies stellt Unternehmen als datenschutzrechtliche Auftraggeber iSv § 4 Z 4 DSG 2000 (In der Folge Auftraggeber) vor das Problem, auf die Angaben des Cloud-Anbieters als datenschutzrechtlichen Dienstleister gem § 4 Z 5 DSG 2000 (in der Folge Anbieter genannt) zu vertrauen. Prinzipiell ist der Auftraggeber verpflichtet, für die

¹Bundesgesetz über den Schutz personenbezogener Daten (Datenschutzgesetz 2000 - DSG 2000) BGBl I 1999/156 idF BGBl I 2013/83.

ordnungsgemäße Sicherung der Daten Sorge zu tragen. Auch sämtliche Meldepflichten über die Verwendung der Daten obliegen dem Auftraggeber (§ 17 DSGVO 2000). Dem Anbieter werden die Daten nur zur Speicherung überlassen. Eine andere Verwendung oder Bearbeitung der Daten durch den Anbieter ist nicht zulässig.

4.2 Vertrag

Um unangenehmen Überraschungen vorzubeugen, müssen bereits im Vertrag sämtliche Unklarheiten beseitigt und Zuständigkeiten definiert werden. Die wichtigsten Vertragsbestandteile setzen sich aus den folgenden Kapiteln zusammen.

4.2.1 Ort der Speicherung

Werden die Daten direkt beim Anbieter gespeichert, oder in einer Zweigstelle im In- oder Ausland? Wenn Daten im Ausland gespeichert werden, ist dies zulässig solange im entsprechenden Land angemessene Datenschutzbestimmungen bestehen. Die Speicherung in EU Mitgliedsstaaten stellt kein direktes Problem dar, bedarf aber je nach Art der Daten eine Meldung. Solange der Datenschutz nicht EU-weit vollständig harmonisiert ist, und weiterhin nationale Umsetzungen der Datenschutzrichtlinien anzuwenden sind, kann es in bestimmten Fällen sein, dass ausländische Regelungen strikter oder weniger strikt sind Anwendung finden. Es ist allerdings immer österreichisches Recht anzuwenden, wenn die Datenverwendung zu Zwecken des Auftraggebers erfolgt und die Haupt- oder Zweigniederlassung in Österreich gelegen ist². Auch die Speicherung von Daten in den Vereinigten Staaten ist prinzipiell möglich, was jedoch österreichische KMUs nur selten betrifft (Safe Harbor³).

4.2.2 Art der Speicherung

Wie sind die Daten abgesichert? Gibt es ein Backup System oder nicht. Wie wird ein Fremdzugriff verhindert? Werden die Datensicherheitsmaßnahmen gemäß § 14 DSGVO 2000 eingehalten oder nicht? Gibt es eine Verschlüsselung und Zugriffskontrollen?

²Vgl. §3 DSGVO 2000.

³<http://www.export.gov/safeharbor/?partnerid=>

4.2.3 Beendigung des Vertrags

Was passiert mit den Daten nach Beendigung des Vertrags? Werden diese sicher vernichtet? Wie bekommt der Auftraggeber nach Beendigung des Vertrags seine Daten zurück?

4.2.4 Insolvenz / Veräußerung des Anbieters

Was geschieht mit den Daten wenn der Anbieter insolvent wird oder von einer anderen Firma übernommen wird. Dieser Punkt ist besonders wichtig, da es eventuell zu einem Datenverlust kommen kann. Bei ausländischen Anbietern ist hier auch auf die Regelungen des Heimatlandes des Anbieters einzugehen.

4.2.5 Verfügbarkeit

Die Verfügbarkeit der Dienste des Anbieters kann für Ihr Unternehmen eine wichtige Rolle spielen. Im Rahmen des Vertrags sollte daher eine Klausel über die Verfügbarkeit der Services integriert werden. Marktüblich ist eine Verfügbarkeit von 99%.

4.3 Sicherheitsaudits

Um sicher zu gehen dass der Anbieter die vertraglich zugesicherten Punkte auch einhält, sollten unregelmäßige Sicherheitsaudits durch den Auftraggeber oder vom Auftraggeber beauftragte Unternehmen durchgeführt werden können. Auch dies ist vertraglich festzuhalten. Sollte der Anbieter aus irgendeinem Grund einen Subanbieter benötigen, welcher eventuell Zugriff auf die Daten bekommen könnte, so haftet der Anbieter gemäß §1313a ABGB⁴ auch für diese.

4.4 Zertifizierung

Da der Auftraggeber für die Einhaltung des Datenschutzes verantwortlich ist, sollte die Auswahl des Anbieters einigen Qualitätskriterien entsprechen. Um gewisse Qualitätsstandards bieten zu können, sind manche An-

⁴Allgemeines bürgerliches Gesetzbuch für die gesamten deutschen Erbländer der Oesterreichischen Monarchie, JGS 1811/946 idF BGBl 2015/35.

bieter zertifiziert. Ein entsprechendes anerkanntes Industriezertifikat gibt Aufschluss, ob der Anbieter die Sicherheitsmaßnahmen und die Bestimmungen des Datenschutzes einhält oder nicht. Um ein solches Zertifikat zu erlangen muss der Anbieter sich regelmäßigen Audits unterziehen und die eingesetzte Soft- und Hardware wird – je nach Anbieter – auf Aktualität und Sicherheitslücken überprüft. Werden die vorausgesetzten Kriterien nicht mehr erfüllt, wird das Zertifikat entzogen. Entsprechende Zertifizierungsanbieter finden sich im gesamten europäischen Raum, unter anderem werden Zertifikate vom TÜV in Österreich⁵ und Deutschland angeboten. Eine ISO Norm⁶ für die Implementierung eines Informationssicherheits-Managementsystems wurde bereits vor einigen Jahren von der *International Organization for Standardization* herausgegeben. Ist ein Anbieter nach dieser Norm zertifiziert, kann zumindest ein Basisschutz sowie die Einhaltung europäischer Datenschutzrichtlinien vorausgesetzt werden.

⁵<http://www.tuev.at/start/browse/de/Webseiten/TUV%20Austria%20Cert>
(zuletzt überprüft am 26.02.2015).

⁶ISO/IEC 27001 <http://www.iso.org/iso/home/standards/management-standards/iso27001.htm> (zuletzt überprüft am 26.02.2015).

5 Understanding the social web – Soziale Medien in KMU

Franz Niederl, Elmar Krajnc und Michael Ulm

5.1 Einleitung

Social Media ist aus dem alltäglichen Leben nicht mehr wegzudenken. Waren soziale Netze im Internet anfänglich Projekte von Studenten um Jahrbücher online abzubilden (wie Facebook 2004) oder von Angestellten in der IT um private Videos zu teilen (wie Youtube 2005) so sind diese heute wichtige Kanäle für Kommunikation und Information. Dieser Beitrag präsentiert mit aktuellen Zahlen wie weit Social Media verbreitet ist. Der grundlegende Aufbau wird erläutert und mit Hilfe von Modellen gezeigt, wie Social Web funktioniert. Abschließend wird erläutert, welche Marketing-Maßnahmen verwendet werden können um Social Media in Unternehmen einzusetzen.

5.2 Definition von Social Media

„Social Media dienen der – häufig profilbasierten – Vernetzung von BenutzerInnen und deren Kommunikation und Kooperation über das Internet.“¹

Diese Definition von Social Media ist eine Verallgemeinerung des Begriffes Social Web, welche das WWW als notwendiges Kriterium enthält. Social Media hingegen umfasst alle Applikationen, ob webbasiert oder nicht, die zum Austausch von Informationen und zur Erstellung dieser dienen. Dazu zählen Facebook, Twitter, WhatsApp, Snapchat, LinkedIn,

¹<http://wirtschaftslexikon.gabler.de/Definition/soziale-medien.html>

Xing, Google+ etc (Szugat, Gewehr und Lochmann, 2006). Des Weiteren ist die Reputation des Senders und des Empfängers von großer Bedeutung, sodass die Pflege des eigenen Profils und das Setzen von Aktivitäten sowie die Reputation der eigenen Kontakte, die Akzeptanz und Glaubwürdigkeit eines Profils stärken oder schwächen. Beziehungen, die im realen oder virtuellen Leben aufgebaut wurden oder werden, sind mit diesen Medien über längere Zeiten und größere Distanzen aufrecht zu erhalten. Vernetzung und Kooperation über das Internet bedeutet, dass digitale Verbindungen diesen Kollaborationen zugrunde liegen.

Im Zentrum dieser Medien stehen die BenutzerInnen bzw. die Gruppen und deren Kommunikation untereinander. Die Kommunikation kann uni- oder bidirektional erfolgen. Ein wichtiger Aspekt ist die Transparenz aller Vorgänge, die Personen, Beziehungen, Inhalte und Bewertungen betreffen können. (Ebersbach, Glaser und Heigl, 2010) All diese userbasierten Aktivitäten werden gemeinsam mit dem Benutzerprofil und dem Freundesnetzwerk im Social Container gespeichert. Dieser bildet einen Ausgangspunkt für viele Social-Media Applikationen, um eine neue große Gruppe an Benutzern zu erreichen. (LeBlanc, 2011)

Die Website „Garys Social Media Count“² veranschaulicht die Anzahl der in einer Sekunde neu hinzugekommen Likes auf Facebook (2 700 000 000 pro Tag), Videos in Stunden auf Youtube (142 500h pro Tag), Anzahl der gesendeten Tweets (4 000 000 000 pro Tag) und viele weitere Social Media Aktivitäten. Diese Zahlen sind nur Richtwerte, da die exakten Werte nicht verfügbar sind, trotzdem erhält man einen guten Einblick in die Aktivitäten der Benutzer.

Die Bedeutung von Social Media ist aber schon seit Jahren über den privaten Bereich hinausgewachsen. Barack Obama war der erste Präsident, der im Wahlkampf 2007 eine für die damalige Zeit sehr fortschrittliche und ausgefeilte Social Media Kampagne geführt hat. Mittlerweile haben sich auch alle anderen Politiker darin mit unterschiedlichen Erfolgen versucht. Negatives Beispiel ist der Social Media Auftritt von Werner Faymann, der mit „Marionetten-Accounts“ gefüllt worden sein soll³. Perso-

²<http://www.personalizemedia.com/garys-social-media-count/>

³http://diepresse.com/home/politik/innenpolitik/709480/Facebook_Falsche-Freunde-fur-Faymann

nen des öffentlichen Lebens nutzen hingegen sehr oft das Social Medium Twitter, um Kurznachrichten zu versenden. Auch Unternehmen sind schon seit den Anfängen in den Social Media vertreten, wobei der Erfolg nicht zufällig eintritt, sondern nur durch exakte Planung, Ausdauer und etwas Glück erreicht werden kann (siehe Kapitel 5.5).

Eine aktuelle IFES Studie im Auftrag des Public Relations Verbands Austria besagt, dass 83% der Befragten der Meinung sind, dass gekaufte Postings den Leser hinters Licht führen. Firmen die Postings kaufen, stehen ebenfalls bei 79% in einem schlechten Licht dar. Auch die Agenturen, die Postings verkaufen, bekommen von 75% Prozent ein negatives Image angeheftet⁴. 2011 wurden im arabischen Frühling Blogs und Foren erfolgreich zur Organisation des Umbruchs eingesetzt. Die Nachhaltigkeit dieser Veränderungen und die Auswirkungen für die Bevölkerung sind leider nicht in allen Staaten positiv zu beurteilen.

Der Social Graph, wie die Verknüpfung aller Benutzer einer Social Media Plattform bei Facebook bezeichnet wird, wird immer komplexer und es können immer mehr Parameter eingebunden werden. Das bedeutet, dass man seine große Anzahl an Freunden nach beliebigen Angaben wie dem Betriebssystem des Smartphone, der Körpergröße oder der Anzahl der gesetzten Likes gruppieren kann. Auf dieser Gruppierung aufbauend können klassische Recommendersysteme (McIlwraith, Marmanis und Babenko, 2015) eingesetzt werden, um zielgerichtete Aktivitäten zu setzen. Choudary⁵ entwickelt den Social Graph weiter, sodass dieser auch Beziehungen zwischen Unternehmen darstellen kann. In seiner Darstellung wird der Unternehmensname, die Art der Beziehung (Käufer oder Verkäufer) und eine Bewertung („Rating“) des Unternehmens durch andere einbezogen. Mit diesem Social Business Graph kann dieselbe zentrale Idee (individuelle Angebote) wie beim Social Graph im B2B-Bereich umgesetzt wären.

Abbildung 5.1 zeigt den Social Media Effect. Der einmal erstellte Inhalt wird über die unterschiedlichen Social Media Plattformen und über diverse Suchdienste weiterverbreitet. Beachtenswert ist in dieser schematischen Darstellung, dass die Verteilung eines Beitrages über viele Medien weitergetragen wird. Diese Ausbreitung zeigt mit welchem geringen Aufwand

⁴Horizont 5/2015

⁵<https://hbr.org/2015/02/the-rise-of-social-graphs-for-businesses>

gut vernetzte Beitragsschreiber die Artikel an viele Personen bringen.

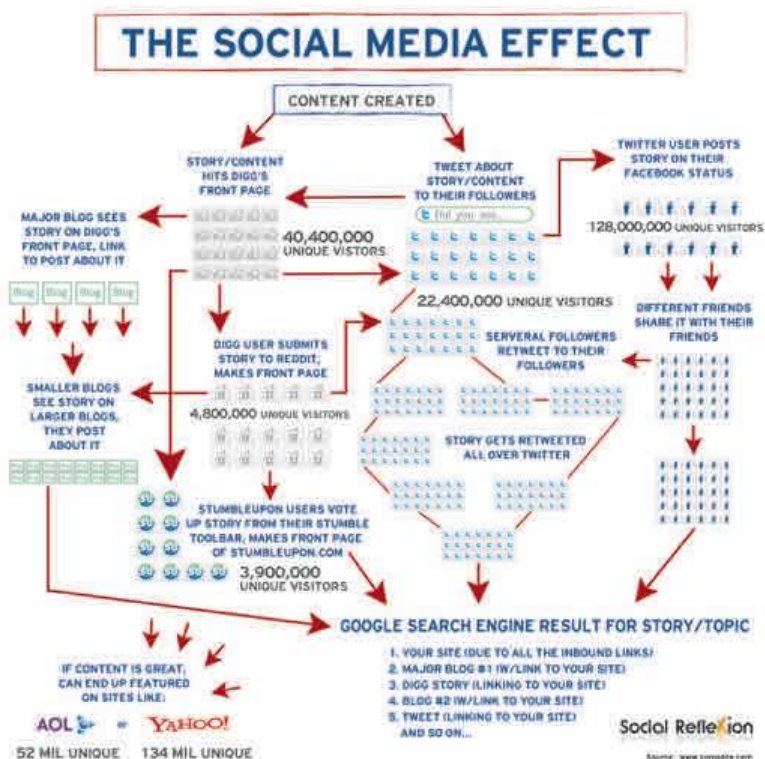


Abbildung 5.1: The Social Media Effect

Die Bedeutung bzw. der monetäre Wert eines Social Mediums ist stark mit der Größe der Community korreliert. Die Benutzerzahlen werden auch in den nächsten Jahren noch weiter ansteigen. Weltweit wird diese Zahl von 2015 mit ca 2 Milliarden 2018 auf ca 2.5Milliarden ansteigen⁶. Das Durchschnittsalter der Social Media Benutzer liegt bei 37 Jahre, ist also deutlich über der als sehr technologieaffin geltenden Kategorie der 15 bis 30jährigen. In dieser Kategorie der 15 bis 30jährigen hat der Marktführer Facebook bereits rückläufige Entwicklungen bei den Benutzerzahlen⁷. Ein Grund dafür kann sein, dass die Generation die Plattform Facebook wieder verlässt, da generationenübergreifende Benutzer vorhanden sind. Weil Jugendliche, Eltern und Großeltern dasselbe Medium nutzen, ist die Pri-

⁶<http://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/>

⁷<http://strategylabs.com/2014/01/3-million-teens-leave-facebook-in-3-years-the-2014-facebook-demographic-report/>

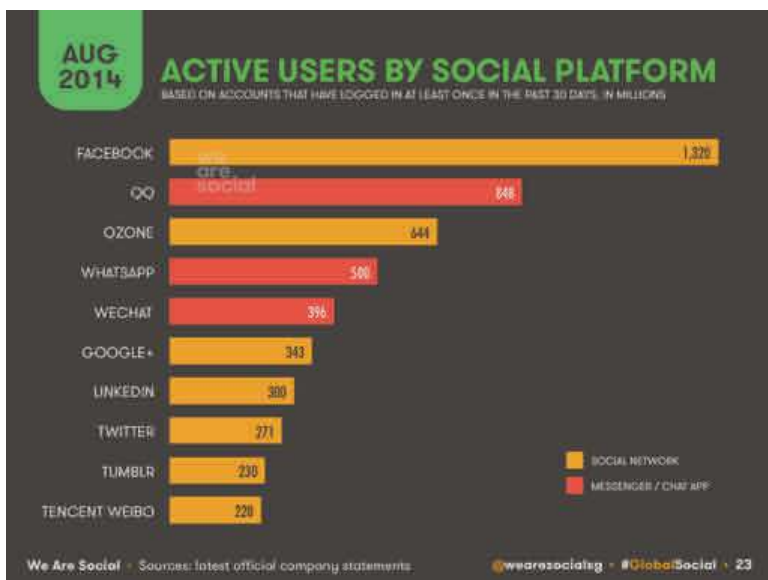


Abbildung 5.2: Aktive Benutzer im August 2014⁸

vatsphäre einer Generation gegenüber einer anderen Generation nicht mehr gesichert. Das olympische Motto „Dabei sein ist alles!“ eingeschränkt auf die richtige Gruppe gilt auch bei Social Media. Wenn alle überall dabei sind wird das Dabeisein gleichermaßen uninteressant. (Zeger, 2009)

In Österreich sind aktuell 3.400 000 Facebook, 130 000 Twitter und 600 000 LinkedIn-Benutzer registriert. Betrachtet man die Anzahl der Benutzer in den verschiedenen Altersgruppen, so ist diese Anzahl bei Facebook von jung nach alt fallend und bei LinkedIn steigend⁹. Die unterschiedliche Struktur spiegelt sich in der Ausrichtung der beiden Social Media wieder. Welche Social Media für welche Aufgaben am besten gewählt werden, wird in Abschnitt 5.4 detaillierter erläutert.

5.3 Mobile Nutzung von Social Media

„Facebook ist ein mobiles Unternehmen“ hat Mark Zuckerberg 2013 verlautbart. Die aktuellen Zugriffszahlen zeigen auch ein entsprechendes Bild.

⁸<http://was-sg.wascdn.net/wp-content/uploads/2014/08/GLOBAL-SOCIAL-STATS.png>

⁹<http://www.socialmediaradar.at>

Die Studien von eMarketer.com belegen dieses Statement mit Zahlen. 2015 sollen 986,9 Millionen (70,1%) Benutzer und 2018 schon 1,34 Milliarden (75,1%) mittels mobile Device auf Facebook zugreifen¹⁰. Begründet wird diese Entwicklung damit, weil die Facebook-Benutzer über alle Endgeräte ein ähnliches Benutzer-Verhalten und ähnliche Erfahrungen aufweisen. Betrachtet man die Aussage Mark Zuckerbergs in finanzieller Hinsicht, so werden laut derselben Studie bereits 2015 die Einnahmen im mobilen Bereich 73% (10,90 Milliarden) der Gesamteinnahmen ausmachen.

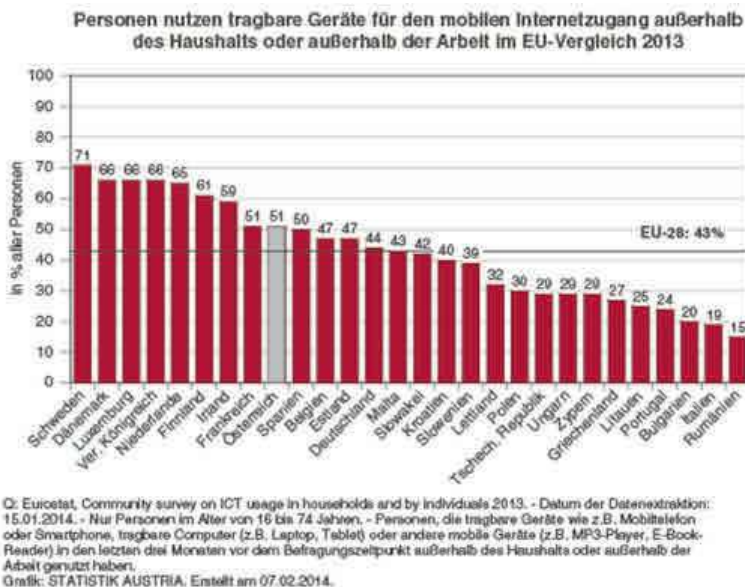


Abbildung 5.3: Mobile Internetzugänge

In Österreich nutzen laut einer Umfrage der Statistik Austria 51% aller Personen tragbare Geräte für den mobilen Internetzugang außerhalb des Haushaltes bzw. der Arbeit. Damit sind wir knapp über dem EU-Durchschnitt von 43%. Teilt man die Bevölkerung nach Altersgruppen ein, so sind die 16-24jährigen mit über 90% deutlich vor den anderen Altersstufen, wo die Nutzungsrate mit zunehmendem Alter abnimmt. Dies ist wie in so vielen Fällen ein klassisches Generationsproblem, das sich in 10-20 Jahren von selber behoben haben wird. (siehe Abbildung 5.3)

¹⁰<http://uk.businessinsider.com/emarketer-2016-facebook-mobile-revenue-and-mobile-user-estimates-2015>
r=US

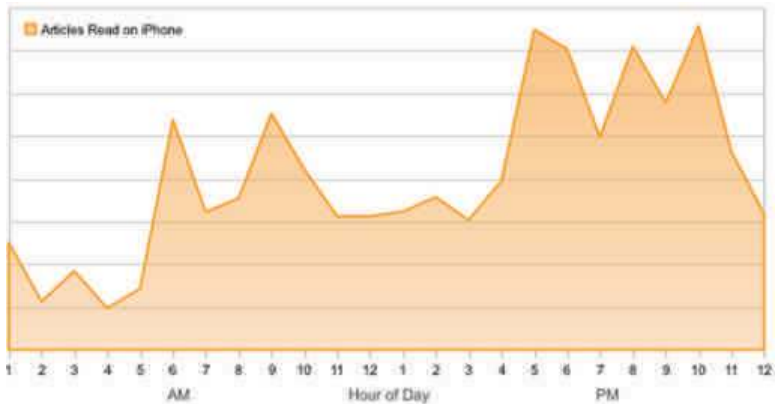


Abbildung 5.4: mobile Lesezeiten¹¹

Die mobile Nutzung von Social Media unterscheidet sich deutlich vom klassischen Desktop Einsatz. Wichtige Fragen sind wo und wann man mobil online ist. 84% verwenden zu Hause ihre mobilen Endgeräte, 80% nutzen diese in den Zwischenzeiten, 74% in Warteschlangen und 64% in der Arbeit. Mobile Verwendung von Social Media Apps erfolgt meistens in einer Umgebung, die ebenfalls Aufmerksamkeit benötigt (U-Bahn, Auto, ...) Abbildung 5.4 zeigt die mobilen Lesezeiten. Die Spitzen sind dabei 6 Uhr morgens beim Frühstück, 9 Uhr morgens beim Arbeitsstart und Abends ab 5 Uhr. Diese Zeitpunkte sind auch die optimalen Zeitintervalle zum Versenden von Informationen über unterschiedliche Social Media Kanäle.

Social Media Aktivitäten wie das Anklicken eines Links, das Posten von Beiträgen oder dem Ansehen von Fotos unterscheiden sich ebenfalls nach der Nutzung. Interaktionen mittels mobiler Geräte müssen effizient und effektiv erledigt werden können. Liken ist daher beliebter als Posten eines Beitrages.

Zusammenfassend ergibt sich, dass der Inhalt für mobile Nutzung optimiert werden muss, um die richtige Zielgruppe mit der besten Interaktion zum richtigen Zeitpunkt anzusprechen.

¹¹Quelle: <http://readitlaterlist.com/blog/2011/01/is-mobile-affecting-when-we-read/>

5.4 Modelle der Plattformen

Um Social Media auch für Marketing-Zwecke zu nutzen, ist es durchaus wertvoll zu verstehen, nach welchem Modell eine Plattform funktioniert. Wenn Unternehmen wissen, wie ein Netzwerk funktioniert, welche Bedürfnisse die Nutzer haben und welche Art von Beziehungen bestehen, dann kann man auch die Menschen die Social Media nutzen erreichen. Im folgenden Abschnitt werden die grundlegenden Konzepte der Relationship-Modelle (vgl. LeBlanc, 2011) und des Honeycomb-Modells (vgl. Kietzmann u. a., 2011) erläutert.

5.4.1 Relationship Modelle

Einer der wichtigsten Bestandteile aller Medien im Social Web ist die Möglichkeit, dass sich die NutzerInnen untereinander vernetzen. Die Art der Vernetzung kann dabei aber sehr unterschiedlich sein. Nicht immer ist ein Verbindung im Social Web eine 1:1-Verbindung.

Follower Modell

In diesem Modell ist die Interaktionen mit vielen Personen zur gleichen Zeit vorgesehen. Ein Teilnehmer erstellt den Inhalt und teilt diesen mit seinen Followern, ähnlich wie ein Rundfunksender der sein Programm ausstrahlt und die Zuseher können dieses verfolgen. In Abbildung 5.5 ist das Schema dieses Modells ersichtlich.

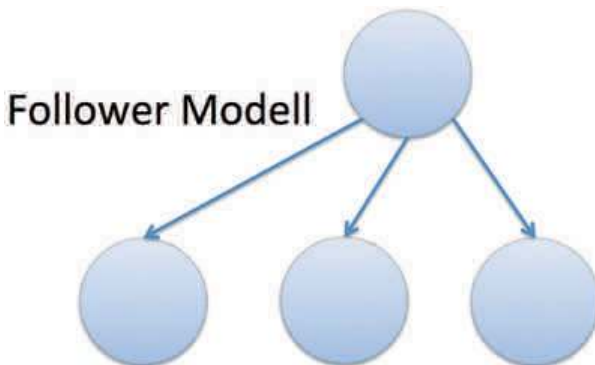


Abbildung 5.5: Follower-Modell

Vorteile dieses Modells sind für den Absender die mögliche große Reichweite und es muss zu den Empfängern keine direkte Verbindung bestehen. Die Nutzer können sich sehr gezielt aussuchen, wem sie folgen wollen und somit genau diese Informationen auswählen die man bekommen möchte. Typische Beispiele im Social Web für dieses Modell sind zum Beispiel Twitter, Youtube oder Instagram.

Connection Modell

Im Connection-Modell sind die Beziehungen immer zwischen zwei Teilnehmern vorhanden. Die Verbindung besteht dabei gleichwertig in beide Richtungen und beide Teilnehmer müssen diese Verbindung meist aktiv bestätigen. Beide haben meist nach der Zustimmung sehr detaillierte Informationen über das andere User-Profil. Das Berechtigungssystem zum Schutz der Privatsphäre ist oft sehr schwierig einzustellen. Ein weiterer Gesichtspunkt in diesem Modell ist der „Freund-meines-Freundes“-Aspekt. Dabei kann man herausfinden über welche Verbindungen man zu Nutzern mit denen ich nicht „befreundet“ bin verbunden ist (siehe Abbildung 5.6).

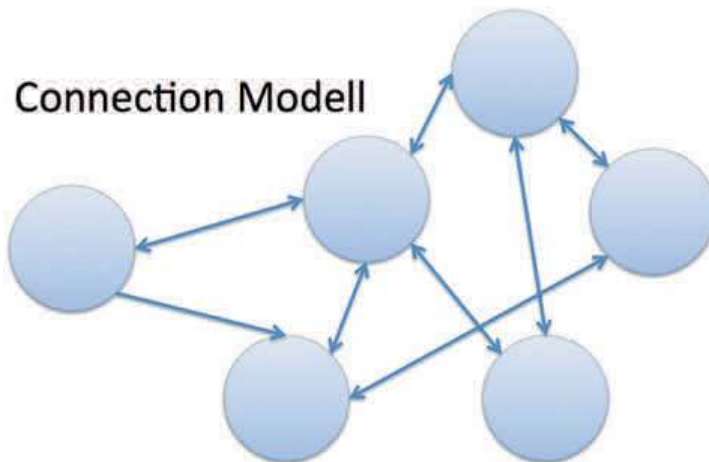


Abbildung 5.6: Connection-Modell

Group Modell

Ein anderer Ansatz der Verbindung wird im Group-Modell verfolgt. Die Teilnehmer sind oft nicht direkt miteinander verknüpft, aber über Gemeinsamkeiten wie z.B. gleiche Interessen, Hobbies oder auch Ansichten, in der selben Gruppe. Dieses Modell bietet Interaktion mit einer kleinen Gruppe von Personen zur gleichen Zeit und auch Interaktion innerhalb der Gruppe. Die Teilnehmer können auch mehreren Gruppen angehören. Ein repräsentatives Group-Modell wird auch in Abbildung 5.7 gezeigt.

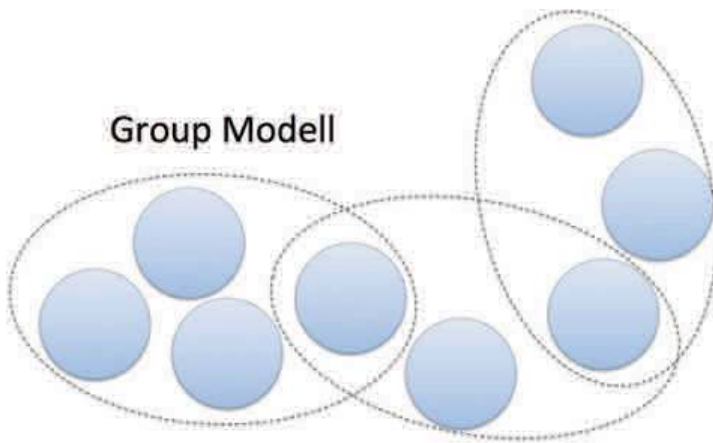


Abbildung 5.7: Group-Modell

5.4.2 Honeycombmodell

Mit Hilfe dieses Modells soll die Komplexität von Social Media nähergebracht werden, indem unterschiedliche Bauteile betrachtet werden aus denen die verschiedenen Plattformen aufgebaut sind. Die Grundbestandteile – Identität, Konversation, Teilen, Präsenz, Beziehungen, Reputation und Gruppenbildung – sind in allen Social Media vorhanden aber nicht überall gleich stark ausgeprägt.

Blöcke des Honeycomb-Modells:

- **Identität**

Dieser Block zeigt an wie viel der User von seiner richtigen Identität preisgibt. In vielen Fällen werden Name, Alter, Ort oder be-



Abbildung 5.8: Blöcke des Honeycomb-Modells¹²

ruffliche Tätigkeit angeben. Wird die Verlässlichkeit dieser Angaben überprüft, so erhöht dies die Bedeutung dieses Blockes für das betroffene Netzwerk. Wenn Kunden und Unternehmen in sozialen Netzen interagieren, soll die Balance zwischen dem Teilen von persönlichen Daten und der Wahrung der Privatsphäre geachtet werden.

- **Konversation**

Inwieweit kann die Plattform zu Kommunikation genutzt werden und wie verwendet die Zielgruppe diese Kanäle. Auch die Art und die Frequenz der Kommunikation sind hierbei ein wichtiges Kriterium. Man muss Konversationen an die übliche Form der jeweiligen Plattform anpassen z.B. bei Twitter sind Nachrichten mit maximal 140 Zeichen sehr kurz aber dafür wird eine höhere Frequenz erwartet.

- **Teilen**

Seit Web 2.0 ist der User nicht nur passiver Konsument sondern erstellt und teilt Inhalte auch aktiv. Der Block gibt Auskunft darüber in wie weit dies vom Medium unterstützt und von den Teilnehmern

¹²www.youtube.com/watch?v=BCjCbqYs.j0

erwartet wird.

Unternehmen können Social Web als Vertriebsweg für (zusätzlicher) Information, Videos, Bilder nutzen, aber auch von ihren Kunden Feedback zurückbekommen.

- **Präsenz**

Wie verfügbar sind die Teilnehmer des Mediums? Sind sie in der virtuellen oder realen Welt an einem Ort und wie rasch reagiert man auf die Interaktion? Mit der ständigen Verfügbarkeit der Medien wird auch von den Teilnehmern erwartet, dass auf Beiträge rasch reagiert wird.

- **Beziehungen**

Wie schon bei den Relationship Modellen gibt es unterschiedliche Arten, wie die Teilnehmer des Social-Nets miteinander verbunden sind.

- **Reputation**

Die Reputation zeigt an wie hoch das Ansehen oder die Stellung des Teilnehmers ist. Kann ich darauf vertrauen, dass die virtuelle Repräsentation einer Person oder eines Unternehmens tatsächlich echt ist?

- **Gruppenbildung**

Wie im echten Leben tendieren die Menschen dazu sich in Gruppen zu formieren und so ist es auch in Online. Gruppen in den Social Media können offen oder geschlossen sein, privat oder öffentlich.

Diese Merkmale oder Bausteine sind in den verschiedenen Plattformen des Social Webs sehr unterschiedlich ausgeprägt. In Abbildung 5.9 werden die beiden Plattformen Facebook und Youtube anhand der Einteilung durch das Honeycomb-Model gegenüber gestellt. Wie man im Diagramm sehen kann ist der wichtigste Bestandteil bei Facebook der Beziehungsblock, weniger stark sind die Blöcke Präsenz, Identität, Konversation und Reputation. Die Bereiche „Teile“ und „Gruppenbildung“ sind in diesem Netzwerk eher im Vergleich zu den anderen schwächer ausgeprägt.

Bei der Plattform Youtube sieht die Ausprägung anders aus. Hier steht das Teilen von Inhalten im Vordergrund. Reputation, Konversation und auch Gruppen sind schwächer vertreten und die Präsenz, die Identität und

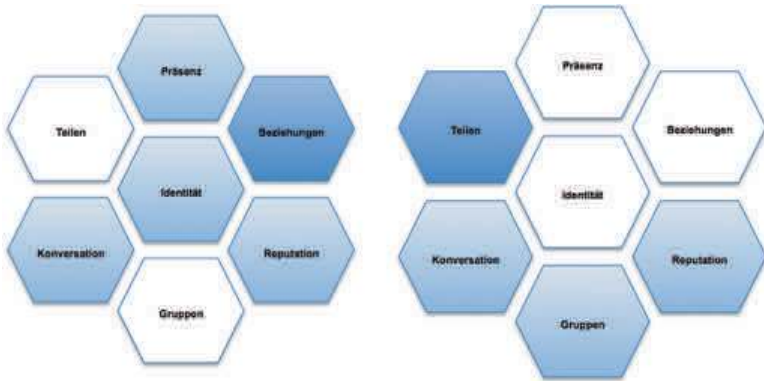


Abbildung 5.9: Facebook (links) und Youtube(rechts) im Vergleich

Beziehungen sind schwach vorhanden.

Um ein Social Media als Unternehmen zu nutzen, muss man um die Vor- und Nachteile der verschiedenen Plattformen Bescheid wissen. Man soll die Plattform wählen, die zum Unternehmen und auch zu den Teilnehmern passt, um Social Media zielgerecht nutzen zu können.

5.5 Einsatz von Social Media

Kein Auftritt auf Facebook, YouTube oder Instagram? Twitter, Google+ und LinkedIn werden auch nicht genutzt? Warum sind soziale Medien für Unternehmen notwendig? Diese Frage stellen sich immer wieder Unternehmen aufs Neue und müssen für sich herausfinden, ob sie Social Networks für sich einsetzen wollen oder nicht.

Dabei ist nicht nur die Frage dafür oder dagegen zu klären, sondern auch Punkte wie z.B. wer soll das Unternehmen vermehrt im Internet vertreten? Wie soll das genau geschehen? Wie oft sollten Neuigkeiten „gepostet“ werden? Und viele weitere Fragen tauchen in der Nutzung von Sozialen Netzwerken auf.

Einige dieser Fragen werden in diesem Abschnitt im Detail besprochen und geklärt.

5.5.1 Posting-Strategie

Durchdachte Inhalte sind eine gute Basis einer Posting Strategie. Inhalte müssen vorbereitet und ebenso aufbereitet werden. Texte müssen für eine bessere Lesbarkeit im Internet optimiert werden sowie auch für mobile Endgeräte passend gekürzt werden. Neben dieser textlichen Optimierung muss auch der sogenannte „Content“ sprich Inhalt auf das Unternehmen abgestimmt werden. Ein Zusammenhang von Unternehmen und geteilten Inhalte sollte teilweise bewusst und andererseits auch unbewusst den Kunden näher an das Unternehmen bringen (Beispiele siehe Abschnitt 5.5.2).

Der allseits bekannte rote Faden sollte sich durch alle Aktivitäten in sozialen Netzwerken ziehen. Die Auswahl kann ruhig buntgemischt sein, doch sollte ein bestimmtes Ziel stets verfolgt werden. Dieses Ziel gilt es je nach Tätigkeiten im Unternehmen bzw. nach strategischer Ausrichtung zu definieren. Konsistentes Handeln im Internet ist wichtiger denn je. Durch das „nicht vergessende“ Internet, ist es schließlich jedem möglich, auch ältere Inhalte abzurufen und mit neuen Meldungen, Aktivitäten oder Vorhaben zu vergleichen. Daher gilt es eine klare Zieldefinition in der Posting-Strategie zu verfolgen.

5.5.2 Unternehmen vs. Mensch

„Das Unternehmen begrüßt mit Freude seinen neuen Kunden Fa. XY“ oder „Wir begrüßen unseren neuen Kunden Fa. XY aus der Region Hochsteiermark und freuen uns mit regionalen Anbietern zusammenzuarbeiten“. Zwei Aussagen und derselbe Kerninhalt: Dass ein neuer Kunde begrüßt wird. Auch könnten beiden Textbausteine kombiniert werden, sodass eine gleich motivierende Aussage entsteht. Doch trotzdem unterscheiden sich beide Aussagen voneinander. Einerseits wird der Kunde von der dritten Person, dem Unternehmen und andererseits von den Mitarbeitern, dem Herz des Unternehmens, begrüßt.

Unternehmen vs. Mensch? Eine Entscheidung des Managements oder des Marketings? Oder doch eine einfache Entscheidung für mehr oder weniger Kundennähe? Jedes Unternehmen muss für sich diese Entscheidung treffen und auch in den Social Media Auftritt integrieren. Auch wird oftmals um die Anrede ob „du“ oder „Sie“ diskutiert. Dabei findet meist die

Entscheidung aufgrund des Alters der Kunden statt. Bei jungen Kunden tendiert man eher in Richtung „du“ und bei älteren Kunden eher zum „Sie“. Eine allgemein gültige Antwort kann man dazu nicht geben, sondern man sollte sich eher überlegen wie der Kunde im Tagesgeschäft angesprochen wird, sodass kein Bruch zwischen den normalen Kontakt und den Auftritt in einem Social Media Netzwerk vorhanden ist.

5.5.3 Beschwerdemanagement

Der Schritt ins Social Media bringt auch gleichzeitig eine neue Form des Beschwerdemanagements mit sich. Kunden können ihrer Unzufriedenheit sofort freien Lauf lassen und über diverse Kanäle dem Unternehmen sowie auch der breiten Öffentlichkeit ihre Meinung mitteilen. Dabei ist es wichtig zu reagieren statt zu ignorieren. Entgegennahme der Kritik, interne Analyse und Veröffentlichung einer durchdachten Antwort sind die drei wesentlichsten Schritte in dieser Form des Beschwerdemanagements.

Schnelligkeit, qualifizierte Antworten und vor allem keine Ausreden sollten integraler Bestandteil des Beschwerdemanagements im Sozialen Netzwerk sein. Manchmal ist es dabei auch geschickter einen Fehler einzugestehen anstelle den Fehler abzustreiten. Negative Stimmen „schreien“ schneller auf als einer positiven Zustimmung der bisherigen Leistungen, denn es geht schließlich um genau den einen Fehler aus 1000 sonst erfolgreichen Tätigkeiten. Alternativ zum Abstreiten ist auch eine Strategie, nach Verbesserungsvorschlägen zu fragen stets angebracht. Nur die wenigsten machen sich wirklich auch Gedanken, wie eine Verbesserung angestrebt werden kann und somit kann der erste Ärger einmal beseitigt werden.

5.5.4 Interaktion

Führen Sie keinen Monolog mit Ihren Kunden. Binden Sie Ihre spezielle Zielgruppe in Ihre Aktivitäten im Social Network mit ein. Stellen Sie offene Fragen, fragen Sie nach den Erfahrungen der Kunden, oder fragen Sie nach deren Meinung. Doch seien Sie aufmerksam und verwenden Sie keine kritischen Themen wie Religion oder Politik wo leicht eine Diskussion in eine ungewollte Richtung gehen könnte.

Oberstes Ziel jedoch ist: Seien Sie Mitglied der Community! Sehen

Sie sich nicht selbst als externen Beobachter sondern nutzen Sie Ihre eigenen Services oder Produkte. Schreiben Sie Artikel, die Sie selbst lesen würden und werden Sie ein Teil der Community. Schließlich wollen Sie Ihren Kunden näher kennenlernen und auch von ihm lernen. Gehen Sie auf die Bedürfnisse ein und interagieren Sie mit Ihren Kunden. Hauptsache Sie haben Spaß bei Ihren Social Media Aktivitäten.

5.5.5 Monitoring

Beiträge schreiben alleine reicht nicht! Verfassen Sie Antworten, geben Sie Bewertungen ab z.B. „liken“ Sie andere Seiten und Artikel, und stellen Sie Verlinkungen zu bekannten oder nützlichen Seiten her. Sie müssen neben dem Veröffentlichen von neuen Inhalten auch stets die „alten“ Inhalte im Auge behalten und diese fortlaufend überwachen.

Gibt es Veränderungen? Welche Artikel haben eine Langzeitwirkung auf Ihr Unternehmen? Analysieren, bewerten und Adaptieren Sie Ihre vordefinierten Ziele. Nur weil Sie JETZT sagen oder bestimmen wohin Sie wollen, heißt es nicht, dass es in 12 Monaten, sprich 365 Tagen immer noch so ist. In dieser langen Zeit können sich neue Trends ergeben und Ihr Ziel könnte ein anderes sein; so auch im Social Web. Die Community erweitert sich und entwickelt sich mit deren Meinungen, Anforderungen und Erfahrungen.

Bleiben Sie dennoch konsequent und behalten Sie Ihr grundsätzliches Ziel im Auge. Vielleicht müssen Sie ja nur die Wegbeschreibung zum Ziel ändern.

5.5.6 Verantwortlichkeiten

Es muss jemanden geben, der Social Media als seine Verantwortlichkeit ansieht. Doch bestimmen Sie nicht den Verantwortlichkeiten sondern finden Sie heraus, wer im Unternehmen dafür am besten geeignet ist.

Stehen Sie hinter ihren Aktivitäten, denn nur so können Sie auch auf Anfragen, Anregungen und anfallender Kritik adäquat antworten. Sind Sie nicht selbst von Ihren Vorhaben überzeugt, so wäre der Schritt ins Soziale Netzwerk nicht der richtige Weg, mit Ihren Kunden in Kontakt zu treten.

5.5.7 Zeitaufwand

Online präsent zu sein bedeutet, mit einem bestimmten Zeitaufwand zu kalkulieren. Sie fragen sich wie hoch dieser ca. sein wird? Dann gehen Sie ein paar Unterkapitel zurück und lesen sich all die Themen erneut durch. Social Media bedeutet Zeit zu investieren und mit der Community zu interagieren.

Tägliche Aktivitäten sind einzuplanen und auch abzuarbeiten. Tools wie z.B. BufferApp oder Hootsuite können Ihnen helfen bestimmte Veröffentlichungen regelmäßig durchzuführen um somit omnipräsent zu wirken. Doch eine manuelle Nachkontrolle und individuelle spontane „Postings“ sind nicht zu vernachlässigen.

Unterschätzen Sie daher nicht den notwendigen Zeitaufwand das soziale Netzwerk für sich arbeiten zu lassen. Entwickeln Sie Ihre eigene Social Media Posting Strategie und lernen aus Ihren Aktivitäten.

5.6 Zusammenfassung

Die Verwendung von Social Media in Unternehmen soll gut vorbereitet und überlegt sein. Als ersten Schritt jedoch sollte man versuchen die Potentiale zu finden und die Szenarien davon abzuleiten. Dazu sollten aber die gesamten Eigenschaften der unterschiedlichen Plattformen erkannt und verstanden werden. Eine Analyse der Stärken und Schwächen mit dem Honeycombmodell sowie des Social Containers der einzelnen Benutzer inklusive des dazugehörigen Relationship-Modells unterstützt dieses Vorhaben.

Der endgültige Einsatz von Social Media im Unternehmen kann unterschiedliche Ziele verfolgen. Meistens jedoch wird damit Marketing betrieben und versucht eine Community aufzubauen. Die Initialisierung, Instanziierung und Betreuung dieser Aktivitäten muss einem Konzept folgen, um die Aussicht auf Erfolg zu erhöhen.

In Zukunft werden sich keine Unternehmen den Social Media entziehen können. Die Frage ist, ob man als Unternehmen das Verständnis für die darin ablaufenden Prozesse hat und entsprechend vorbereitet ist oder

ob man durch ein ungewolltes Ereignis von den Benutzern in diese Prozesse unfreiwillig involviert wird.

Literaturverzeichnis

- Ebersbach, Anja, Markus Glaser und Richard Heigl (2010). *Social Web. 2.* überarb. Aufl. UTB, Stuttgart. ISBN: 978-3825230654.
- Kietzmann, Jan H., Kristopher Hermkens, Ian P. McCarthy und Bruno S. Silverstre (2011). „Social media? Get serious!“ In: *Business Horizons* 54.3, S. 241–251. DOI: 10.1016/j.bushor.2011.01.005.
- LeBlanc, Jonathan (2011). *Programming Social Applications*. 1st. O’Reilly Media. ISBN: 978-1-4493-9491-2.
- McIlwraith, Douglas G., Haralambos Marmanis und Dmitry Babenko (2015). *Algorithms of the Intelligent Web*. Second Edition. Manning Publications Co. ISBN: 9781617292583.
- Szugat, Martin, Jan Erik Gewehr und Cordula Lochmann (2006). *Social Software*. entwickler.press. ISBN: 978-3939084099.
- Zeger, Hans g. (2009). *Paralleluniversum Web 2.0: wie Online-Netzwerke unsere Gesellschaft verändern*. 1st. Kremayr & Scheriau. ISBN: 978-3-218-00792-4.

6 Bring your own device – BOYD

Kaja Unger

***Zusammenfassung:** Grundsätzlich stellen Unternehmen die IT-Infrastruktur zur Verfügung, doch immer öfter wünschen sich MitarbeiterInnen, ihre eigenen Endgeräte einsetzen zu dürfen. Ob dies zulässig und sinnvoll ist, wird in diesem Beitragsowohl aus Sicht der Unternehmen als auch der MitarbeiterInnen dargestellt.*

6.1 BYOD – Chancen nutzen – Risiken vermeiden

BYOD – „bring your own devices“ – ist eines der modernsten Schlagworte in der IT und für Klein- und Mittelbetriebe oft die Erfüllung eines Traums der MitarbeiterInnen. Meist wissen KMU jedoch gar nichts von diesem Schlagwort – für sie ist es nichts anderes, als dass ihre MitarbeiterInnen ihre eigenen IT-Endgeräte nutzen und das KMU sich die Kosten für die Anschaffung und auch Wartung dieser Geräte erspart. Damit ist auch schon der Begriff erklärt: Es handelt sich um die dienstliche Nutzung privater digitaler Geräte, wie Smartphones, Notebooks oder Tablets am Arbeitsplatz.

Allein das regelmäßige Mitbringen der privaten Endgeräte in das Unternehmen stellt noch keine Gefahr dar – problematisch wird es, wenn die Endgeräte mit dem Firmennetzwerk verbunden werden oder auf den Endgeräten Kundendaten, Betriebsgeheimnisse oder sonstiges Know-How des Unternehmens gespeichert oder verarbeitet werden. Dass damit auch Gefahren für das Unternehmen verbunden sind, wird vielen erst dann richtig bewusst, wenn Sicherheitsprobleme auftauchen. Etwa weil Kundendaten plötzlich verloren sind oder auch bei der Konkurrenz bekannt sind. Damit stellt sich auch die Frage, ob BYOD ein Segen oder ein Fluch ist.

Denn BYOD bietet Unternehmen auch wesentliche Vorteile. So ersparen sich die Unternehmen etwa den Einkauf von Smartphones oder Tablets, was das IT-Budget selbstverständlich entsprechend entlastet und Ressourcen für andere Anschaffungen ermöglicht. Darüber hinaus kann das BYOD-Modell auch zu einer höheren Produktivität führen, weil die MitarbeiterInnen durch die Nutzung ihrer privaten Endgeräte nicht mehr an den Arbeitsplatz gebunden sind und ihrer Arbeit daher auch anderen Orten und außerhalb der Öffnungszeiten der Unternehmen nachgehen können. Schlussendlich sind MitarbeiterInnen auch motivierter, wenn sie ihre eigenen Geräte nutzen können und nicht an die vom Unternehmen vorgegebenen Endgeräte gebunden sind, die meist auch wesentlich „uncooler“ sind als die neuen, leistungsfähigen „Minicomputer“.

6.2 Technische Gefahren

Welche Gefahren für Unternehmen entstehen können ist derzeit noch nicht absehbar. Auf den ersten Blick jedoch gibt es verschieden Szenarien, die zu den Sicherheitsrisiken zählen. Etwa das Einschleusen von schadhafter Software (sogenannte Malware), das Einschleusen von Trojanern, die von den privaten Endgeräten ausgehen und vom Firmennetzwerk nicht sofort erkannt werden, der Diebstahl von Firmendaten – also von Betriebs- und Geschäftsgeheimnissen, aber auch ganz einfach der Datenverlust, weil Daten lokal auf den privaten Endgeräten gespeichert und damit nicht durch Backups gesichert werden. Ganz einfach können die genannten Gefahren auch dadurch auf die Geräte gelangen, indem eine App heruntergeladen wird, die im Hintergrund Einlass zu den eigenen Endgeräten zulässt – sprich die Nutzer wissen von dieser App nicht, dass sie „böse“ ist.

Wie können sich Unternehmen nunmehr vor eventuellen Gefahren schützen? Nun zum einen ganz sicher durch das Verbot des Verwendens privater Endgeräte für Firmenzwecke. Allerdings muss hier eingeräumt werden, dass das Unternehmen damit an Attraktivität für die MitarbeiterInnen verliert, da es kaum ein Unternehmen gibt, das seinen MitarbeiterInnen immer die neueste, „hippe“ Hardware zur Verfügung stellen kann.

Eine Möglichkeit besteht auch darin, die Nutzung privater Endgeräte zuzulassen, die Daten jedoch zentral in den Firmennetzen zu sichern und den Zugriff auf Firmendaten lediglich mittels VPN oder Client-Software

zuzulassen. Nachteilig ist dabei jedoch der hohe administrative Aufwand, da die privaten Endgeräte regelmäßig überprüft und gewartet werden müssten. Damit hier die Kosten überschaubar bleiben, gibt es bereits IT-Lösungen, die am Markt angeboten werden, etwa mittels eines eigenen USB-Sticks, der eine Software enthält, die dazu führt, dass das private Endgerät aus einer geschlossenen Umgebung heraus gestartet und damit auch eine sichere Verbindung zum Firmennetzwerk aufgebaut wird. Das bedeutet, dass das lokale Betriebssystem des privaten Endgerätes gar nicht zur Verwendung kommt. Damit soll einerseits der Angriff durch schadhafte Software ausgeschlossen und andererseits das Einschleusen von Malware verhindert werden. Damit der USB-Stick nicht durch „Nichtberechtigte“ verwendet wird, sollte er zudem durch ein Authentisierungssystem geschützt sein. Da normalerweise die gesamte erforderliche Software auf dem USB-Stick befindlich ist, benötigt es auch keine gesonderte Installation und/oder Konfiguration auf den privaten Endgeräten, wodurch viel Administrationsaufwand eingespart wird. Um sich als Unternehmen aber 100%ig abzusichern, sollten die Nutzer des USB-Sticks auch vertraglich dazu verpflichtet werden, das Authentisierungssystem nicht zu unterlaufen, in dem sie es etwa – aus Bequemlichkeit – deaktivieren.

Eine andere Variante könnte darin bestehen, einen verschlüsselten Container auf dem privaten Endgerät einzurichten, in dem dann die Regeln des Unternehmens gelten. Dies muss den MitarbeiterInnen auch mitgeteilt werden und diese verpflichten sich ihrerseits zur Einhaltung der unternehmensinternen Regeln, wobei die Nichteinhaltung mit Schadenersatzforderungen verbunden sein kann. Der auf dem privaten Endgerät eingerichtete Container wird dabei mit einer hohen Verschlüsselung und einem Passwort versehen. Damit sollen die Firmendaten sicher sein, da kein Austausch zwischen dem privaten und dem Unternehmensbereich möglich sein soll. Auch kann das Unternehmen von außen auf das Gerät zugreifen, wenn auch nur auf den Bereich des gesicherten Containers. Damit das Handling relativ einfach ist, muss der Nutzer des privaten Endgerätes im Vorfeld allerdings angeben, ob er das Gerät nun zu privaten oder zu dienstlichen Zwecken nutzt. Blackberry hat eine dementsprechende Variante entwickelt: Immer wenn der/die Mitarbeiter/in das Blackberry privat nutzen möchten, ist der berufliche Account zu schließen und es ist ein separates App zu öffnen, um das Blackberry privat zu nutzen. Dies wird von den

Nutzern als eher unhandlich und zeitraubend eingestuft. Ob es Erfolg haben wird, wird sich daher erst zeigen.

6.3 Arbeitsrechtliche Gefahren

Aus arbeitsrechtlicher Sicht besteht das Problem des BYOD darin, dass die Arbeit mit der Freizeit verschwimmt. Anrufe, Mails, aber auch Arbeit werden mit nach Hause genommen und es gibt keinen klar abgegrenzten Bereich mehr, in dem sich Mann oder Frau nicht mit dem Unternehmen, in dem sie tätig sind, beschäftigen. Dies insbesondere dann, wenn für private und berufliche Mails auch ein und derselbe E-Mail-Account verwendet wird. Das Abschalten fällt damit schwer und führt auch vermehrt zu Stress und psychischen Belastungen. Dies wiederum führt zu Erschöpfungszuständen (im Volksmund auch BURN OUT genannt), die nur mehr durch längere Auszeiten vom Unternehmen geheilt werden können und nicht nur den Unternehmen erhebliche Kosten verursachen. Eine Möglichkeit dem entgegenzuwirken besteht darin, unternehmensinterne Regeln aufzustellen, die es den Mitarbeiter/innen erlauben, zB auf Mails in angemessenen Zeiten zu reagieren. So hat etwa die Firma Anton Paar ein eigenes Regelwerk herausgegeben, das es Mitarbeiter/innen in der Arbeitswoche erlaubt auf E-Mails innerhalb einer Frist von 48 Stunden zu antworten. Dementsprechend gibt es keinen Stress ständig online zu sein. Der/die Mitarbeiter/in hat somit auch die Möglichkeit sich den Arbeitstag so einzuteilen, dass pro Tag ein gewisses Zeitfenster für die Bearbeitung von E-Mails reserviert wird.

Gibt es keine dementsprechenden Regelungen im Unternehmen, besteht – wie vorhin schon erwähnt – die Gefahr, dass der Arbeitgeber in einen Konflikt gerät, den er eigentlich nur verlieren kann: Auf Grund seiner Fürsorgepflicht hat er die Arbeit für seine Mitarbeiter/innen so einzuteilen, dass diese die Arbeit auch innerhalb ihrer Arbeitszeit schaffen können. Außerhalb der Arbeitszeit soll Freizeit und Zeit zur Erholung herrschen. Das heutzutage leider sehr beliebte Modell der „All-In-Verträge“ ist jedoch alles andere als ressourcenschonend, gehen doch manche Personalisten noch immer davon aus, dass dies bedeutet, dass nicht nur die 20 oder 25 Überstunden pro Monat „abzuarbeiten“ sind, sondern darüber hinaus auch weitere Überstunden, die eben nicht entlohnt werden. Dem spielt natürlich in die Hände, wenn privaten Endgeräte für berufliche Zwe-



Abbildung 6.1: Arbeitsplatz mit mobilen Devices

cke verwendet werden und somit eine Trennung zwischen Arbeit und Privat gar nicht mehr möglich ist. Dies wird dann spätestens im Urlaub klar, wenn „nur mal kurz“ die E-Mail angeschaut wird. Von Erholung und „arbeitsfreien“ Tagen bleibt dann meist nichts mehr – außer dass der Haussegen schief hängt. Daher sollte sich jedes Unternehmen bevor es private Devices zulässt, auch immer überlegen, wie es trotz der Zulässigkeit der Nutzung privater Endgeräte Arbeits- und Freizeit trennen kann. Am einfachsten wird dies wohl gelingen, wenn entsprechend lange Reaktionszeiten unternehmensintern festgelegt werden.

6.4 Support

Eine weitere Herausforderung, die sich durch das Thema BYOD stellt, ist der technische Support im Unternehmen. Gerade in KMU gibt es meist nur ein kleines, die firmeninterne EDV betreuendes Supportteam. Dieses wird kaum die Möglichkeit haben, sich alle Kenntnisse über alle privaten Devices zuzulegen und diese warten zu können.

Darüber hinaus sind private Devices oft nicht mit der Unternehmenssoftware verbunden und – handelt es sich um geschlossene Systeme, wie das iPad – es besteht keine technische Möglichkeit auf diese Geräte remo-



Abbildung 6.2: Support privater Devices - Ein Problem?

te zuzugreifen. Damit steht aber auch schon fest, was aus Kostengründen wohl unvermeidbar ist: Private Endgeräte können vom Unternehmen nicht gewartet werden, so dass auch eine softwaretechnische Kontrolle diese Geräte nicht möglich ist.

6.5 Rechtsrahmen

Eine der wichtigsten rechtlichen Bestimmungen im Zusammenhang mit BYOD ist sicherlich das Datenschutzgesetz. Denn nach § 14 DSGVO ist der Auftraggeber, also jene Person in deren Interesse Daten automatisationsunterstützt verarbeitet werden, verpflichtet angemessene Datensicherheitsmaßnahmen zu ergreifen. Somit ist der Auftraggeber verpflichtet, Daten vor Zerstörung und Verlust zu schützen und er muss sicherstellen, dass die Daten ordnungsgemäß verwendet werden und vor dem Zugriff Unbefugter geschützt sind. Gerade die Vermischung von privaten und betrieblichen Endgeräte führt jedoch dazu, dass gerade der oben erwähnte Sicherheitsaspekt kaum mehr erfüllt werden kann, da private Endgeräte regelmäßig nicht mit den unternehmenseigenen IT-Sicherheitsaspekten ausgestattet sind. Damit im Fall des Falles geklärt ist, wer nun welche Verantwortung zu tragen hat, empfiehlt sich vor Einsatz privater Endgeräte am Arbeitsplatz eine Konsultation der eigenen Rechtsabteilung bzw. des

Firmenanwaltes. Hier wäre zunächst zu klären, welche relevanten gesetzlichen Bestimmungen – neben dem Datenschutzgesetz – durch BYOD betroffen sein können. Weiters sollte dabei auch Blick über die Landesgrenzen hinaus geworfen werden, da in der Regel auch der Patriot Act der USA zu berücksichtigen sein wird, der es den Bundesbehörden der USA erlaubt, Kommunikation mitzuhören, zu speichern und zu verwenden. Relevant wird dieser, da eine Vielzahl der Endgeräte bzw. der darauf befindlichen Software aus den USA stammen und dort ein vollkommen anderer Rechtsrahmen zugrunde gelegt wird als er in Europa üblich ist. Auch sollte überlegt werden, wie betriebliche Daten im Falle des Verlusts oder Diebstahls des privaten Endgerätes geschützt werden können. Wichtig ist es hierbei die Mitarbeiter/innen eindringlich darauf hinzuweisen, dass jeder Verlust/Diebstahl unverzüglich zu melden ist. Denn nur bei einer raschen Meldung kann überhaupt auf die Gefahr des Datenverlustes reagiert werden. Insbesondere sollte der/die Mitarbeiter/in mittels schriftlicher Vereinbarung zur Löschung sämtlicher über das private Endgerät erreichbarer Daten verpflichtet werden, um sicherzugehen, dass firmenrelevante Daten nicht an Unberechtigte gelangen. Besser wäre es noch, wenn das Unternehmen selbst einen Fernzugriff auf das private Endgerät hätte und damit sicherstellen kann, dass die Löschung der Daten auch wirklich erfolgt ist. Naturgemäß mit dem Risiko, dass der Verlust/Diebstahl eventuell erst zu einem späteren Zeitpunkt gemeldet wird, da noch immer die Hoffnung auf ein „Wiederauffinden“ des Endgerätes besteht.

7 Der Weg zur eigenen App - Herausforderungen im Mobile App Development

Johannes Feiner, Elmar Krajnc, Franz Niederl

Zusammenfassung: *Bei der Entwicklung mobiler Apps stehen die EntwicklerInnen vor großen Herausforderungen. Sie möchten ihren KundInnen Applikationen in die Hand geben, welche diese gerne nutzen und weiterempfehlen. Für die Firmen soll durch die Apps ein Mehrwert entstehen, der sich einerseits durch gesteigerte Reputation wie auch finanziell in vergrößertem Gewinn zeigt. In diesem Artikel werden einzelne Schwierigkeiten bei der Softwareentwicklung herausgenommen und diskutiert. Im Fokus stehen dabei die User-Experience als Erfolgsfaktor, Vorteile durch menschenzentriertes Design und möglichst frühes Prototyping, mehrere Arten von Tests, der Mehrwert von Sensoren am Smartphone, Aspekte von Werbung in Social Media, die Dynamik bei der Entwicklung in agilen Teams und ein Ausblick auf kommende Entwicklungen des mobilen Sektors.*

7.1 Herausforderungen bei der Entwicklung mobiler Apps

In der heutigen Zeit werden immer mehr Applikationen auf Smartphones verwendet. Es wird prognostiziert, dass in Zukunft die mobile Nutzung die Verwendung eines PCs obsolet machen wird¹. Das bedeutet, dass der Markt und der Bedarf von mobiler Software in allen Bereichen anwachsen wird.

¹<http://searchengineland.com/analyst-mobile-to-overtake-pc-for-local-search-by-2015-119148>

Viele Firmen stehen derzeit vor der Herausforderung ihre Kunden mit mobilen Apps zu versorgen. Dieser Trend kommt aus dem privaten Bereich, wo für jede Kleinigkeit des täglichen Lebens eine App existiert (vgl. Maierhofer, 2014). Das Motto *There is an App for it*² geht auch an den österreichischen Klein- und Mittelbetrieben nicht spurlos vorbei und aktuell versuchen die Firmen ihre Produkte mit mobilen Apps zu ergänzen. Leider bleibt es oft beim Versuch, denn App-Entwicklung erfordert neben finanziellem Aufwand auch profundes Know-How auf vielen – möglicherweise für manche Firmen gänzlich neuen – Gebieten, wie von mobiler Softwareentwicklung für verschiedene Plattformen, von eCommer- ce, mCommerce³ und Social Media oder es fehlt ganz generell die Erfahrung im Markt mobiler Apps.

Im Folgenden werden einige Aspekte der mobilen Softwareentwicklung ausgewählt und diskutiert. Zu Beginn wird der Fokus auf die Kundin, den Kunden gelegt, welche hohe Erwartungen an die *User Experience*⁴ haben. Das heisst, die Nutzung einer App sollte nicht nur ein Problem schnell und zielgerichtet lösen, sondern auch noch Spaß machen. Mobile Apps sollten gerne und oft gestartet und im Idealfall nach erfolgreicher Benutzung vielleicht sogar an Freunde weiterempfohlen werden.

Der nächste Abschnitt betrachtet die Teams, welche Software entwickeln. Wie sollten diese zusammengesetzt sein und wie kann gemeinsam eine zyklische Entwicklung mit ständiger Verbesserung und Optimierung gelingen? Daraufhin werden auch die Schwierigkeiten und Möglichkeiten des Testens behandelt, insbesondere die verschärften Anforderungen bei Smartphones durch Sensoren und Aktoren. Oft ist eine einfache Simulation am Desktoprechner zu wenig; so sind zum Beispiel bei *location based services* eine Durchführung der Tests im Sinne des *Mobile Usability Testing* anhand Real-World-Szenarios mit mehreren Typen von Smartphones, instabilen mobilen Netzwerken und deaktiviertem GPS unerlässlich.

Schlussendlich wird noch ein kleiner Ausblick in die mögliche Zukunft

²*There's An App For That* war bereits beim Apple iPhone 3G Commercial 2009 das Leitthema. Siehe <https://www.youtube.com/watch?v=szrsfeyLzyg>.

³Aktuell wird ein Lehrgang zur Ausbildung von m-Commerce Fachkräften an der FH JO- ANNEUM in Kapfenberg mit internationalen Partnern im Rahmen eines EU Projektes entwickelt. Dieser Lehrgang soll die MitarbeiterInnen von KMUs befähigen, die aktuellen e-Commerce Aktivitäten in die mobile Welt zu bringen.

⁴Als einige der Kernelemente für User Experience nennt zum Beispiel Hartson und Pyla, 2012 die Aspekte *sability*, *usefulness* und *emotional impact*.

im Bereich der mobilen Apps gegeben. Mobile Bezahlsysteme, „Augmented Reality“ und „Embedded Computing“ sind aktuell noch Stichworte, welche aber durchaus schon bald Realität werden könnten.

7.2 Was schätzen die BenutzerInnen?

Am Beginn jedes Softwareprojektes geht man von funktionalen und nicht-funktionalen Anforderungen aus. Speziell auf mobilen Anwendungen ist die nicht-funktionale Anforderung einer guten Bedienbarkeit ein wichtiges Kriterium.

Im Sinne der Usability⁵ sollte der Fokus einer Apps insbesondere auf einer erfreulichen Nutzung liegen. Was heisst hier *erfreulich* und was bedeutet das für die EntwicklerInnen? Vorrangig wollen die NutzerInnen einer App ein vorhandenes Problem lösen. So sollte man bei der Fahrplan-App⁶ ein Ticket kaufen können oder bei der Park-App⁷ in Wien einen Parkschein lösen können. Aber neben dieser offensichtlichen, vordergründigen Grundbedingung spielen noch viele weitere Aspekte eine Rolle. Effektivität, also das tatsächliche Lösen des Problems wie zum Beispiel – ein Bahnticket zu kaufen – kann alleine zu wenig sein. Eine Lösung muss, damit eine App akzeptiert wird, auch effizient erfolgen. Müsste man sich für jeden Ticket-Kauf neu anmelden und die Kreditkartennummer und Adresse erneut eingeben, wäre die App zwar effektiv, aber nicht sonderlich effizient. Als Muss-Kriterien für Apps gelten vor allem a) erfolgreiches Erreichen einer Lösung in b) vernünftig kurzer Zeit ohne Anleitung, also c) selbsterklärend und d) konform bezüglich des jeweiligen Betriebssystems.

Am letzten Punkt scheitern (multi-platform) Applikationen manchmal schon, wenn zum Beispiel die Basis System Funktion *Zurück* nicht Android bzw. iOS spezifisch implementiert wurde. Vorhersehbares, gewohntes und daher erwartetes natives Verhalten bei der Bedienung der Applikation ist wichtiger Teil guter Usability. Die Bewertung von Benutzbarkeit erfolgt durch gut messbare Kriterien wie Zeitaufwand, Anzahl der Touches oder Anzahl der einzugebenen Touches pro Aufgabenstellung.

Weiters erwarten sich UserInnen, dass sie nach Download und Installa-

⁵Die ISO Norm Travis, 2010 definiert für Usability die Kriterien Effektivität, Effizienz und Zufriedenheit.

⁶Scotty: http://www.oebb.at/de/Reiseplanung/Fahrplanauskunft/Mobile_Dienste

⁷Parking in Vienna: <http://www.parkeninwien.at/en/app.html>

tion einer App für diesen Aufwand auch entsprechend *belohnt* werden. Die Benutzung soll Erwartungskonform sein und positive Empfindungen anregen. Somit werden neben freundlichem, grafischem Design, durchaus mal *sinnlose* Funktionalitäten⁸ wie das Teilen der zuletzt gelaufenen Strecke auf Facebook zum eigentlichen Erfolgsfaktor⁹.

Nach dieser Betrachtung der Anforderungen aus EndbenutzerInnen-sicht wird in den nächsten Abschnitten der Blickpunkt auf die Entwicklungsseite, sowie das Softwareentwicklungs- und Programmiererteam gelegt.

7.3 Wie entsteht eine mobile App?

Nach dem Überblick über die Komplexität und Diversität der Anforderungen an ein Softwareentwicklungsteam folgt nun eine kurze, exemplarische Beschreibung, welche Schritte während der App-Entwicklung nötig sind. Bei der ersten Idee für eine App hat man meist ein Problem, einen Anwendungsfall im Hinterkopf, welchen man für eine bestimmte Zielgruppe lösen muss. Günstig ist es, wenn ein Problem in Teilprobleme aufgeteilt werden kann, um danach diese kleineren Einzelprobleme gut – nein eigentlich ausserordentlich gut – zu lösen. Es ist empfehlenswert die Anzahl der Features für die Erstversionen zu minimieren und erweiterte Möglichkeiten oder Erweiterungen für später vorzumerken. Vielleicht stellt sich heraus, dass diese¹⁰ gar nicht nötig sind. Das Prinzip KISS – keep it simple and stupid – ist eine erfolgreiche Strategie in der Softwareentwicklungsbranche. Nun folgt die Eingrenzung der Zielgruppe, oder Zielgruppen. Empfehlenswert ist die Erarbeitung von Personas, das heisst, unterschiedliche Profile von exemplarischen NutzerInnen zu erstellen. Diese erfundenen PseudoanwenderInnen werden mit Namen, Alter, Hobbies und je nach Anwendungsfall mit weiteren wichtigen Eigenschaften versehen. Eine Persona alleine ist

⁸Vergleiche Social Media Sharing bei Runtastic <https://www.runtastic.com/en/apps/runtastic>.

⁹J. Nielsen und Budiu, 2013 zeigt im Global Snapshot des Consumer Reports, dass in den untersuchten Ländern zum Beispiel die Social-Networking-Nutzung am Smartphone in beinahe allen untersuchten Ländern weit über 70% beträgt, was die Integration ebensolcher Social Media Features in andere Apps nahelegt.

¹⁰Laut <http://de.wikipedia.org/wiki/Flickr> entstand Flickr als Zusatzfeature des Spieles *Game Neverending*, wobei nach dem großen Erfolg dieses Bildhochladen-Features das ursprüngliche Spiel überflüssig und somit eingestellt wurde.

zuwenig um unterschiedliche, divergente Nutzertypen abzudecken, mehr als sechs bis acht¹¹ sind aber wegen des Aufwandes ebenfalls nicht empfehlenswert. In Szenarios und Stories werden nun Anwendungsfälle beschrieben und am besten gleich mit Abnahmekriterien, also Kriterien wann ein Use-Case¹² durch die App geeignet abgebildet werden kann und die Entwicklung erfolgreich war, versehen. Im Sinne des User-Centred Design – siehe Krajnc, Feiner und Schmidt, 2010 über die Definition und den Einsatz von UCD vor allem mit Paperprototypen aber auch Wireframes und Mockups – sollte spätestens in dieser Phase der Entwicklung *echten* EndnutzerInnen Skizzen oder Zeichnungen der einzelnen Bildschirminhalte vorgelegt werden. Erste Test-BenutzerInnen¹³ simulieren die Benutzung der App durch Betrachten der geplanten Inhalte, der Screens, und navigieren von Zettel zu Zettel durch die App.

Die technische Umsetzung, die Implementierung (siehe auch Abbildung 7.1) ist für eine einzelne native App grundsätzlich sehr ähnlich konventioneller Software. Wie generell auch sonst in der Softwareentwicklung, sollte von Unit-Tests bis Continuous Integration schon früh auf Qualitätssicherung gesetzt werden. Verstärkt durch die globalisierten Marktgegebenheiten werden internationalisierte Apps¹⁴ benötigt, um in den internationalen Stores zu reüssieren.

¹¹Vergleiche auch die Studie von L. Nielsen und Hansen, 2014, wo von 13 UX DesignerInnen nur ein einziger mehr als acht Personas erstellte.

¹²Unter einem Use-Case versteht man eine technische Sicht auf Interaktionen von BenutzerInnen oder Systemen. Dabei beschreibt man diese Interaktion relativ formell mittels Titel, erwünschtem Ziel, Akteur, Bedingungen, sowie mögliche Wege ein genanntes Ziel zu erreichen. Währenddessen meint man mit einer User-Story eine Kurzbeschreibung einer Aktion aus Kundensicht wie zum Beispiel *Als Benutzer der Flickr Website möchte ich ein Album mit meiner Bekannten teilen, damit diese die Fotos meines Fotoalbums betrachten kann.*

¹³J. Nielsen, 1994 erläutert in seinem Artikel über *Guerrilla-HCI* das Kosten/Nutzen Verhältnis von Test-User und ermittelt das beste Verhältnis bei bereits cirka drei bis fünf Usern.

¹⁴Mehrsprachigkeit erfordert nicht nur einfache Wort für Wort Übersetzungen, was schwierig genug sein kann, sondern Anzeige in verschiedenen Währungen, unter Umständen auch verschiedene Symbole und Bilder, rechtlich angepasste Texte und Verkaufsbedingungen.



Abbildung 7.1: Typischer Arbeitsplatz für mobile Entwicklung.

7.4 Warum im Team entwickeln?

Obwohl es bei einfachen Apps durchaus möglich wäre diese gänzlich alleine zu entwickeln, beruht die kommerzielle Entwicklung von Software praktisch immer auf umfangreicher Teamarbeit. Für alle Arten von Applikationen ist da einerseits das Wissen über die *Domäne* wichtig. So kann zum Beispiel eine Sportlerin die Bedürfnisse an eine Mountainbike-Training-App besser nachvollziehen als ein Couchpotato. Andererseits, inkludiert die Umsetzung einer App dann oft auch ein umfangreiches Wissen über die Infrastruktur im Hintergrund. Eine *location based* Shopping App der Floristen Oberösterreichs wird möglicherweise ein Karten Service zur Visualisierung der Positionen der Shops sowie ein Routing-Service zur Berechnung des Anfahrtsweges benötigen. Dabei sind eine Kombination aus vorhandenem Web Service wie Google Maps¹⁵ und aus selbst implementiertem Business-Backend Service, welches Öffnungszeiten und Lagerbestand der Verkaufsstellen liefert, nötig. Zur Umsetzung kommen dann wei-

¹⁵Mehr über die Google-APIs für Kartendienste findet man auf <https://www.google.com/intx/de/work/mapsearch/>.

tere Anforderungen an die Softwareentwicklung – unter Umständen auf mehreren Plattformen wie Android, iOS und Windows Phone – zum Tragen. Kenntnis von verschiedenen Programmiersprachen¹⁶, umfangreichen herstellerabhängigen Frameworks und Entwicklungsumgebungen erzwingen quasi die Abwicklung im Team. Für audiovisuelle Umsetzung – bei Spielen ein Muss – sind dann weitere ExpertInnen mit grafischen und musikalischen Know-How gefragt. Zur Sicherstellung der Gesamtqualität und deren Kontrolle kommen dazu noch Security Aspekte – von Netzwerksicherheit bis Datenschutz – oder auch Accessibility (Krajnc, Feiner und Schmidt, 2010), Know-How und Usability-Tests. Diese werden wiederum von speziell ausgebildeten und erfahrenen Teammitgliedern beigetragen. Ein fächerübergreifendes, multi-domain Team und agile Entwicklungsmethoden ersetzen das veraltete, archetypische Bild des einsamen Softwarehackers.

7.5 Wann ist die Software fertig?

Softwareprojekte werden selten zeitgerecht fertiggestellt¹⁷, beziehungsweise ausgelieferte Software wird von den AnwenderInnen selten als *fertig* empfunden. Aber im Rahmen einer agilen Entwicklung können den KundInnen regelmäßig verbesserte und lauffähige Applikationen bereitgestellt werden. Die Endanwender können diese Applikationen testen und melden Probleme oder Fehler während der Anwendung an das Entwicklerteam zurück und bringen im Idealfall auch weitere Ideen und zusätzliche Wünsche ein. Dieses Feedback wird ausgewertet, priorisiert und für weitere Iterationen berücksichtigt. So entsteht Schritt für Schritt, Iteration für Iteration, eine den Ansprüchen der BenutzerInnen noch besser entsprechende Software auf der Basis dieser zyklische Entwicklung und auf der Basis des Einsatzes agiler Methoden.

¹⁶Android Applikationen werden auf Java Basis erstellt, iOS Applikationen mit XCode unter Swift oder Objective-C und Windows Phone Applikationen im Visual Studio mit XAML und C#. Cross-Platform-Development mit Phone Gap erfordert Know-How in HTML5 und weiteren Web-Technologien.

¹⁷Laut <http://www.projectsmart.co.uk/why-software-projects-fail.php> sprengen über 50% der Softwareprojekte den Kostenrahmen, haben über 200% Zeitüberziehung, wobei nur rund 60% der Funktionalität implementiert wurden.

7.5.1 Deployment und Marketing

Ein spezielles Thema bei mobilen Apps ist das Deployment, also der Prozess die mobile App auf das mobile Endgerät möglichst vieler Enduser zu bekommen. Web Applikationen benötigen dazu nur einen Link und können wie eine Website angesurft werden. Echt native Applikationen, aber auch Cross-Platform Entwicklungen¹⁸ werden durch die App-Stores vertrieben. Dies wiederum kann große Vorteile, aber auch einige Einschränkungen bedeuten. Einerseits vertrauen die EndnutzerInnen der Software aus den großen Stores von Apple und Microsoft eher, da diese Software vor der Bereitstellung restriktiven Tests und Qualitätskontrollen unterliegt. Andererseits werden Apps über den Play Store von Android nicht so rigide wie der teilweisen willkürlich anmutenden Apple/Microsoft Firmenpolitik¹⁹ entsprechend, vorselektiert oder gar zensiert. Auch unterliegen sie weniger willkürlichen und technischen Einschränkungen. Im Fall der Google Android Platform geben die Kommentare und Bewertungen von Apps den Endusern ein bißchen Sicherheit, ob eine App auch wirklich funktionsfähig oder nützlich ist, beziehungsweise keine Schadsoftware enthält.

Durch das Vorhandensein von hunderttausenden Apps ist das Marketing in den App-Stores in den letzten Jahren immer aufwendiger und umfassender geworden. Es ist erforderlich, dass spezielle Werbemaßnahmen, zum Beispiel via Social Media Kanäle, geplant und begleitend durchgeführt werden. Aufwands- und abrechnungstechnisch sind die Stores einerseits ein zentraler, leicht zu findender Ankerpunkt der potentielle KundInnen um mobile Applikationen downloaden, installieren und bezahlen zu können. Andererseits behalten sich die Anbieter dieser Services ca. 30% des Umsatzes²⁰ ein. Ein nicht geringer Teil des Umsatzes, welchen Apple und Co hier für die Bereitstellung dieser Store-Infrastruktur beanspruchen. So ist es nicht mehr einfach von der App Entwicklung zu leben oder gar reich zu werden, wie es in den Anfängen noch einigen Spieleentwick-

¹⁸Cross-Platform Entwicklung mit Frameworks wie Phonegap ermöglicht Apps (Corral u. a., 2011) in HTML5 und JavaScript zu erstellen, welche dann allerdings in eine native App eingebettet und als solche vertrieben wird. Man findet auf <http://cordova.apache.org> beziehungsweise auf <http://phonegap.com> sowohl das Open Source Framework, als auch die kommerzielle Adobe Enterprise Version.

¹⁹Man vergleiche die Entwicklerrichtlinien bei Apple: <https://developer.apple.com/app-store/review/guidelines/>.

²⁰You pick the price, you get 70% of sales revenue: <https://developer.apple.com/programs/ios/distribute.html>.

lern gelungen ist. Der Erfolg ist von der guten Idee abhängig, Ausnahmeerscheinungen wie Instagram, SnapChat, WhatsApp haben den Nerv der Community getroffen und mit wenigen MitarbeiterInnen einen überraschend hohen Marktwert erzielt.

Oft stellt eine mobile App nur das Gratis-Add-On für vorhandene Services und Produkte dar. Zum Beispiel stellen Pelletsöfenhersteller zusätzlich zur stationären Steuerung im Haus für die mobile Ofensteuerung²¹ der Heizung eine native, gratis App bereit.

7.5.2 Mobile-Usability-Testing

Zurück zur Qualität und zum Testen von mobilen Applikationen. Dieser Bereich wird immer noch etwas stiefmütterlich betrachtet und Entwicklerinnen stellen ungläubig und erstaunt die Unbedienbarkeit ihrer Apps oft erst dann fest, wenn die App bereits im Store zum Download verfügbar ist. Frühzeitiges Einbinden der EndbenutzerInnen wurde oben schon im Zusammenhang mit User-Stories und Personas beim Prototyping erwähnt. Echte Feldtests im Real-World Betrieb sind auch nötig. So sollten Usability-Evaluierungen, welche Schwachstellen in der Software aufzeigen, nicht mehr im klassischen Labor durchgeführt werden. Besser ist es, die Software während tatsächlicher Verwendung zu testen, das heisst, während man unterwegs ist, am Fahrrad, an der Bushaltestelle, im Auto. Dieser speziellen Form des Usability-Testings, das *mobile usability testing* und -reporting (Feiner und Andrews, 2012) kommt noch größere Bedeutung zu, wenn es um Applikationen geht, welche Sensoren und Akteure nutzen. Eine Routing-App, ein Schüttel-Sensor, ein Bluetooth-Device oder kontaktloses Bezahlen am POS (Point of Sales) kann nur unzureichend simuliert werden. Eine der Möglichkeiten ist ein Thinking-Aloud-Test, wo Benutzerinnen während sie mit der Software einzelne Aufgaben durchführen, laut mitsprechen und auch ihre aktuellen Gedanken formulieren. Zugleich werden sie für eine spätere Analyse bei der Interaktion mit der App gefilmt und auch die Aktivitäten am Bildschirm werden mitgeschnitten²².

²¹ÖkoFen myPelletronic App für iOS und Android <http://touchdemo.oekofen.info>.

²²Im Rahmen der Community Konferenz UX Day Graz entstand folgendes Video über Mobile Usability Testing an der FH JOANNEUM <https://www.youtube.com/watch?v=d8REps9pww0>.

7.6 Wer übernimmt Pflege und Wartung?

Die Entwicklung für Software ist, wie im vorigen Abschnitt dargelegt, keine triviale Aufgabe. Dazu kommt nun auch noch die Weiterentwicklung und die nötigen Updates für Software. Im mobilen Bereich sind Updates häufiger nötig. Beispielsweise müssen durch Security-Updates die entwickelten Applikationen upgedated und verbessert werden, wobei gerade bei internationaler Verbreitung eventuelle Schwachstellen von BetrügerInnen schnell ausgenutzt werden könnten. Durch das always-online, also das ständige Verbundensein mit dem Internet, ist die Gefährdung weiters erhöht. Auch erwarten sich UserInnen regelmäßige Updates, um das Gefühl zu haben die Software wird weiter entwickelt und gepflegt. Im mCommerce und im eMarketing Bereich bedarf es weiters einer ständigen Präsenz in den Social Media Kanälen. Der Segen des Marketings via Facebook wird schnell zum Fluch von Social Media, wenn der erste Shitstorm auf die Firma einprasselt.

7.7 Native App oder Web-App?

Eine wichtige Entscheidung bei der Entwicklung von mobilen Applikationen ist, ob man eine Native-App oder eine Web-App entwickelt. Wie in Abschnitt 7.3 ausführlich behandelt wird, gehören dazu die Kriterien wie der Entwicklungsprozess, der benötigten Gerätesensoren, der Performance, der Offline-Verfügbarkeit, der Distribution, der Updates und der Wirtschaftlichkeit. Tabelle 7.1 von Geymayer, 2014 zeigt eine Gegenüberstellung anhand dieser Kriterien. Die Stärken einer Native-App liegen im Zugriff auf die Sensoren des Gerätes, der Performance, der Offline Verfügbarkeit und der Distribution.

Nachteile hingegen gibt es bei der Entwicklung und bei der Aktualisierung von Apps. Die Wirtschaftlichkeit ist abhängig davon, ob die App für eine Plattform (iOS, Android, etc) entwickelt wird oder für viele. Für welche Art man sich entscheidet, ist zu guter Letzt vom Problem der Zielgruppe abhängig.

7.8 Was ist an mobiler Entwicklung so schwierig?

Zusammenfassend nochmals die Herausforderungen an mobiler App Entwicklung, welche diese so anders und auch schwieriger im Vergleich zu

	Native	Web
Entwicklungsprozess	★★★★(★)	★★★★★
Device-API	★★★★★★	★★★★(★★)
User-Interface	★★★★★	★★★★★
Performance	★★★★★★	★★★★
Offline-Verfügbarkeit	★★★★★★	★★★★★(★)
Distribution	★★★★★★	★★(★★★★)
Updates	★★★★★	★★★★★★
Wirtschaftlichkeit	★★★★★(★)	★★★★(★★)

Tabelle 7.1: Gegenüberstellung der Stärken und Schwächen von Native- und Web-Applikationen.

herkömmlicher Desktop Software machen. *Multiplatform*: Native App-Entwicklung ist für die verschiedenen Betriebssystem Plattformen unterschiedlich und erfordert beachtlichen Aufwand, was am Beispiel Windows Phone und Apple iOS Entwicklung leicht nachvollzogen werden kann²³. *Sensors*: von Apps wird erwartet, dass diese die internen Services sowie die Aktoren und Sensoren des Smartphones wo nur immer möglich auch wirklich ausnutzen um auf das Adressbuch, die Kamera(s), den Helligkeitssensor, die Tonausgabe, Vibration, GPS-Signal, NFC-Funk und mehr zu verwenden. Dieser Kontext erleichtert und vereinfacht die Benutzung, weil manch nötige Daten – zum Beispiel der aktuelle Ort – nicht mehr extra vom User eingegeben werden müssen, sondern bereits über GPS automatisch bereitstehen. *Social media*: Einerseits sollten Apps erlauben sich mit Facebook oder Google+ Accounts anzumelden, Daten über Twitter zu sharen und andererseits müssen im Sinne der Werbung und Vermarktung die Register von Social Media Marketing gezogen werden. *Mobile testing*: Es ist sicher zu stellen, dass eine App den Anforderungen für kurzzeitige Benutzung während des Wartens auf den Bus, die möglichst ungestörte Weiternutzung –auch wenn ein Telefongespräch angenommen wurde – aber auch einer einfachen Bedienung während des Radfahrens mit *One Eyeball*, *One Thumb*²⁴, also quasi mit nur einem Auge und dem Daumen genügt. *Web services*: Für vorhandene Services – Wetter-, Übersetzungs-,

²³Vgl. auch den generischen Ansatz bei der App Entwicklung (Krajnc, Zugaj u. a., 2014)

²⁴Luke Wroblewski beschreibt auf <http://www.lukew.com/ff/entry.asp?1664> dass man bei Apps auch die mobile Anwendung berücksichtigen soll.

Karten- und Routingdienste – wie auch als Schnittstelle zu den firmeneigenen Systemen – Lagerstand und Bestellabwicklung – müssen Dienste angeboten werden. Dabei sind meist bereits vorhandene Backendsysteme mit zusätzlich neuen cloud basierten Diensten²⁵ zu verknüpfen.

7.9 Was bringt die Zukunft?

Vergleicht man die technologische Entwicklung bei Autos mit den von mobile devices, so stehen der Gesellschaft noch umwälzende Veränderungen bevor. Der herkömmliche Desktop-PC ist jetzt bereits in manchen Teilbereichen unwichtig oder gänzlich überflüssig geworden. Genauso wie die Betriebssystemplattformen ihre Relevanz verloren haben und für die EndnutzerInnen de facto der Browser das Betriebssystem darstellt. Globale Nutzung, internationale Vernetzung, mobile EndanwenderInnen – ausgestattet mit mehr und mehr Sensoren auf den flexible display Smartphones (Mone, 2013; Ra u. a., 2012) – geben die Richtung vor. Augmented Reality, wie Live Einblendungen von Informationen via Google Glass werden alltäglich. Das Bargeld wird durch *mobile payment* abgelöst und auch der klassische Einkauf in der Boutique kann fallweise schon von Handy-Apps, welche die eigenen Körpermaße kennen und die Schuhe, das T-Shirt an einem virtuellen Modell, dem Robotic-Mannequin (Shein, 2014), live visualisieren, unterstützt werden.

Die Anzahl der angebotenen mobilen Lösungen für ähnliche Probleme, wie Applikationen zur Navigation, wird weiterhin stark steigen. Dadurch werden sich erfolgreiche Apps mittels Originalität beziehungsweise optimierter User Experience abheben müssen. Entwicklungsteams erfordern daher erhöhte Interdisziplinärität um diesen Anforderungen gerecht zu werden, so sollten in zukünftigen Teams neben den SoftwareentwicklerInnen auch Domainexperten sowie WerbetexterInnen oder PsychologInnen vertreten sein.

Damit kleinere Firmen die Kosten einer Entwicklung nativer mobiler Applikationen für die unterschiedlichen Plattformen aufbringen können,

²⁵Man denke hier zum Beispiel an *Push Notifications* <https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/ApplePushService.html>, wo die User von Statusänderungen am Server über ein kleines Icon oder über System Nachrichten direkt am Handy informiert werden, auch ohne die entsprechende App gestartet zu haben.

werden zukünftige Systeme im Baukastenprinzip design- und konfigurierbar sein. So werden nicht mehr einzelne Interface Komponenten wie ein Button angeordnet, sondern umfassendere Funktionalitäten, wie eine Liste inklusive Daten, Detailansichten und Interaktion assembliert und danach im Sinne des Cross-Platform Developments für Endgeräte exportiert. Die Entwicklungszeiten und -Kosten werden geringer und jede Firma kann – vergleichbar mit einer eigenen Website – auch eine eigne App anbieten. Die Zukunft stellt uns vor spannende Herausforderungen!

Literaturverzeichnis

- Corral, Luis, Alberto Sillitti, Giancarlo Succi, Alessandro Garibbo und Paolo Ramella (2011). „Evolution of Mobile Software Development from Platform-Specific to Web-Based Multiplatform Paradigm“. In: *Proc 10th SIGPLAN Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*. Portland, Oregon, USA: ACM, S. 181–183. ISBN: 978-1-4503-0941-7. DOI: 10.1145/2048237.2157457.
- Feiner, Johannes und Keith Andrews (2012). „Usability Reporting with UsabML“. In: *Proc 4th International Conference on Human-Centered Software Engineering (HCSE 2012)*. Hrsg. von Marco Winckler, Peter Forbrig und Regina Bernhaupt. Bd. 7623. Springer Berlin / Heidelberg, S. 342–351. ISBN: 3642343465. DOI: 10.1007/978-3-642-34347-6_26.
- Geymayer, Daniel (2014). *Mobile Apps – Native vs. Web*. Bachelorarbeit. FH JOANNEUM, Internettechnik.
- Hartson, Rex und Pardha S. Pyla (2012). *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*. Elsevier. ISBN: 9780123852410.
- Krajnc, Elmar, Johannes Feiner und Stefan Schmidt (2010). „User Centred Design Interaction Design for Mobile Application Focused on Visually Impaired and Blind People“. In: *Proc. 6th International Conference on HCI in Work and Learning, Life and Leisure: Workgroup Human-Computer Interaction and Usability Engineering (USAB 2010)*. Hrsg. von Gerhard Leitner, Martin Hitz und Andreas Holzinger. Bd. 6389. Springer, S. 195–202. ISBN: 3642166067. DOI: 10.1007/978-3-642-16607-5_12.
- Krajnc, Elmar, Wilhelm Zugaj, Mathias Knoll und Johannes Feiner (2014). „Generic Mobile Applications for Small and Medium Enterprises“. In: *International Scientific Conference UNITECH'2014*. Bd. II. Technical University of Gabrovo, S. 363–367.

- Maierhofer, Michaela (2014). *Evaluierung von mobilen Helper Tools für KMUs*. Bachelorarbeit. FH JOANNEUM, Internettechnik.
- Mone, Gregory (2013). „The Future is Flexible Displays“. In: *Commun. ACM* 56.6, S. 16–17. DOI: 10.1145/2461256.2461263.
- Nielsen, Jakob (1994). *Guerrilla HCI – Using Discount Usability Engineering to Penetrate the Intimidation Barrier*. URL: <http://www.nngroup.com/articles/guerrilla-hci/>.
- Nielsen, Jakob und Raluca Budiu (2013). *Mobile Usability: Für iPhone, iPad, Android, Kindle*. 1. mitp, S. 236. ISBN: 978-3826695032.
- Nielsen, Lene und Kira Storgaard Hansen (2014). „Personas is Applicable – A Study on the Use of Personas in Denmark“. In: *Proc SIGCHI Conference on Human Factors in Computing Systems CHI2014*. Toronto, Ontario, Canada: ACM, S. 1665–1674. ISBN: 978-1-4503-2473-1. DOI: 10.1145/2556288.2557080.
- Ra, Moo-Ryong, Bin Liu, Tom F. La Porta und Ramesh Govindan (2012). „Medusa: A Programming Framework for Crowd-sensing Applications“. In: *Proc 10th International Conference on Mobile Systems, Applications, and Services*. Low Wood Bay, Lake District, UK: ACM, S. 337–350. ISBN: 978-1-4503-1301-8. DOI: 10.1145/2307636.2307668.
- Shein, Esther (2014). „Computing What Fits“. In: *Commun. ACM* 57.11, S. 16–19. DOI: 10.1145/2667113.
- Travis, David (2010). *ISO 9241 Part 11: Guidance on Usability*. Userfocus. URL: <http://www.userfocus.co.uk/resources/iso9241/part11.html> (besucht am 12.09.2010).

8 Cross-Platform-App-Development

Marvin Puchmüller

8.1 Was ist Cross-Platform-App-Development?

Grundsätzlich können App Entwickler für jede mobile Plattform eine sogenannte „native“ App entwickeln, was bedeutet, dass man für jede Plattform, die man unterstützen will eine eigene App entwickeln müsste, weil alle mobilen Plattformen mit verschiedenen Programmiersprachen und auch verschiedenen Systemarchitekturen arbeiten. Die Disziplin der Cross-Platform-Entwicklung beschäftigt sich damit denselben Code auf verschiedenen Plattformen nutzen zu können und so dieselbe App in den verschiedenen Stores anbieten zu können.

Diese Technik bringt den Entwicklern den Vorteil, dass sie nicht alle Plattformen und auch nicht die nativen Programmiersprachen sämtlicher Plattformen beherrschen müssen, sondern nur das Cross-Platform-Tool das eingesetzt wird. Durch diesen Vorteil sinkt nicht nur die Entwicklungszeit von Apps, die auf verschiedenen Plattformen laufen sollen sehr stark, sondern auch der Wartungsaufwand und die Einarbeitungszeit neuer Entwickler. Für Klein- und Mittelunternehmen entsteht dadurch die Möglichkeit ihre Unternehmensapplikationen mit geringeren Kosten entwickeln zu lassen und sie trotzdem für eine breitere Masse zugänglich zu machen.

8.2 Welche Plattformen müssen berücksichtigt werden?

Wenn man heutzutage von mobilen Plattformen spricht gibt es eigentlich nicht sehr viele große Plattformen. Die folgende Grafik¹ (siehe Abbildung 8.1) zeigt den weltweiten Marktanteil der jeweiligen Plattformen:

¹<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>

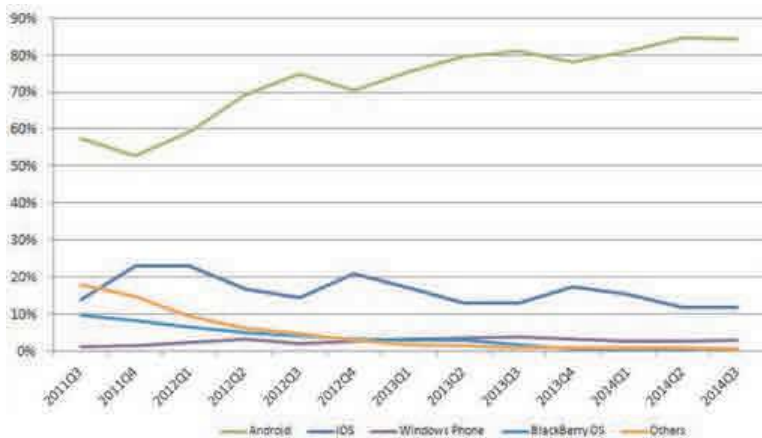


Abbildung 8.1: Worldwide Smartphone OS Market Share (Share in Unit Shipments)

Wie man aus dieser Grafik entnehmen kann gibt es nur zwei wirklich große Plattformen: Google Android und Apple iOS. Auch Microsoft Windows Phone sollte nicht vernachlässigt werden, weltweit liegt es zwar unter 5% Marktanteil, in Österreich ist es allerdings mit ca. 7% Marktanteil² durchaus beliebter.

8.3 Möglichkeiten Cross-Plattform-Apps zu entwickeln

Damit derselbe Code der Entwickler auf mehreren Plattformen funktioniert, muss eine von zwei Strategien ausgewählt werden: Entweder native Code-Generierung oder eine Komponente auf den jeweiligen Plattformen finden, die alle gemeinsam haben und den Code ausführen kann.

Die native Code-Generierung ist vom Konzept her der einfachere Weg. Hierbei ist die Grundidee, dass der von den Entwicklern erzeugte gemeinsame Code für jede Plattform als Input für das Generierungstool dient, dieses Tool den Code in nativen Code umwandelt, der von der jeweiligen Runtime der Plattform verstanden wird. Somit hat man eine native App entwickelt, die auf dem Gerät auch nicht von einer „normalen“ App unterschieden werden kann. Diese Technik funktioniert allerdings nur unter bestimmten Umständen, da jede Transformation oder Generierung von

²<http://www.format.at/technik/mobile/microsoft-windows-10-phone-5325093>

Code ein sehr generischer Ansatz ist, der nicht optimalen Code erzeugt. Fast immer wirkt sich das auf die Größe der App aus, da generierter Code oft sehr aufgeblasen ist. Auch die Performance der Applikation wird nie gut sein, wie bei einer wirklich nativen Applikation. Ein weiteres Problem ist, dass diese Technik sehr bald an ihre Grenzen stößt, da man auf die Funktionalität des Code-Generators angewiesen ist. Kann dieser z.B. einen bestimmten Geräte-Sensor wie z.B. den GPS-Sensor im Gerät nicht auslesen, dann können die Entwickler diesen auch nicht in der App benutzen. Möchte man den Sensor dennoch auslesen, müssten die Entwickler wiederum nativ darauf zugreifen, was wiederum dem Prinzip der Cross Platform Entwicklung widerspricht.

Die zweite Strategie, die sich mit dem Benutzen einer Gemeinsamkeit der verschiedenen Plattformen beschäftigt, ist da durchaus vielversprechender. Grundsätzlich geht es darum etwas zu finden, das Code ausführen darf und auf allen Geräten der verschiedenen Plattformen bereits installiert ist. Eine solche gemeinsame Komponente ist der Web Browser. Zwar ist auf den verschiedenen Plattformen nicht derselbe Web Browser vorhanden, dennoch sind sie alle in der Lage HTML darzustellen, dieses mit CSS zu verschönern und vor allem können sie alle JavaScript Code ausführen. Diese Idee ist nicht neu, sogenannte „Web-Apps“ benutzen diese bereits seit vielen Jahren. Der große Unterschied zu Web-Apps ist allerdings der, dass Web-Apps nur über den Web-Browser aufgerufen werden können, wenn eine aktive Internetverbindung besteht, und die App nicht auf dem lokalen Gerät verfügbar ist, sondern alles am Web-Server gespeichert wird. Bei der lokalen Benützung des Web-Browsers ist die Cross-Platform-App wirklich als App auf dem Gerät vorhanden und kann ohne Internetverbindung ausgeführt werden. Allerdings gibt es hier den schwerwiegenden Nachteil, dass hier nur der Web Browser benutzt werden kann und somit keine Geräte-Sensoren oder z.B. die gespeicherten Kontakte etc. in der App benützt werden können. Man könnte also nur Webseiten anzeigen, die eben lokal gespeichert sind. Daher gibt es für diese Strategie eine Weiterentwicklung, die sogenannten „Hybrid-Apps“.

Hybrid-Apps funktionieren im Grunde wie die Browser-Apps, können aber auf Geräteinformationen zugreifen. Sie bieten somit die Vorteile der Code-Generierungs Apps zusammen mit der lokalen Darstellung von Webseiten im Gerätebrowser.

8.4 Funktionsweise hybrider Apps

Hybride Apps bestehen aus zwei Komponenten, einem nativen „Mantel“ und dem Web-Browser. Am Gerät hat man somit eine echte App mit einem Icon und beim Starten der App wird auch wirklich eine eigene Applikation gestartet. In dieser Applikation wird dann der Web-Browser gestartet und in dieser Instanz des Web-Browsers werden dann die lokalen HTML Seiten mit dem dazugehörigen CSS und dem JavaScript Code ausgeführt. Das Herzstück einer solchen Applikation aus der Sicht der Entwickler ist hierbei der JavaScript Code. Jeder Web-Browser kann JavaScript ausführen, daher bietet es sich an, den Hauptteil der Applikation mit JavaScript zu erzeugen. Allerdings kann man mit JavaScript allein nicht auf die Gerätesensoren oder andere Informationen zugreifen. Hierfür wird dann der native „Mantel“ ausgenutzt. Da es sich bei Hybrid-Apps schließlich um eine eigene Applikation handelt, hat dieser native Teil der Hybrid-App ebenso Zugriff auf alle vom Benutzer erlaubten Informationen. Hierbei ergeben sich wiederum zwei Probleme, erstens muss man wieder nativen Code entwickeln um auf diese Geräte-Spezifika zugreifen zu können, und zweitens muss man irgendwie wieder die Verbindung zum Web-Browser und dem darin laufenden JavaScript Code herstellen. Das erste Problem kann nicht einfach gelöst werden, man muss tatsächlich nativen Geräte-Code entwickeln für jede Plattform, auf der die App laufen soll. Allerdings können fast alle Fälle bereits durch die Cross-Platform-Frameworks, die es gibt, abgedeckt werden, es ist nur sehr selten notwendig, selbst ein solches „Plugin“ zu entwickeln. Das zweite Problem, nämlich die Informationen der Sensoren an das JavaScript zurückzugeben, ist wesentlich einfacher zu lösen. Jede Plattform, die einen Web-Browser besitzt, hat auch eine Schnittstelle zwischen nativem Code und dem Web-Browser. Wenn man nun auf z.B. den GPS-Sensor zugreifen möchte, muss man im JavaScript Code den nativen Wrapper ansteuern. Dieser greift dann auf den Sensor zu, liest die Daten aus, und liefert die Daten dann an das JavaScript zurück.

8.5 Herausforderungen bei der Entwicklung und beim Testen hybrider Apps

Die Entwicklung solcher Hybrid Apps sollte mit Hilfe von dazu geeigneten Tools und Frameworks durchgeführt werden, da es eine komplexe Aufgabe ist, die Brücke zwischen nativem Code und dem JavaScript Code

im Webbrowser richtig aufzubauen. Es gibt zwar einige Frameworks, die diese Aufgabe übernehmen, jedoch sind nur zwei davon wirklich hervorzuheben, Apache Cordova³ (auch als PhoneGap⁴ bekannt) und Appcelerator Titanium⁵. Beide Frameworks sind in der Lage, hybride Apps zu erzeugen, wobei Cordova den direktesten Zugang für Entwickler bietet, da man den Application-Code wirklich mit HTML, CSS und JavaScript so erzeugt, wie er dann auch auf den Geräten ausgeführt wird. Titanium geht da einen anderen Weg, nämlich durch Einsatz eines XML-basiertem Markups und des eigenen Alloy⁶-Frameworks. Dieses Framework kann zwar nicht so direkt getestet werden wie das „reine“ JavaScript des Cordova Frameworks, jedoch bietet es sehr viele API-Funktionen an, die den Entwicklern sehr viel Arbeit im Vorhinein abnehmen und gleichzeitig durch den Einsatz von getesteter Software die Qualität der App erhöhen. Eine objektive Reihung welcher der beiden Frameworks der bessere ist, kann nicht erstellt werden, beide können hybride Apps erzeugen und jede hat eigene Vor- und Nachteile. Es gibt zwar noch weitere Frameworks für hybride Apps, wie u.a. Oracle ADF Mobile⁷ oder IBM Worklight⁸, allerdings benutzen diese Frameworks im Hintergrund wieder Cordova als finales Tool zur Erzeugung der hybriden App, daher kann man davon ausgehen, dass Cordova in mehr Apps genutzt wird, als Titanium.

Der hybride Ansatz zur Entwicklung von Cross-Platform Apps bietet neben den offensichtlichen Vorteilen wie der Zeitersparnis bei der Entwicklung und Wartung auch noch zwei weitere Vorzüge, die nicht außer Acht gelassen werden sollten. Der erste Pluspunkt ist, dass man den Großteil des Codes mit Hilfe von Desktop-Browsern implementieren bzw. testen kann, da diese natürlich auch mit JavaScript umgehen können. Dies gilt zwar „nur“ für Code der nicht auf diverse Sensoren oder andere gerätespezifische Schnittstellen zugreifen möchte, allerdings machen solche Zugriffe in den meisten Applikationen eher einen Bruchteil aus, der Hauptteil besteht u.a. aus dem Anzeigen dieser Informationen, Abfragen

³<http://cordova.apache.org/>

⁴<http://phonegap.com/>

⁵<http://www.appcelerator.com/titanium/>

⁶http://docs.appcelerator.com/titanium/3.0/#!/guide/Alloy_Framework

⁷<http://www.oracle.com/technetwork/developer-tools/adf-mobile/overview/adfmobile-1917693.html>

⁸<http://www.ibm.com/developerworks/downloads/ws/worklight/>

von User-Eingaben und Warten auf Interaktionen. Zusätzlich ist es auch möglich, gewisse Geräte-Zugriffe im Browser nachzustellen bzw. zu *mocken*⁹. Sollte es dennoch notwendig sein zu testen oder Fehler darin zu beheben, kann man dies mit Hilfe von echten Geräten und den dazugehörigen Desktop-Browsern ebenfalls durchführen. Um beispielsweise eine hybride App auf einem Android-Gerät zu testen, muss man die App nur starten, das Gerät an den Entwicklungscomputer anschließen und auf diesem Computer mittels Google Chrome auf den Browser des Gerätes zugreifen und schon kann man die hybride App testen bzw. verbessern und kann es direkt am Gerät ausprobieren. Dieselbe Funktionalität bietet Apple mit dem iPhone bzw. iPad und dem Desktop-Safari Browser an. Microsoft Windows Phone allerdings bietet diese Funktionalität nicht an. Generell muss bei hybriden Apps die im Web-Browser ausgeführt werden darauf hingewiesen werden, dass Microsoft eigener Web-Browser Internet-Explorer¹⁰ bei weitem nicht mit seinen Konkurrenten Google Chrome, Apple Safari oder Mozilla Firefox mithalten kann wenn es um den Einsatz von JavaScript geht. Sehr viele Web-Browser APIs oder auch JavaScript Libraries können im Internet-Explorer zu Problemen führen, da dieser vieles noch nicht implementiert hat und auch die *JavaScript VM*¹¹ nicht auf demselben Stand ist wie die der Konkurrenten. Wenn man die eingesetzten Web-Browser vergleicht, wie es z.B. von Diensten wie *CanIUse*¹² gemacht wird, kann man sehr schnell feststellen, dass der Internet-Explorer nicht der optimale Web-Browser ist und das man bei hybriden Apps, die auf Android bzw. iOS fehlerfrei laufen, im Windows Phone häufig auf Probleme stossen wird. Es ist allerdings mit keinem der Frameworks und auf keiner der Plattformen möglich auszusuchen, welchen Browser man einsetzen möchte, da dies von Betriebssystem der Plattform vorgegeben ist und nicht verändert werden kann. Ein ähnliches Problem hat man beim Einsatz der Apps auf älteren Android Geräten, die noch nicht den Chrome als Standard Browser eingesetzt hatten. Diese Geräte benutzen den für Android entwickelten Stock-Browser der nie für den Einsatz von hybriden Apps gedacht war und deshalb auch keine Tools zum Testen bzw. Debuggen am Gerät

⁹Mock-Objekte (eng. „to mock“ – „etwas vortäuschen“) sind Objekte, die als Ersatz für echte Objekte beim Testen dienen, wenn diese nicht verfügbar sind

¹⁰<http://www.microsoft.com/de-at/download/internet-explorer.aspx>

¹¹Die JavaScript Virtual Machine ist ein Programm im Web-Browser, dass dafür zuständig ist, JavaScript Code auszuführen und diesen davor zu optimieren.

¹²<http://caniuse.com/>

zur Verfügung stellt. Man kann sich aber auch beim Stock-Browser bzw. beim Internet-Explorer mit dem Einsatz von z.B. WeinRe¹³ oder anderen Remote-Debugging-Tools behelfen um zumindest schwerwiegende Fehler zu finden und zu beheben. Diese Tools sind, wie zu erwarten, bei weitem nicht so gut integriert und daher auch weniger gut geeignet als die Tools für Chrome oder Safari.

Wie bereits angedeutet kann es bei der Entwicklung von hybriden Apps zu interessanten Fehlern kommen, die bei nativen Apps nicht aufgetreten wären, oder nur auf einer Plattform oder sogar nur auf einem speziellen Testgerät festzustellen sind. Klarerweise liegt hier das Hauptproblem darin, dass die verschiedenen Plattformen unterschiedliche Betriebssysteme haben und daher eventuell denselben Code anders ausführen oder bestimmte Berechtigungen etc. nicht für die App zur Verfügung stellen. Außerdem haben alle Plattformen eigene Browser, die wiederum eigene JavaScript VMs besitzen und verschiedene APIs anbieten. Auch die Betriebssystem-Version bzw. die Version des Browsers kann zu neuen Problemen führen bzw. alte Probleme lösen. Bei nativen Applikationen hat man speziell die Web-Browser- und JavaScript Komplikationen nicht, da sie ohne den Web-Browser funktionieren und direkt mit der plattformspezifischen Programmiersprache umgesetzt werden. Dies sollte garantieren, dass dieselbe App auf verschiedenen Geräten der selben Plattform funktioniert. Es kann zwar vorkommen, dass sich gewisse APIs mit der Version ändern, allerdings sind diese Änderungen in der Regel sehr gut dokumentiert und können sehr schnell nachgezogen werden. Wenn man diese native Variante mit hybriden Apps vergleicht, sieht das ganze schon erheblich schlimmer aus, da man auf verschiedenen Ebenen auf Inkompatibilitäten stoßen kann. Es kann z.B. schon ein Problem sein, dass derselbe hybride Code auf einer Plattform funktioniert und auf der anderen nicht. Dies kann an dem Berechtigungsschema der Plattformen liegen, die zwar bei allen sehr ähnlich sind, aber nicht gleich. Es kann auch vorkommen, dass eine neue Version des Betriebssystems zu neuen Fehlern führt, weil sich die API des Betriebssystems geändert hat, oder weil die Version des Frameworks zu alt bzw. zu neu sein kann. Auch der Web-Browser kann in neuen Versionen zu Fehlern führen, oder aber auch etwaige JavaScript Libraries die im Einsatz sind.

Trotz der Vielzahl an potenziellen Fehlerquellen sollte man sich

¹³<http://people.apache.org/~pmuellr/weinre-docs/latest/>

dennoch nicht davon abschrecken lassen, da Fehler aufgrund von Betriebssystem-Updates oder neuen Browser Versionen sehr selten und meist auch in kurzer Zeit behebbar sind. Außerdem sind die Frameworks wie Cordova oder Titanium immer auf einem aktuellen Stand und lösen viele dieser Probleme bereits für die Entwickler.

8.6 Aufbau einer App – HTML, CSS und JavaScript

Da man bei hybriden Apps hauptsächlich den Web-Browser als „Unterlage“ benutzt, sollte man auch dessen Funktionen für die Entwicklung der Apps nutzen. So kann man den Aufbau einer *View*¹⁴ statt mit komplexen XML Aufbau wie z.B. bei Android und Interface-Buildern bei Apple mit einfachem HTML erzeugen, wie es bei jeder Webseite eingesetzt wird. Es gibt sehr viele Tools die das Entwickeln von HTML erleichtern, von einfachen Texteditoren über spezielle Entwicklungsumgebungen bis hin zu grafischen Editoren, die mittels Drag&Drop neue Objekte hinzufügen. Einer der großen Vorteile von HTML als Auszeichnungssprache ist, dass es auf allen gängigen Web-Browsern fast identisch verstanden wird. Damit die App auch gut aussehen und die User Experience (vgl. Garrett, 2011) gesteigert werden kann, kann man diese mit HTML erzeugte Struktur mithilfe von CSS verschönern. Wie auch HTML wird CSS von jedem Web-Browser unterschiedlich interpretiert. CSS, das für einen Web-Browser speziell erzeugt wurde, kann auf anderen Web-Browsern zu einem vollkommen anderen Ergebnis führen als erwartet. Hier spielen einige Faktoren eine Rolle. Zunächst werden Regeln, die für CSS spezifiziert wurden, bei der Implementierung für die Web-Browser unterschiedlich verstanden. Dies führt dazu, dass die Kombination von CSS Regeln zu interessanten Erlebnissen für die User führen kann. Hier kommt auch noch ein historisches Problem hinzu: Viele Web-Browser Hersteller wollten ihrer Zeit voraus sein, indem sie spezielle Regeln schon unterstützten, bevor diese zum Standard wurden. Damit diese eigenen Regeln von den Standard-Regeln unterscheidbar blieben, wurden sie mit einem Web-Browser spezifischen Präfix gekennzeichnet. Auch wenn viele dieser Regeln mittlerweile zum Standard hinzugefügt wurden, sind diese eigenen Regeln immer noch vorhanden und verhalten sich kurioserweise

¹⁴Eine View ist ein Container für Objekte die für den User sichtbar sind. Es ist der Bereich einer App der mit den Usern interagiert und der am Bildschirm gesehen werden kann.

oft nicht wie die Standard Regel. Ein Beispiel: Die Standard CSS Regel `border-image` kann dafür eingesetzt werden, dass ein Element mit einem Rahmen umgeben wird, der mit einem speziellem Bild gefüllt ist. WebKit¹⁵ Browser, wie z.B. Apple Safari und Google Chrome, benutzen hierfür eine eigene Regel, `-webkit-border-image`. Diese eigenen Regeln gibt es für alle Browser, nicht nur für WebKit Browser, und jeder Hersteller hat sein eigenes Präfix. Nachfolgend ein weiteres Beispiel mit allen Präfixes:

```
1 #image {
2   -webkit-transform: rotate(5deg); /* Chrome, Safari */
3   -moz-transform: rotate(5deg);   /* Firefox */
4   -o-transform: rotate(5deg);     /* Opera */
5   -ms-transform: rotate(5deg);    /* Internet Explorer */
6   transform: rotate(5deg);        /* CSS3 Standard */
7 }
```

Dieses Beispiel zeigt sehr gut auf, wie die Text-Transformation von allen Herstellern anders implementiert wurde. Möchte man nun sicher stellen, dass diese Regel in allen Web-Browsern einigermaßen dasselbe Ergebnis liefert, muss man, wie in diesem Beispiel gezeigt, alle eigenen Regeln ansprechen, anstatt nur die Regel des CSS3 Standards.

Um die hybride App auf möglichst vielen Geräten wie Smartphones, Smartwatches oder Tablets gut benutzen zu können, muss auch das CSS auf diese verschiedenen Displays und Auflösungen angepasst werden. Hier spricht man von Responsive (Web) Design (vgl. Gardner, 2011). Dabei handelt es sich um die Disziplin, denselben Inhalt für die verschiedenen Größen der Displays so anzupassen, dass der Inhalt immer so gut aufbereitet ist, dass der User damit arbeiten kann. Ein gutes und responsives Design zu erarbeiten ist eine langwierige Aufgabe, für die sehr viel Geduld und/oder Erfahrung nötig ist, aber wenn das Design steht und auf den verschiedenen Geräten gut benutzbar ist bietet es ein sehr reiches User Interface, das gerne benutzt wird. Schlechtes Design führt oft dazu, dass potenzielle User bzw. Kunden frühzeitig die App löschen bzw. die Webseite verlassen, da sie die Geduld verlieren, daher ist diese Disziplin nicht zu unterschätzen. Im Vergleich zum Design von nativen Apps ist es in etwa

¹⁵<http://www.webkit.org/>

gleich aufwendig ein responsives User Interface zu erstellen, egal ob mit CSS und HTML oder mit den nativen Möglichkeiten. Eine Ausnahme hiervon sind allerdings Apple iOS Apps, da diese ein durchaus einheitliches Look&Feel besitzen und im SDK ein Interface Builder dabei ist, mit dem das Erstellen von responiven Apps für iOS sehr einfach ist. Hier kommt auch noch hinzu, dass iOS auf einer sehr geringen Anzahl an Geräten mit verschiedenen Auflösungen läuft und daher die nativen Tools genau auf diese verschiedenen Bildschirmauflösungen ausgelegt sind. Im Vergleich dazu ist Android auf sehr vielen verschiedenen Geräten mit verschiedenen Auflösungen, Formaten und Größen zu finden und daher ist es bei nativen Anwendungen sehr viel schwieriger ein Design zu erzeugen, das für alle passt. Die folgende Abbildung¹⁶ 8.2 verdeutlicht das Problem noch einmal.

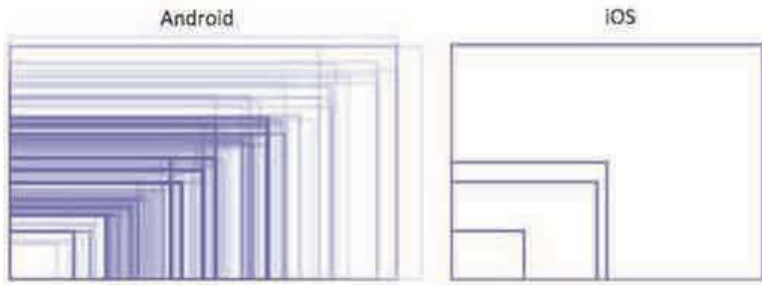


Abbildung 8.2: Verschiedene Auflösungen und Größen

Zu dem Problem der verschiedenen Auflösungen und Größen kommt auch noch hinzu, dass die Geräte drehbar sind, d.h. plötzlich soll der Inhalt statt im Hochformat im Querformat angezeigt werden und trotzdem gut präsentiert werden.

Hat man nun ein passendes Design erzeugt, das mit dem HTML zusammenpasst und auch noch in allen relevanten Web-Browsern funktioniert, hat man erst die erste Hälfte im Entwicklungsprozess geschafft, denn es fehlt noch die gesamte Logik, damit die App mehr ist als ein schönes Bildschirmposter. Für den Großteil der Business Logik wird JavaScript eingesetzt. Teile werden zwar wieder, wie zuvor beschrieben, mit den nativen Programmiersprachen umgesetzt werden müssen, aber trotzdem sollte man versuchen, so viel Logik wie möglich direkt im JavaScript umzusetzen, damit man sich Arbeit ersparen kann. Eine der größten Stärken von

¹⁶<https://ristola.wordpress.com/2014/03/27/markkinakatsaus-q12014-mobiilikayttojarjestelmat/>

JavaScript ist, dass es eine sehr schwach typisierte Sprache ist, die den Entwicklern kaum Regeln vorgibt wie sie verwendet werden soll. Diese Stärke ist gleichzeitig eine große Schwäche, ganz besonders für Neulinge im JavaScript Bereich, da oft Code entwickelt wird, der irgendwann nicht mehr zu gebrauchen ist. Normalerweise wird JavaScript für Webseiten benutzt um Kleinigkeiten zu verbessern oder um irgendwelche Effekte oder Überprüfungen durchzuführen. Bei einer hybriden App wird dieser Rahmen mehr als gesprengt und man sieht sich plötzlich vor der Herausforderung des Software Designs. Für normale Webseiten reicht es meistens völlig aus, wenn der JavaScript Code direkt im HTML als `<script>`-Tag eingesetzt wird und dort einmalig ausgeführt wird. Wird dieselbe Seite erneut aufgerufen, wird auch derselbe Code wieder ausgeführt. Bei hybriden Apps sollte diese Technik nicht eingesetzt werden, da sehr bald zu viel Logik und damit zu viele Zeilen von JavaScript Code direkt im HTML eingebettet sein würden und damit unwartbar würden. Die weitaus bessere Variante ist es das JavaScript in eigene Dateien mit der Endung „.js“ abzuliegen und diese dann nach Bedarf ins HTML einbetten, ebenfalls über den `<script>`-Tag.

Auch wenn das JavaScript ausgelagert und im HTML nur referenziert wird muss man sich bei hybriden Apps vor Beginn überlegen, welche Architektur man verwenden möchte. Für JavaScript eignen sich besonders gut zwei Architekturen, das MVC-Pattern (vgl. Bucanek, 2009) oder eine modulare Architektur. Das Model-View-Controller-Pattern ist eine der grundlegendsten Architekturen im Bereich der Softwareentwicklung. Grundsätzlich geht es darum die Logik in drei Bereiche einzuteilen, die dann nach genau definierten Regeln miteinander interagieren. Das Model ist zuständig für die Business-Logik und die View für die Darstellung der Daten, die im Modell enthalten sind. In einer hybriden App wäre die View grundsätzlich das HTML, aber auch JavaScript kann dafür notwendig sein, beispielsweise beim Hinzufügen oder Entfernen von Daten, die angezeigt werden sollen. Da User mit den Elementen interagieren, die in der View angezeigt werden, muss es auch den Controller geben, der dafür zuständig ist, die Daten im Model entsprechend anzupassen. Diese sehr einfache Architektur lässt sich mit HTML und JavaScript sehr gut umsetzen und vereinfacht die Entwicklung einer App enorm.

Wenn man allerdings viele Apps entwickelt, kommt es häufig vor dass man bestimmte Teile wiederverwenden möchte, vielleicht sogar mehrmals

in derselben App. Hier bietet es sich an, eine modulare Architektur zu verwenden, damit man verschiedene Module immer wieder ohne sie zu verändern einbauen kann. Ein gutes Beispiel hierfür wäre ein Kalendermodul, das HTML bleibt immer gleich da man immer dasselbe anzeigen möchte, die Logik wie Vor- oder Zurückblättern oder ein Datum auswählen etc. bleibt auch gleich und die Daten im Model sind auch dieselben. Daher wäre es hier angebracht ein eigenes Modul daraus zu machen welches dann als Komplettpaket mit:

```
var kalender = new Kalender();
```

aufzurufen ist und fertige Funktionalität liefert. Hierbei muss allerdings erwähnt werden, dass somit das HTML nicht statisch in einer Datei vorhanden sein kann, da es dynamisch zur View hinzugefügt werden soll. Daher muss das HTML im JavaScript Code zusammengesetzt und anschließend an das DOM¹⁷ angeheftet werden.

Eine passende Architektur zu wählen ist schwierig, da man nicht immer nur Module hat und auch nicht immer das MVC-Pattern optimal passt, daher empfiehlt es sich beide Architekturen zusammen zu benutzen, so kann z.B. ein komplexes Modul durchaus in sich wieder mit dem MVC-Pattern umgesetzt werden.

Der Einsatz von diesen Architekturen bringt zwar neue Komplexität zur hybriden App und wird anfangs die Umsetzung etwas verzögern, aber sobald ein Projekt eine gewisse Größe erreicht hat, kann man die Vorteile der Architektur erkennen, dadurch, dass es eine strikte Trennung zwischen Code Segmenten gibt, sei es durch Module oder durch das MVC-Pattern. Allerdings gibt es bei JavaScript auch noch weitere Punkte, die man beachten muss, wenn man eine gut funktionierende hybride App bauen will.

Einer dieser Punkte ist, dass JavaScript eine sehr stark Event-basierten Aufbau hat. Wenn z.B. ein Knopf gedrückt wird, mit einem Finger über den Bildschirm gewischt wird oder sich das Gerät dreht, gibt es zu allem ein Event-Objekt, das genau dann ausgelöst wird, wenn das Event passiert. Diese Event-Objekte kann man mit Event-Listnern abfangen und daraus Konsequenzen ziehen. Diese Event-Listener sind im JavaScript nur normale Funktionen, die auch anders ausgelöst werden könnten. Dieser Event-basierte Aufbau ermöglicht es den Entwicklern alle Eventualitäten

¹⁷Document Object Model, bezeichnet eine Struktur aus Objekten, Eigenschaften und Methoden in dem alle Elemente auf einer Webseite enthalten sind.

abzudecken und eine sehr vielseitige App zu entwickeln. Allerdings wird sehr oft vergessen, dass Event-Objekte eine sehr spezielle Eigenschaft besitzen, die zu Fehlern führen kann, nämlich das *Bubbling*¹⁸. Beim Bubbling geht es darum, dass ein Event in einem Knoten des DOM ausgelöst wurde und sich das Event der DOM-Hierarchie weiter nach oben arbeitet, bis es schließlich an der obersten Ebene angekommen ist. Wenn jetzt z.B. zwei Event-Listener sich im Bubbling-Pfad des Events befinden, werden nacheinander beide Funktionen ausgeführt. Dieses Verhalten wird sehr oft von Entwicklern übersehen und führt zu seltsamem Verhalten. Man kann dieses Verhalten zwar sehr leicht unterbrechen, oder auch für die Funktionalität nutzen, aber durch schlechtes Design oder unüberlegten Einsatz von weiteren Event-Listener kann früher oder später ein Bubbling-Fehler passieren.

Der Umgang mit JavaScript als „echter“ Entwicklungssprache kann, wie beschrieben, zu einigen interessanten Situationen führen, wenn man allerdings ein Gefühl dafür entwickelt hat wie JavaScript am besten und saubersten eingesetzt werden kann, ermöglicht es den Entwickler, damit großartige Applikationen mit sehr einfachen Mitteln zu schreiben.

8.7 Die Zukunft liegt im Browser

Die Mottos „The future is in your Browser“¹⁹ und „HTML5 Rocks!“²⁰ stehen in engem Zusammenhang mit hybriden Apps. Grundlage für beide Mottos ist das sogenannte HTML5, die neueste Version der Auszeichnungssprache HTML. Im Gegensatz zu seinen Vorgängern, die sich hauptsächlich damit beschäftigten, welche Tags mit welchen Attributen etc. für HTML zulässig sind und welche Schlussfolgerungen daraus zu ziehen sind, hat sich die Version 5 weiterentwickelt und befasst sich zusätzlich mit der optimierten Einbindung von JavaScript in die HTML Struktur. War früher ein Web-Browser ein Werkzeug um mittels Hyperlinks von einer Seite zur nächsten zu gelangen und dort Texte zu lesen und Bilder anzusehen, ist es heute möglich darin Musik und Videos abzuspielen, Spiele zu spielen, per Drag&Drop Objekte zu verschieben und vieles mehr, und das alles ohne Flash oder andere Browser Plugins. Auch hier liegt

¹⁸<http://javascript.info/tutorial/bubbling-and-capturing>

¹⁹<http://ig.obsglobal.com/2013/04/html5-the-future-is-in-your-browser/>

²⁰<http://www.html5rocks.com/de/>

das Geheimnis im JavaScript, da die strikte Trennung zwischen HTML und JavaScript nicht mehr gegeben ist. Diese Vielfalt an neuen Funktionen des Web-Browsers waren eine der wichtigsten Entwicklungen die notwendig waren, hybride Apps entwickeln zu können. Ohne diese Funktionen würden viele Apps nicht hybrid umsetzbar sein, da die App dann nicht das gewünschte Resultat liefern könnten und man wieder nativ entwickeln müsste. HTML5 bietet nicht nur neue Funktionen die von den Usern direkt erlebt werden können, sondern auch neue Techniken um den Code besser zu machen. So gibt es mit HTML5 sogenannte Web-Worker, die im Prinzip das Multithreading im Web-Browser endlich erlauben. Bis dato war die Ausführung von JavaScript auf einen einzigen Thread begrenzt, nämlich den UI-Thread. Wenn nun eine Berechnung oder das Herunterladen von zusätzlichen Ressourcen mit JavaScript länger gedauert hat, merkte man das, dass die Benutzeroberfläche einfrore, also nicht auf Eingaben jeglicher Art reagierte. Web Worker erlauben es nun solche Operationen auf einen neuen Thread auszulagern und dann im Hauptthread auf das Ergebnis zu warten. Speziell auf mobilen Geräten, wo die Browser nicht so viel Computing Power zur Verfügung haben wie am Desktop, waren solche „Freezes“ ein weit verbreitetes Problem und führte dazu, dass die User unzufrieden waren.

Ein Problem, das bei normalen Webseiten nicht aufgetreten ist, ist das Persistieren von Daten im Web-Browser, da Webseiten alle relevanten Daten am Server speichern und sich nicht auf den Web-Browser verlassen. Bei einer hybriden App allerdings hat man kaum andere Möglichkeiten Daten zu sichern. Man könnte zwar Daten in Dateien am Dateisystem des Gerätes speichern, dies kann aber aufgrund von fehlenden Berechtigungen etc. zu Problemen führen. Dank dem W3C²¹ und dessen Bemühungen um Standards für Web-Browser gibt es in allen Web-Browsern Möglichkeiten Daten zu persistieren. Die einfachste und oft eingesetzte Variante ist der LocalStorage. Es ist ein einfacher Key-Value-Mapping Speicher, der Strings, Zahlen und boolesche Werte speichern kann. Damit ist es möglich einfache Daten wie z.B. den Usernamen etc. zwischenzuspeichern, damit man diesen dann für den User in einem Feld eintragen kann etc. Wenn man allerdings echte JavaScript Objekte persistieren möchte muss man diese zuvor zu einem String parsen²², den man dann im LocalStorage abspei-

²¹<http://www.w3.org/>

²²Parsen (engl. „to parse“ – „analysieren, zerteilen“) ist ein Schritt um aus dem Input eines

chern kann. Dieses Parsen kann mit dem JSON-Prototypen bewerkstelligt werden, indem man `JSON.stringify()` zum Umwandeln des Objektes in einen String bzw. `JSON.parse()` um aus dem String wieder ein Objekt zu erhalten. `LocalStorage` reicht somit für sehr viele Anwendungsfälle aus, speziell, wenn verschiedene nicht zusammenhängende Daten darin abgelegt werden sollen. Eine Erweiterung dieses `LocalStorage` ist die Datenbank `IndexedDB`²³, die auf demselben Prinzip basiert, allerdings den Zugriff auf Werte in der Datenbank mithilfe von `Indexes` wesentlich schneller macht. Möchte man allerdings viele Daten, die in Beziehungen zueinander stehen, dann reicht dieses `Key-Value-Mapping` nicht aus, da es z.B. bei 1-zu-mehr Relationen entweder zu Inkonsistenz in der Datenbank oder zu duplizierten Werten führen würde. Daher gibt es noch eine Alternative zum `LocalStorage`, nämlich die `WebSQL` Datenbank²⁴. Dies ist eine echte relationale Datenbank, die auf `SQLite` basiert und wie andere relationale Datenbanken zu benutzen ist. Der Zugriff im Code ist zwar etwas komplexer als beim Zugriff auf `LocalStorage`, aber man hat die Möglichkeit komplexe Strukturen und Relationen zwischen Daten damit zu persistieren und diese dann im Web-Browser darzustellen. Das größte Manko der `WebSQL` Datenbank ist, dass sie leider *deprecated* wurde, also offiziell von der Verwendung abgeraten wurde, und sie somit in späteren Browserversionen eventuell nicht mehr zur Verfügung steht. Diese *Deprecation* wurde nicht ausgesprochen weil sie schlecht wäre oder veraltet, sondern weil es nur eine Implementierung der Spezifikation gibt, die von allen Browser Herstellern gleich ist. Da das W3C bei all seinen Spezifikationen darauf besteht, dass es verschiedene Implementierungen geben muss wurde `WebSQL` dann *deprecated*. Da es aber kein vergleichbares Produkt für den Einsatz im Web-Browser gibt, wird auch `WebSQL` noch länger in den Web-Browsern zu finden sein und kann daher auch immer noch für hybride Apps eingesetzt werden.

Wenn man absieht von `HTML5` und der W3c muss man schließlich noch den Einsatz von `Third-Party-Libraries` für `JavaScript` erwähnen. Die Frage, die man sich stellen muss ist: „Will ich das Rad neu erfinden, oder wurde es vielleicht schon von jemand anderem erfunden, getestet und stetig weiterentwickelt?“. Genau darum geht es beim Einsatz von solchen

oder mehrerer Datentypen einen Output eines anderen Datentypen zu erzeugen, mit dem dann weitergearbeitet werden kann.

²³<http://www.w3.org/TR/IndexedDB/>

²⁴www.w3.org/TR/webdatabase/

Libraries. Gerade im Bereich der JavaScript Entwicklung ist die Anzahl von guten Libraries sehr hoch, egal für welches Einsatzgebiet. So gibt es z.B. jQuery²⁵, eine Library, die sehr viele Funktionalität für alle möglichen Szenarien und Anwendungsfälle bietet, oder dessen Nebenprojekt jQuery Mobile²⁶, das ähnlich wie Sencha Touch²⁷ oder Kendo UI²⁸ viele nützliche Funktionen und Objekte bietet, die speziell für den Bereich der mobilen Webseiten und den Gebrauch in Web-Browsern auf mobilen Geräten gedacht sind. Zu diesen großen und sehr breit gefächerten Libraries gibt es auch noch solche, die auf einen bestimmten Anwendungsfall spezialisiert sind, wie z.B. Leaflet.js²⁹, eine Library die für die Darstellung von Kartenmaterial zuständig ist, oder Epoch³⁰, eine Library zur Darstellung von Graphen. Der Einsatz solcher Libraries ist durchaus anzuraten, da sie komplexe Entwicklung dieser Features erspart. Diese Libraries sind sehr gut getestet und optimiert. Allerdings muss man darauf achten, welche und vor allem wie viele Libraries in einer hybriden App eingesetzt werden sollen, denn jede zusätzliche Library kostet der JavaScript VM zusätzliche Zeit diese korrekt zu interpretieren und auszuführen. Daher sollte man gerade bei der Optimierung für mobile Geräte genau abwägen, welche Library wirklich wertvolle Zusatzfunktionalität bietet und auf welche verzichtet werden kann. Eine Patentlösung für dieses Problem gibt es allerdings nicht.

Dank den Techniken von HTML5, den Spezifikationen des W3C und durch JavaScript-Libraries wurde es erst ermöglicht gut funktionierende und auch von der Funktionalität brauchbare hybride Apps zu entwickeln, denn sonst wäre es impraktikabel Zeit und Geld in die Entwicklung einer hybriden App zu stecken die kaum Funktionalität bietet.

Auch wenn es viele Vorteile bringt hybride Apps anstatt nativer zu entwickeln muss ein Punkt noch hervorgehoben werden, nämlich die Performance, also die Leistung der App während der Ausführung. Da hybride Apps zum Großteil innerhalb eines Web-Browsers eingesetzt werden, kann eine hybride App niemals und auf keiner Plattform mit der Performance einer vergleichbaren nativen mithalten. Der Hauptgrund dafür ist, dass native Apps kompiliert werden und JavaScript interpretiert wird. Kompilierter

²⁵<http://jquery.com/>

²⁶<http://jquerymobile.com/>

²⁷<http://www.sencha.com/products/touch/>

²⁸<http://www.telerik.com/kendo-ui>

²⁹<http://leafletjs.com/>

³⁰<https://github.com/fastly/epoch>

Code ist wesentlich besser für die Runtime, in der er ausgeführt werden soll optimiert, außerdem können native Apps auf spezielle Funktionen zugreifen, die im Kernel, dem Kern des Betriebssystems, verfügbar sind. Hybride Apps, oder zumindest der JavaScript Code darin, können dies nicht, da sie vom integrierten Web-Browser ausgeführt werden und daher aus Berechtigungssicht ein Zugriff auf den Kernel untersagt ist. Hinzu kommt auch noch, dass der Kernel auf keinem Gerät mit JavaScript kompatibel ist, daher wäre es auch mit den entsprechenden Berechtigungen nicht möglich darauf zuzugreifen.

Aufgrund dieser Nachteile bei der Performance sollte man rechenintensive Applikationen wie z.B. Spiele oder Apps zur Bilderkennung und -bearbeitung oder zur Berechnung komplexer Algorithmen nicht als hybride Apps umsetzen, sondern diese nativ implementieren, auch wenn dank HTML5 viele Fälle bereits umgesetzt werden könnten. Da die meisten Apps allerdings eher einfach gehalten sind, beispielsweise Abfragen von Userdaten, Fotos machen, Kommunikation mit anderen Usern etc., können die meisten Apps dennoch hybrid entwickelt werden, ohne dass bemerkenswerte Performanceeinbußen zu bemerken wären.

8.8 Sind Cross-Platform Apps die Zukunft?

Cross-Platform Apps sind nicht nur die Zukunft, sondern bereits Realität, es wird geschätzt, dass etwa 50 bis 86% aller Apps den internen Web-Browser benutzen³¹. Zwar sind nicht alle davon Cross-Platform-Apps, manche werden den Web-Browser für kleine Aufgaben in der App einsetzen, aber diese Zahlen sind ein guter Indikator dafür, dass die Zahl der Cross-Platform-Apps durchaus mit der Anzahl nativer Apps konkurrieren kann. Da die Entwicklung im Bereich der Cross-Platform Frameworks auch fortlaufend weiterbetrieben wird, bekommen Entwickler immer weitere Möglichkeiten großartige Apps mit Hilfe dieser Frameworks zu erstellen, und auch die Web-Browser bzw. die mobilen Geräte werden immer leistungsstärker, daher werden auch Performanceprobleme bei JavaScript-basierten Applikationen immer seltener bzw. fallen kaum noch auf. Der Trend geht eindeutig in Richtung Cross-Platform Apps, vor allem mit Hilfe von hybriden Apps, große Unternehmen weltweit wie T-Mobile, Cisco,

³¹http://www.cis.syr.edu/~wedu/Research/paper/webview_acsac2011.pdf siehe Seite 8

E-Bay, PayPal und noch viele weitere³² bedienen sich bereits der Vorteile, die Cross-Platform-Apps mit sich bringen. Nicht nur große Unternehmen, sondern auch KMUs aber auch Privatpersonen nutzen vermehrt Cross-Platform Frameworks um ihre Apps zu entwickeln, oft aus dem Grund dass es keine oder zu wenige Entwickler im Unternehmen gibt die alle relevanten Plattformen nativ abdecken können und es somit billiger ist, eine App für alle Fälle zu erarbeiten. Hier kommt auch noch hinzu, dass man nicht unbedingt spezialisierte App-Entwickler einsetzen muss sondern auch jene Entwickler eingesetzt werden können, die sich im Webbereich spezialisiert haben, generell sind für die meisten Fälle nur mittelmäßige JavaScript- und HTML-Kenntnisse erforderlich um zumindest einen Prototypen zu entwickeln. Der Lernerfolg ist im JavaScript-Bereich sehr hoch, da es sehr viele Lernprogramme, sehr viel Dokumentation und Tutorien gibt, mit denen man die eigenen Fähigkeiten steigern kann.

Abschließend kann davon ausgegangen werden, dass Cross-Platform Entwicklung ein immer größeres Thema für viele Unternehmen werden wird. Die Vorteile wie geringere Entwicklungskosten, geringerer Wartungsaufwand, einfachere und offenere Programmiersprachen und Design mittels HTML und CSS stechen die Nachteile wie mögliche Performanceeinbußen, unerwartete Plattform- und Browserprobleme, etc. einfach aus. Auch die Anzahl an Cross-Platform Apps die bereits in den Stores zur Verfügung stehen zeigt, dass sich diese Art der App-Entwicklung bereits bewährt hat. Besonders wichtig ist auch die Tatsache, dass User qualitativ keine Unterschiede feststellen können, ob sie nun eine native App benutzen oder bereits eine Cross-Platform-App.

³²<http://www.appcelerator.com/customers/app-showcase/> bzw. <http://phonegap.com/app/feature/page1/>

Literaturverzeichnis

- Bucanek, James (2009). *Learn Objective-C for Java Developers*. APRESS. ISBN: 978-1-4302-2369-6.
- Gardner, Brett S. (2011). „Responsive Web Design – Enriching the User Experience“. In: *Sigma noblis* 11.1, S. 13–19. URL: <http://www.noblis.org/media/2dd575c1-2de9-4d92-9bdb-f72ad9fb9a19/docs/SigmaDigEco2011.pdf>.
- Garrett, Jesse James (2011). *The Elements of User Experience – User-Centered Design for the Web*. 2nd. New Riders Publishing. ISBN: 9780321683687.

9 Modernes Web Design

Mathias Knoll, Franz Niederl

9.1 Einleitung

Die Entwicklung des Webs schreitet in raschem Tempo voran. Dadurch ändern sich auch die Ansprüche an modernes Web Design. Einige Parameter haben sich im Laufe der Zeit als konstante Werte etablieren können. Dazu gehören in erster Linie das „ins Zentrum stellen“ der Benutzer-Interessen aber auch die Positionierung von Elementen in einer Website wie Navigation oder Suche. Viele Elemente im Webdesign entsprechen nur mehr einige Jahre dem aktuellen Stand der Technik und müssen daher in immer kürzer werdenden Zyklen erneuert werden. Auszeichnungssprachen wie HTML¹ wurden aufwärtskompatibel konzipiert. Browser ignorieren neue Elemente, die sie nicht verstehen, sodass eine Website trotzdem noch angezeigt werden kann. So blendet der Browser etwa beim Fehlen eines Bildes alternative Texte ein. Bei entsprechender Umsetzung der HTML-Seite können anstelle von Flash-Filmen Bilder angezeigt werden. Ebenso kann in Javascript² so entwickelt werden, dass DOM³-Operationen durch ein erneutes Laden der Seite ersetzt werden. Diese Eigenschaft nennt sich „Graceful Degredation“ („würdevolle Herabstufung“) oder „Fault Tolerance“ (Fehlertoleranz) und ist Teil einer Praxis,

¹HTML5 ist die Sprache in der Webseiten geschrieben werden. Es ist eine Auszeichnungssprache (Hypertext Markup Language), die neben den Inhalten einer Webseite auch deren Struktur und Format definiert. Sie wurde 1990 von Tim Berners-Lee entwickelt.

²JavaScript ist eine Programmiersprache, die im Browser abläuft. Der Quellcode wird wie die HTML Seite oder die CSS Angaben am Browser abgearbeitet.

³Das „Document Object Model“ (DOM) ist eine Schnittstelle zur HTML Seite, mit der auf alle Objekte darin zugegriffen werden kann.

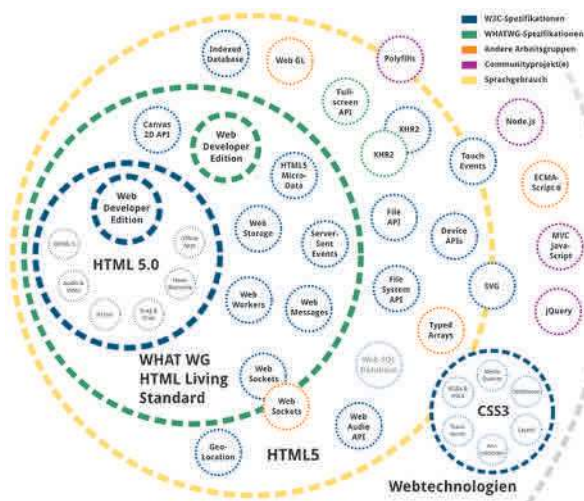


Abbildung 9.1: HTML-Technologie Stack (<https://github.com/SirPepe/SpecGraph>)

nach der für ältere Geräte, die moderne Funktionen nicht unterstützen und sich darauf verlassen, dass die neueren Funktionen einfach ignoriert werden und automatisch eine grundlegendere Version dargestellt wird.

„At this stage, we will be looking for a systems which are future-proof: portable, or supported on many platforms, extendable to new data formats.“

*Tim Berners-Lee,
„Information Management: A Proposal“
3/1989, 5/1990*

Desktop-Computer, die den Markt nach wie vor dominieren, finden ihren Einsatz sowohl in Heimwerkplätzen als auch in Firmen und bestehen meistens aus einem Rechner in einem Gehäuse, einem Bildschirm, einer Maus und einer Tastatur. Mit der Entwicklung der Grafik ging auch die Entwicklung der Computerbildschirme einher. In den Anfängen des *World Wide Webs*⁴ (WWW) sahen die Websurfer⁵ noch in Röhrenmonitore. Die

⁴Das WWW (World Wide Web) ist das weltweite Netz aus Webseiten, dessen Konzept von den CERN Mitarbeitern Tim Berners-Lee und Robert Cailliau stammt.

⁵Der Begriff „Surfen“ wurde durch den Artikel „Surfing the Internet“ einer US-amerikanischen Bibliothekarin namens Jean Armour Polly geprägt (Juni 1992 im Wil-

Ablösung geschah im Laufe der Zeit durch Flüssigkristallbildschirme (Liquid Crystal Display oder kurz LCD). Mit diesen folgte auch der Trend zu breiten Formaten und vielen unterschiedliche Auflösungen. Mit dem Siegeszug der mobilen Endgeräte (Handys, Tablets & Co) vermehrte sich auch das Browsen über jene Geräte. Laut der Studie von Morgan Stanley Research (siehe Abbildung 9.1) sollte die Anzahl an Surfern auf mobilen Browsern diejenige der Internetsurfer auf Desktop-PCs überflügeln.

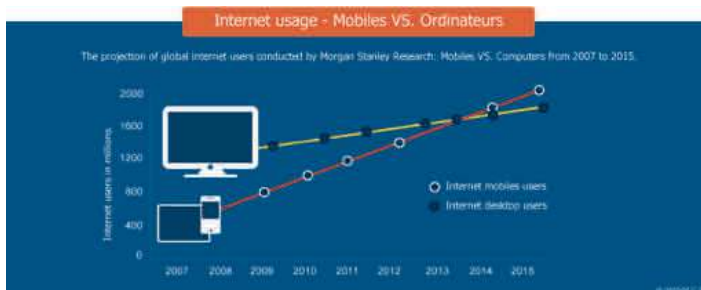


Abbildung 9.2: Quelle Morgan Stanley Research (<http://www.hangar17.com/images/stories/responsive-web-04-en.jpg>)

Die neuen Herausforderungen mobiler Geräte bestehen in den verschiedensten Displaygrößen, der Navigation mit dem Finger und nicht mehr mit Maus und Tastatur und nicht zu wissen, wie, von welcher geografischen Position und in welcher Situation der Benutzer eine Website besuchen wird. Gleichzeitig steigen die Ansprüche. Benutzer wechseln häufig zwischen Handy, Tablet und PC und erwarten eine „*seamless user experience*“. Eigenständige mobile Websites können diese Erwartungen häufig nicht mehr erfüllen.

Mit der heterogenen Landschaft an webfähigen Geräten und um den Anforderungen gerecht zu werden, entwickelten sich Webstandards wie HTML5, CSS3⁶ und JavaScript im letzten Jahrzehnt zu äußerst vielseitigen Werkzeugen. Sie ermöglichen nicht nur die Gestaltung von Webseiten für verschiedenste Endgeräte. Vielmehr nutzen moderne Websites auch die

son Library Bulletin erschienen) und beschreibt das (oberflächliche) Betrachten mehrere Webseiten hintereinander.

⁶CSS sind Stilangaben zu HTML Seiten (Cascading Stylesheets). In der Version 3 sind diese sehr mächtig und lassen sehr komplexe und responsive Layouts zu.

Möglichkeiten jener Endgeräte und liefern damit maßgeschneiderte Informationen mittels diverser Designstrategien.

9.1.1 Responsive Web Design (RWD)

In der ehemaligen Mailingliste „*A list apart*“, nunmehr ein beliebtes „Webzine“, veröffentlichte der Webdesigner und -entwickler Ethan Marcotte im Jahre 2010 einen Artikel, in dem er „responsive Architekturen“ mit der Gestaltung von Webinhalten in Verbindung brachte. Die architektonische Praxis, nach der über Untersuchungen der Umwelt und über Sensorerfassungen die Form, die Farbe und der Charakter von Gebäuden adaptiert werden, lässt sich mit aktuellen Webtechnologien in ein „responsives Webdesign“ übertragen. Das responsive Webdesign (RWD) lässt sich am besten als „reaktionsfähig“ beschreiben. Es betrachtet den Aufbau eines Webauftritts und versucht diesen für alle erdenklichen Fälle zu optimieren. Das RWD versucht Webdesigns für jeden Bildschirm und jedes Gerät in einem fließenden Übergang darzustellen.

9.1.2 Adaptives Web Design (AWD)

Adaptives Webdesign (AWD) ist ebenfalls ein Überbegriff für verschiedene Techniken und Herangehensweisen. Geprägt wurde AWD von Aaron Gustafson, der in seinem Buch „Adaptive Web Design: Crafting Rich Experiences with Progressive Enhancement“ (Gustafson, 2011) im Jahre 2011 weitere Strategien beschrieb. AWD optimiert Webseiten für verschiedene Grenzen vordefinierter Auflösungen und bietet unterschiedliche Layouts für diese an. RWD hingegen passt sich fließend auch innerhalb dieser Vorgaben an. Man kann sagen, dass AWD versucht, Webdesigns für eine vordefinierte Gruppe von Geräten und Bildschirmen maßzuschneidern.

Die Gemeinsamkeit der zwei Methoden liegt im Ziel, Webseiten auf mobilen Endgeräten und verschiedenen Bildschirmgrößen anzubieten. Die beiden unterscheiden sich in der Art, wie sie ihre responsiven / adaptiven Strukturen aufbauen: RWD basiert auf flexibler und fließender Rastern und AWD auf fix festgelegter Bildschirmgrößen. RWD braucht mehr an Programmieraufwand und Strategien zur Implementierung von fließenden Layouts, CSS und eine flexible Basis. AWD hingegen ermöglicht eine

überschaubare Anzahl an Layouts und darin eine (pixel)genaue Darstellung der Elemente.

9.1.3 Responsive Design und Server Side Components (RESS)

Allzu oft fließen die Grenzen der beiden Designparadigmen ineinander - werden doch beide in der Regel für modernes Web Design verwendet. Diese übliche Hybridlösung verbindet das clientseitige RWD mit Teilen von adaptiven Layouts sowie serverseitigen adaptiven Aufgaben. Als Beispiel für diese Aufgaben dient hier das Projekt „Adaptive Images“⁷, mit dessen Hilfe am Server unterschiedlich große Bilder für unterschiedliche Geräte ausgegeben werden.

9.1.4 Progressive Verbesserung

Sowohl RWD, AWD als auch RESS kehren das Prinzip der „Grateful Degredation“ um und wenden sich der progressiven Verbesserung zu. Diese folgt der Prämisse der Barrierefreiheit, der semantischen Auszeichnung und der Trennung von Information und Präsentation mit dem Ziel eine Website auch für Endgeräte mit eingeschränkten (unterschiedlichen) Funktionen benutzbar zu machen. Webseiten sollten grundsätzlich mit jedem Webbrowser und jeder Art Internetverbindung in ihrer grundlegendsten Form – der Bereitstellung von Information – zugänglich sein. Benutzern mit hoher Bandbreite und modernen Browsern wird eine mit zusätzlichen Funktionen ausgestatte Version der Seite dargestellt.

9.2 Strategien zur Umsetzung

Die Herangehensweise an eine aktuelle Designumsetzung in unserer „*Mul-ti - Geräte - Welt*“ ist denkbar schwer. Man versucht ein Webdesign zu entwickeln, das derart flexibel ist, um auch auf unbekanntem Geräten optimal dargestellt zu werden. Die erste strategische Frage lautet: Wie flexibel, wie zukunftsfreundlich bin ich? Welche Geräte unterstütze ich? Welche Zielgruppe habe ich?

⁷Siehe <http://adaptive-images.com/>

9.2.1 Herkömmliche Entwicklung

Die Erstellung von „normalen“ Webauftritten lief bis dato nach dem klassischen Wasserfallmodell (siehe Abbildung 9.3) ab. Für jede der einzelnen sich ablösenden Entwicklungsphasen gab es eine thematisch abgeschlossene Verantwortlichkeit. Zu Beginn stand immer eine Designphase. In dieser erarbeitete der Webdesigner ein Feinkonzept, welches Komponenten (Kopf, Navigation, Fuß, etc.) in einem Gitter anordnet und grafisch visualisiert. Der Webdesigner gab danach seine Verantwortlichkeit ab und es folgte die Umsetzung eines Prototyps in Form einer statischen Webseite⁸ durch einen Webprogrammierer. In dieser Phase wurden Bilder und Strukturen der Designphase in HTML und CSS „übersetzt“. Sobald diese Vorlage geschaffen war, konnte sie in der folgenden Phase vom Webentwickler in einem CMS (Content Management System) als „Theme“ oder „Template“ umgesetzt werden. Im herkömmlichen Modell steht nach jeder Phase eine Abnahme, die unter Umständen zu einem Rücklauf führt, um Änderungen an der vorangegangenen Phase vorzunehmen. Ein Rücklauf über mehrere Phasen ist hier unerwünscht, da dies zu einem erhöhten Aufwand führt.

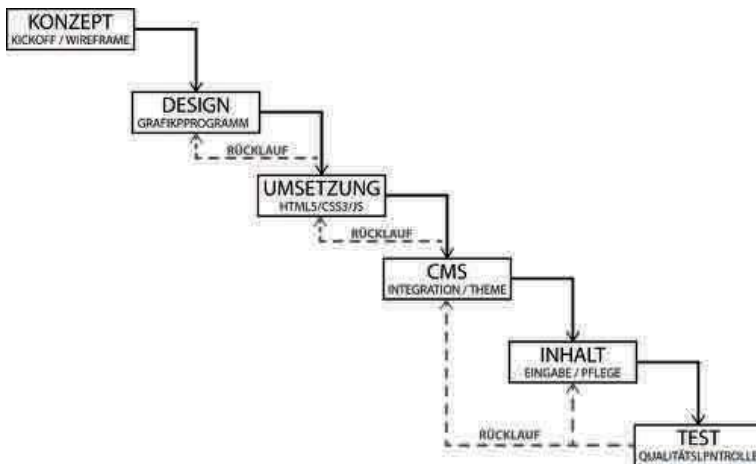


Abbildung 9.3: Wasserfallmodell zur herkömmlichen Webentwicklung (Knoll)

⁸Eine statische Webseite beinhaltet nur clientseitige Komponenten. Sie benötigt keinen Webserver sondern nur einen Browser zur Anzeige.

9.2.2 Moderne Entwicklung

Ein modernes Design kann nicht einfach in seine fundamentalen Bestandteile aufgeteilt werden, die der Reihe nach abgearbeitet werden. Von Anfang an sieht sich ein Projektteam mit dem Ineinandergreifen verschiedener Technologien und Techniken konfrontiert. Es gibt verschiedene Strategien sich der Aufgabe RWD/AWD/RESS zu stellen.

Beim „Content First“ werden Benutzer und Inhalte in den Vordergrund gerückt. Bevor Oberflächen überhaupt entworfen werden, steht am Anfang das Erfassen der Benutzergruppen und Benutzungsszenarien im Vordergrund. Aus diesen Erkenntnissen lassen sich Inhalte und Dienste filtern, die auf Anwendungsfälle referenzieren (siehe Abbildung 9.4).

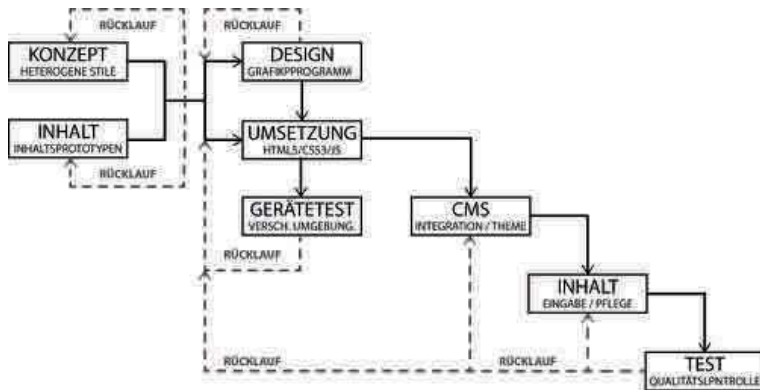


Abbildung 9.4: Webentwicklung unter modernen Gesichtspunkten (Knoll)

Auf Basis der Ergebnisse entscheidet sich, ob man „Mobile First“ oder „Desktop First“ anwendet. „Mobile First“ bedeutet, dass mobile Endgeräte bei der Layoutentwicklung priorisiert werden, während bei „Desktop First“ zu Beginn der Entwicklung der Fokus auf Desktop-Geräten liegt.

Der nächste, logische Schritt liegt in der Inhaltsverteilung und Priorisierung abhängig vom Benutzer und vom Szenario. Danach folgt ein iterativer Designprozess, dessen Stadien aus dem Entwurf, der groben Struktur, dem visuellen Design und einem Prototypen samt Tests bestehen. Ziel ist, eine Vorlage zu schaffen, die in ein CMS eingebaut werden kann. Durch die Verwendung eines modernen Designs müssen in diesen

Iterationen Teams mit heterogenem Umfeld (Designer, Programmierer, Framework Spezialist, etc.) zusammenarbeiten.

Die Anordnung von Websitekomponenten ordnet sich der Anzeige auf verschiedenen Geräten mit unterschiedlichen Auflösungen unter. Priorisierungen verschiedener Teilbereiche, wie zum Beispiel ortsabhängige Dienste bei mobilen Geräten oder detaillierte Darstellungen bei Desktopsystemen bieten Funktionalitäten, welche dem Benutzer, wenn er die Site öffnet, in seiner augenblicklichen Situation von Nutzen sind. Medienabfragen und dazugehörige Rahmenwerke unterstützen ein Design bei dieser Aufgabe. Konkret konzipiert man nun nicht nur ein Design, sondern eine Iteration eines Grunddesigns.

Inhaltsprototypen erschließen die Wichtigkeit der einzelnen Komponenten in Bezug zu Größe und Ausrichtung der Website schon früh in der Entwicklung. Der Kreis schließt sich mit der Optimierung nach den Tests und führt sich bis zu einer annähernd fehlerfreien Site fort. Durch einen Inhaltsplan kann innerhalb des CMS eine reguläre Pflege der Inhalte erfolgen.

9.3 Kosten

Bei der Budgetplanung eines Designs stellt sich die Frage, ob man Webdesign überhaupt effektiv kalkulieren kann. Je nach Vorstellung darüber, wie ein Auftritt aussehen sollte und wie viele Features enthalten sind, variieren die Kosten. Die Frage, ob ein für spezielle Geräte maßgeschneidertes Design (AWD) eventuell günstiger als RWD wäre, ist schwer zu beantworten. Je nach Detailierung des Entwicklungsprozesses kann man aber auch die Kosten einer Site mit modernem Design kalkulieren. Demnach bildet die Aufschlüsselung der Struktur einer Website einen signifikanten Faktor in der Berechnung der Kosten. Die frühe Auseinandersetzung mit Inhalten und Komponenten im Rahmen der „Content First“ Strategie erleichtert die Budgetplanung.

9.3.1 Fortbildung

Neben der Hauptarbeit an einem Design darf nicht außer Acht gelassen werden, dass viele Webdesigner noch nicht mit den Grundlagen eines

RWD/AWD/RESS vertraut sind und Fortbildung, aber auch Recherche zu den neuesten Entwicklungen und Trends unumgänglich wird. Durch den geänderten Workflow in der Entwicklung entstehen mehr Durchgänge in denen mehr Menschen kollaborieren. Das herkömmliche Wasserfallmodell, nachdem Design und dessen technische Umsetzung hintereinander standen, wird in der modernen Entwicklung durch ein agiles Modell ersetzt, welches verlangt, dass sich Programmierer und Designer immer öfter austauschen.

9.3.2 Qualitätssicherung

Früher bestanden Tests und Qualitätssicherung darin, dass die Darstellung und Funktionen einer Site an einer Handvoll Browser geprüft wurden. Durch die vielen zusätzlichen Faktoren des modernen Designs erhöht sich sowohl der Aufwand an Zeit für das Abdecken der vielen Möglichkeiten in einem modernen Design als auch der Einsatz von Ressourcen wie Testgeräten und Testumfelder.

9.4 Faktoren

Webseiten für unterschiedliche mobile Geräte können fast nur über moderne Webdesigns wie RWD, AWD oder RESS dargestellt werden – zu sehr unterscheiden sich die Geräte in Form, Funktion, Ausstattung und Betriebssystem. Aus diesen Hauptfaktoren schälen sich zwei der wichtigsten für die Umsetzung eines modernen Web Designs heraus:

Mobile Endgeräte besitzen verschiedenste Auflösungen mit unterschiedlichen Pixelgrößen beziehungsweise unterschiedlich große Pixeldimensionen. Vergleicht man einen Full HD⁹ Fernseher mit 60 Zoll Diagonale und ein Full HD Handy mit 4 Zoll Diagonale, so besitzen beide die gleiche Anzahl von Pixeln in Höhe und Breite – beim Fernseher jedoch ist ein Pixel 0,691mm breit und beim HD Handy 0,046mm (<http://lcdtech.info/de/data/pixel.size.htm>).

Ein weiterer wichtiger Faktor liegt in den unterschiedlichen Browserversionen auf den verschiedenen Geräten und deren Abdeckung bei der

⁹Full HD entspricht einer Auflösung von 1920 x 1080 Pixel im Bildverhältnis 16:9

Unterstützung von aktuellen Webstandards. Grundsätzlich gibt es Regeln und Normen im Internet – die Umsetzung dieser ist aber von Browser zu Browser in den verschiedenen Versionen und Betriebssystemen unterschiedlich (<http://caniuse.com>).

Die notwendigen Technologien hinter einem modernen Design sind wie eingangs erwähnt, die Kombination von HTML5 zum Strukturieren einer Webseite, CSS3 zur Formatierung und JavaScript für clientseitige Logik. Da die Kombination und Programmierung jener Programmier- und Auszeichnungssprachen sehr schnell komplex und zeitintensiv werden kann, bieten diverse Programmbibliotheken Unterstützung an und bilden somit eine weitere Technologieschicht.

9.5 Werkzeuge

Um ein Gefühl für Webdesign zu bekommen, stehen bereits zahlreiche Internetauftritte, als Beispiel zur Verfügung (zum Beispiel „Starbucks“¹⁰). Hinter jeder dieser erfolgreichen Sites steht ein Entwicklungsprozess bei dem ein Team von Webdesignern und Programmierern die Komponenten und das Verhalten bei unterschiedlichen Endgeräten festlegen und anhand von Papierprototypen oder Visualisierungen mit Bildbearbeitungs- und Grafikprogrammen vordefinieren.

9.5.1 Prototypen

Eine gängige Methode der Entwicklung von Designs ist mittlerweile das „In-Browser-Prototyping“. Aktuelle Browser wie Firefox, Internet Explorer oder Chrome sind mittlerweile zu mächtigen Werkzeugen herangewachsen, die es nicht nur erlauben die Struktur von geladenen Webseiten detailliert zu erforschen sondern auch Elemente zu ändern oder Funktionalitäten wegzuschalten oder umzuschreiben. Mit fast allen Browsern lassen sich zudem clientseitige JavaScript-Programme debuggen¹¹.

¹⁰<http://www.starbucks.com/>

¹¹Beim „Debuggen“ werden Fehler von einem Programm diagnostiziert und aufgefunden. Der englische Ausdruck stammt noch aus der Zeit, wo Ungeziefer (engl. Bugs) noch wirklich ein Problem für Großrechner war.

9.5.2 Bibliotheken

Oft arbeiten hinter einem aktuelle Webdesign Musterbibliotheken („Pattern Libraries“) wie zum Beispiel „Bootstrap“¹² oder „Foundation“¹³ mit Rahmenwerken („Frameworks“). Fast alle Frameworks setzen sich aus einer Kombination von CSS und JavaScript zusammen. Nicht selten durchlaufen die Standardformate für die Darstellung von HTML Elementen die Logik innerhalb eines Frameworks, der dann die Darstellung entsprechend der Benutzerdaten anpasst. Ein Referenzwerk informiert meistens über Anwendung von CSS Klassen, JavaScript Methoden sowie die Verwendung der Bausteine („Boiler Plates“) eines Frameworks.

9.5.3 Entwicklungsumgebungen

Wenn man in der Entwicklung über die Prototypenentwicklung hinausgeht, reichen Texteditor und Webbrowser nicht mehr als Entwicklungswerkzeug. In diesem Fall muss man auf eine integrierte Entwicklungsumgebung (IDE – Integrated Development Environment) zurückgreifen, die die Navigation in den unterschiedlichen Dateien und die Fehlersuche erheblich erleichtert. Wichtig bei der Wahl der IDE ist die Unterstützung der verwendeten Technologien. HTML, CSS und JavaScript sollten auf jeden Fall unterstützt werden. Hinzu kommt in den meisten Fällen noch die Fähigkeit, serverseitige Script Sprachen wie JSP („Java Server Pages“), PHP („Pre Hypertext Processing“) oder ASP („Active Server Pages“) zu verstehen und zu „debuggen“. Je nach Voraussetzung kann so die IDE „Eclipse“ (<http://eclipse.org>), PHPStorm der Firma JetBrains (<https://www.jetbrains.com/phpstorm/>) oder eine andere, passendere Entwicklungsumgebung nützlich sein.

9.6 Layout

Bei der Gestaltung einer Website steht zu Beginn die Frage nach dem Layout.

¹²<http://getbootstrap.com/>

¹³<http://foundation.zurb.com>

9.6.1 Historie

In den Anfängen des „World Wide Web“ bildeten die Stand- oder Desktop-PCs die Mehrheit der Endgeräte, die auf Webserver zugegriffen. Die Unterstützung der Auflösungen für Monitore und deren Formate beschränkte sich auf eine Handvoll Möglichkeiten. Die Proportion war in den meisten Fällen 4:3 und die Auflösung hielt sich im Bereich von 800x600, 1024x768 und 1280x1024 Pixel auf.

9.6.2 Statisches Layout

Aus diesem Grund wurde gerne ein statisches Layout für Webseiten gewählt. Dieser Typ definierte im Grunde eine bestimmte Breite an Pixeln für eine Webseite in der dann sämtliche Elemente eingepasst wurden. Als gängiges Maß galt lange Zeit eine Breite von 920 Pixel. Bei diesem Beispiel wurde damit gerechnet, dass der Großteil der Internet Surfer Monitore mit Auflösungen größer-gleich 1024x767 Pixel besäßen. Nach Subtraktion des Browserrahmens, der möglichen Scroll Balken und anderer grafischer Elemente erschien eine Site mit der Breite von 920 Pixeln auf diesen Monitoren optimal. Bei größerer Auflösung wird die Site zentriert angeordnet und der Freiraum links und rechts mit Hintergrundgrafiken gefüllt.

9.6.3 Fluid Layout

Im Gegensatz dazu passt sich das fließende Layout der Größe des Browserfensters an. Elemente vergrößern sich, strecken sich bis an den Rand oder brechen in die nächste Zeile um. Diese Verfahrensweise verwendet auf der einen Seite den gesamten Platz, der im Browser zur Verfügung steht, hinterlässt aber auf der anderen Seite bei zu wenig Inhalten und zu hohen Auflösungen viele Löcher und die Seite wirkt wie auseinandergerissen. Verstärkt wird diese „Zerrissenheit“ des Layouts vor allem mit dem Trend zu breiten Bildschirmen mit Proportionen von 16:10 oder 16:9 und mit der damit einhergehenden Pixeldichte bei der Breite.

9.6.4 Dynamisches (modernes) Layout

Aktuell mischen sich Desktop-PCs, Tablets und Handys mit Hoch- und Querformat und den unterschiedlichsten Auflösungen. Die Besucher einer Website gehen davon aus, dass die Site auf ihrem Gerät ansehnlich und

optimal benutzbar ist. Die Erfüllung dieser Erwartungshaltung kann weder mit statischem noch mit fließendem Layout oder einer Kombination der zwei erreicht werden. Die Lösung liegt im Einziehen einer dynamischen Schicht welche alle Paradigmen der Gestaltung einer Website verwaltet und zielgerichtet ausliefert. Es handelt sich dabei längst nicht mehr um ein Design, sondern eher um ein Programm mit Bedingungen, Bausteinen und Methoden. Programmierung und Verständnis für logische Zusammenhänge erweitern die Gestaltung eines einfachen Layouts zur eigenen reaktionsfähigen und/oder adaptiven Anwendung.

9.7 Feststellen des Mediums („Media Queries“¹⁴)

Im vorigen Kapitel wurden Bedingungen als neues Element von Layouts erwähnt. Diese helfen beim Festlegen der unterstützenden Formate. In CSS3 gibt es die Möglichkeit mit so genannten „Media Queries“ Aussagen über das Zielgerät zu machen. Neben der unterstützten Auflösung, der Pixelproportion und -größe lassen sich auch Aussagen zur vertikalen oder horizontalen Ausrichtung treffen. Probleme gibt es damit trotzdem. Ältere Browser bieten unter Umständen keine Unterstützung dafür oder liefern andere Werte als erwünscht. Als Alternative zu CSS3 lassen sich Informationen auch über JavaScript erfragen. Viele RWD Frameworks verwenden sowohl CSS3 als auch JavaScript, um Rückwärtskompatibilität zu gewährleisten.

9.7.1 Media Features

Media Queries bieten die Möglichkeit gerätespezifische Informationen mittels logischen Operatoren zu filtern. Des Weiteren unterstützen diese die Angabe von Minimalwerten, Maximalwerten (min-,max-) und Gleichheitsoperatoren (>,<) zur näheren Definition der Eigenschaften. In der Tabelle (siehe Tabelle 9.1 auf Seite 116) werden übliche Eigenschaften und Beispiele aufgezählt.

¹⁴Siehe <http://www.w3.org/TR/css3-mediaqueries/>

Eigenschaft	Beschreibung	Beispiel
color	Gibt über die Anzahl der Farbbits Aufschluss	<code>@media all and (min-color:4) { ... }</code>
color-index	Anzahl der Farben in der Farbtabelle des Browsers	<code>@media all and (min-color-index: 256) { ... }</code>
aspect-ratio	Verhältnis Breite/Höhe des angezeigten Fensters	<code>@media screen and (min-aspect-ratio: 1/1) { ... }</code>
device-aspect-ratio	Verhältnis Breite/Höhe des Bildschirms	<code>@media screen and (device-aspect-ratio: 16/9) { ... }</code>
device-height (device-width)	Bildschirmhöhe in Pixel (Bildschirmbreite in Pixel)	<code>@media screen and (max-device-height : 799px) { ... }</code>
grid	Konsole oder graphischer Bildschirm (0/1)	<code>@media handheld and (grid) { ... }</code>
monochrom	Monochrome Displays	<code>@media all and (monochrome) { ... }</code>
orientation	Orientierung (Portrait/Landscape)	<code>@media all and (orientation: portrait) { ... }</code>
resolution	Auflösung in dpcm ¹⁶ oder dpi ¹⁷	<code>@media print and (min-resolution: 300dpi) { ... }</code>
scan	Scantyp bei Fernsehgeräten	<code>@media tv and (scan: progressive) { ... }</code>
width (height)	Breite (Höhe)	<code>@media handheld and (min-width: 20em) { ... }</code>

Tabelle 9.1: CSS3 Media Query Features

9.7.2 Browserspezifische Features

Neben den vom W3C¹⁵ aufgeführten Eigenschaften unterhalten verschiedene Browser ihre eigenen Eigenschaften. Eines der wichtigsten ist die „pixel-ratio“, welche ähnlich der „resolution“ in dppx („dots per pixel“) rechnet. Demnach müssen bei entsprechenden Stilangaben beide Eigenschaften verwendet werden.

¹⁶dpcm = dots per centimeter

¹⁷dpi = dots per inch

¹⁵W3C ist eine Abkürzung für das World Wide Web Konsortium (<http://w3c.org>), welches Standards für das WWW verwaltet.

Browser	Eigenschaft (nicht standardisiert)
Mozilla Firefox	-moz-device-pixel-ratio
Google Chrome	-webkit-min-device-pixel-ratio
Opera	-o-device-pixel-ratio
Internet Explorer	-device-pixel-ratio

Tabelle 9.2: Pixel Ratio

Die Lösung ist in vielen Fällen die konsequente Verwendung sämtlicher Browservarianten. Glücklicherweise beginnt Mozilla Firefox¹⁸ als einer der meistbenutzten Browser bereits mit der Demontage der proprietären Eigenschaften hin zu einer standardisierten Version.

9.8 Typographie

Eine zu kleine Schrift wandelt sich oft zur grauen Fläche während eine zu große Schrift nicht alle Informationen auf einen Blick darstellt. Mitunter die wichtigste Entscheidung ist die Schriftart und deren Standardgrößen innerhalb eines Webauftritts.

Der Webdesigner hat viele Möglichkeiten eine Schrift zu definieren. Zu Beginn steht der Typ. Die Entscheidung, ob mit Serifen¹⁹ oder ohne, wird durch Design-Richtlinien vorgegeben: Serifen sollten auf Monitoren vermieden werden. Serifen sind kleine Linien, die an den Kanten von Schriftzeichen anschließen und sind für gedruckte Zeichen gedacht. Sie machen Texte einfacher lesbar, weil sie einzelne Zeichen und Zeilen unterscheidbarer machen. Dadurch wird der horizontale Lesefluss vereinfacht und man ermüdet beim Lesen weniger. Bücher haben in der Regel eine Auflösung von 1000dpi (d.h. Punkte pro 2,54cm).

Monitore hingegen kommen auf ungefähr 100dpi. Selbst das Retina Display von Apple hat nur 300dpi. Mit dieser Auflösung lassen sich vor allem kleine Buchstaben mit Serifen schwer darstellen und verkommen schnell zu einer unleserlichen Fläche. Zeichen ohne Serifen sind einfacher

¹⁸Ein freier Webbrowser des Mozilla-Projekts. Siehe <https://www.mozilla.org>

¹⁹Als Serifen werden die feinen Linien bezeichnet, die Buchstaben abschließen. Sie sollen u.a. bei gedruckten Werken im Finden der richtigen Zeile helfen.

und lassen sich besser skalieren. Auch bei Kinderbüchern dominieren serifenlose Schriften auf Grund der besseren Erkennbarkeit der einzelnen Zeichen.

Bei der Größe der Schrift bieten sich für den Webdesigner mehrere Möglichkeiten zur Auswahl. Er kann entweder eine fixe Größe in Pixel oder Längeneinheiten wie Zentimeter, Millimeter oder Inch angeben oder er wählt prozentuelle (%) beziehungsweise proportionale Angaben. Letztere besitzen die Einheit „em“ und werden zur Bestimmung der Zeichenbreite in Abhängigkeit von der Schriftgröße verwendet. Die Breite der Einheit „em“ soll der Breite des Buchstaben „M“ (daher der Name) entsprechen. Dies ist die Standardeinheit eines aus dem Buchdruck bekannten „Geviert“, einer quadratischen Fläche, die Maßgrundlage für Zeichen und Abstände bildet. (So ist zum Beispiel ein Leerzeichen ein Viertel-Geviert.) Sowohl das Konsortium für das WWW (welches den Standard CSS definiert) als auch für RWD empfiehlt es sich diese Maßeinheit für skalierbare Formatangaben zu verwenden. Eine weitere Möglichkeit relative Angaben zu machen ist die X-Höhe („ex“), welche die Höhe des kleinen „x“ als Maßeinheit verwendet.

Bei der Wahl der Schriftart hat man nun zwei Möglichkeiten. Man kann eine so genannte „System Font“ verwenden. Ist diese auf einem Gerät nicht vorhanden, so wird eine ähnliche, verfügbare Schrift verwendet. Eine andere Möglichkeit ist die Verwendung von Webfonts²⁰, die aus dem Internet zum Zeitpunkt der Wiedergabe der Webseite geladen werden. Dieser Vorgang erhöht die Ladezeit der Website. Sofern Schriftarten ausgeliefert werden, sollte auch die Lizenzfrage geklärt werden.

9.9 Bilder

Bilder bilden einen integralen Bestandteil von Webseiten. Zum einen sind sie Teil des Layouts und dienen der Wiedererkennbarkeit einer Site und sind Teil von Artikeln, Blogbeiträgen oder Bildergalerien. Andererseits werden Bilder auch oft zur Darstellung von Informationen verwendet.

Das RWD entscheidet hier, in welcher Auflösung ein Bild angezeigt

²⁰<http://www.fontshop.de/webfonts/>

oder nicht angezeigt wird. Beim AWD können je nach Anwendung bereits Bilder in verschiedenen Auflösungen vorhanden sein oder werden dynamisch vom Server in der richtigen Größe skaliert und geliefert. Wahlweise reduziert die Bildkompression die Datenmenge für mobile Endgeräte weiter. (z.B.: <http://adaptive-images.com>).

Einfache Richtlinien helfen, im Vorfeld Probleme zu vermeiden. Große Hintergrundbilder sollten bei mobilen Endgeräten vermieden werden. Bei Symbolen oder einfachen Grafiken wechselt man mittlerweile zu Bildern in Vektorgrafik in SVG („Scalable Vector Graphics“) oder verwendet Bilder aus einem (vektororientierten) Schriftsatz.

9.10 Komponenten

Der Erfolg eines modernen Webdesigns liegt in der Anzeige jener Komponenten, die für den Benutzer gerade wichtig sind. Entscheidend ist eine frühe Planung wann und wo diese auf der Website angezeigt werden.

9.10.1 Navigation

Die Hauptkomponente einer Website ist die Navigation oder das Hauptmenü. Bei großen Auflösungen kann man sämtliche Punkte sehen und sofort und zielgerichtet navigieren. Bei mobilen Anwendungen jedoch tritt die umfangreiche Navigation in den Hintergrund. Wichtige Komponenten sollten unmittelbar im Blickfeld liegen und das Menü sollte nur bei Bedarf aufrufbar sein. In vielen Fällen reduziert sich bei Handys ein Menü auf einen Button, der, wenn er berührt wird, wichtige Punkte zur Auswahl gibt.

9.10.2 Werbung

Viel Freifläche in einer Site bietet auch Platz für Werbung. Bei kleinen Formaten kann Werbung zum blockierenden Element werden. Dezentere Werbung in kleinen Bereichen oder am Ende der Webseite verringert den Störfaktor. Auf Grund von Media-Queries können sich großflächige Animationen und Videos in kleine Thumbnails verwandeln. Die Behandlung von Videos innerhalb einer Webseite kann auf zweierlei Arten geschehen. Einerseits können Videos in unterschiedlichen Formaten am Webserver

vorrätig sein, andererseits können sie vom Webserver auf entsprechendes Format gestutzt und komprimiert werden. Ähnliche Verfahrensweisen gelten für Bilder und Bildergalerien.

9.10.3 Modale Dialoge

Nicht unerwähnt sollten modale Dialoge bleiben. Diese können in mobilen Geräten nur sehr schwer implementiert werden und müssen gegebenenfalls in fixe Dialogseiten umgewandelt werden. Hierbei ändert sich nicht nur das Design sondern auch die Struktur und die Logik innerhalb eines Content Management Systems.

9.10.4 Tabellen

Tabellen sind sehr schwierig in einem modernen auf mobile Geräte getrimmten Design. Zum einen beinhalten sie viele Daten, welche zum anderen reduziert aber immer noch vergleichbar angezeigt werden müssen. Oft stellt sich schon zu Beginn die Frage nach der Hauptrichtung: Vergleicht man Reihen oder Spalten? Weiteres: Welche Information ist wichtig, welche unwichtig? Mit der Antwort zu diesen Fragen kann die Tabellenfläche neu angeordnet werden.

9.10.5 CMS und Unterstützung

Viele Content-Management-Systeme (CMS) bieten bereits Architekturen an, die für moderne Designs vorbereitet sind. Joomla!²¹ integriert so zum Beispiel standardmäßig das Bootstrap Framework²². Des Weiteren unterstützen fast alle CMS modulare Mechanismen zur Gestaltung des Aussehens mit Hilfe von Templates. Diese kann man sowohl kostenlos als auch über kommerzielle Anbieter beziehen. Während CMS wie Wordpress, Joomla!, Drupal und Typo3 aus der „Desktop First“ Zeit stammen und aus dieser in die mobile Welt gewachsen sind, wurden neuere CMS wie „MoBoom²³“ oder „RespondCMS²⁴“ bereits mit dem Gedanken an responsive/adaptive Designs für mobile Endgeräte entwickelt.

²¹siehe<http://joomla.org>

²²siehe<http://getbootstrap.com>

²³siehe<http://moboomb.com/>

²⁴siehe<http://respondcms.com/>

9.11 Testen

Ein eigener Bereich im Webdesign ist sowohl die Überprüfung der Site-Struktur als auch der Formatierung und des Layouts. Mit modernen Designs skalieren sich die Testfelder um den Variantenreichtum an unterschiedlichen Endgeräten. Der erste Test einer Website betrifft die korrekte Anwendung zugrundeliegender Architekturen und Technologien. Zu diesem Zweck bietet das W3C²⁵ sowohl für HTML als auch für CSS Validatoren²⁶ an. Oft werden zugunsten eines Designs Fehler bewusst in Kauf genommen, um die gewünschte bzw. vorgegebene Darstellung zu erreichen.

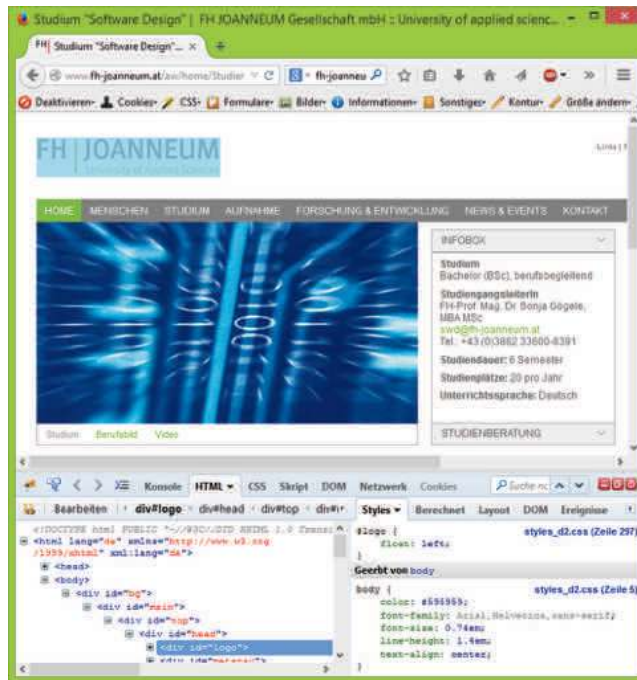


Abbildung 9.5: Der Browser Firefox mit aktiviertem Firebug

Um das Verhalten einer Website unter verschiedenen Auflösungen und Formaten zu überprüfen reicht es den Ansichtsbereich („Viewport“) entsprechend in der Größe zu ändern. In manchen Fällen unterstützen Entwicklungswerkzeuge in Browsern (wie „Firebug“²⁷ oder die fix inte-

²⁵siehe <http://www.w3c.org>

²⁶siehe <http://validator.w3.org> und <http://jigsaw.w3.org/css-validator>

²⁷siehe <https://addons.mozilla.org/de/firefox/addon/firebug/>

grierten Webentwicklertools unter Mozilla Firefox).

Abbildung 9.5 auf Seite 121 zeigt den Browser Firefox mit aktiviertem Firebug Plug-In. In der Dokumentstruktur links unten wurde ein Seitenelement gewählt, welches durch Firebug auf der Webseite hervorgehoben wird.



Abbildung 9.6: Der Browser Firefox mit aktiviertem, integrierten Entwicklungswerkzeug

Ein anderes Beispiel zeigt die Abbildung 9.6 , in der dieselbe Webseite mit dem integrierten Entwicklungswerkzeug des Browsers angezeigt wird. In diesem Bild sieht man neben der individuellen Wahl der Auflösung auch Informationen zu den Dimensionen der Seitenelemente.

Browser wie der Internet Explorer von Microsoft ® oder Safari von Apple ® bieten ähnliche Werkzeuge für Webentwickler. Als eine weitere Möglichkeit der schnellen Überprüfung eignen sich des Weiteren Online-Portale, welche Webseiten auf Eignung für verschiedene Endgeräte evaluieren. Neben den Tests im Browser findet man im Internet eine

Reihe an Gratisangeboten wie <http://responsivetest.net>, welche verschiedene Testfälle nebeneinander darstellen.

Für wirklichkeitsgetreuere Tests bieten Entwicklungsumgebungen wie das „Android Studio“ oder „Microsoft Visual Studio“ Emulatoren ihrer mobilen Betriebssysteme an. Nichtsdestotrotz sind Tests an realen Endgeräten anstatt deren Emulationen notwendig, da Eingabemethoden wie „Multitouch“ oder unterschiedliche Tastaturlayouts nicht immer realitätsgetreu simuliert werden können. Dafür bieten sich Pilotsites an, die auf eine kleine Benutzergruppe eingeschränkt sind, um Feedback zu geben.

9.12 RWD in der Praxis

Die Frage wie viele der wichtigsten Websites bereits dem RWD-Modell folgen hat Guy Podjarny (<http://www.guypo.com/rwd-2014/>) 2013 und 2014 untersucht.

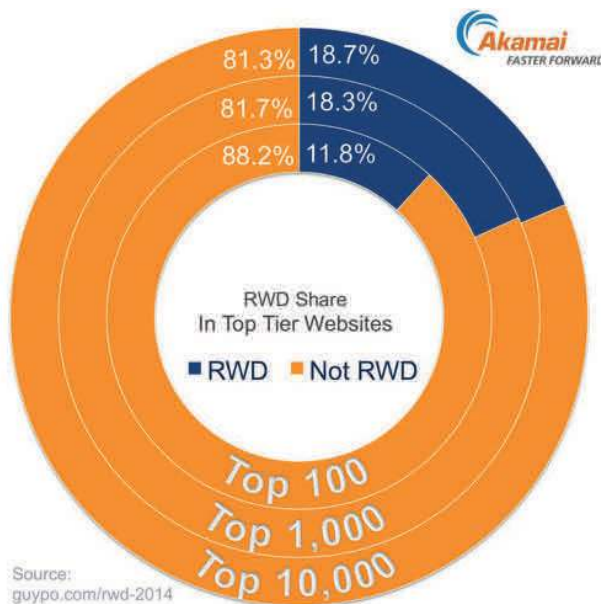


Abbildung 9.7: Untersuchung RWD-Modell

Die Grafik (Abb. 9.7) zeigt, dass unter den wichtigsten 100, 1000 und 10000 Websites erst 18,7% diesen modernen Layout Richtlinien fol-

gen. Ein Grund dafür kann sein, dass sich diese Website-Betreiber eigene Websites für unterschiedliche Endgeräte erstellt haben. Eine weitere Begründung könnte sein, dass die Umstellung aufgrund der Komplexität des CMS nicht einfach möglich ist. Betrachtet man die Änderungen im letzten Jahr, so haben 8% der wichtigsten 10000 Websites ihre Website auf RWD umgestellt.

In der Studie wurden auch die Auswirkungen von Umstellungen in den Aspekten Performance und Fußabdruck der Websites untersucht. RWD basierende Websites haben im Vergleich zu Desktop Websites in vielen Aspekten eine bessere Performance, verglichen mit dedizierten mobilen Websites besitzen sie aber einen größeren Fußabdruck auf mobilen Geräten.

Warum die Umstellung so zögerlich erfolgt, ist sicher auf viele Gründe zurückzuführen. Ein Grund bei KMUs ist sicher der Kosten-Nutzenfaktor. Da für viele der Nutzen einer RWD basierenden Website noch nicht ersichtlich ist, werden in diesem Segment die Websites erst im Zuge eines Retrofittings durch RWD basierende Websites ersetzt werden. In „How to



Abbildung 9.8: Links ursprüngliche Darstellung, mitte und rechts die RWD basierte Darstellung (Pösendorfer, 2014)

make your Website mobile“ untersucht Pösendorfer (Pösendorfer, 2014)

die notwendigen Schritte für eine Umstellung einer Website in RWD. Anhand einiger Beispiele von KMU-Websites wird unter Zuhilfenahme des erstellten Leitfadens der Umstieg erläutert. Die mit wenigen Handgriffen erzielten Ergebnisse sind ein rascher Anfang und sollten im Detail noch optimiert werden. Vergleicht man jedoch die Ansicht der Ursprungsseite mit den Ergebnisseiten, so wird schnell der Vorteil in der Darstellung auf mobilen Endgeräten ersichtlich (siehe Abbildung 9.8).

Eine automatisierte Umstellung von Websites auf das RWD-Modell könnte den Umstellungsprozess ebenfalls unterstützen oder beschleunigen. Aber in fast allen Fällen ist dies aufgrund der komplexen Template-Strukturen und CSS-Vorlagen oft nur beschränkt möglich und eine manuelle Nacharbeitung unabdingbar. (Frühwirth, 2014)

Abbildung 9.9 zeigt ein Ablaufdiagramm zur automatischen Analyse der vorhandenen HTML-Struktur. Mithilfe einer automatisierten Ana-



Abbildung 9.9: Flowchart des Analyse- und Umrechnungsvorganges (Frühwirth, 2014)

lyse können vorhandene absolute Pixelangaben in HTML und CSS gefunden und durch relative Werte ersetzt werden. Ebenso können notwendige Viewport-Angaben²⁸ und der Einsatz eines aufklappbaren Menüs automatisch hinzugefügt werden. Die Einführung von Breakpoints und das ändern von Umbrüchen und der Anordnung von Blöcken ist aber nicht möglich, dafür wird eine manuelle Nachbearbeitung benötigt. Der ersparte Arbeitsaufwand schwankt je nach Komplexität und Programmierstil zwischen 40% und 60%.

²⁸Viewport ist der für die Darstellung zur Verfügung stehende Bereich im Webbrowser.

9.13 Beispielprojekt einer RWD – Adaptierung

Eine Adaptierung einer Website auf RWD kann in den folgenden Schritten erfolgen

1. Analyse der bestehende Website
 - (a) Layout
 - (b) Inhalt
 - (c) Performance
 - (d) Auswertung der Webstatistik
2. Erstellen eines Konzeptes
3. Umsetzen des Konzeptes
4. Testen auf multiplen Endgeräten

Beispielhaft sei die Adaptierung der Website der Stadt Kapfenberg erwähnt, wo die Umstellung nach gründlicher Analyse und Auswertung nur drei Zeilen HTML-Code benötigen würde.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">
...
<label for="show-menu" class="show-menu">☰ Menü</label>
<input type="checkbox" id="show-menu" role="button">
```

Abbildung 9.10: Adaptierung mittels Viewport

Alle weiteren Anpassungen erfolgen als Ergänzung im bestehenden CSS-File.

Der Aufwand Analyse, Konzepterstellung und Umsetzung in einer Teststellung betragen ca. 100 Stunden.

Das Ergebnis der Anpassungen ist in Abbildung 9.11 auf Seite 127 ersichtlich. Die Abbildung zeigt links die bestehende Website ohne Anpassung und rechts jene mit den Veränderungen. Zwei der drei Zeilen HTML-Code sind für das Erscheinen des „Menü-Buttons“ in der Kopfzeile notwendig. Die Meta-Angabe zum Viewport veranlasst den Browser anstatt

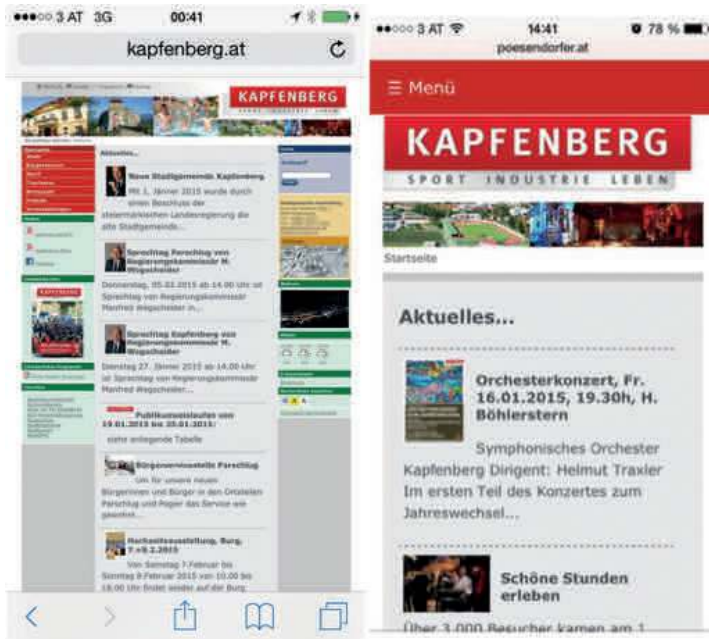


Abbildung 9.11: Anpassung Webpage Kapfenberg

der Annahme einer virtuellen Bildschirmbreite und –höhe nur die real vorhandene Bildschirmbreite und -höhe als Grundlage für die Darstellung der Website zu verwenden.

9.14 Zusammenfassung und Ausblick

Webdesign ist seit 20 Jahren in einer stetigen Entwicklung und hat immer neue Herausforderungen zu bewältigen. Entstandene Konventionen, erfahrene Benutzer und ein breites Angebot an Inhalten und Funktionen prägen das WWW. Wie lange die Tendenzen in der Layoutgestaltung und im aktuellen Flat-Design beibehalten werden, ist nicht vorhersehbar.

Modernes Webdesign ist ein Begriff, der vielfältige Gebiete umfasst. Die Umsetzung erfolgt daher nicht durch einzelne Personen, sondern durch ein interdisziplinäres Team bestehend aus Grafiker, Texter bis hin zu Entwicklern und Usability Experten. Neben dem Team hat sich aber auch der Entwicklungsprozess von einem starren, Wasserfall ähnlichem Vorgehen hin zu einem agilen Prozess gewandelt. Die dritte große Änderung betrifft die Wiedergabe-Geräte. Einerseits wur-

den die anfänglichen Röhrenbildschirme mit geringer Auflösung durch hochauflösende LCDs abgelöst, wodurch die Auflösungsoptimierung für eine bestimmte Bildschirmauflösung schon immer suboptimal gewesen ist. Ergänzt werden diese Evolutionen durch mobile Endgeräte, die unterschiedliche Auflösungen und in der Regel andersartige Eingabemöglichkeiten (zB Multitouch Gesten) anbieten. Sensoren schaffen neuartige Situationen in der User Experience.

Die Professionalisierung der Webseitenentwicklung ist speziell an den Test- und Deploymentprozessen²⁹ erkennbar, die ähnlich wie bei Softwareprojekten immer unerlässlicher werden. Die Testumgebung Browser in allen Versionen und auf allen Betriebssystemen stellt für Entwickler immer noch einen hohen Aufwand dar. Der Einsatz von CMS und umfangreichen Bibliotheken wie JQuery³⁰ oder Bootstrap erleichtern die Cross-Browser-Entwicklung von Websites erheblich.

Die Webentwicklung der Zukunft wird aus programmiertechnischer Sicht für den Einsteigerbereich mit besseren und anpassbaren Baukastensystemen einfacher und robuster werden. Stark individualisierte Oberflächen werden jedoch weiterhin einen aufwendigen Prozess durchlaufen müssen.

²⁹Das Ausrollen von Software

³⁰siehe <http://www.jquery.com>

Literaturverzeichnis

- Frühwirth, Alexander (2014). *Analyse von Websites und automatische Erstellung eines responsive Designs*. Bachelorarbeit. FH JOANNEUM, Internettechnik.
- Gustafson, Aaron (2011). *Adaptive Web Design: Crafting Rich Experiences with Progressive Enhancement*. 1st. Easy Readers, LLC. ISBN: 9780983589549.
- Pösendorfer, Lisa (2014). *How to make your website mobile?* Bachelorarbeit. FH JOANNEUM, Internettechnik.

10 KMU IT-Security-Scan

Klaus Gebeshuber

Die Größenklasse der Kleinstunternehmen sowie der kleinen und mittleren Unternehmen (KMU) setzt sich aus Unternehmen zusammen, die weniger als 250 Personen beschäftigen und die entweder einen Jahresumsatz von höchstens 50 Mio. EUR erzielen oder deren Jahresbilanzsumme sich auf höchstens 43 Mio. EUR beläuft (Gemeinschaft, 2006).

Das in dieser Arbeit vorgestellte System zielt auf sehr kleine und kleinste Unternehmen ab, deren IT-Systemen kaum Beachtung finden. Die Auswirkungen eines Ausfalls sind aber auch hier gravierend. Die durch Viren, Malware und automatisierte Hackerangriffe verursachten Ausfälle nehmen – neben den klassischen Hard- und Software Problemen – stetig zu.

Der *KMU IT-Security-Scan* bietet Unternehmen die Möglichkeit, sehr einfach eine Vielzahl von möglichen Angriffspunkten für Hacker und automatisierte Angriffssysteme zu identifizieren und damit entsprechende Maßnahmen zu setzen um die eigene IT-Sicherheit zu erhöhen.

Dabei wird das eigene Netzwerk aus dem Internet gescannt und nach bekannten Schwachstellen überprüft. Die Prüfung erfolgt mit Methoden, die auch Hacker und automatisierte Tools für eine Auswahl von potentiellen Angriffszielen nutzen.

10.1 KMU-IT-Landschaft

Oft wird die IT-Landschaft in sehr kleinen Unternehmen nach der Erstinstallation kaum mehr beachtet, die Wartung und Überwachung des Betriebs erfolgt durch externe Systembetreuer. Das IT-Know-How im Unternehmen ist auf Anwenderebene beschränkt und das Verständnis für IT-Sicherheit und mögliche Gefahren aus dem Internet ist nicht vorhanden.

IT-Systeme sind in jedem Unternehmen meist selbstverständlich, die Folgen eines Ausfalls werden oft erst beim Eintreten eines Schadensfalles bewusst. Man denke nur an den Verlust von Kundendaten, Vertragsvereinbarungen oder auch nur von einfachen Bestellungen. Ebenso kann das Unerreichbarsein per Telefon oder Email über längere Zeit einen großer Schaden für eine Firma bedeuten.

Technisch gesehen handelt es sich hier um Firmennetzwerke, die genau einen Kontaktpunkt in das Internet haben (DSL-Router). Teilweise wird ein externer Zugang in das Firmennetzwerk benötigt (VPN, Remote Desktop) und eventuell betreibt die Firma auch einen WEB- oder FTP-Server für Datenabfragen von Außendienstmitarbeitern. Die Abbildung 10.1 zeigt eine derartige Netzwerkstruktur. Im Gegensatz dazu bie-

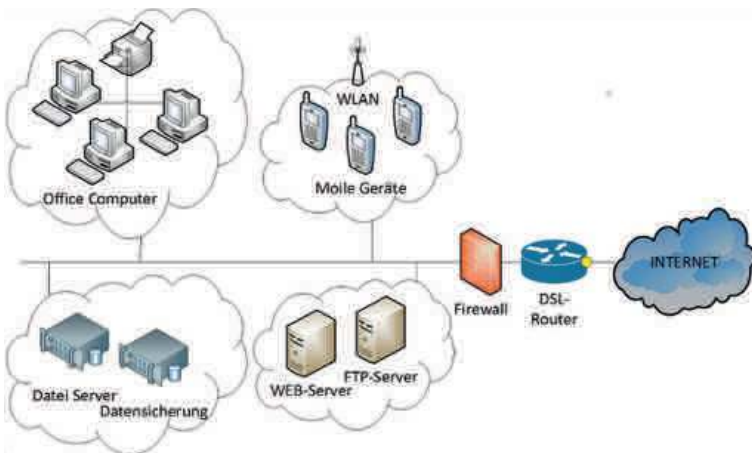


Abbildung 10.1: Firmennetzwerk mit einem Kontaktpunkt ins Internet

ten klassische Heim-Netzwerke kaum Dienste in das Internet an, wodurch

die potentielle Angriffsfläche für direkte Angriffe aus dem Internet reduziert wird. Unabhängig davon, ob extern erreichbare Dienste angeboten werden oder nicht, sind Systeme für Angriffe von innen kaum geschützt (Download von Malware, Viren, etc.). In diesem Fall hilft nur ein sauberes Patch-Management um mit der eingesetzten Software immer am aktuellsten Stand zu sein und verfügbare Sicherheits-Updates zu nutzen.

10.2 Bedrohungsszenarien

Bedrohungen aus dem Internet sind allgegenwärtig. Die Motivation der Angreifer ist von unterschiedlichster Art und reicht von reiner Neugier oder Spaß bis hin zu organisierter Kriminalität. Eine Vielzahl der Angriffe erfolgt automatisiert über sogenannte Bot-Netze. Dabei handelt es sich um einen Verbund von bereits gehackten oder durch Viren infizierte Computersysteme, die durch Scannen von ganzen Bereichen des Internets versuchen, neue Systeme zu infizieren oder anzugreifen. Aus diesem Grund ist jedes einzelne System im Internet ein potentieller Angriffspunkt. Aussagen wie „Wer interessiert sich denn schon für meine kleine Firma“ sind aus Sicht eines automatisierten Angriffs wertlos.

Ein typischer, externer Angriff läuft wie folgt ab:

- Identifikation von Systemzugriffspunkten aus dem Internet – dabei werden bestimmte Ports in einem IP-Adressbereich gescannt
- Sind offene Ports gefunden, so erfolgt eine weitere Analyse, ob die dahinterliegenden Systeme angreifbar sind. Das kann durch veraltete Softwareversionen oder einfach durch eine Fehlkonfiguration der Systeme möglich sein
- Ausnutzen von Sicherheitslücken/Schwachstellen
- Schaffung eines stabilen, wiederverwendbaren Zugriffspunktes (Backdoor)
- Diebstahl/Manipulation/Zerstörung von Daten oder auch die Nutzung des Systems für Angriffe auf andere Systeme (zur Verschleierung des Ursprungs eines Angriffes)

- Beseitigen von Spuren des Angriffs durch Löschung oder Manipulation von Log-Dateien

Die Beobachtung des Netzwerkverkehrs aus dem Internet zeigt, dass es auch bei privaten Anschlüssen mehrmals täglich Angriffsversuche (Port Scans) gibt. (Research und (GREAT), 2013)

Im IT-Grundschutzkatalog des deutschen Bundesamts für Sicherheit in der Informationstechnik (BSI) (Bundesamt für Sicherheit in der Informationstechnik, 2013) wird sehr detailliert auf mögliche Bedrohungsszenarien eingegangen. Darunter fallen Themen wie:

- G 0 Elementare Gefährdungen
 - Ausspähen von Informationen / Spionage (G0.14)
 - Unbefugtes Eindringen in IT-Systeme (G0.23)
 - Software-Schwachstellen oder -Fehler (G0.28)
 - Schadprogramme (G0.39)
 - Verhinderung von Diensten – Denial of Service (G0.40)
 - ...
- G 3 Menschliche Fehlhandlungen
 - Fehlerhafte Administration von IT-Systemen (G3.9)
 - Fehlerhafte Konfiguration von Routern und Switches (G3.64)
 - Fehlerhafte Administration von VPNs (G3.90)
 - Fehlerhafte Konfiguration von Exchange (G3.60)
 - ...
- G 5 Vorsätzliche Handlungen
 - Missbrauch von Remote-Zugängen für Managementfunktionen von Routern (G5.61)
 - Manipulation von Managementparametern (G5.86)
 - Angriffe auf Exchange-Systeme (G5.163)
 - ...

Daraus ergeben sich typische Angriffspunkte, deren Ursachen meist veraltete Software, Fehlkonfigurationen auf der Firewall oder auch nur „vergessene“ Testzugänge zu internen Services sind.

Unternehmen müssen sich über die eigenen, zu schützenden Informationen im Klaren sein. Was sind die eigenen Wirtschaftsgüter, deren Verlust oder Manipulation existenzbedrohlich sein kann? (z. B.: Pläne, Kundendaten, Produktionsdaten, Verträge, ...)

10.3 Systemanforderungen

Die Zielgruppe für den KMU IT-Security-Scan sind kleine und mittlere Unternehmen, wo sowohl IT- und Security-Wissen kaum vorhanden sind. Dementsprechend ergeben sich folgende spezifische Anforderungen:
Benutzeranforderungen:

- Das System soll von Personen mit geringem IT Know-How bedienbar sein (Usability)
- Keine spezielle Konfiguration / Parametrierung notwendig (1 Klick Bedienung)
- Keine Installation von spezieller Software
- Nutzung am Mobiltelefon
- Nutzung am Desktop PC
- Einfache Darstellung der Ergebnisse und Unterstützung bei der Ableitung von Maßnahmen
 - Ist Handlungsbedarf gegeben J/N
 - Bereitstellung von ausreichend Information, damit ein Systembetreiber Korrekturen am IT-System vornehmen kann
- Mehrsprachfähigkeit

Betreiberanforderungen:

- Geringer Aufwand für die Integration neuer Client-Systeme (Neue Betriebssysteme, veränderte Bildschirmauflösung)

- Modernes Design (Responsive)
- Der Service darf nicht als anonyme Scanner/Hackerplattform missbraucht werden
- Rechtliche Absicherung der Betreibers vor Missbrauch
- Einfache Erweiterbarkeit für neue Funktionalitäten

Kostenlose Nutzung:

- Der Service soll kostenlos zur Verfügung gestellt werden

10.4 Systemarchitektur

Die Abbildung 10.2 zeigt schematisch den Aufbau und damit auch das mögliche Wirkungsumfeld des KMU IT-Security Scanners. Die

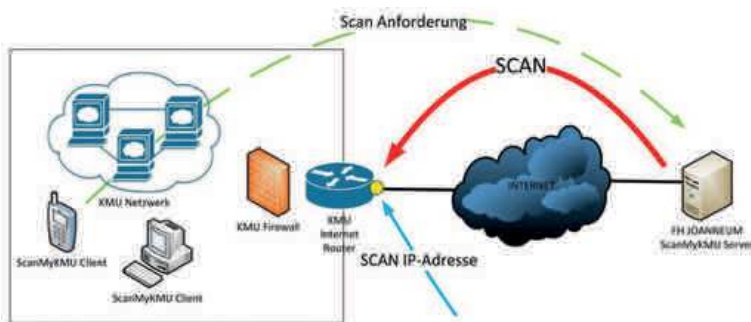


Abbildung 10.2: KMU IT-Security Scanner

Überprüfung der Netzwerkgrenze eines Unternehmens in das Internet kann nicht aus dem eigenen Netz erfolgen. Es ist erforderlich, dass der Scan von außerhalb des Netzwerkes erfolgt. Im konkreten Fall stellt die FH JOANNEUM ein Scan-System in deren eigener IT-Infrastruktur zur Verfügung. D.h., jeder Scan wird von ein und derselben Absenderadresse, die der FH JOANNEUM zugeordnet ist, durchgeführt¹. Damit ist auch eine Unterscheidung des Scans in Firewall-Logfiles von anderen Angriffen

¹FH JOANNEUM, KMU IT-Security Scan, 2014 <https://kmu.fh-joanneum.at>
IP: 178.189.85.118

möglich.

Einschränkungen:

- Eine Überprüfung der internen IT-Systeme ist mit dem vorliegenden System nicht möglich. Dazu wäre ein Zugang in das interne Netz notwendig.
- Um Missbrauch des Systems zu verhindern, wird nur die Adresse gescannt, mit der der Aufruf zum Scan erfolgt ist (im Normalfall ist das die externe Adresse des Routers, mit dem die Verbindung in das Internet hergestellt wird). Eine Auswahl einer beliebigen IP-Adresse ist nicht vorgesehen. Bei Netzwerkstrukturen, mit mehreren Zugriffspunkten aus dem Internet ist daher nur ein eingeschränktes Bild im Ergebnis sichtbar.

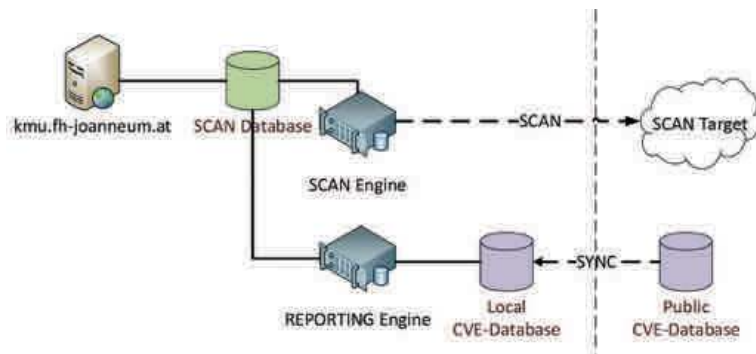


Abbildung 10.3: Architektur des KMU IT-Security Scanners

Der WEB-Server `kmu.fh-joanneum.at` enthält die Benutzerschnittstelle, bestehend aus Registrierung, Auslösung eines Scans, Betrachten von Reports und Statistiken. Die SCAN-Engine führt die Security Überprüfungen durch. Die REPORTING-Engine überprüft, ob für die identifizierten Systeme bekannte Sicherheitslücken vorhanden sind, dabei wird aus Gründen der Systemunabhängigkeit und aus Geschwindigkeitsgründen eine lokale Kopie der CVE-Datenbank² verwendet. Die Synchronisation mit der öffentlichen CVE-Datenbank erfolgt zweimal täglich. Die Ergebnisse aus SCAN und REPORTING werden in der SCAN-Datenbank

²CVE Search Engine: <https://github.com/wimremes/> und <https://github.com/adulau/>

abgelegt und stehen dem WEB-Server für die weitere Interaktion mit dem Benutzer zur Verfügung. Die vorliegende Architektur erlaubt eine einfache Parallelisierung von SCAN und REPORTING, um in einer späteren Ausbaustufe auch eine große Anzahl von gleichzeitigen Anfragen bedienen zu können.

10.5 Funktionsumfang

Für die Nutzung des KMU IT-Security-Scans ist eine Registrierung unter <https://kmu.fh-joanneum.at> erforderlich. Aus rechtlichen Gründen erfolgt die Einschränkung des Scans auf in Österreich registrierte IP-Adressen.

Der KMU IT-Security-Scan bietet die folgende Funktionalität:

Basis Überprüfung:

- DNS Namensauflösung
- Überprüfung der Antwort der Zielsystems auf PING-Anfragen (ICMP)
- Abfrage der WHOIS Registrierungsinformation

Serviceüberprüfung:

- TCP Port Scan auf eine einstellbare Anzahl von Ports (z.B. Scan der 20 am häufigsten verwendeten Ports)
- Serviceidentifikation bei offenen Ports
- Suche nach bekannten Sicherheitslücken der identifizierten Services in der CVE (Common Vulnerabilities and Exposures) Datenbank³

Detaillierte Serviceüberprüfung (Deep Scan):

- Port 80 (http) Überprüfung auf ca.4000 gängig Datei und Verzeichnisnamen
- Port 21 (ftp) Überprüfung ob Anonymous FTP aktiviert ist

Für nicht registrierte Benutzer steht ein Demo-Modus zur Verfügung. In diesem Fall wird ein virtuelles Zielsystem mit zahlreichen bekannten Sicherheitslücken gescannt.

³CVE Details siehe <http://www.cvedetails.com>

10.6 Bedienung

Nach erfolgter Registrierung auf <https://kmu.fh-joanneum.at> erhält der Benutzer eine Bestätigung per E-Mail und kann sich im WEB-Portal einloggen.

Vor dem Start muss die Anzahl der zu scannenden Ports eingestellt werden. Die Einstellmöglichkeit geht von den 20 am häufigsten verwendeten Ports bis hin zur Maximalanzahl von 65535. Die Top-10 enthalten ca. 50% aller offenen Ports, die TOP-1000 enthalten ca. 93% aller offenen Ports. Die Reihung erfolgt anhand gewonnener Statistiken des Open-Source-Port Scanners NMAP⁴.



Abbildung 10.4: KMU IT-Security Scans

Nach Bestätigung der Hinweise zur Nutzung (siehe Abb. 10.5) des Services startet der Scan.

Die Dauer eines Scans hängt vorwiegend von der Anzahl der offenen Ports ab und liegt etwa im Bereich von 30 Sekunden bis 10 Minuten. Werden Services wie HTTP oder FTP gefunden, so erfolgt eine weitere, detaillierte Analyse des Services (Deep Scan). Während des Scan Vorgangs erhält der Benutzer Informationen über den aktuellen Fortschritt und etwaige gefundene Sicherheitslücken (siehe Abb. 10.6).

Nach Abschluss des Scans erhält der Benutzer einen detaillierten Bericht über den Scan (siehe Abb. 10.7), die gefundenen Sicherheitslücken

⁴NMAP Security Scanner siehe: <http://nmap.org/>

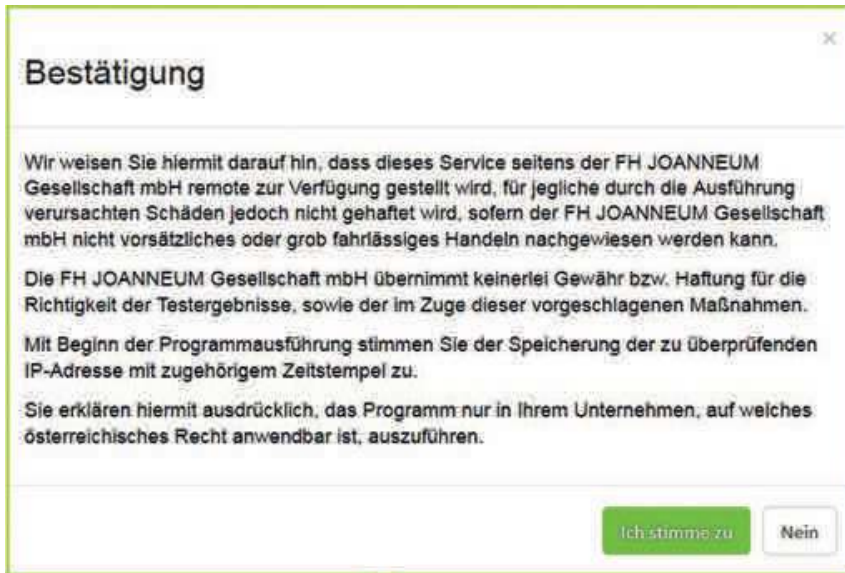


Abbildung 10.5: Hinweise zur Nutzung



Abbildung 10.6: Sicherheitslücken



Abbildung 10.7: 4 offene Ports identifiziert

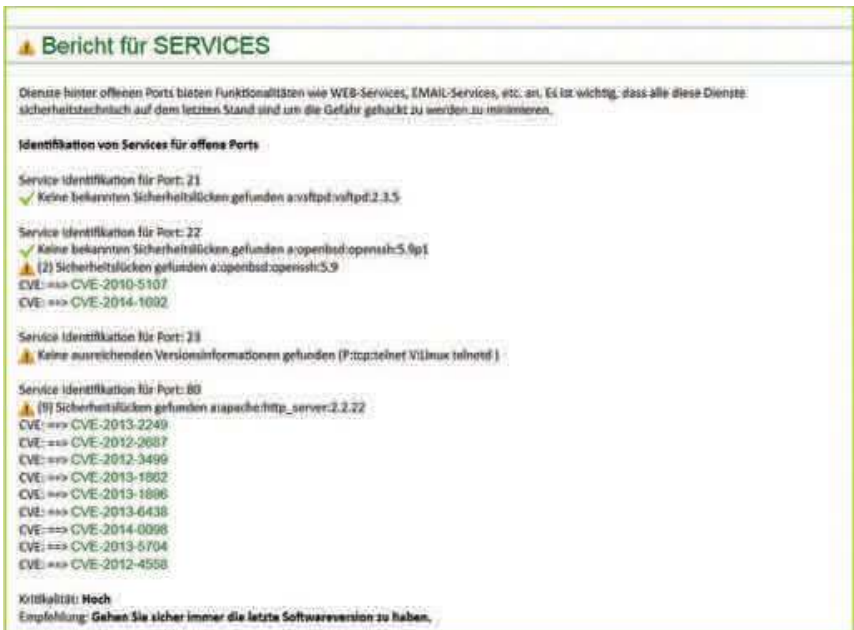


Abbildung 10.8: Services mit bekannten Sicherheitslücken gefunden (Open SSH 5.9 (2), Apache 2.2.22 (9))



Abbildung 10.9: Dateien und Verzeichnisse am WEB-Server gefunden

(siehe Abb. 10.8), eine Einschätzung der Kritikalität (siehe Abb. 10.9) und eine Empfehlung für die weitere Vorgehensweise.

Der Bericht kann als PDF exportiert werden und dem IT-Verantwortlichen vorgelegt werden. Einige der gefundenen Punkte können mit sehr einfachen Mitteln behoben werden (Deaktivierung nicht verwendeter Services, Entfernen von nicht verwendeten Dateien etc.)

10.7 Ergebnis nach durchgeführten Sicherheitsmaßnahmen

Die folgenden Bilder zeigen das Zielsystem nach Durchführung der notwendigen Sicherheitsmaßnahmen. Die Wirksamkeit der Maßnahmen wurde durch einen erneuten Scan verifiziert. Es konnten zahlreiche Mängel behoben werden.



Abbildung 10.10: Deaktivierung Port 23

Deaktivierung Port 23 (TELNET, siehe Abb. 10.10): Der Telnet-Dienst sollte aus Gründen der Sicherheit nicht mehr verwendet werden (Übertragung von Benutzernamen und Passwort im Klartext über das Netzwerk). Im konkreten Fall wird die volle Funktionalität durch den verschlüsselten SSH-Dienst abgedeckt.

Update SSH-Server auf Open SSH 6.6: Für die, trotz Update aufgeführte Sicherheitslücke liegt derzeit noch kein Sicherheits-Patch vor, in diesem Fall ist man auf die Lieferung von Sicherheitsupdates durch den Hersteller angewiesen.



Abbildung 10.11: Update Apache WEB-Server

Update Apache WEB-Server auf 2.2.27 (siehe Abb. 10.11): Derzeit liegen keine bekannten Sicherheitslücken vor.

Entfernung der Verzeichnisse und Dateien aus dem Root-Verzeichnis des WEB-Servers (siehe Abb. 10.12). Damit stehen mögliche Informationsquellen für Angreifer nicht mehr zur Verfügung.

10.8 Ergebnisse

Derzeit haben etwa 100 Firmen und Privatpersonen den KMU IT Security Scan genutzt. Dabei wurden bei 55% der durchgeführten Scans offene Ports gefunden (siehe Abbildung 10.13).

Ein offener Port in Verbindung mit aktueller Software im Hintergrund muss noch kein Problem darstellen. Wird allerdings ein alter Softwarestand

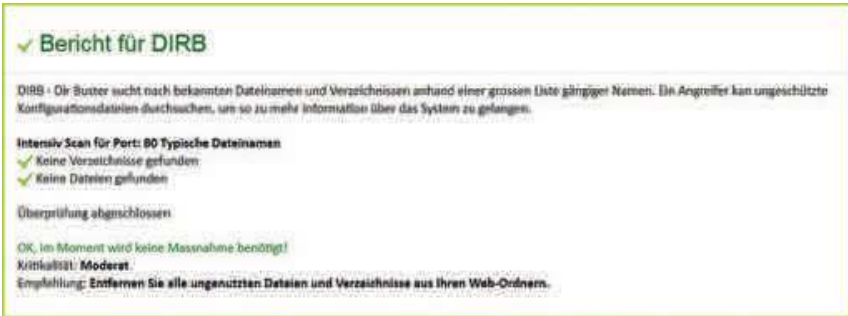


Abbildung 10.12: Root Verzeichnis des WEB-Servers

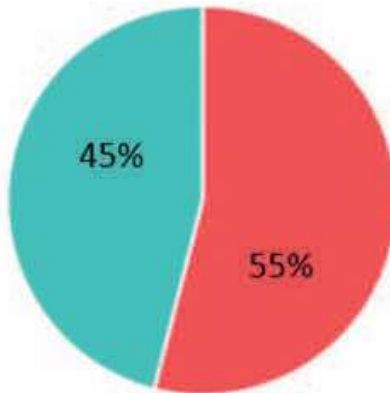


Abbildung 10.13: Gefundene offene Ports

mit bekannten Sicherheitslücken verwendet, so ist das System potentiell angreifbar.

Die größte Anzahl von mehr als 300 bekannte Sicherheitslücken wurde im Bereich der WEB-Server (Port 80 und Port 443) identifiziert. Port 80 ist auch der am häufigsten verwendete Port.

Die Abbildung 10.14 zeigt die prozentuelle Verteilung der identifizierten Services (grün) und die Summe der bekannten Sicherheitslücken zusammengefasst über alle Versionsnummern (rot).

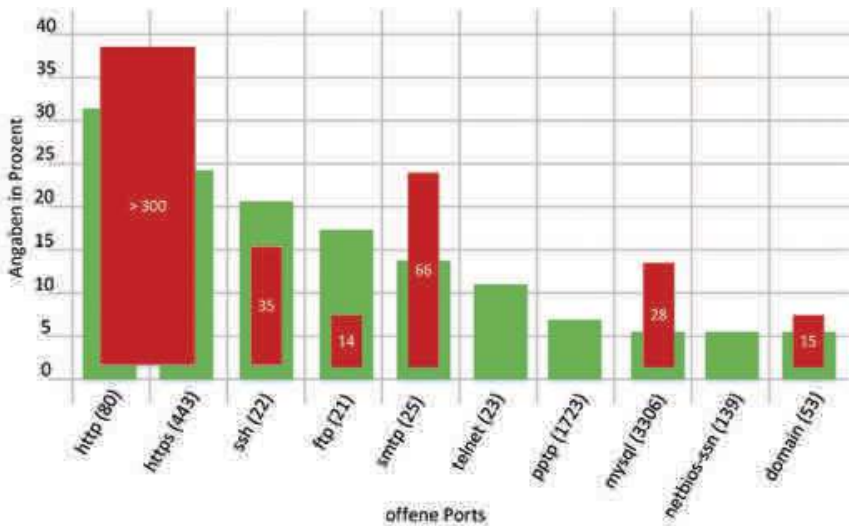


Abbildung 10.14: Identifizierte Services und bekannte Sicherheitslücken

Literaturverzeichnis

- Bundesamt für Sicherheit in der Informationstechnik (2013). *BSI Grundschrift*. URL: https://www.bsi.bund.de/DE/Themen/ITGrundschrift/StartseiteITGrundschrift/startseiteitgrundschutz_node.html (besucht am 2014).
- Gemeinschaft, Europäische (2006). *Die neue KMU-Definition*. URL: http://ec.europa.eu/enterprise/policies/sme/files/sme_definition/sme_user_guide_de.pdf (besucht am 2014).
- Research, Global und Analysis Team (GREAT) (2013). *Security Bulletin 2013*. URL: http://media.kaspersky.com/pdf/KSB_2013_EN.pdf (besucht am 2014).

11 Mobile Usability Testing

Johannes Feiner, Elmar Krajnc

Zusammenfassung: *Seit Smartphones und Tablets gilt Usability als selbstverständlich und es kann schon beinahe von jedem Kleinkind erwartet werden, dass es anspruchsvolle Anwendungen am Handy bedienen kann. Diese Einfachheit zu erreichen, bedeutet aber auch viel Vorleistung und Wissen bei der Gestaltung von Software für mobile Geräte. Usability Testing kann – wenn es früh im Entwicklungszyklus eingesetzt wird – sicherstellen, dass die Entwicklung von Softwareprogrammen nicht vorbei an den Menschen geht. Im Bereich der mobilen Smartphone-Applikationen wird von Mobile-Usability-Testing gesprochen, einem anspruchsvollem und schwierigerem Teilgebiet des Usability-Engineering.*

Im diesem Artikel wird auf die – im Vergleich zu herkömmlicher Desktop Software – veränderten Anforderungen der Usability und auch des Usability-Testing auf Smartphones eingegangen. Einerseits sind die technischen, interaktiven aber auch sozialen Möglichkeiten und Features auf mobilen Devices enorm gestiegen, und andererseits werden Aufgaben, Tasks, Probleme heute durch kürzere, unterbrochene Workflows zeitlich, und durch ständige Internetverbindung zu Cloud-Web-Services auch inhaltlich meist völlig anders gelöst. Es kommen mehrere kleine Applikationen zum Einsatz, anstatt wie früher einzelne Branchenlösungen am Desktop Rechner im Büro. Diese kleinen Apps werden vor Ort beim Kunden am Tablet, auf dem Weg zur Arbeit im Auto, oder in der Freizeit beim Joggen am Handy bedient.

11.1 Intro

Usability auf Smartphones klingt nach einem gelösten Problem. Mit dem ersten iPhone¹ kamen neue Geräte auf den Markt, welche mit wenigen Tasten auskommen und intuitiv zu bedienen sind. Plötzlich konnte jedes Kind jede beliebige Software bedienen. Und als im nächsten Schritt das iPad die Wohnzimmer eroberte, bedienen auch Hauskatzen spielerisch jedes Programm². Im Gegensatz zu herkömmlicher Desktop Software – wie zum Beispiel individuelle Branchenlösungen – funktioniert eine zielgerichtete Verwendung ohne jegliche Einschulung, ohne weitere Erklärungen oder Handbücher. Was macht Software aus, dass sie einfach bedienbar ist und ohne große Hemmschwelle von Jung und Alt so selbstverständlich und gerne verwendet wird? Was müssen Entwicklungsteams tun, um dieses ungeschriebene *Usability-Versprechen* an die EndwanwenderInnen auch einlösen zu können?

11.2 User-Experience und Benutzbarkeit

Benutzbarkeit wird klassischerweise auf verschiedenste Arten definiert, von DIN-Normen³, über die *Regeln von Shneidermann*⁴ hin zu den Nielsen Heuristiken⁵ (siehe auch Abb 11.1). Die Bewertung von Benutzbarkeit von Software kann wiederum aufgrund von Checklisten und Kriterienkataloge⁶

¹Das Apple iPhone kam in der ersten Generation im Jänner 2007 auf den Markt und gilt gemeinhin als Urmutter aller modernen Multitouch Smartphones. Siehe auch Artikel in der MacWorld <http://www.macworld.com/article/1054769/iphone.html>.

²Friskies bietet eigene *Games for Cats* im iTunes Store <https://www.gamesforcats.com> an, welche laut Herstellerseite bereits mehr als 2 Millionen Downloads verbuchen.

³DIN 9241 Ergonomie für Software, siehe zum Beispiel Standards *Ergonomic requirements for office work with visual display terminals* aus dem Jahre 1997 <https://www.iso.org/obp/ui/#iso:std:iso:9241:-1:ed-2:v1:en> wo auf Tastatur, Display, aber auch auf Dialoge, Dialoge mit Formularen, Menüs, Kommando-Dialoge aber auch auf Usability Richtlinien im Allgemeinen (Part 11) eingegangen wird.

⁴In den *Acht Goldenen Regeln* extrahiert Ben Shneidermann <https://www.cs.umd.edu/users/ben/goldenrules.html> seit 1985 aus den unzähligen User-Interface-Design-Prinzipien heraus.

⁵Als Quasi-Standard gelten die *Zehn Usability Heuristiken* von Jakob Nielsen: 1995b Nielsen, 1995 <http://www.nngroup.com/articles/ten-usability-heuristics/> aus dem Jahre 1995.

⁶Siehe zum Beispiel den KUI KeevilKeevil, 1998 Usability Index von Benjamin Keevil <http://www3.sympatico.ca/bkeevil/sigdoc98/> oder eine weiterentwickelte, verbesserte

erfolgen.

- | | |
|------------------------------------------------------------|--------------------------------------|
| 1. Visibility of system status | 1. Strive for Consistency |
| 2. Match between system and the real world | 2. Cater to Universal Usability |
| 3. User control and freedom | 3. Offer Informative Feedback |
| 4. Consistency and standards | 4. Design Dialogs to Yield Closure |
| 5. Error prevention | 5. Prevent Errors |
| 6. Recognition rather than recall | 6. Permit Easy Reversal of Actions |
| 7. Flexibility and efficiency of use | 7. Support Internal Locus of Control |
| 8. Aesthetic and minimalist design | 8. Reduce Short-term Memory Load |
| 9. Help users recognize, diagnose, and recover from errors | |
| 10. Help and documentation | |

Abbildung 11.1: Die zehn Usability Heuristiken von Jakob Nielsen und die acht goldenen Regeln von Ben Shneidermann.

Diese klassischen Betrachtungsweisen – oft auf Web Seiten Evaluierung optimiert – müssen aktualisiert, adaptiert und gegebenenfalls erweitert werden, um den neuen zusätzlichen Ansprüchen der mobilen Applikationen zu entsprechen. Fragen wie *Ist ein Glossar, eine Sitemap vorhanden?*, *Werden den Benutzern Rollen zugewiesen?* oder *Sind die Icons mit erläuternden Texten versehen?* sind auf Smartphoneapplikationen einfach nicht anwendbar oder gar kontraproduktiv.

Für die Usability von Smartphone Anwendungen müssen andere Bereiche in Betracht gezogen werden. Apps auf mobilen Geräten werden

te Version, den WUI Web Usability Index <http://www.cheval-lab.ch/was-ist-usability/usabilitymethoden/kriterienkataloge/wui/>, welcher nach Ausfüllen eines Excel Sheets einen schnellen (grafischen) Überblick über eventuelle Usability Mängel von Web Seiten bietet.



Abbildung 11.2: Durch Positionierung mittels GPS werden *location-based-services* möglich. Für die BenutzerInnen erübrigt sich nicht nur die Eingabe des Standortes, es könnte auch eine Navigations-Applikationen aufgrund der Geschwindigkeit auf die aktuelle Geschwindigkeit und den User-Typ (AutofahrerIn) schließen. Abhängig von der aktuellen Situation könnte der Weg zum nächsten *Point of Interest* vorgeschlagen werden.

natürlich auch mobil eingesetzt⁷. Man muss daran denken, dass der oder die BenutzerIn von ihrer Umgebung stark abgelenkt sein kann. Durch den mobilen Kontext ist auch nicht immer eine zuverlässige Verbindung zu Netzwerken oder anderen Geräten gewährleistet. Die geringe Gerätegröße schlägt sich auch bei der Darstellung von Bildern, Texten sowie Bedienelemente am Touchdisplays nieder. Im Gegensatz zu herkömmlichen Computern sind auch die Eingabemöglichkeiten stark eingeschränkt (vgl. Harrison, Flood und Duce, 2013).

Die klassischen Usability-Richtlinien berücksichtigen auch nicht neue Möglichkeiten wie Multitouch-Interaktion (Swipe- oder Zoomgesten siehe Abbildung 11.2) zur Navigation, die Verwendung von Sensoren zur automatischen Bestimmung des Standortes oder gar der Blickrichtung. Auch die Verwendung der eingebauten Kamera(s), des Telefonbuches und SMS-Funktionalität war früher schlicht und einfach nicht vernünftig möglich und kommt in diesen Bewertungsschemata daher auch nicht vor.

Eine Erweiterung des Usability-Focus findet heute im Speziellen auf die Felder Accessibility und User-Experience statt. Die Bedeutung der fehlenden Accessibility merkt man erst, wenn Menschen mit verschiedenen körperlichen und mentalen Einschränkungen wie Sehschwächen oder gar Blindheit die Software bedienen. Oder man versucht bei Straßenarbeitern im Winteraussendienst, welche bei Minusgraden mit Handschuhen arbeiten, eine Touch-Display-App (siehe Abbildung 11.3 (Quelle Wikimedia⁸) zu etablieren. Aber auch spezielle Benutzergruppen wie ältere Personen, Kleinkinder oder Analphabeten bedienen Software am Handy – je nach deren Möglichkeiten – oft auf durchaus unerwartete Weise.

User Experience (UX) wiederum betont die Wichtigkeit der zufriedenen Personen. Man versucht durch User Experience Design (UXD oder auch UED) positive Emotionen bei den BenutzerInnen zu wecken, indem funktionelle Usability nicht genügt, sondern die Software insofern anreichert, das sich ein positives Gesamterlebnis, beziehungsweise eine subjektive Zufriedenheit bei den EndanwenderInnen einstellt.

⁷Vergleiche *One Eyeball- One Thumb* im Artikel von Wroblewski, 2012, welches die verminderte Aufmerksamkeit bei der mobilen Benutzung von Apps beschreibt.

⁸Multitouchgesten https://upload.wikimedia.org/wikipedia/commons/0/08/Open_source_gesture_library.pdf

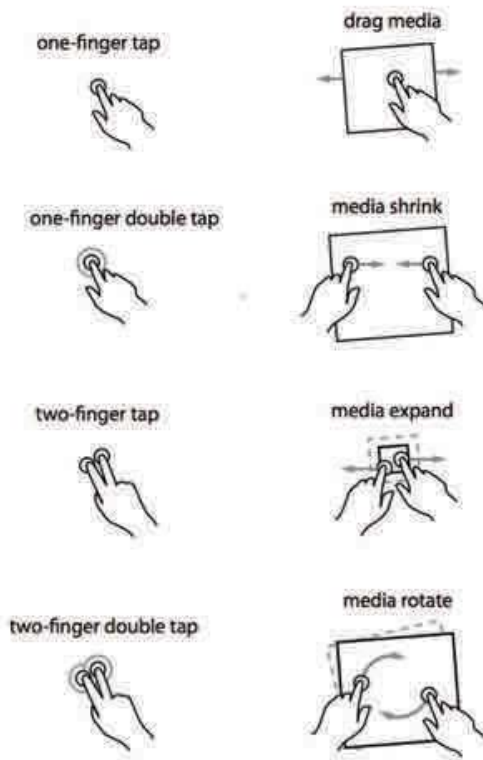


Abbildung 11.3: Verschiedene Arten von Multitouchgesten. Diese sind auf klassischen Desktop Applikationen nicht verfügbar.

11.3 Beispiele aus dem täglichen Leben

Die Bedienbarkeit und somit auch Nützlichkeit⁹ von Apps sei nun an ein paar ausgewählten Beispielen verdeutlicht. Ein klassischer Anwendungsfall ist der Ticketkauf am Handy. Ob Flugticket, Bahnticket, oder Parkticket, der Nachweis über den getätigten Kauf wird problematisch, wenn das Netz ausfällt, wenn der Akku zur Neige geht, oder wenn die Software abstürzt und das Handy neu gebootet werden muss. Genauso verärgert es NutzerInnen, wenn die Kundenkarte, oder der Ermäßigungs-Gutschein, welche am POS (Point of Sales) eingescannt wird, wieder mal nicht funk-

⁹Vergleiche Donald Norman in The Design of Everyday Things Norman, 2013.

tioniert, weil das Display spiegelt, zu dunkel ist, oder die Barcodescanner ausfallen.

Ein Spezialfall unter den problematischen Apps sind jene Applikationen, welche ursprünglich am Desktop für Maus-Bedienung – dem Point-and-Click Paradigma – entworfen und erst später auf Tablets, oder Smartphones, transferiert wurden. Üblicherweise leidet auf diese Art portierte Software einerseits unter der ursprünglichen Notwendigkeit von exakter, pixelgenauer Mauspositionierung oder auch den Context-Menüs welche mit Rechtsklick aktiviert werden, und andererseits fehlenden Möglichkeiten in der Software wie *Multitouch Gesten* oder Sensoren.

Zu kleine und auch überladene Schaltflächen stellen ein großes Problem für die Bedienung dar, laut einer Studie (vgl. Nielsen und Budiu, 2013) beträgt die ideale Größe für einen Button rund 1 mal 1cm.

Ein gelungenes Beispiel für gelungene *Wiedererfindung* ist Videoschnittsoftware am Smartphone. Vor kurzem war so eine App aus mehreren Gründen noch komplett unvorstellbar. Einerseits aus Sicht der geringen Rechenleistung der mobilen Telefone, andererseits auf Grund der schwierigen, komplexen Interaktion, welche Videoschnitt bislang erforderte. In Kombination mit großzügigen Cloud Speichermöglichkeiten¹⁰, schnellen Netzen¹¹ (wie LTE¹²), high-performance graphischen Chips und vereinfachter Touch-Interaktion¹³ hat man das Unmögliche umgesetzt.

¹⁰Dropbox, Google, Apple, Microsoft bieten jeweils gratis Speicherplatz im Umfang von mehreren Gigabyte in der Cloud an.

¹¹Sowohl größere Bandbreite, als auch geringe Latenz erlauben quasi real-time Response der entsprechenden Web-Services.

¹²Einige spannende geplante LTE, Long Term Evolution, Erweiterungen von machine-type communications bis zu indoor positioning findet man zum Beispiel auf <http://www.3gpp.org/news-events/3gpp-news/1628-rel13>.

¹³Zwei-Finger Zoom und einfaches Ziehen der Endposition ermöglicht zum Beispiel das Zuschneiden eines Movie-Clip.

11.4 Usability verbessern! Aber wie?

Mit User-Centred Design (UCD¹⁴) und User Experience (UX, siehe zum Beispiel Hartson und Pyla, 2012) Design begegnet das agile Softwareentwicklungsteam¹⁵ den gesteigerten Anforderungen. Diese Ansätze stellen sicher, dass sehr früh die Endnutzer eingebunden werden, und die Entwicklung auf wenige, nützliche und der Gesamtzufriedenheit zu Gute kommende Features gelegt wird.

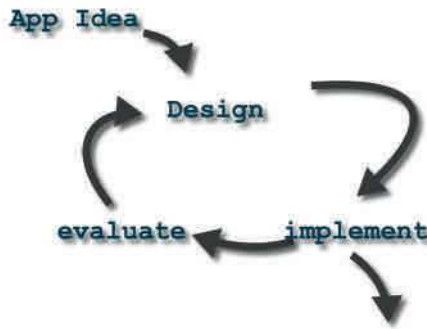


Abbildung 11.4: User Centred Design Zyklus.

Entwicklungstechnisch gilt es den völlig veränderten Anforderungen auch Rechnung zu tragen. Einerseits werden Apps heute kürzer, aber dafür bei jeder Gelegenheit verwendet (siehe zum Beispiel Mobile Interaction Design von Banga und Weinhold, 2014). Dieses *burst-like* Verhalten meint, dass man das Handy zückt um kurze Zeitspannen, wie das Warten auf den nächsten Bus, zu überbrückt. Auch der aktuelle Zustand muss auf jeden Fall immer bewahrt werden, denn durch *auto-save-state* ist sichergestellt, dass eine Applikation im jeweils letzten Zustand bleibt, also keine Änderungen oder Eingaben verloren gehen, wenn ein Anruf entgegengenommen wird und die App in den Hintergrund gedrängt, oder be-

¹⁴User Centred Design wurde durch Donal Norman 19988 *The Psychology of Everyday Things* (die aktuelle Ausgabe von Norman, 2013 nennt sich *The Design of Everyday Things*, siehe auch Abbildung 11.4) aus User Centred System Design entwickelt und bekannt gemacht. Es konzentriert sich auf zuvor nebensächliche Aspekte wie Ästhetik und fordert *Freude am Bedienen* von Objekten und Software.

¹⁵Jeff Sutherland stellte mit Kollegen Ken Schwaber 1995 auf der Konferenz OOPSLA die formale Grundlagen von SCRUM (siehe zum Beispiel Schwaber und Beedle, 2002) vor. Beide sind auch Mitunterzeichner des Agilen Manifests <http://agilemanifesto.org>.

endet wird. Moderne Applikationen benötigen keinen *Save*- Button mehr, alle relevanten Informationen werden automatisch gespeichert und bei Bedarf wiederhergestellt. Fehlerhafte Eingaben sollten allerdings leicht durch *Undo* rückgängig gemacht werden können.

Der Mobile-First-Ansatz¹⁶ priorisiert bei der Entwicklung von Software das Handy vor dem normalen PC. Zuerst muss eine Handy-(Web)App funktionieren, erst dann wird eine Desktop Version erstellt. Dieses Fokussieren auf den kleinen Handybildschirm mit wenig Platz erzwingt das Herausarbeiten der wichtigsten Abläufe und der dazu nötigen Interaktionen. Dies ist um einiges schwieriger, als eine großflächige Applikation mit zahlreichen Features zu versehen.



Abbildung 11.5: Usability-Evaluierungen sind mit mobilen Geräten besonders herausfordernd.

Als wichtiger Puzzlestein im zyklischem Entwicklungsprozess wird Usability-Evaluation (siehe Abbildung 11.5) mit effizientem Usability Reporting¹⁷ angesehen. So wird sichergestellt, dass vorhandene Schwächen

¹⁶Mobile First bei IBM <http://www.ibm.com/mobilefirst/>.

¹⁷Zum Beispiel versucht man mit UsabML Feiner und Andrews, 2012, das Reporting durch ein vorgegebenes Format für die Weiternutzung im Feedbackzyklus (Bugtracker, pdf-Reports, XML-Exchange) zu normieren und dadurch zu vereinfachen.

auch in geeigneter Weise an das Entwicklerteam zurückgemeldet werden. Das schließt ein, dass im Gegenzug die Entwickler auch positives Feedback über erfolgreich getätigte Verbesserungen erhalten.

11.5 Zukünftige Entwicklungen

Zukünftige Smartphones werden vermutlich noch mehr und erweiterte Interaktionen¹⁸ mit den Menschen unterstützen. So existieren schon seit mehreren Jahren sich selbst leicht verformbare – etwas dicker zu werden und sich aufzublasen, wenn eine neue Nachricht eintrudelt – Handys und solche, welche die Temperatur – vielleicht aufgrund der *warmen Stimmung* einer erfreulichen Smiley-Nachricht – verändern oder den inneren Schwerpunkt verändern können, um uns den Weg – jetzt links um die Ecke gehen – anzeigen könnten.

Die große Revolution der *wearables* kommt möglicherweise etwas schleichender als gedacht. Zum Beispiel über die tragbaren Smartphone-Uhren oder Kopfhörer, welche in Jacken verbaut sind. Diese Entwicklungen werden dann wiederum neue und spannende Herausforderungen für die Usability-Testing und Usability-Reporting Forschung darstellen.

¹⁸Seit einigen Jahren existieren bereits (3D-)Brillen wie Oculus Rift oder Google Glass, welche neue Visualisierungen und Interaktionen ermöglichen. Auch hat Microsoft im Jänner 2015 für das neue Windows 10 die HoloLens vorgestellt.

Literaturverzeichnis

- Banga, Cameron und Josh Weinhold (2014). *Essential Mobile Interaction Design: Perfecting Interface Design in Mobile Apps*. 1st. Pearson Education, Inc (Addison-Wesley). ISBN: 9780321961570.
- Feiner, Johannes und Keith Andrews (2012). „Usability Reporting with UsabML“. In: *Proc 4th International Conference on Human-Centered Software Engineering (HCSE 2012)*. Hrsg. von Marco Winckler, Peter Forbrig und Regina Bernhaupt. Bd. 7623. Springer Berlin / Heidelberg, S. 342–351. ISBN: 3642343465. DOI: 10.1007/978-3-642-34347-6_26.
- Harrison, Rachel, Derek Flood und David Duce (2013). „Usability of mobile applications – Literature review and rationale for a new usability model“. In: *Journal of Interaction Science* 1.1, S. 1–16. DOI: 10.1186/2194-0827-1-1.
- Hartson, Rex und Pardha S. Pyla (2012). *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*. Elsevier. ISBN: 9780123852410.
- Keevil, Benjamin (1998). „Measuring the Usability Index of Your Web Site“. In: *Proc 16th International Conference on Computer Documentation (SIGDOC 1998)*. Quebec, Quebec, Canada: ACM, S. 271–277. ISBN: 1-58113-004-X. DOI: 10.1145/296336.296394.
- Nielsen, Jakob (1995). *10 Usability Heuristics for User Interface Design*. (Besucht am 01. 01. 1995).
- Nielsen, Jakob und Raluca Budiu (2013). *Mobile Usability: Für iPhone, iPad, Android, Kindle*. 1. mitp, S. 236. ISBN: 978-3826695032.
- Norman, Donald A. (2013). *The Design of Everyday Things: Revised and Expanded Edition*. Basic Books. ISBN: 9780465072996.
- Schwaber, Ken und Mike Beedle (2002). *Agile Software Development With Scrum*. Prentice-Hall.

Wroblewski, Luke (2012). *Testing One Thumb, One Eyeball Mobile Use*.
URL: <http://www.lukew.com/ff/entry.asp?1664> (besucht am
26.11.2012).

12 Datenqualität - KMU - Mobile Anwendung: Ein auflösbarer Widerspruch?

Wilhelm Zugaj

Datenqualität ist doch eigentlich kein Thema – nicht wahr? Aktuelle Informationssysteme liefern alle Informationen auf Knopfdruck. Dank des durchschlagenden Erfolgs mobiler Clients sind diese nun immer und überall verfügbar. Die Qualität dieser Daten – kein Thema. Entweder wird das Informationssystem von gut geschulten Fachexperten gespeist, oder aber – Web 2.0 sei hier gedankt – die Community hat ein Auge darauf, und in wenigen Minuten werden nicht ganz exakte Daten korrigiert. Zudem leistet man sich doch in erfolgreichen Firmen teure IT-Abteilungen; ISO-zertifiziert in der Methode, ISO-zertifiziert die eingesetzten Produkte. Da können sich die Verantwortlichen doch zufrieden zurücklehnen – Datenqualität ist bei uns kein Thema.

12.1 Ein offensichtlicher Widerspruch

Allem Wunschdenken zum Trotz belegen zahlreiche Studien, dass es um die Datenqualität nicht zum Besten steht. Vielmehr bestätigen Sie, was der einfache Standardnutzer eines Informationssystems aus täglicher, leidvoller Erfahrung nur zu gut weiß: Datenqualität ist ein Stiefkind der IT, fehlende, falsche oder irreführende Daten kosten jede Menge Zeit und Geld.

Dies sei an dieser Stelle mit den Problemen untermauert, die beim Ticket-Kauf über das Informationssystem „Scotty“ der Österreichischen Bundesbahnen entstehen können und von help.orf.at dokumentiert sind: Verkauf von 1.Klasse-Tickets auf Strecken, auf denen keine 1.-Klasse

Züge verkehren¹, im System hinterlegte, falsche Streckenlängen und daraus resultierende, falsche Fahrkartenpreise² und fehlende Hinweise auf reservierungspflichtige Züge, die teure Nachzahlungen verursachen³.

Ausgezeichnet geschulte IT-Experten, Zertifizierungen, Fachexperten die einen guten Teil ihrer Zeit der Datenpflege widmen – diese Dinge sind für große Firmen oft schon ungeliebte Ausgabenposten, für Klein- und Mittelbetriebe liegen sie gänzlich außerhalb des Möglichen. Für diese Unternehmen ist es oft schon eine Herausforderung maßgeschneiderte mobile Anwendungen für den eigenen Geschäftsbereich, oder als Frontend für die Kunden zu entwickeln (bzw. entwickeln zu lassen). Datenqualität – das ist dann maximal ein Kundenthema. Wenn es Probleme gibt, soll der Kunde sich melden, dann schauen wir uns das an!

Haben Sie gerade ein Déjà-vu? Als Softwarequalität vor 20 Jahren noch kein Thema war, da hat man doch ähnliches gehört. Ein Absturz des Programms, unerwartetes Verhalten? Man kann nicht alles testen, der Kunde soll es melden, dann beheben wir die Probleme schon. Dies ist doch eine Gelegenheit um Wartungsverträge zu verkaufen. Wie war doch gleich der Ausdruck dafür – ah ja **Bananensoftware!**

KMUs sehen sich in einer Zeit des mobilen Zugriffs auf große Informationssysteme im Bereich der Datenqualität einer immensen Herausforderung gegenüber. Der Widerspruch zwischen der Größe dieser Herausforderung und den begrenzten Kapazitäten und Möglichkeiten scheint unauflösbar. Motivation genug um Probleme und mögliche Lösungswege einer genaueren Analyse zu unterziehen.

12.2 Datenqualität – eine Definition

Die Auswirkungen schlechter Datenqualität sind dramatisch. Sie kosten Zeit (Verbesserung der Daten selbst, Rückgängig-Machung der Auswirkungen der durch falsche Daten fehlgeleiteten Geschäftsprozesse). Noch schwerwiegender als dies ist der drohende Verlust von Kundenvertrauen

¹Beitrag unter <http://help.orf.at/stories/1699742/>

²Beitrag unter <http://help.orf.at/stories/1681755/>

³Beitrag unter <http://help.orf.at/stories/1683212/>

und Reputation.

Die wissenschaftliche Definition des Begriffs Datenqualität unterscheidet zwischen den Begriffen „quality of design“ und „quality of conformance“. Ersteres beschreibt die Übereinstimmung von Anforderung und Spezifikation von Daten und Datenstrukturen, letzteres jene von Spezifikationen und den tatsächlichen Ausprägungen in real existierenden Datenverwaltungssystemen. Datenqualität selbst wird als durch die Daten erhaltene Befähigung zum effektiven Einsatz von Systemen definiert („fitness to use“).

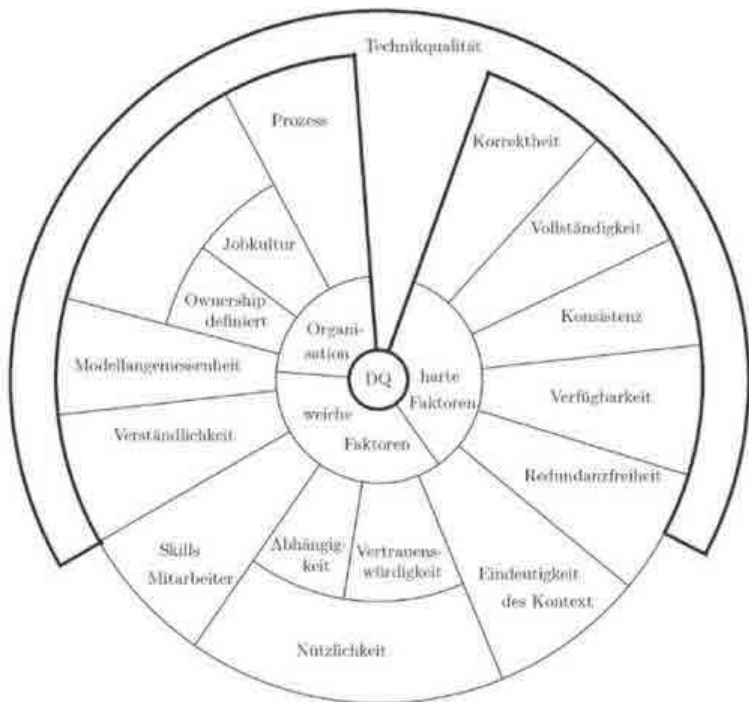


Abbildung 12.1: Datenqualitätsradar nach Würthele

Sie selbst ist aber keine einfache Eigenschaft, die Daten innewohnt, zum Teil innewohnt, oder eben nicht innewohnt. Vielmehr sind zwei Dinge zu beachten: Erstens hat Datenqualität viele Facetten. Wissenschaftliche Definitionen umfassen Begriffe wie Korrektheit, Vollständigkeit, Konsistenz, Verfügbarkeit, etc. aber auch Nützlichkeit, Verständlichkeit und „definierte Ownership“ (Würthele, 2003). Zudem variieren diese Begriffe

und deren Definitionen von Autor zu Autor (Batini u. a., 2009).

Zweitens ist Qualität immer eine kontextbezogene Eigenschaft. Die Vollständigkeit von Daten ist zum Beispiel nur definiert, wenn der Kontext der Anwendung für die diese benötigt werden gegeben ist. Wenn von einer Person keine Adresse vorliegt, ist dies im Kontext eines Online-Spiels nicht nur kein Mangel, sondern sogar gewünscht, im Kontext einer Online-Bestellung mit anschließendem Zustellungsvorgang hingegen ein Datenqualitätsproblem.

Die Messung konkreter Qualitätswerte geschieht über Datenqualitätsmetriken (Heinrich, Kaiser und Klier, 2007). Dies sind Funktionen, definiert auf Daten bzw. Datenstrukturen, die diesen Zahlen zuordnen und dabei die Kriterien „Normalization, Interval Scale, Interpretability, Aggregation, Adaptivity and Feasibility“ erfüllen. Der Wert dieser Metriken besteht darin, dass sie als Basis zur Definition von Methodologien zur Sicherstellung von Datenqualität dienen können. Kern dieser ist wiederum die „Data Quality Loop“ (Heinrich 2008). Diese funktioniert nach folgendem

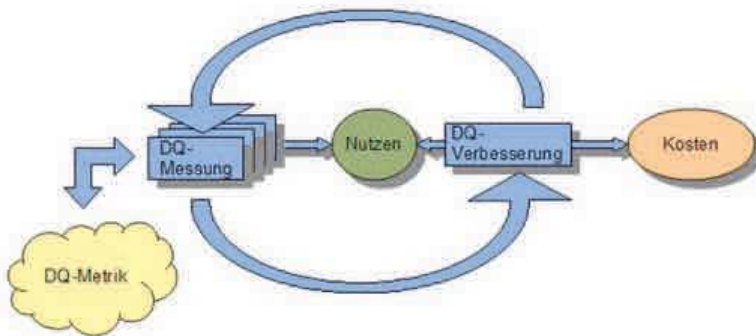


Abbildung 12.2: "Data Quality Loop" nach Heinrich

Prinzip: Datenqualitätsmetriken berichten laufend über den Zustand ausgewählter Qualitätskategorien im Kontext der jeweiligen Anwendung. Verbesserungen der Qualität verursachen Kosten, erzeugen aber auch entsprechenden Nutzen, der durch erneute Datenqualitätsmessung bestätigt wird. Die Abwägung der Kosten gegen den Nutzen bestimmt die Anzahl der Zyklen bestehend aus Messung, Verbesserung und Bestätigung die durchlaufen werden, um einen optimalen Kosten/Nutzen Zustand zu erreichen.

Für KMUs ist dieser Prozess nur bedingt praxistauglich. Das Erstellen von Datenqualitätsmetriken bedingt den Einsatz von wissenschaftlich geschultem Personal und ist ohne finanzielle Förderung durch die öffentliche Hand für KMUs kaum praktikabel. Das Einrichten einer „Data Quality Loop“ ist somit ohne externe Hilfe kaum möglich.

12.3 Datenqualität auf dem Prüfstand

Im Jahr 2002 hat „The Datawarehousing Institute“ eine richtungsweisende Untersuchung zum Thema Datenqualität durchgeführt⁴.

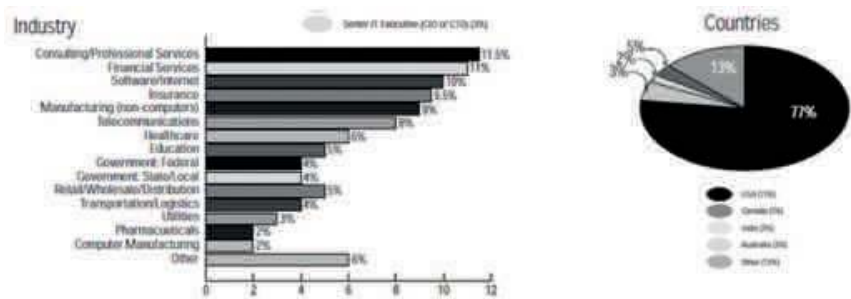


Abbildung 12.3: Verteilung der Studienteilnehmer auf Industrien und Länder

Führungspersonen und IT-Verantwortliche von Industriebetrieben – überwiegend aus den USA – wurden zum Umgang mit Datenqualität und den Auswirkungen und Problemen die ein Mangel an Qualität verursacht, befragt. Abbildung 12.3 zeigt die Verteilung auf Länder und Industrien der Studienteilnehmer.

Die folgenden beiden Abbildungen (Abbildung 12.4 und Abbildung 12.5) zeigen die Verteilung der häufigsten genannten Auswirkungen schlechter Datenqualität bzw. den Nutzen hoher Qualität.

Zusammenfassend belegt diese Studie einen Schaden von 611 Milliarden US-\$ pro Jahr alleine durch zusätzliches Porto, Druck und Arbeitszeit, resultierend aus fehlerhaften Dokumenten, die aus mangelnder Datenqualität entstehen. Zusatzkosten für Reputationsverlust, Kundenverlust

⁴Eckerson W. 2002, Data Quality and the Bottom Line, in TDWI Report Series, 2002.

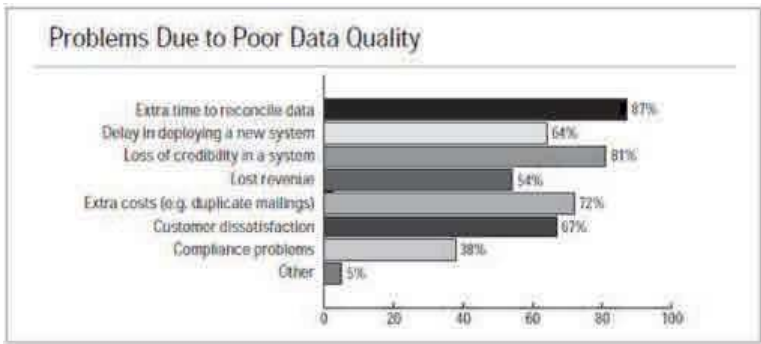


Abbildung 12.4: Häufigste durch geringe Datenqualität verursachte Probleme

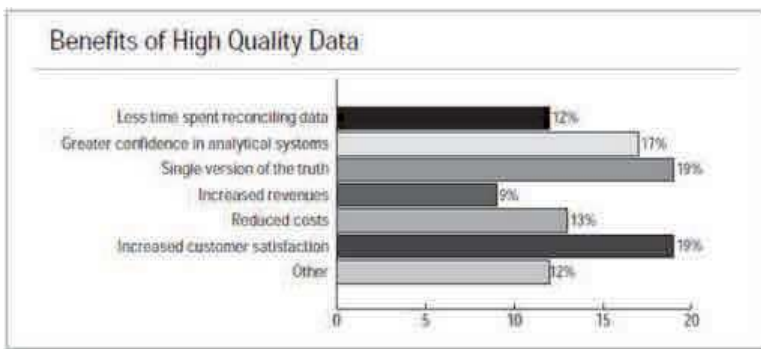


Abbildung 12.5: Häufigste Vorteile, bedingt durch hohe Datenqualität

Produktionsausfall, etc. bleiben dabei unberücksichtigt.

Legen wir diese Situation auf die Steiermark um, so würde sich durch direkten Vergleich des steirischen Bruttoinlandsproduktes mit jenem der USA ein hochgerechnetes Schadenspotential von 1,1 Milliarden Euro ergeben. Details dazu finden sich in (Zugaj, 2014).

Eine Untersuchung von „Awareness“ und Qualitätsmaßnahmen in Bezug auf Datenqualität unter 31 steirischen Betrieben – fast alle KMUs – ergab ein ernüchterndes Bild. Die Situation ist praktisch identisch mit jener der amerikanischen Industrie vor mehr als zehn Jahren (ebd.).

Es mangelt an Bewusstsein, an systematischer Erfassung von Datenqualitätsproblemen, an Maßnahmen zur Messung dieser und vor allem an Werkzeugen zur automatisierten Evaluierung und Korrektur. Im Falle von KMUs ist dies also nicht nur ein Problem des Willens und der Bereitschaft, sondern auch ein Problem der Sach- und Personalressourcen.

Anzustrebende Lösungsszenarien müssen daher genau auf den Aspekt der limitierten Ressourcen von Klein- und Mittelbetrieben abgestimmt werden.

12.4 Lösungsszenarien für Klein- und Mittelbetriebe

Damit KMU Informationssysteme mit mobilen Front-Ends und mit entsprechender Datenqualität – sowohl für den Eigenbedarf als auch als Angebot für ihre Kunden – zur Verfügung stellen können, bedarf es zweierlei: Die Möglichkeit ressourcen-schonend derartige Front-Ends erstellen zu können, sowie der Zugriff auf einfache und kostengünstige Werkzeuge zur Datenqualitätssicherung. Für die Back-Ends moderner Informationssysteme hingegen gibt es heute hingegen eine Vielzahl an freien, open-source oder kostengünstigen kommerziellen Systemen.

Im Rahmen des Projekts „KMU goes Mobile“ hat das Institut Internet-Technologien & -Anwendungen der Fachhochschule JOANNEUM eine Lösung für den Themenkreis „Mobile Front-Ends“ für KMUs erarbeitet. Diese ist unter anderem in (Feiner:2014b) beschrieben. Betreffend den

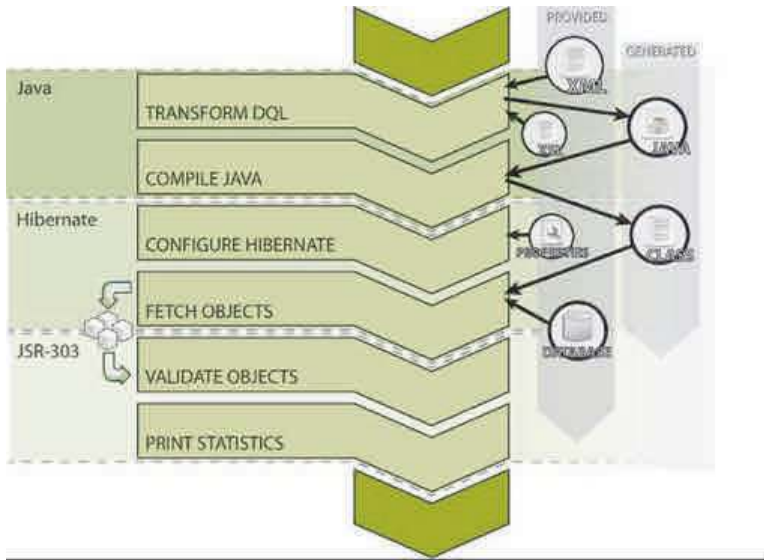


Abbildung 12.6: Architektur von Qualitator

Themenkreis „Datenqualität“, erarbeitete es unter anderem den Prototyp „Qualitator“. Dieser ist ein kostenloses Werkzeug zur Definition und Anwendung von Datenqualitätsmetriken. Zu überprüfende Qualitätskriterien werden in einer eigenen einfachen XML-Beschreibung (MetricML) hinterlegt. Mittels standardisierter Technologien (XSLT, JSR 303 und JPA) ist eine schlanke Anwendung in der Lage, in einem relationalen Datenbanksystem gespeicherte Daten gegen definierte Qualitätskriterien zu prüfen und so Messwerte für den Qualitätszustand der Daten bereit zu stellen.

Qualitator wurde gezielt für den Einsatz in ressourcen-knappen Unternehmen entworfen. Die einfache Definition von Qualitätsbedingungen und Bedienbarkeit zielt auf den effektiven Einsatz im Umfeld von Klein- und Mittelunternehmen ab. An dieser Stelle soll nicht unerwähnt bleiben, dass für die bewusste Einfachheit der Beschreibung der Qualitätskriterien mittels MetricML ein Preis zu zahlen ist. MetricML erfüllt nicht alle der bereits erwähnten Axiome einer Datenqualitätsmetrik. Eine detaillierte Beschreibung der Thematik findet sich in (Zugaj, 2013).

Ein weiteres Lösungsszenario zur Hebung der Qualität in relationalen Datenbanken stellt „Database-Refactoring“ dar. Database Refactoring

ist definiert als eine Änderung von Daten und Datenstrukturen, die die Semantik von Verhalten und Information eines Informationssystems unverändert lässt (Ambler, 2006).

Die manuelle Durchführung von Database-Refactoring ist für KMUs aufgrund der Ressourcen-Intensität keine Option. Hier bedarf es der Automatisierung. Im Vordergrund steht dabei aber nicht die konkrete Änderung von Daten oder Datenstrukturen einer Datenbank. Vielmehr handelt es sich um einen Prozess, der in mehrere Phasen unterteilt, von passenden Werkzeugen betreut werden muss.

Wir unterscheiden dabei eine Analysephase, in der Problem der Qualität von Daten und Datenstrukturen ausfindig gemacht werden müssen. Danach eine Definitionsphase, in der die anzuwendende Verbesserung – das konkrete Refactoring – ausgewählt wird. Nachdem Änderungen an Daten und Strukturen durchgeführt wurden, müssen die (mobilen) Anwendungen, die auf diese zugreifen, angepasst werden. Diese Phase wird „Transitions-Phase“ genannt, da hier die alten Daten bzw. Datenstrukturen noch nicht entfernt werden dürfen. Die Anpassung aller mit den Daten arbeitenden Anwendungen muss erst abgewartet werden. Nach deren Abschluss folgt die „Audit-Phase“ in der Zugriffe auf die alten Datenzustände vor dem Refactoring nicht mehr zulässig sind. Diese werden daher geloggt und gemeldet.

Auch hier war es im Rahmen anwendungsorientierter Forschung des Institut Internet-Technologien & -Anwendungen der FH JOANNEUM möglich, eine prototypische Lösung zu entwickeln. Der in (Zugaj, 2014) vorgestellte Prototyp ist in der Lage die Datenbankadministratoren in allen vier der gerade eben beschriebenen Phasen zu unterstützen (siehe Abbildung 12.7).

Feldversuche haben gezeigt, dass volle Automatisierung von Datenstrukturänderungen bei Datenbankadministratoren nicht erwünscht ist. Die Möglichkeit von manueller Kontrolle und manuellen Eingriff muss bestehen bleiben. Zu diesem Zweck erzeugt der Prototyp in den einzelnen Phasen nur Skript-Dateien, deren Ausführung manuell freigegeben werden muss. In Summe steht dem KMU so ein Werkzeug zur Verfügung, das ausgefeilte Strukturänderungen zur Qualitätsverbesserung mit minimalem

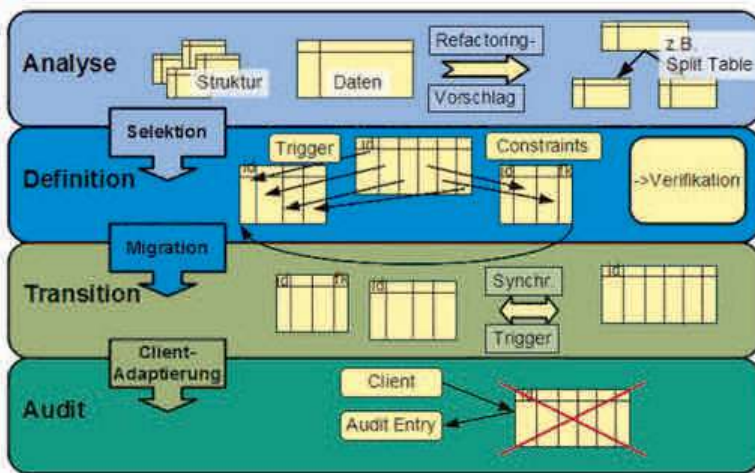


Abbildung 12.7: Phasen und Übergänge beim Database-Refactoring

Ressourcenaufwand ermöglicht.

12.5 Conclusio – ein auflösbarer Widerspruch

Die Anforderungen von KMUs an Informationssysteme mit mobilen Front-Ends werden stark durch begrenzte Ressourcen und Kapazitäten diktiert. Datenqualität kann in diesem Kontext vorrangig nur durch frei zugängliche, einfach zu bedienende Werkzeuge unterstützt werden. Dass dieses Thema trotzdem nicht vernachlässigt werden darf, bestätigen internationale und nationale Untersuchungen nachdrücklich. Forschungsarbeiten und Prototypen des Instituts Internet-Technologien & -Anwendungen zeigen auf, wie der scheinbare Widerspruch zwischen KMUs und mobilen Informationssystemen aufgelöst werden kann. Der nächste Schritt – dieses Wissen in die Klein- und Mittelbetriebe hineinzutragen und diese beim Prozess der kritischen Erhöhung der Wettbewerbsfähigkeit durch systematische Datenqualitätssicherung zu begleiten – kann nun begonnen werden.

Literaturverzeichnis

- Ambler, Scott J. (2006). *Refactoring Databases: Evolutionary Database Design*. Prentice Hall. ISBN: 9780321774514.
- Batini, Carlo, Cinzia Cappiello, Chiara Francalanci und Andrea Maurino (2009). „Methodologies for Data Quality Assessment and Improvement“. In: *Computing Surveys* 41.3, 16:1–16:52. DOI: 10.1145/1541880.1541883.
- Heinrich, Bernd, Marcus Kaiser und Mathias Klier (2007). „Metrics for Measuring Data Quality – Foundations for an Economic Data Quality Management“. In: *Proc. 2nd International Conference on Software and Data Technologies ICSOFT 2007*. Hrsg. von Joaquim Filipe, Boris Shishkov und Markus Helfert. Bd. ISDM/EHST/DC. INSTICC Press, S. 87–94. ISBN: 978-989-8111-07-4.
- Würthele, Volker Gerhard (2003). „Datenqualitätsmetrik für Informationsprozesse“. dissertation. Der Eidgenössischen Technischen Hochschule Zürich. ISBN: 3-8334-0345-4. DOI: 10.3929/ethz-a-004650156. URL: <http://e-collection.library.ethz.ch/eserv/eth:26911/eth-26911-01.pdf>.
- Zugaj, Wilhelm (2013). „Ensuring Data Quality with Hibernate and JSR 303“. In: <http://unitech.tugab.bg/docs/doc101.pdf>. Bd. 2. Technical University of Gabrovo, S. 289–294.
- (2014). „Enhancing data quality in the field - towards applied database refactoring“. In: *International Scientific Conference UNITECH 2014*. Bd. II. Technical University of Gabrovo, S. 269.

„Wir von der ERA Elektro Recycling Austria haben uns als KMU entschieden mit der FH JOANNEUM „mobile“ zu gehen. Da wir im Auftrag von Elektro- und Elektronikherstellern für die Sammlung und Verwertung von Elektrogeräten in Österreich verantwortlich sind, stellen gesetzliche Vorgaben ziemlich große Anforderungen an Logistik und Datenmanagement. Mit Hilfe von KMU goes mobile gelingt es uns neuerlangtes Know-How unserer Community mit einem „Fingerwisch“ zur Verfügung zu stellen.“

DI Thomas Maier

Geschäftsführer ERA GmbH

Johannes Feiner, Projektleiter von KMU goes mobile, studierte Telematik an der TU Graz. Er unterrichtet und forscht am Institut für Internet-Technologien & -Anwendungen an der FH JOANNEUM. Seine Forschungsschwerpunkte sind Mobile Software Development und Usability.