

How does an App Store / Market work?

Thomas Behrens
Betreuer: Dipl.-Inf. Marc-Oliver Pahl
Seminar Future Internet SS2012
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: behrenst@in.tum.de

KURZFASSUNG

Ein Application Store (App Store) ist ein internetbasiertes Vertriebssystem für Software, das vor allem durch die steigende Verbreitung von Smartphones und Tablet-PCs Beliebtheit erlangt. Es ermöglicht dem Anwender sein Gerät mit zusätzlichen Programmen (Apps) funktional zu erweitern. Die vorliegende Arbeit stellt einzelne Bestandteile des Systems „App Store“ vor und geht auf Fragestellungen und Konzepte hinter diesem System ein. Diese sind für die Implementation eines App Store wichtig, aber auch um die Komplexität eines solchen Systems zu verstehen. Außerdem wird auf Vertreter sowie deren Unterschiede und Gemeinsamkeiten eingegangen.

Schlüsselworte

App Store, App Market, Google Play, Nokia Ovi Store, Microsoft Marketplace, BlackBerry App World, App

1. EINLEITUNG

Der erste App Store wurde im März 2008 [1] von Apple unter dem Namen „Apple App Store“ eröffnet. Seitdem wird von verschiedenen Unternehmen versucht eigene App Stores zu implementieren und das Konzept zu verbessern. Viele Konzepte, auf denen der App Store beruht, existierten schon vorher und wurden von Apple lediglich weiterentwickelt und auf mobile Endgeräte portiert. Seit 2001 wird versucht mit der Einführung neuer Geschäftsmodelle auf die hohe Anzahl an Raubkopien von Musik und anderen digitalen Inhalten im Internet zu reagieren. Die ersten Vertreter von sogenannter Electronic Software Distribution (ESD) sind die Musikanbieter Napster (seit 2001) [2], Musicload (seit 2003) [3] und Apples iTunes Store (seit 2003) [4]. Auch werden Computerspiele seit dieser Zeit über eine Form von App Store für PCs verteilt (Steam: seit 2003) [5]. Die stetig steigenden Downloadzahlen u.a. in Apples App Store zeigen wie erfolgreich das Konzept ist. [6]. Die Anzahl der Apps vervielfacht sich seit der Einführung des Stores jedes Jahr, genauso wie die Zahl der Entwickler und der mit dem Store generierte Umsatz, der 2011 bei 1,78 Milliarden US-Dollar lag [7, 8, 9].

Als Folge von Apples Erfolg mit diesem Konzept und der daraus resultierenden Nachfrage nach deren Geräten [10, 11] gab es bald viele Konkurrenten. Im August 2008, im selben Jahr wie Apple, eröffnete Google den Android Market (seit 2012 Google Play). Neben diesen zwei größten, gibt es mehrere kleinere App Stores, im Juni 2010 circa 80 [12]. Da nur wenige dieser App Stores wirklich erfolgreich wurden, muss es Unterschiede geben, die den Erfolg wesentlich beeinflus-

sen. Solche Unterschiede gibt es in Realisierungskonzepten, die sich u.a. im Tätigkeitsfeld der App Store-Betreiber unterscheiden. Damit werden bestimmte Spielräume für Entwickler und Möglichkeiten für Endanwender festgelegt. Auch gibt es für App Stores technologische Grundkonzepte und Funktionen, die vorhanden sein müssen, aber verschieden realisiert werden können.

Im Folgenden werden zunächst Grundtechnologien von App Stores erläutert und aktuelle Entwicklungen zu deren Verbesserung vorgestellt. Kapitel 3 beschäftigt sich mit Vergütungsmodellen, also den verschiedenen Möglichkeiten für App Store-Betreiber und App-Entwickler, mit ihrer Arbeit Geld zu verdienen. Darauf folgend werden in Kapitel 4 Betreibermodelle vorgestellt und verglichen. In Kapitel 5 werden die bekanntesten App Stores vorgestellt und auf die größten Unterschiede und Gemeinsamkeiten zwischen diesen eingegangen. Die Arbeit endet mit der Vorstellung von ähnlichen Arbeiten und einem Ausblick auf die Entwicklung von App Stores.

2. TECHNISCHE GRUNDLAGEN EINES APP STORES

Ein App Store ist ein System zur Erweiterung eines Geräts mit Apps und deren Verwaltung. Die Funktionalität eines App Stores beschränkt sich aber nicht auf die Bereitstellung von zum jeweiligen Gerät kompatible Anwendungen. Dieser Abschnitt beschreibt verschiedene Funktionen, die ein App Store typischerweise bereitstellt, bzw. die bei der Realisierung zu berücksichtigen sind. Falls es neuartige Entwicklungen zu bestimmten Funktionen gibt werden diese vorgestellt. Dabei wird vor allem auf die Nutzerfreundlichkeit und die Einfachheit der einzelnen Funktionen geachtet. Dies liegt darin begründet, weil das Ziel eines App Stores es ist, durch möglichst alle Nutzergruppen bedienbar zu sein, egal wie hoch das technische Verständnis ist. Apple bewirbt seinen Mac App Store beispielsweise damit, dass man sich „die gewünschten Apps einfacher holen [kann] als je zuvor. Keine Verpackung, keine weiteren CDs oder DVDs und keine zeitraubende Installation. Ein Klick reicht, um Apps auf deinen Mac zu laden und zu installieren.“ [13]

Die ausführliche Vorstellung der Grundfunktionen in App Stores wird mit drei wesentlichen Funktionen eines App Stores, dem Finden einer bestimmten App, dem Herunterladen und der anschließenden Installation begonnen. Anschließend werden die Aktualisierung und die Funktionen für Bewertungen und Kommentare vorgestellt und erläutert, welchen

Einfluss letztere innerhalb des App Stores haben. Darauf werden Sicherheitskonzepte vorgestellt, die bei der Realisierung eines App Store mitberücksichtigt werden sollten. Dabei wird auf Kernunterschiede in aktuellen Realisierungen eingegangen. Danach werden Möglichkeiten vorgestellt, App-Entwickler zum Programmieren für die eigene Plattform zu motivieren. Darauf folgen eine Vorstellung von Möglichkeiten der Qualitätssicherung der Apps, geräteunabhängige Apps und Vorteile, die sich durch die Unterstützung eines Stores von mehreren Geräten ergeben.

2.1 Suchen

Die Suchfunktion ist eine Grundvoraussetzung um aus vielen tausenden Apps die Gewünschte finden. Mit einer stetig steigenden Zahl von derzeit über 500.000 Apps im Apple App Store ist dies keine leichte Aufgabe [14]. Ziel dieser Funktion ist es ohne spürbare Wartezeit ein möglichst gutes Ergebnis anzuzeigen. Damit wird Frust verhindert, der ansonsten bei Nutzern zu einer ablehnenden Haltung in Bezug auf den App Store führen würde. Heutige App Stores optimieren dies, indem sie schon während der Eingabe des Suchbegriffs (inkrementelle Suche) per Autovervollständigung versuchen, das richtige Suchergebnis anzuzeigen. In dieselbe Richtung geht der Versuch, Rechtschreibfehler im Suchbegriff zu erkennen und mit Hilfe einer Autokorrekturfunktion zu berichtigen [15]. Des Weiteren lassen sich Bewertungen dazu einsetzen, die Suchergebnisse in ihrer Relevanz zu sortieren und damit zu verbessern. Als weitere Eingabemöglichkeit für Suchbegriffe ist teilweise eine Spracheingabe vorhanden [16].

2.2 Herunterladen

Wenn der Anwender eine gewünschte App gefunden hat, muss sie als danach aus dem Internet heruntergeladen werden. Da eine Benutzerinteraktion für diesen Vorgang überflüssig ist und außerdem zusätzliche Nutzerdialoge nur zu mehr Komplexität führen würden, erfolgt der Download automatisch. Um diese Funktion dauerhaft gewährleisten zu können, muss eine ausreichende Infrastruktur zur Verfügung stehen. Es gibt mehrere Entwicklungen, die die Infrastruktur entlasten und damit die Kosten für App Store-Betreiber verkleinern. Dazu zählen z.B. dynamisch erweiterbare Cloud-Server, die Verwendung von Content Delivery Networks (CDN) und GeoTargeting.

2.3 Installieren

Um die heruntergeladene App auszuführen, ist deren Installation nötig. Darin wird das komprimierte, heruntergeladene Paket entpackt und für die Ausführung auf dem Gerät konfiguriert. Für den Anwender bietet die Installation durch einen App Store den Vorteil, dass sie einfacher ist gegenüber anderen Installationen. Während z.B. unter Microsoft Windows in mehreren Dialogfenstern u.a. der vom Anwender gewünschte Speicherort und Umfang der Installation abgefragt wird, ermöglicht beim App Store eine festgelegte Programmierschnittstelle (API), auf nötige Benutzerinteraktionen während der Installation vollständig zu verzichten. Ein Nachteil davon sind die mangelnden Konfigurationsmöglichkeiten, die es bei Smartphones auf Dateiebene aufgrund von eingeschränkten administrativen Möglichkeiten kaum gibt. Diese werden aber nicht benötigt, da in den Spezifikationen festgelegt ist, wie Apps auf dem Gerät installiert werden sollen, also wie die empfangenen Daten entpackt und

in das Dateisystem integriert werden sollen [17]. Außerdem wird durch die API eine Laufzeitumgebung erzeugt, welche es ermöglicht, dass Apps auf verschiedenen Geräten mit unterschiedlicher Hardware gleich laufen. Die API stellt dabei für App Entwickler einheitliche Funktionen zur Verfügung, die Gerätehersteller an die jeweilige Hardware übersetzen müssen. Deshalb sind gerätespezifische Konfigurationen nur noch in Ausnahmen erforderlich.

2.4 Aktualisieren

Mit dem Aktualisieren der Apps nimmt der App Store dem Anwender eine sonst notwendige Aufgabe ab. Dies ist notwendig, da immer wieder Schwachstellen in Apps oder kritische Fehler auftreten und mit so behoben werden können. Aktualisierungen werden durch eine Bestätigung des Anwenders gestartet und laufen ab dem Zeitpunkt automatisch. Möglich ist auch wie bei Google Play die Option anzubieten, auf die Bestätigung zu verzichten und automatisch zu aktualisieren. Ein App Store nimmt dem Anwender bei der Aktualisierung viele Aufgaben ab. Das lässt sich am besten am Beispiel eines Nutzers mit einem Betriebssystem ohne App Store z.B. Microsoft Windows verdeutlichen. Dieser muss, wenn er seine Programme aktuell halten will, die aktuelle Version über das Internet suchen, diese mit der installierten vergleichen und im Fall eines vorhandenen Updates die neue Version heruntergeladen. Danach muss die alte Version deinstalliert und in mehreren Installationsschritten die neue Version installiert werden. App Stores führen diese Schritte automatisch ohne Eingreifen des Anwenders durch. Die Version jeder App wird im App Store mit der jeweils aktuellen verglichen und, falls eine neuere zur Verfügung steht, die Möglichkeit gegeben, ein Update durchzuführen.

2.5 Bewerten & Kommentieren

Um andere Nutzer vor schlechten Apps zu warnen bzw. sie auf gute aufmerksam zu machen, lassen sich Apps bewerten. Bei den zwei größten App Stores, dem Apple App Store und Android Play, aber auch anderen, finden Bewertungen auf einer Skala von 0 bis max. 5 Sternen statt. Der Durchschnitt dieser Bewertung wird in den Bestenlisten, sowie vor jeder Installation auf der Infoseite der App angezeigt. Es gibt aber eine Bewertungsverzerrung, da Nutzer vor allem besonders gute und besonders schlechte Apps bewerten. Außerdem bewerten nur eine sehr geringe Zahl an Nutzern eine App, in einem repräsentativen Test durchschnittlich nur zwischen 0,008 und 0,056 Prozent [18]. Um diesem Problem zu begegnen, kann man zusätzlich zu den oben beschriebenen expliziten Bewertungen implizite Bewertungen einführen. Dabei nimmt der App Store oder ein im Hintergrund laufendes Programm das Benutzerverhalten auf und sendet es zur Auswertung an einen Server. Implizite Bewertungen setzen sich zusammen aus Ereignissen wie z.B. Installationen, Aktualisierungen und Deinstallationen, sowie Erkenntnissen über die Installationsdauer, die Anzahl der durchgeführten Updates sowie die Häufigkeit und Dauer der jeweiligen Ausführung einer App. Diese Informationen werden an einen Server geschickt, auf welchem die Informationen dann zusammen mit denen von anderen Nutzern ausgewertet und abgeglichen werden. Damit ist es u.a. möglich Empfehlungslisten zu verbessern. Diese können einzelnen Anwendern Apps empfehlen, die andere User mit einem ähnlichen Verhalten und ähnlichen Apps gerne nutzen. [18]

Eng mit dem Rating verknüpft ist die Kommentarfunktion. Hier können Nutzer über ihre Erfahrungen mit der App berichten und bei ihnen aufgetretene Fehler melden. Der Entwickler kann so erkennen, was die Nutzer an der App kritisieren und an diesen Stellen die App verbessern [19]. Das ist z.B. bei Google Play wichtig, da Android auf vielen verschiedenen Plattformen von verschiedenen Herstellern läuft. Diese haben jeweils die Android-Version angepasst und verwenden zum Teil unterschiedliche Displayauflösungen, was teilweise zu Problemen und Layout-Fehlern bei bestimmten Apps führen kann [20].

Ein Problem stellt die Zusammengehörigkeit von Wertung und Kommentar dar [21]. Wird eine App in einer früheren Version mit vielen Fehlern schlecht bewertet, ist es für den Entwickler sehr schwer, eine gute Gesamtbewertung zu erlangen, obwohl die bewerteten Fehler schon behoben wurden. Die Möglichkeit, dass der Entwickler überholte Kommentare und Bewertungen entfernen kann, würde jedoch dazu beitragen, dass Bewertungen manipulierbar wären. Eine Lösung ist es, implizite Bewertungen zu sammeln und solche Wertungen weniger stark zu gewichten. Eine andere Möglichkeit ist, eine Bewertungsfunktion für Kommentare einzusetzen und besonders schlecht bewertete Kommentare zu entfernen oder abwertend zu gewichten.

2.6 Sicherheitskonzepte

Da Smartphones und andere mobile Geräte vollwertige Computer sind und viele Informationen über den Anwender speichern, müssen sie gut geschützt werden [22].

Zugriff auf die Daten kann ein Angreifer z.B. durch das Ausnutzen einer Sicherheitslücke bekommen. Um solche schnell zu schließen, ist die automatisierte Verteilung von Patches bzw. Updates über den App Store sinnvoll. Das bedeutet, die Updates werden aktiv dem Anwender gemeldet, ohne dass dieser nach Updates suchen muss. Ein Beispiel hierfür ist Steam, da sich hier Anwendungen nur starten lassen, wenn sie sich vorher auf den aktuellen Stand gebracht haben.

Ein anderes Sicherheitsfeature kann die Erstellung von Backups im App Store und deren Wiederherstellung sein. Entweder es werden dabei nur die installierten Apps gespeichert oder zusätzlich noch Einstellungen und gespeicherte Inhalte hinzugefügt. Man könnte nur speichern, welche Apps man installiert hat oder noch die Einstellungen und gespeicherten Inhalte dazu nehmen.

Ein großes Risiko steckt für den Anwender im Download von bösartigen Apps, z.B. Spyware oder Viren, aus dem App Store. Es gibt verschiedene Ansätze das Risiko durch solche Apps in den Griff zu bekommen. Google Play verwendet ein Suchprogramm mit dem Namen „Bouncer“. Das analysiert neu in den App Store eingestellte und vorhandene Apps bzw. deren Code und überprüft Entwickler-Accounts auf Auffälligkeiten. Laut Google ließ sich damit der Anteil dieser Apps innerhalb von 2011 um 40% reduzieren. [23]

Apple hingegen lässt nur zusätzlich von Personal überprüfte Apps in den Store [24]. Dazu waren im Mai 2010 40 Leute angestellt, die jeweils zu zweit jedes App vor der Zulassung in den Store überprüften [25]. Da diese 40 Personen aber an einem Tag ca. 2.000 Apps überprüfen, kann es vorkommen, dass sie trotz der zusätzlichen Prüfung schadhafte Apps übersehen. Zu den Prüfungsmechanismen sind kei-

ne Details bekannt, da diese dann leichter umgangen werden könnten. Es gibt aber so genannte Review Guidelines, die auflistet aus welchen Gründen eine App zurückgewiesen werden kann. Denen ist zu entnehmen, dass u.a. Abstürze und gewalttätige, rassistische oder jugendgefährdende Inhalte zum Ausschluss führen. Außerdem muss sie für den Anwender einen Mehrwert bieten und sich von bestehenden Apps unterscheiden. Kritisiert wird Apple dafür, dass Apps nach deren Ermessen und aus unverständlichen Gründen abgelehnt werden, z.B. weil sie sich zu wenig von existierenden und vor allem firmeneigenen Apps unterscheiden. [24]

Trotz aufwendiger Prüfungen gelangen, u.a. durch die große Zahl an neuen Apps, immer wieder Schadprogramme in den App Store. Deshalb schottet Android die Apps mit Hilfe von Sandboxing voneinander ab, so dass der Absturz einer App keine Auswirkung auf andere Apps hat. Somit gibt es auch jede App nur auf ihre eigenen Daten und nicht die anderer Apps zugreifen. [26]

Des Weiteren gibt es in Android ein Sicherheitsmodell, in dem jede App vor der Installation nach den erforderlichen Zugriffsrechten auf die Hardware fragt (Permission Modell) [26]. Wenn z.B. ein Spiel die Rechte für den Zugriff auf die Kontaktdaten erhalten möchte, sind Zweifel über die Absichten des Apps angebracht. Der Anwender kann so dubiose Apps erkennen und daraufhin entscheiden, ob er der App den Zugriff gestatten will. Kritisiert wird, dass die letzten beiden Schutzmechanismen bei Android nicht zwingend zwischen Anwendungen des gleichen Entwicklers gelten. Diese können über sogenannte Intents Daten austauschen, wodurch sich Zugriffsrechte kombinieren lassen und somit umgangen werden können. [26]

Deshalb wird an Software geforscht, die es dem Endanwender erlaubt zu sehen, auf welche Daten und Hardwarefunktionen das App wann zugreift, und welche Daten es versendet. Dazu wurde von der TU Berlin für Android eine Sandbox entwickelt (AASandbox), die es dem Anwender ermöglicht, für eine beliebige App eine statische und dynamische Analyse durchzuführen und bösartige Programme automatisch zu erkennen. Die statische Analyse wird dabei auf den Installationsdateien ausgeführt, die dynamische über den von Google zur Verfügung gestellten Android Emulator. [27]

Ein weiteres Sicherheitsfeature erlaubt es Google Play, als Malware erkannte Apps per Fernzugriff von allen Geräten zu löschen, auf denen diese App installiert ist [26].

2.7 Entwicklerunterstützung

Der Erfolg eines App Stores hängt neben der Qualität auch von der Anzahl der angebotenen Apps ab. Deshalb ist es für App Store-Betreiber wichtig, Entwicklern Anreize zu schaffen und sie bestmöglich zu unterstützen, Apps für die eigene Plattform zu programmieren.

Dazu gehört die Bereitstellung eines Software Development Kits (SDK), mit dem die Entwickler arbeiten können. Der Umfang der Programme in diesem Paket ist stark abhängig vom App Store-Betreiber. Im Idealfall finden sich ein guter Debugger, Bibliotheken, ein Emulator, eine ausführliche Dokumentation, Beispielcode und Anleitungen (Tutorials). Die Bibliotheken sind wichtig, um den standardisierten Zugriff auf Hardware zu vereinfachen und um die Entwick-

lungszeit zu verkürzen. In ihr enthalten sind Grundfunktionen des Geräts, wie z.B. Interface-Elemente, Zugriffe auf Telefonbuch/Kamera/Sensoren, die mit wenigen Befehlen in die eigene App integriert werden können. Eine ausführliche Dokumentation der zur Verfügung stehenden Schnittstellen und Richtlinien ist auch von Vorteil, da es z.B. die Einarbeitungszeit verkürzt. Mit vorhandener API und Bibliothek lässt sich u.a. auch die Benutzerführung homogener gestalten, was den Vorteil hat, dass Anwender schneller die Bedienung einer beliebigen App erlernen.

Unter bestimmten Voraussetzungen ist es für Entwickler einfacher mit einer App erfolgreich zu werden. Grundsätzlich sollte eine Nachfrage nach neuen, innovativen Apps vorhanden sein, also der Markt noch nicht übersättigt sein. Je mehr Anwender eine Plattform hat, umso mehr steigt die Wahrscheinlichkeit, mit der eigenen App erfolgreich zu sein. Die Betreiber von App Stores mit weniger Anwendern versuchen diesen Wettbewerbsnachteil u.a. durch direkte Zahlungen an gute Entwickler, auf die sie im Konkurrenzkampf dringend angewiesen sind, auszugleichen. Hier lassen sich Nokia und Microsoft nennen, die im März 2012 ein Förderungsprogramm für App-Entwickler in Höhe von 18 Mio. Euro beschlossen haben [28].

Des Weiteren ergeben sich durch eine Zusammenarbeit von App-Entwicklern und dem App Store-Betreiber beim Marketing für beide Seiten Vorteile. Der App Store-Betreiber hat das Ziel, seinen Store mit möglichst herausragenden Apps zu bewerben. Der App-Entwickler strebt eine möglichst hohe Aufmerksamkeit für seine App an. Es kann z.B. der App Store-Betreiber in einer Kategorie „Featured Apps“ für die App werben oder direkt im App Store Werbung dafür schalten und dafür z.B. einen höheren Anteil am Umsatz dieses Apps erhalten.

Für die Entwicklung von Apps ist es von Vorteil, die Anzahl der Geräte und deren Betriebssystemversionen zu kennen, die zurzeit auf den Store zugreifen. Damit kann der Entwickler abschätzen, wie groß die Zielgruppe bei bestimmten Hardwareanforderungen an die App ist. Außerdem kann damit die Entscheidung der Betriebssystemversion erleichtert werden, die zwar Zugriff auf zusätzliche Bibliotheksfunktionen liefert, aber inkompatibel zu älteren Geräten ist. Das ist auch für die Entwicklung von Updates interessant, wobei die Statistik über die Geräte auf denen die App installiert wichtiger ist. Dazu gibt es z.B. von Google das „Android Developer Device Dashboard“ [29] oder von Steam die „Steam Hardware & Software Survey“ [30].

Noch besser als nur die Geräte zu kennen, ist die Möglichkeit für unterschiedliche Versionen eines Betriebssystems seine App zur Verfügung zu stellen. Damit kann man Nutzern eines besseren Geräts eine anspruchsvollere Version zur Verfügung stellen, ohne die Anwender mit schwächeren Geräten zu verlieren.

Weil die Ablehnung einer App durch den App Store-Betreiber in vielerlei Hinsicht ein schwerer Schaden für den Entwickler ist, sollte außerdem der Review-Prozess so klar definiert und durchgeführt werden, dass eine Ablehnung für den Entwickler kalkulierbar und nachvollziehbar ist.

2.8 Qualitätssicherung

Es genügt aber nicht nur eine große Anzahl an neuen Apps, diese sollten auch die qualitativ hochwertig sein. Denn wenn der Großteil der Apps nur einen geringen Mehrwert bietet, als Negativbeispiel lässt sich hier eine „Farting App“ [24] nennen, ist der von einem Anwender gewonnene Gesamteindruck auch negativ. Wie bereits dargelegt, kann man die Zulassung zum Store verschärfen, und damit neben Viren und Trojanern auch schlecht programmierte Apps herausfiltern. Es ließe sich aber auch z.B. ein Zertifizierungsverfahren durchführen, bei dem sich Apps von einem Kontrollgremium gegen eine kleine Gebühr nach festen Kriterien auf Qualität von Software und Inhalt prüfen lassen. Dem Anwender kann damit die Möglichkeit geben werden, sich nur zertifizierte Apps anzeigen zu lassen. Eine solche Zertifizierung wäre aber recht aufwendig zu realisieren, da im optimalen Fall nicht nur überprüft wird, ob die App die Qualitätsstandards des App Stores erfüllt, sondern z.B. auch ob Anwendung für ihren Zweck effizient programmiert ist und wie hoch die Benutzbarkeit und Bedienbarkeit ist. Der TÜV Süd bietet z.B. eine App-Zertifizierung für derartige Kriterien an [31].

Zur Qualitätssicherung hat der Verband der deutschen Internetwirtschaft e.V. eine Umfrage durchgeführt. Es wurde ermittelt, „wie [man] die Qualitätssicherung bei mobilen Anwendungen in Shops sicherstellen [kann]“ [32]. Dabei sprachen sich über 70% der Befragten für eine Qualitätssicherung bei der Annahme in den App Store aus, fast 20% hielten eine Qualitätssicherung für nicht erforderlich und ca. 8% befürworteten verschiedene Qualitätsstufen, die entsprechend Geld kosten. Da aber solche Zertifizierungen sehr teuer sind, und damit die Vorteile des leichten Marktzugangs verschwinden würden, ist die Umsetzung, wie schon die Umfrage gezeigt hat, fragwürdig.

2.9 Geräteunabhängige Apps

Ein Problem der heutigen Smartphones ist die Beschränkung eines Apps auf normalerweise nur ein einzelnes Betriebssystem. Damit ist es für Entwickler unmöglich, ohne Adaption des Apps auf andere Betriebssysteme den Großteil des Marktes zu erreichen. Um diese Beschränkungen zu umgehen werden geräteunabhängige Programmiersprachen, wie z.B. Java, Flash oder HTML5 verwendet.

HTML5 ist die Weiterentwicklung der von Webseiten bekannten HTML-Sprache, bietet aber weitere Möglichkeiten der Programmausführung. Dabei findet nur eingeschränkter Hardwarezugriff statt, aber auf Funktionen wie Kamera und GPS kann über API-Schnittstellen trotzdem zugegriffen werden. Zulassungsbeschränkte App Stores wie z.B. der von Apple weigern sich u.U. solche plattformunabhängigen Apps aufzunehmen. Das liegt u.a. daran, dass diese Apps nicht auf eine bestimmte Plattform optimiert sind.

2.10 Unterstützung mehrerer Geräte

Ein weiteres häufig beworbenes Feature ist die Vernetzung mehrerer Geräte über den App Store. Derzeit ist standardmäßig die Lizenz, die man beim Kauf einer App erhalten hat, sowie eine Liste der kostenlos installierten Programme beim App Store-Betreiber gespeichert [33]. Das hat den Vorteil, dass beim Kauf eines Geräts mit demselben App Store auf Wunsch die bereits auf dem anderen Gerät installierten Programme überspielt werden können.

Ein weiterer Schritt in diese Richtung kommt durch die starke Verbreitung von Cloud-Diensten. Es ist möglich nach jedem Beenden einer App, im Falle von ausreichend Netzwerkkapazitäten (WLAN), diese in auf einen Server bzw. in die Cloud hochzuladen. Von dort wird diese an die anderen eigenen Geräte verteilt. Somit lässt sich an allen Geräten immer an dem aktuellen Stand und den aktuellen Einstellungen weiterarbeiten.

3. VERGÜTUNGSMODELLE

Zur Motivation von Entwicklern gehört es auch finanzielle Anreize zu schaffen. Ein App Store sollte deswegen attraktive Vergütungsmodelle anbieten, die für den Entwickler einfache Kalkulationen ermöglichen, um sein Risiko und seinen möglichen Gewinn einschätzen zu können. Allgemein gibt es bei der Vermarktung von Apps fünf Ansätze. Der Verkauf von Daten über den Anwender, Werbung, Partnerschaften, das Einwerben von Spenden und der Verkauf der Apps. Weil Spenden unüblich und unkalkulierbar sind und sich Partnerschaften immer zusätzlich zu den Ansätzen abschließen lassen, ergeben sich daraus drei Hauptmodelle, die zurzeit von den größten App Stores angeboten werden.

1. Kostenpflichtige App:

Eine Möglichkeit mit Apps Geld zu verdienen, ist die App zu verkaufen. Dabei lässt sich ab einem gewissen Bekanntheitsgrad des Entwicklers, z.B. „EA Games“, die App auch zu niedrigen Preisen verkaufen, um einen guten Umsatz zu erzielen. Das gilt auch für Apps mit einem wirklichen Mehrwert, z.B. als Ersatz für ein Navigationssystem. Damit erklärt sich auch der niedrige Durchschnittspreis für kostenpflichtige Apps von unter zwei US-Dollar in den größten Stores [34].

2. Kostenlose App mit der Möglichkeit, In-App-Inhalte zu kaufen:

Da der Kauf einer kostenpflichtigen App mit einer gewissen Überwindung verbunden ist, wurde das Modell der In-App-Käufe eingeführt. Dabei ist die App für den Anwender kostenlos, kann aber über zusätzliche Inhalte erweitert werden. Damit der App Store-Betreiber aber trotzdem am Gewinn des Entwicklers beteiligt bleibt, ist hier die gleiche Kostenaufteilung wie im kostenpflichtigen Modell üblich. Diese Art von Apps werden vor allem dort eingesetzt, wo die App selber als Store fungiert, z.B. zum Kauf von E-Books oder Musikstücken.

3. Sonstige kostenlose App:

Apps, bei denen der Anwender zu keinem Zeitpunkt zahlen muss, erwirtschaften den Gewinn anders. So können über eine Werbefläche, die während der App-Ausführung angezeigt wird, Kleinstbeträge verdient werden, die sich aber bei der vielfachen Verwendung der App zu großen Beträgen aufsummieren. Diese Art von Apps bieten sich für Entwickler an, die noch nicht so erfolgreich sind. Wenn es dem Entwickler gelingt viele Nutzer zu gewinnen, kann er sich über die vielen Kleinstbeträge aus den Werbeeinnahmen finanzieren.

Der App Store-Betreiber behält meist einen Anteil vom Gewinn des Entwicklers, um sich zu finanzieren. Im App Store

und bei Google Play erhält der Entwickler 70% und der Betreiber 30% der Einnahmen [35]. In anderen Stores kann die Aufteilung z.B. je nach Zahlungsart variieren.

Der App Store-Betreiber sollte verschiedene Zahlungsmöglichkeiten anbieten, z.B. Kreditkarte, Micropayment-Dienste und Prepaid. Zusätzlich kann bei Kooperation zwischen den Betreibern von Mobilfunk und App Store die Zahlung per Handyrechnung vereinbart werden. Je mehr Zahlungsmöglichkeiten es gibt, umso leichter wird es dem Anwender fallen zu zahlen. Bei den meisten Betreibermodellen muss der App Store-Betreiber für die Abwicklung der Zahlungen diese erst absichern.

4. BETREIBERMODELLE

Es existiert eine Vielzahl an unterschiedlichen App Store-Betreibermodellen in den zu diesen Themen entstandenen technischen Studien (siehe verwandte Arbeiten). Dabei sind die Unterschiede zwischen den Modellen gering. Der Grund für die vielen Modelle ist, dass sie nach unterschiedlichen Gesichtspunkten entstanden sind und deshalb auf unterschiedliche Weise auseinander dividiert wurden. Im Folgenden wird auf ein Modell eingegangen, das an der Freien Universität Brüssel entstand [36]. Es unterscheidet die Plattformen nach der Kontrolle des App Store-Betreibers über die Inhalte seines Stores und die Kontrolle über die Kundenbeziehung zum Endanwender.

4.1 Enabler Plattform

Die Enabler Plattform kontrolliert die Inhalte im App Store, um den eigenen Wert sicherzustellen. Es findet aber nicht zwingend eine Kundenbeziehung statt, der Anwender kann also auch über andere Quellen seine Apps beziehen. Für diesen Typ von Plattform ist es wichtig, dass der Betreiber zum einen gute Beziehungen zur Industrie hat, zum anderen aber auch Wissen und Erfahrung besitzt, um für Verbraucher und Entwickler attraktive Plattformen zu entwickeln. Es sollte also eine IT-Infrastruktur und Werkzeuge zur Verfügung gestellt werden, die den App-Entwickler während des gesamten Entwicklungsprozesses unterstützen. Damit versucht der Betreiber, die Plattform möglichst attraktiv und damit beliebt für Entwickler, Industriepartner und auch Anwendern zu machen. Eingesetzt wird dieser Plattfortmty häufig für mobile Betriebssysteme. Beispielsweise stellt Google mit Android und Microsoft mit Windows Mobile eine Enabler Plattform zur Verfügung. [36]

4.2 System Integrator Plattform

Wie man in Tabelle 1 sehen kann, kontrolliert die System Integrator Plattform Anwender und Inhalte und muss damit die meisten Dienste der hier vorgestellten Plattformen übernehmen. Zusätzlich zur Entwicklung der Plattform und deren Support werden Aufgaben des Geräteherstellers, die Kundenbetreuung sowie die aktive Vermarktung für das Anwerben neuer Entwicklern übernommen. Dabei muss der Plattformbetreiber nicht unbedingt das Gerät selbst fertigen, er kann es z.B. auch bei einem Hersteller in Auftrag geben und dann unter seinem Markennamen (Branding) vertreiben. Verfügt der Betreiber einer Plattform über genügend Kompetenzen, um die unterschiedlichen Dienste zur Verfügung zu stellen, kann er u.U. mit diesem Modell erfolgreich werden, was z.B. Plattformen wie Apples iPhone bzw.

iOS, Microsoft Windows Phone und Nokia Ovi gelungen ist. Vorteil ist, dass weniger Absprachen zwischen den einzelnen Teilnehmern nötig sind, da die meisten Kompetenzen beim Plattformbetreiber gebündelt sind. [36]

4.3 Neutral Plattform

Die Neutralen Plattformen bieten vom Typ her den anderen Teilnehmern am wenigsten Dienste an, sind also komplementär zum System Integrator-Modell zu sehen. Es gibt weder eine Kontrolle über die Inhalte noch über die Endanwender. Neutrale Plattformen stellen für die Entwicklung der Apps grundlegende Werkzeuge, meist auf Open Source-Basis, zur Verfügung um eine einfache Softwareentwicklung zu fördern. Den Entwicklern werden aber ansonsten keine weiteren Dienste zur Verfügung gestellt und es wird auch keine Kundenbeziehung aufgebaut. Die Neutrale Plattform ist häufig ein Zusammenschluss von Mobilfunknetzbetreibern (Mobile Network Operators, MNOs), Geräteherstellern und Gerätevertreibern. Dies beruht darauf, dass die Plattform selbst nicht auf viel Gewinn ausgelegt ist, aber alle Beteiligten die Möglichkeit haben, zusätzliche Dienste anzubieten, mit denen sie jeweils Gewinn erzielen können. Haupterfolgsfaktor ist die Zusammenarbeit zwischen den beteiligten Unternehmen und das Potential der einzelnen Unternehmen. Das Ziel des Zusammenschlusses kann zusätzlich auch das Erreichen neuer Standards und der Gewinn an technologischem Fortschritt sein. Beispielplattformen hiervon sind Bondi und die Linux Mobile (LiMo). [36]

4.4 Broker Plattform

Der Unterschied zwischen Enabler Plattform und Neutral Plattform ist die hinzugekommene Kontrolle über den Anwender. Hier hat der Betreiber keine Kontrolle mehr über die Inhalte in seinem Store, bietet diese aber den Endkunden an und erhält für jede Verkaufsabwicklung eine Provision. Die angebotenen Apps können je nach Betreiber unabhängig vom Endgerät sein, also mehrere Endgeräte unterstützen. Typischerweise setzen MNOs diese Art von Plattform um, weil sie Kunden mit unterschiedlichen Geräten haben und die Geschäftsbeziehung zu diesen Kunden aufrechterhalten wollen. Vertreter dieses Modelltyps sind der Vodafone Apps Shop, Handango und GetJar. [36]

Plattformtypen:		Kundenverhältnis zum Endanwender	
		Ja	Nein
Kontrolle über Inhalte	Ja	Enabler P.	System Integrator P.
	Nein	Neutral P.	Broker P.

Tabelle 1: Überblick über vorgestellten Plattformtypen

Die Entscheidung für den Plattformtyp hängt also stark von den Kompetenzen und den Zielen des App Store-Betreibers ab. Die klare Strukturierung lässt sich im Vergleich der vorhandenen App Stores gut veranschaulichen.

5. APP STORES IM VERGLEICH

Im Folgenden werden die bekanntesten App Stores vorgestellt und jeweils Merkmale herausgestellt, mit denen sie sich von ihrer Konkurrenz abheben.

5.1 Apple App Store

Bis März 2012 wurden ca. 25 Milliarden Apps heruntergeladen, was ihn zu einem der erfolgreichsten App Stores macht [37]. Da der Umsatz und damit die Verkäufe von iPhones, iPads und Macs immer weiter ansteigen [10, 11], ist auch kein Einbrechen der guten Zahlen zu erwarten. Dabei schafft Apple über iCloud die Möglichkeit, Apps zwischen diesen Produkten zu transferieren, und somit seine Kunden stärker an sich zu binden. Konkurrenz bekommt der App Store von alternativen App Stores wie z.B. Cydia, die über einen sogenannten Jailbreak auf Apple-Geräten installiert werden können. Diese Konkurrenz entstand u.a. aus verschiedenen Kritikpunkten, wie z.B. den für viele Entwickler unverständlichen Zugangsbeschränkungen und Zensur. Als Beispiel lässt sich die Entfernung von mehreren Apps des Konkurrenten Google im Nachhinein nennen oder die Verweigerung Apps aufzunehmen, die nach Ansicht von Apple eine Konkurrenz gegenüber den eigenen Produkten und Sicherheitsrichtlinien darstellen. Auch die umfassende Datensammlung wird stark kritisiert [38].

5.2 Android

Die größte Konkurrenz zu Apple ist momentan Googles Open Source-Plattform Android. Die Anzahl an Smartphones dieses Betriebssystems wächst aufgrund von verschiedenen Herstellern und Modellen in verschiedenen Preissegmenten sehr viel stärker als die von Apple iPhones [39]. Im Juli 2011 war der Marktanteil von Android mit über 44% schon mehr als doppelt so hoch wie der von iOS [40].

5.2.1 Google Play

Google verfolgt ein offeneres Konzept mit seinem offiziellen App Store als Apple. E dürfen zusätzliche App Stores auf den Geräten installiert und verwendet werden. Apps können z.B. auch ohne App Store installiert werden und Entwickler auf verschiedenen Wegen bei der Entwicklung unterstützt werden. Aus dem App Store werden Apps nur bei gemeldeten Verstößen oder einer Erkennung durch den Bouncer entfernt (siehe Kapitel 2.6)

5.2.2 Amazon Appstore for Android

Dieser App Store weitet Amazons Online-Vertriebssystem auf mobile Anwendungen aus und stellt zusätzliche Angebote dessen Gerät Kindle Fire zur Verfügung. Dabei ist es von Vorteil, dass viele schon einen Benutzeraccount bei Amazon haben und dort Zahlungsmittel hinterlegt haben. Amazon versucht bei der Zulassung zum Store ähnlich wie Apple die Apps zu überprüfen. Kritik kommt vor allem für ein Modell, nach dem Amazon eine beliebige App einen Tag kostenlos anbietet, was ohne Absprache mit dem Entwickler passiert und dieser dafür auch nicht entlohnt wird [41].

5.3 Windows Phone Marketplace

Der Windows Phone Marketplace kam 2010 für Geräte des Windows Phone 7 auf den Markt. Neben den üblichen Funktionen gibt es die Möglichkeit Apps mit begrenzter Laufzeit und Demo-Apps herunterzuladen und so auszuprobieren bevor man sie kauft.

5.4 BlackBerry App World

Ein weiterer Smartphone-App Store ist die von RIM für das Handybetriebssystem BlackBerry OS entwickelte BlackBerry App World. Besonderheiten sind die Möglichkeiten Apps an seine Freunde zu verschenken und neben Kreditkarte auch über Telefonrechnung oder PayPal zu zahlen. Obwohl es den Store erst seit 2009 gibt, ist er für Entwickler hoch profitabel. Mehr als 13% der Entwickler sollen laut einer Studie mit ihrem App bereits mehr als 100.000 Dollar verdient haben. [42]

5.5 Nokia OVI Store

Der mittlerweile in Nokia Store umbenannte App Store bietet Apps für die Betriebssysteme Symbian, Maemo und MeeGo an. Es ist aber auch möglich, andere Inhalte wie z.B. Java ME-, Flash-Anwendungen und Klingeltöne anzubieten. Als Besonderheit lässt sich das „Store-in-Store“-Konzept sehen, mit dem Nokia MNOs die Möglichkeit bietet, eigene Stores dort zur Verfügung zu stellen. [43]

5.6 Andere App Stores

Neben den größten App Stores für mobile Endgeräte gibt es noch ähnliche Konzepte auf anderen Systemen. Da diese einige interessante Besonderheiten aufweisen werden sie im Folgenden kurz vorgestellt werden.

5.6.1 Steam

Steam ist mit über 35 Mio. Benutzerkonten einer der führenden App Stores für Spiele an PCs. Das System unterstützt Raubkopie-Erkennung, also die Überwachung der Spiele auf Manipulationen, Online-Vertrieb, Verteilung und Aktualisierung der Spiele und mehrere soziale Features, wie z.B. die Möglichkeit Gruppen zu bilden und miteinander zu chatten. [5]

5.6.2 Origin

Dieses von Electronic Arts (EA) betriebene System orientiert sich an der Funktionalität des bereits vorgestellten Steam, hat aber mit 3,9 Mio. Nutzern wesentlich weniger Nutzer als dieses. Kritik bekommt das System, weil man für Software von EA gezwungen wurde, dieses System zu benutzen und der Lizenzvertrag EA weitreichende Rechte auf dem Computer des Anwenders einräumte. Dazu gehört z.B. die Sammlung von technischen und verwandten Informationen und deren Übertragung an EA und deren Partner. [44]

5.6.3 Ubuntu Software Center

Dieses für das Betriebssystem Ubuntu konzipierte Software Center ist das erste, was das App Store Prinzip vollständig für PC-Betriebssysteme umsetzt. Das System lässt sich über eine App Store-typische Oberfläche bedienen und greift bei der Installation von neuen Apps auf die Paketverwaltung Synaptic zurück [45]. Dabei fällt der große Nachteil von früheren Linux-Distributionen weg, dass sich Anwendungen nur schwer, z.B. über bestimmte Makefiles, installieren lassen, welche auch nicht immer auf jeder Distribution richtig funktioniert haben. Mit der gewonnenen Einfachheit, bringt das kostenlose System teurere Konkurrenzsysteme wie Microsoft Windows und Apples Mac OS in Zugzwang. Nachdem bereits ein Mac App Store eingeführt wurde [13], wird auch Microsoft für die nächste Windows Version einen App Store anbieten [46].

5.6.4 GetJar

GetJar gehört zu den größten Multi-Plattform Stores mit über 2,5 Mrd. Downloads und über 400.000 Entwicklern [47]. Wesentlicher Erfolgsfaktor ist die Unterstützung sehr vieler Plattformen u.a. Java ME, BlackBerry, Symbian, Windows Mobile und Android.

6. VERWANDTE ARBEITEN

Aufgrund der vielen unterschiedlichen Teilkomponenten von App Stores, konnte im Rahmen der Arbeit nicht auf alle im Detail eingegangen werden. Als verwandte und weiterführende Arbeiten seien folgende empfohlen:

- In [36] wird das unter „Betreibermodelle“ erklärte Modell ausführlicher vorgestellt und dabei u.a. noch genauer auf die nötigen Kompetenzen der Plattformbetreiber eingegangen, die sie für das jeweilige Modell mitbringen müssen.
- Auf das IISI(n)-Modell, ein Modell was andere Blickwinkel an App Stores beleuchtet, wird in [48] genauer eingegangen. Dort werden die zwei Plattformen Nokia OVI und Apple App Store genauer miteinander verglichen, wodurch sehr gut kenntlich gemacht wird welche Aspekte dieses Modell betrachtet.
- Bei [49] handelt sich um eine Arbeit, die für den Erfolg von App Stores wesentliche Aspekte vorstellt. Dabei wird vor allem auf mögliche Geschäftsmodelle eingegangen und ein entwickeltes dreidimensionales Modell vorgestellt, dass die treibenden Faktoren für Erfolg beschreiben soll.
- Wie bereits angedeutet gibt es eine Vielzahl an anderen Betreibermodellen und Aufteilungen in diese Modelle, auf die aufgrund der begrenzten Länge nicht eingegangen werden konnte. Bei Interesse seien die folgenden Arbeiten empfohlen: [50, 51, 52, 53, 54, 55].
- In [26] werden Sicherheitsmechanismen auf Android-Systemen detailliert behandelt. Mögliche Angriffe werden dabei vorgestellt und Maßnahmen aufgezeigt, wie diese verhindert werden können.

7. FAZIT & AUSBLICK

Hinter einem App Store, wie ihn ein Anwender wahrnimmt, stecken eine große Zahl an Funktionen und technischen Hintergründe. Durch die deutliche Vereinfachung für den Anwender im Umgang mit Programmen ist sowohl der App Store, als auch die App an sich zum Erfolgsmodell geworden. Die Grundfunktionen sind größtenteils mit einem Klick ausführbar, man muss sich weder um Updates kümmern, noch darum ob ein Programm überhaupt die Mindestanforderungen für ein Gerät erfüllt. Größter Nachteil von heutigen App Stores ist noch die große Inhomogenität bei den zur Programmierung der Apps für einen Store zugelassenen Programmiersprachen. Die Zukunft wird zeigen, ob sich Entwickler in absehbarer Zeit darauf konzentrieren können, gute Apps und nicht nur eine App in 7 verschiedenen Programmiersprachen zu schreiben. Denn obwohl jede Plattform so viele gute Apps wie möglich haben will, gibt es bislang noch keine großen Kooperationen zwischen den größten App Stores, um eine Unterstützung von Apps anderer Stores zu gewährleisten.

Obwohl es scheint als wäre der Markt an App Stores gesättigt, werden vielleicht noch weitere auf den Markt drängen und das erfolgreiche Modell an verschiedenen anderen Stellen in unserem Alltag integrieren. Ob es irgendwann ganz normal sein wird über App Stores Autos, Drucker oder Kameras um Funktionen zu erweitern, lässt sich nicht sagen. Fest steht, dass es sich um ein Erfolgsmodell handelt, das sich noch stärker verbreiten wird.

8. LITERATUR

- [1] Apple, "Apple Announces iPhone 2.0 Software Beta," 6. März 2008. <http://www.apple.com/pr/library/2008/03/06Apple-Announces-iPhone-2-0-Software-Beta.html>.
- [2] Roxio, "Napster's Launch Party To Feature Hottest New Music Artists," 23. Oktober 2003. http://blog.roxio.com/press/divisioninvestor/2003/10/napsters_launch_party_to_feature_hottest_new_music.html.
- [3] Musicload, "Musicload – Zahlen und Fakten," 15. Juni 2011. <http://musicload.newsroomloads.de/facts/musicload-zahlen-und-fakten/06/2011/>.
- [4] Apple, "Apple Launches the iTunes Music Store," 28. April 2003. <http://www.apple.com/pr/library/2003/04/28Apple-Launches-the-iTunes-Music-Store.html>.
- [5] Valve, "Steam Client Released," 12. September 2003. <http://store.steampowered.com/news/183/>.
- [6] Apple, "Apple's App Store Downloads Top 15 Billion." Online, Juli 2011. <http://www.apple.com/pr/library/2011/07/07Apples-App-Store-Downloads-Top-15-Billion.html>.
- [7] Statista (über 148Apps.biz), "Statistik: Anträge auf Freigabe von neu entwickelten Apps im iTunes App Store von 2008 bis 2011." <http://de.statista.com.eaccess.ub.tum.de/statistik/daten/studie/157931/umfrage/antraege-der-entwickler-auf-veroeffentlichung-von-apps/>.
- [8] Statista (über 148Apps.biz), "Statistik: Anzahl verfügbarer Apps im US iTunes App Store bis Juli 2011 (in 1.000)." <http://de.statista.com.eaccess.ub.tum.de/statistik/daten/studie/157934/umfrage/anzahl-der-apps-im-itunes-app-store-seit-2008/>.
- [9] Statista (über IHS), "Statistik: Globaler Umsatz ausgewählter App-Stores in den Jahren 2009 und 2010 in Millionen US-Dollar." <http://de.statista.com.eaccess.ub.tum.de/statistik/daten/studie/180896/umfrage/weltweiter-umsatz-fuehrender-app-stores-seit-2009/>.
- [10] Statista (über Apple), "Statistik: Absatz von Apples iPhone weltweit vom 3. Geschäftsquartal 2007 bis zum 1. Geschäftsquartal 2012 (in Millionen Stück)." <http://de.statista.com.eaccess.ub.tum.de/statistik/daten/studie/12743/umfrage/absatz-von-apple-iphones-seit-dem-jahr-2007-nach-quartalen>.
- [11] Statista (über Apple), "Statistik: Absatz von Apples iPhone weltweit vom 3. Geschäftsquartal 2007 bis zum 1. Geschäftsquartal 2012 (in Millionen Stück)." <http://de.statista.com.eaccess.ub.tum.de/statistik/daten/studie/12743/umfrage/absatz-von-apple-iphones-seit-dem-jahr-2007-nach-quartalen/>.
- [12] T. Husson, "App store markets overhyped," Juli 2010. <http://cat-iqconference.com/2010/07/26/app-store-markets-overhyped/>.
- [13] Apple, "App Store - Tolle Mac Apps," 30. März 2012. <http://www.apple.com/de/mac/app-store/great-mac-apps.html>.
- [14] Apple, "The App Store," 30. März 2012. <http://www.apple.com/iphone/built-in-apps/app-store.html>.
- [15] Apple, "Apple in Education," 23. März 2012. <http://www.apple.com/education/special-education/>.
- [16] Google, "Just speak it: introducing Voice Actions for Android," August 2010. <http://googlemobile.blogspot.de/2010/08/just-speak-it-introducing-voice-actions.html>.
- [17] Google, "Debugging: Using the Dev Tools App," 20. März 2012. <http://developer.android.com/guide/developing/debugging/debugging-devtools.html>.
- [18] A. Girardello and F. Michahelles, "Explicit and Implicit Ratings for Mobile Applications," in *GI-Jahrestagung 2010*, pp. 606–612, Information Management - ETH Zürich, September 2010.
- [19] A. Hammershoj, "Challenges for mobile application development," in *Intelligence in Next Generation Networks (ICIN), 2010 14th International Conference*, IEEE, Oktober 2010.
- [20] Google, "Android Developers - Dev Guide," 21. März 2012. <http://developer.android.com/guide/appendix/market-filters.html>.
- [21] C. Dellarocas, "The Digitization of Word-of-Mouth: Promise and Challenges of Online Reputation Systems," in *Management Science Vol. 49, No. 10, Special Issue on E-Business and Management Science*, pp. 1407–1424, INFORMS, Dezember 2001. <http://www.jstor.org/stable/4134013>.
- [22] L. H. Marcial, "A comparison of screen size and interaction technique: Examining execution times on the smartphone, tablet and traditional desktop computer," 28. September 2010. http://marcial.web.unc.edu/files/2011/05/Marcial_lit_review_for_cmte.pdf.
- [23] H. Lockheimer, "Android and Security - Official Google Mobile Blog: VP of Engineering, Android," 2. Februar 2012. <http://googlemobile.blogspot.de/2012/02/android-and-security.html>.
- [24] Apple, "App Store Review Guidelines," 30. März 2012. <http://developer.apple.com/appstore/guidelines.html>.
- [25] Heib, A., "Ein Blick hinter die Mauer des App-Stores," 2010. <http://www.tagesanzeiger.ch/digital/mobil/Ein-Blick-hinter-die-Mauer-des-AppStores/story/25463932>.
- [26] R. Fedler, C. Banse, C. Krauß, and V. Fusenig, "Android OS Security: Risks and Limitations," tech. rep., Fraunhofer-Einrichtung für Angewandte und Integrierte Sicherheit (Fraunhofer AISEC), Mai 2012. <http://www.aisec.fraunhofer.de/content/dam/aisec/de/pdf/tech%20reports/AISEC-TR-2012-001-Android-OS-Security.pdf>.
- [27] T. Blaesing, L. Batyuk, A.-D. Schmidt, S. Camtepe, and S. Albayrak, "An Android Application Sandbox

- system for suspicious software detection,” in *Malicious and Unwanted Software, 2010 5th International Conference on*, pp. 55–62, IEEE, Oktober 2010.
- [28] Aalto University: AppCampus, “Microsoft and Nokia to invest up to 18 million euros in mobile application development program at Aalto University,” 26.03.2012. <http://appcampus.aalto.fi/about>.
- [29] Google, “Android Developer Device Dashboard - Platform Versions,” 25.04.2012. <http://developer.android.com/resources/dashboard/platform-versions.html>.
- [30] Valve Corporation, “Steam Hardware & Software Survey: März 2012,” 25.04.2012. <http://store.steampowered.com/hwsurvey>.
- [31] “Softwarequalität - Prüfung nach DIN 12119 / EN 9241 | TÜV SÜD GRUPPE.” www.tuev-sued.de/ps/apps.
- [32] Arbeitskreis Mobile des eco - Verband der deutschen Internetwirtschaft e.V., “Expertenumfrage Mobile Applications,” 2010. http://mobile.eco.de/files/2011/04/Expertenumfrage_Mobile_Applications_22.pdf.
- [33] Apple, “iTunes Store - Bedingungen,” 13. April 2012. <http://www.apple.com/legal/itunes/de/terms.html>.
- [34] P. J. ZDNet, “Apple app prices rebound,” 13. März 2012. <http://www.zdnet.com.au/apple-app-prices-rebound-339318352.htm>.
- [35] C. Schmidt, *Digitaler Softwarevertrieb für mobile Endgeräte am Beispiel des Apple-App-Store*. GRIN Verlag, 26. Juli 2010.
- [36] V. Goncalves, N. Walravens, and P. Ballon, “„How about an App Store?” Enablers and Constraints in Platform Strategies for Mobile Network Operators,” in *Mobile Business and 2010 Ninth Global Mobility Roundtable (ICMB-GMR)*, pp. 66–73, IEEE, Juni 2010.
- [37] Statista (über Apple), “Statistik: Kumulierte Anzahl der weltweit heruntergeladenen Anwendungen aus dem Apple App Store in Milliarden (Stand: 03.März 2012).” <http://de.statista.com/eaccess.ub.tum.de/statistik/daten/studie/20149/umfrage/anzahl-der-getaetigten-downloads-aus-dem-apple-app-store/>.
- [38] Apple, “Apple Q&A on Location Data,” 27. April 2011. <http://www.apple.com/pr/library/2011/04/27Apple-Q-A-on-Location-Data.html>.
- [39] Statista (über Gartner), “Marktanteile der Betriebssysteme am weltweiten Absatz von Smartphones von 2009 bis 2011.” <http://de.statista.com/eaccess.ub.tum.de/statistik/daten/studie/12885/umfrage/marktanteil-bei-smartphones-nach-betriebssystem-weltweit-seit-2009/>.
- [40] comScore, “Android Captures #2 Ranking Among Smartphone Platforms in EU5,” 13. September 2011. http://www.comscore.com/Press_Events/Press_Releases/2011/9/Android_Captures_number_2_Ranking_Among_Smartphone_Platforms_in_EU5.
- [41] R. Kim, “Amazon’s Android Appstore, not so amazing,” 5. Juli 2011. <http://gigaom.com/2011/07/05/amazon-appstore-not-so-amazing/>.
- [42] Evans Data Corporation, April 2012.
- [43] Nokia, “Nokia builds new custom Ovi Store concept for Orange France and Deutsche Telekom,” 12. Mai 2011. <http://press.nokia.com/2011/05/12/nokia-builds-new-custom-ovi-store-concept-for-orange-france-and-deutsche-telekom/>.
- [44] E. A. (EA), “Electronic Arts Reports Q3 FY12 Financial Results,” 01. Februar 2012. <http://investor.ea.com/releasedetail.cfm?ReleaseID=644995>.
- [45] Ubuntu Wiki, “Softwarecenter,” 2012-03-30. <https://wiki.ubuntu.com/SoftwareCenter>.
- [46] Microsoft, “Discover apps in the Windows Store.” <http://windows.microsoft.com/en-US/windows-8/apps>.
- [47] GetJar, “About GetJar,” 30. März 2012. <http://www.getjar.com/about/>.
- [48] V. Tuunainen, T. Tuunainen, and J. Piispanen, “Mobile Service Platforms: Comparing Nokia OVI and Apple App Store with the IISIn Model,” in *Mobile Business (ICMB), 2011 Tenth International Conference on*, pp. 74–83, IEEE, Juni 2011.
- [49] T. Yamakami, “A Three-Dimension Analysis of Driving Factors for Mobile Application Stores: Implications of Open Mobile Business Engineering,” in *Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference*, pp. 885–889, IEEE, März 2011.
- [50] K. Kimbler, “App store strategies for service providers,” in *Intelligence in Next Generation Networks (ICIN), 2010*, pp. 1–5, IEEE, Oktober 2010.
- [51] O. Rugnon, M. Escudero, J. Rueda, and S. Shanmugalingam, “Toward Dynamic Business Models on marketplace environments,” in *Intelligence in Next Generation Networks (ICIN)*, pp. 1–6, IEEE, Oktober 2010.
- [52] J. Laugesen and Y. Yuan, “What Factors Contributed to the Success of Apple’s iPhone?,” in *Mobile Business and 2010 Ninth Global Mobility Roundtable (ICMB-GMR)*, pp. 91–99, Juni 2010. <http://ieeexplore.ieee.org/eaccess.ub.tum.de/stamp/stamp.jsp?tp=&arnumber=5494782>.
- [53] M. Chen and X. Liu, “Predicting popularity of online distributed applications: iTunes app store case analysis,” in *Proceedings of the 2011 iConference, iConference ’11*, (New York and NY and USA), pp. 661–663, ACM, 2011.
- [54] M. Cortimiglia, A. Ghezzi, and F. Renga, “Mobile Applications and Their Delivery Platforms,” *IT Professional*, vol. 13, no. 5, pp. 51–56, 2011.
- [55] M. Cusumano, “Platforms and services: understanding the resurgence of Apple: Combining new consumer devices and Internet platforms with online services and content is proving to be a successful strategy,” *Commun. ACM*, vol. 53, no. 10, pp. 22–24, 2010.