

# Inhaltsverzeichnis

|       |  |    |
|-------|--|----|
| 1     | Einleitung.....                            | 1  |
| 1.1   | Aufgabenstellung.....                      | 1  |
| 1.2   | Gliederung.....                            | 1  |
| 2     | Vorstellung der bestehenden Software.....  | 2  |
| 2.1   | Überblick.....                             | 2  |
| 2.2   | Verwendete Technologien.....               | 2  |
| 2.3   | Kartenformat.....                          | 2  |
| 2.4   | Sensoren.....                              | 3  |
| 3     | Gegenüberstellung mobiler Plattformen..... | 4  |
| 3.1   | Übersicht.....                             | 4  |
| 3.1.1 | Marktanteile.....                          | 4  |
| 3.1.2 | Apps in den Märkten.....                   | 5  |
| 3.2   | Plattformen im Detail.....                 | 6  |
| 3.2.1 | Android 2.x.....                           | 6  |
| 3.2.2 | Windows Phone 7.....                       | 10 |
| 3.2.3 | iOS 4.x.....                               | 14 |
| 3.2.4 | Symbian^3.....                             | 17 |
| 3.2.5 | Blackberry OS 6.....                       | 20 |
| 3.2.6 | Sonstige.....                              | 22 |
| 3.3   | Multi-Plattform Lösungen.....              | 23 |
| 3.3.1 | Übersicht.....                             | 23 |
| 3.3.2 | Java ME.....                               | 23 |
| 3.3.3 | Flash / Adobe AIR.....                     | 24 |
| 3.3.4 | .net Framework / mono.....                 | 24 |
| 3.3.5 | Qt.....                                    | 24 |
| 3.4   | Verfügbare Sensoren für Smartphones.....   | 24 |
| 4     | Analyse der Anforderungen.....             | 25 |
| 4.1   | Allgemeine Anforderungen.....              | 25 |
| 4.2   | Use Cases.....                             | 25 |
| 4.2.1 | Use Cases des Kernsystems.....             | 26 |
| 4.2.2 | Use Cases "Soziale Erweiterung".....       | 31 |

|       |   |    |
|-------|---|----|
| 4.3   | POI Kategorien .....  | 35 |
| 4.4   | Checkliste .....  | 36 |
| 4.5   | Zusammenfassung der Umfrage.....                              | 36 |
| 5     | Evaluierung.....  | 38 |
| 5.1   | Bewertungsrichtlinien.....                                    | 38 |
| 5.1.1 | Kriterien der Bewertung .....                                 | 38 |
| 5.1.2 | Das Bewertungsschema .....                                    | 41 |
| 5.2   | Evaluierung der Plattformen im Kontext der Anforderungen..... | 41 |
| 5.2.1 | Android 2.x .....   | 42 |
| 5.2.2 | Windows Phone 7 .....   | 45 |
| 5.2.3 | iOS 4.x.....  | 48 |
| 5.2.4 | Symbian^3.....  | 50 |
| 5.2.5 | Blackberry OS 6.....  | 53 |
| 5.3   | Ergebnis .....  | 55 |
| 6     | Beispielhafte Portierung.....                                 | 57 |
| 6.1   | Übersicht .....   | 57 |
| 6.2   | Screenshots.....  | 57 |
| 6.3   | Ansätze, Probleme und mögliche Lösungen .....                 | 61 |
| 6.3.1 | Deserialisierung.....   | 61 |
| 6.3.2 | Rendering.....  | 62 |
| 6.3.3 | Sensoren .....  | 62 |
| 6.4   | Ergebnisse der Machbarkeitsstudie .....                       | 62 |
| 7     | Diskussion .....  | 63 |
| 8     | Zusammenfassung .....   | 64 |
| 9     | Anhang .....  | 65 |
| 9.1   | Abbildungsverzeichnis .....                                   | 65 |
| 9.2   | Tabellenverzeichnis .....                                     | 66 |
| 9.3   | Quellenangaben .....  | 67 |

# 1 Einleitung

## 1.1 Aufgabenstellung

Diese Fachstudie beschäftigt sich mit einer möglichen Portierung eines bestehenden Software-Systems (TANIA – Tactile Acoustical Navigation and Information Assistant), welches am VIS (Institut für Visualisierung und Interaktive Systeme) an der Universität Stuttgart entstand. Dabei stellt sich die Frage, welche aktuellen mobilen Plattformen für eine Portierung am besten geeignet sind. Anders als bei TANIA soll das zukünftige System nicht mehr nur speziell sehbehinderte Menschen als Zielgruppe haben, sondern für jeden Studenten der Universität zur Verfügung stehen. Außerdem sollen in diesem Zusammenhang mögliche Anforderungen an das neue System untersucht und ausgearbeitet werden. Im Rahmen dieser Fachstudie sollen daher folgende Punkte untersucht und bearbeitet werden:

- Analyse des existierenden Systems für sehbehinderte Menschen
- Ermittlung der Anforderungen sehender Studierender und Erstellung typischer Use-Cases
- Untersuchung des bestehenden Kartenmaterials
- Mobile Plattformen, die sich für eine Portierung anbieten
- Einsatzmöglichkeiten der auf mobilen Systemen verfügbaren Sensoren
- Untersuchung des Aufwands für die Anpassung für Sehende
- Anforderung an die Benutzerinterfaces
- Beispielhafte Portierung

## 1.2 Gliederung

Das Dokument ist folgendermaßen gegliedert: Das erste Kapitel bietet eine Einführung zum Thema. Im zweiten Kapitel gibt es eine Übersicht der bestehenden Software. Hierbei werden zusätzlich zu allgemeinen Aspekten auch Besonderheiten und der aktuelle Stand der Entwicklung beschrieben. Das dritte Kapitel nimmt eine unabhängige Gegenüberstellung der verschiedenen mobilen Plattformen vor. Im vierten Kapitel werden die zu erwartenden Anforderungen vorgestellt.

Die Kapitel zwei bis vier dienen als Grundlage für das fünfte Kapitel dieser Fachstudie. In diesem Kapitel erfolgt die Evaluation der in Frage kommenden mobilen Plattformen. Dabei werden zuerst die verschiedenen Bewertungskriterien eingeführt und anschließend die einzelnen Plattformen anhand dieser Kriterien analysiert und bewertet. Als Ergebnis dieser Studie wird eine Empfehlung gegeben.

Im sechsten Kapitel folgt die Dokumentation einer beispielhaften Portierung auf die empfohlene mobile Plattform.

## **2 Vorstellung der bestehenden Software**

### **2.1 Überblick**

Das TANIA-System (Tactile Acoustical Navigation and Information Assistant [50]) ist aus dem Sonderforschungsbereich 627 hervorgegangen und ist Teil des Anwendungsprojektes ASBUS (Assistenz für sensorisch Behinderte an der Universität Stuttgart). Es wird am Institut für Visualisierung und Interaktive Systeme der Universität Stuttgart entwickelt.

Ziel des Projektes ist es Personen mit sensorischen Einschränkungen einen barrierefreien Zugang zu allen öffentlich zugänglichen Bereichen der Universität zu ermöglichen.

TANIA ist ein Navigationssystem auf der Basis einer genauen Kartierung der Universität. Zusätzlich können kontextrelevante Beschreibungstexte akustisch oder über eine Braillezeile an den Benutzer übermittelt werden.

Der Funktionsumfang der Software umfasst eine auf die Bedürfnisse von sehbehinderten Benutzern angepasste Navigation innerhalb von Gebäuden. Dabei orientiert sich die Software innerhalb von Gebäuden durch RFID-Sensoren und durch einen Beschleunigungssensor, der die Schritte des Benutzers erkennt und mit Hilfe eines Kompasses den zurückgelegten Weg in der Karte aufzeichnet. Die Software ist so in der Lage Informationen aus der Karte über Räume und Hindernisse an den Benutzer weiterzugeben.

In Zukunft soll die Kartierung der Uni vervollständigt und um studienrelevante Daten ergänzt werden. Es sollen zusätzlich Daten der Stadt Stuttgart und der Verkehrsbetriebe ergänzt werden, um den sehbehinderten Benutzern einen einfacheren und unabhängigeren Einstieg in das Universitätsleben zu ermöglichen.

Für weitergehende Informationen zum Projekt und zu seinem Entwicklerteam wurde deren Projektseite im Literaturverzeichnis verlinkt [50].

### **2.2 Verwendete Technologien**

Die derzeitige Software ist in C++ geschrieben und verwendet zur visuellen Darstellung das GUI Framework "QT". Als Zielplattform wurde bisher ausschließlich Windows fokussiert. Als Entwicklungsumgebung kommt das Visual Studio von Microsoft zum Einsatz. In das Projekt sind Bibliotheken von Drittherstellern eingebunden, darunter z.B. die Bibliotheken des RFID-SDK von Intel.

### **2.3 Kartenformat**

In Verwendung ist ein XML-Format, welches sich stark an den GML/cityGML Standard [1] anlehnt. Es existieren zweidimensionale Karten des Campus Vaihingen, sowie im speziellen des Informatik-Gebäudes. Diese Karten wurden bisher in mehreren Arbeitsschritten aus CAD-Zeichnungen importiert, zuerst in den cityGML Standard [2] exportiert und anschließend zum eigenen Format hin nachbearbeitet. Zum Teil sind in dem zweidimensionalen Kartenmaterial die Stockwerke mit ihren einzelnen Räumen modelliert enthalten. Hierbei wird für jedes Stockwerk eine eigene Kartendatei verwendet.

## **2.4 Sensoren**

Die bestehende Software setzt zur Navigation zu einem großen Teil auf in den Gebäuden angebrachte RFID-Chips, welche Informationen zu einzelnen Räumen enthalten und zur absoluten Positionsbestimmung und Synchronisierung dienen. Ebenfalls unterstützt wird auch ein GPS-Sensor, welcher im Außenbereich zur Positionsbestimmung genutzt wird. Außerdem erlauben Bewegungssensoren eine Approximation der Fortbewegung.

## 3 Gegenüberstellung mobiler Plattformen

### 3.1 Übersicht

Im nachfolgenden Kapitel beschäftigen wir uns mit der Betrachtung der einzelnen mobilen Plattformen. Dabei werden einige Metriken zur Gegenüberstellung der einzelnen Plattformen aufgezeigt. Diese enthalten die Verkaufszahlen der einzelnen Smartphone-Systeme aus dem Bestand der Firma Canalys und Daten zur Anzahl der in den verschiedenen in den zentralen Märkten verfügbaren Apps aus verschiedenen Quellen.

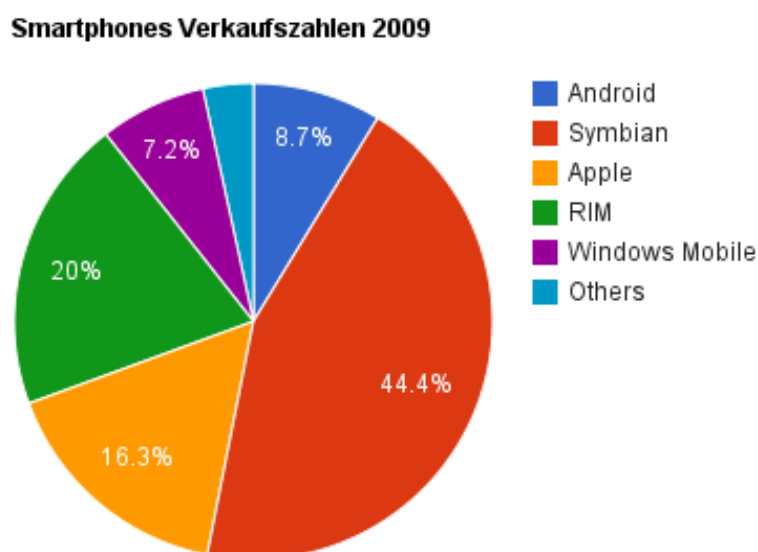
Danach wird dann jede Plattform einzeln im Detail betrachtet. Die Daten werden zur besseren Übersicht pro Plattform tabellarisch aufbereitet. Am Ende jedes Abschnitts folgt eine kurze Auflistung der Vor- und Nachteile jeder Plattform.

#### 3.1.1 Marktanteile

Ein entscheidender Faktor für die Bewertung einer Plattform ist ihre Verbreitung. Genaue und fundierte Zahlen über die Verbreitung der einzelnen Plattformen waren nicht verfügbar.

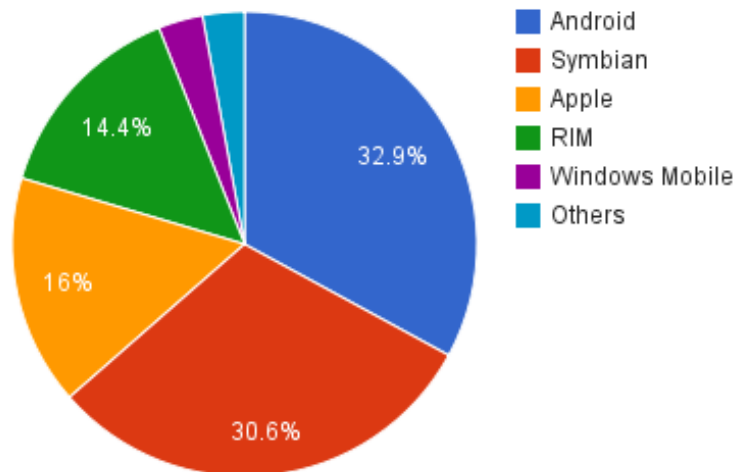
Nach einer Recherche im Internet konnten allerdings verlässliche Verkaufszahlen aus den Jahren 2009 und 2010 von Smartphones der jeweiligen Plattformen ermittelt werden. Diese stammen von der Firma Canalys, einem amerikanischen Unternehmen, das sich auf die Erhebung technischer Marktdaten spezialisiert hat.

Die Zahlen waren angegeben in "Millionen verkaufter Geräte". Zur besseren Übersicht folgt im Anschluss ein Kreisdiagramm mit der prozentualen Aufteilung (Abb. 1,2). Die absoluten Zahlen stehen für jede Plattform jeweils in deren Übersichtstabellen.



**Abbildung 1: Prozentuale Aufteilung der Verkaufszahlen von Smartphones aus 2009 [9]**

**Smartphones Verkaufszahlen 2010**



**Abbildung 2: Prozentuale Aufteilung der Verkaufszahlen von Smartphones aus 2010 [9]**

### **3.1.2 Apps in den Märkten**

Ebenso wie Verkaufszahlen kann die Anzahl der Apps einer Plattform eine Aussage über deren Akzeptanz geben. Dabei wird davon ausgegangen, dass Entwickler vor allem Apps für weitverbreitete Plattformen entwickeln.

Aus den Zahlen der angebotenen Apps wurden Kreisdiagramme für die drei Plattformen (Android, iPhone OS und Windows Phone) erstellt, deren Größe eine Referenz für die Gesamtanzahl ist und deren Einteilung den prozentualen Unterschied zwischen kostenfreien und kostenpflichtigen Apps darstellt (Abb.3).

Für die anderen Plattformen dieser Fachstudie konnten die dafür benötigten Zahlen nicht ermittelt werden.

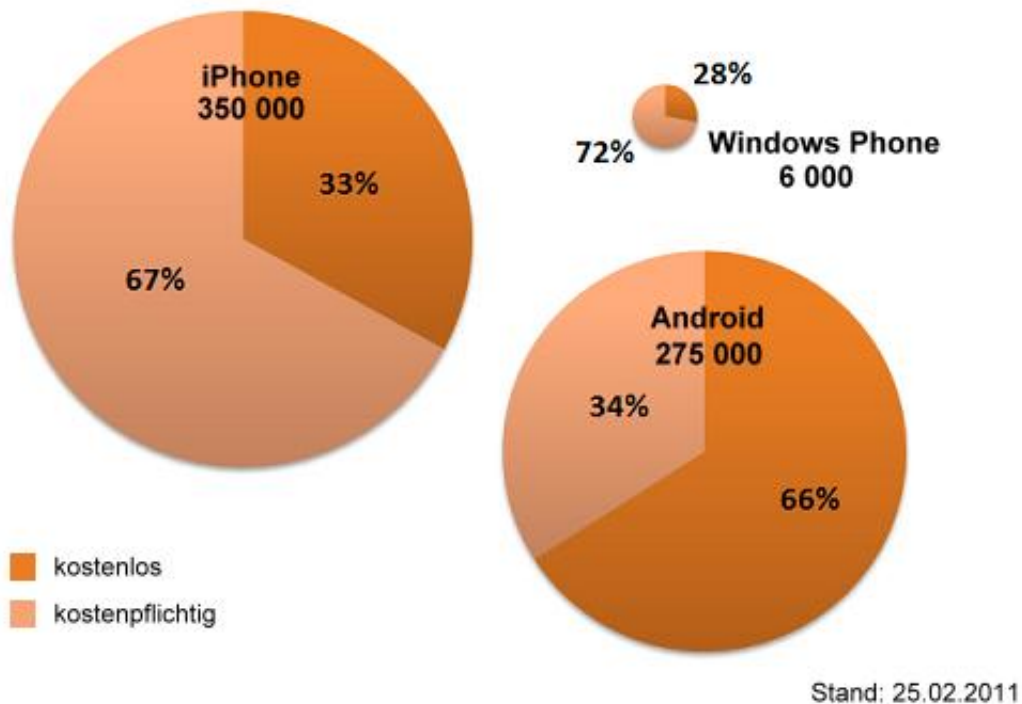


Abbildung 3: Anzahl der Apps für Android, iOS und Windows Phone [4, 5, 6]

## 3.2 Plattformen im Detail

Es folgt eine Untersuchung der verschiedenen Plattformen. Dabei wird jede Plattform für sich unabhängig betrachtet. Für jede Plattform werden die wichtigsten Erkenntnisse notiert und eine einheitliche tabellarische Auflistung der verschiedenen Daten erstellt.

Am Ende jedes Kapitels zu einer Plattform findet sich eine kurze Auflistung der positiven und negativen Merkmale.

### 3.2.1 Android 2.x

Googles Android [10] ist eine vergleichsweise junge Plattform. Erstmals im Oktober 2008 veröffentlicht, verzeichnet sie seitdem ein rasantes Wachstum und ist im Jahr 2010 bereits zu der Plattform mit den höchsten Verkaufszahlen aufgestiegen. Android zeichnet sich insbesondere durch seine vollständige Offenheit aus. Öffentlich und kostenlos zugänglich sind nicht nur die gebräuchlichen Entwicklungswerkzeuge, wie ein SDK, sondern auch der Quellcode des gesamten auf Linux basierenden Betriebssystemkerns. Darüber hinaus bietet Google in einem NDK zusammengefasst entsprechende Werkzeuge, wie einen C-Compiler, um native Systementwicklung, wie die Integration eigener Gerätetreiber oder Systembibliotheken zu ermöglichen.













Für die Anwendungsentwicklung kommt ein auf JAVA basierendes Framework zum Einsatz. Auch in der Wahl des Entwicklerbetriebssystems bietet Google die freie Auswahl zwischen Windows, Linux und Mac, da das entsprechende Eclipse-Plugin auf allen Plattformen lauffähig ist. Im Gegensatz zu manch seiner Konkurrenten, wie z.B. dem iPhone, ist die Trennung zwischen Hardware und Software bei Android recht groß. Google selbst tritt nur in geringem Umfang als Smartphone-Hersteller auf (z.B. "Google Nexus One" – dieses wird von HTC gefertigt und mit einem Google-Branding versehen). Daher hegt Android den Anspruch auf einer äußerst heterogenen Hardwareumgebung stabil lauffähig zu sein. In der Praxis führt dies jedoch zu einer hohen Versionsfragmentierung, da die Hersteller oftmals für ältere Geräte, obwohl möglich, dennoch keine Betriebssystemupdates auf neuere Android-Versionen zur Verfügung stellen.

Zur Verteilung der Anwendungssoftware setzt Android ähnlich wie seine Konkurrenten auf einen zentralen App-Market, über den sowohl kostenlose als auch kostenpflichtige Software bezogen werden kann. Anders als bei den meisten Konkurrenten ist es auch möglich Anwendungspakete (.apk Dateien) direkt auf das Gerät zu übertragen und so manuell zu installieren. Verfügbare Sensoren sind vergleichbar mit denen anderer Plattformen, wegen der vielen unabhängigen Gerätehersteller sind die Differenzen innerhalb der Plattform jedoch größer. Zu dem Standardumfang der meisten Smartphones gehören in der Regel GPS, Kompass, Bewegungs- und Lichtsensoren. Neben der freien Verfügbarkeit aller Quellcodes zur Unterstützung der Entwickler überzeugt Android ebenfalls durch die geringen bis nicht vorhandenen Kosten, um Software zu erstellen und zu veröffentlichen. Allerdings verlangt Google zur Zeit, ähnlich wie seine Konkurrenten, eine anteilige Gebühr von 30% für im Android-Market verkaufte, kostenpflichtige Apps. [11, 52, 53]

| Merkmal                | Daten   |
|------------------------|---|
| Hersteller             | Google  |
| Verkaufszahlen [9]     | 2009: 8,7 % (4,7 Millionen Geräte)<br>2010: 32,9 % (33,3 Millionen Geräte)  |
| IDE                    | Eclipse (Windows, Linux, Mac)   |
| Entwickler-Plattformen | SDK (Java):<br>Link: <a href="http://developer.android.com/sdk/">http://developer.android.com/sdk/</a><br><br>NDK (C/C++):<br>Link: <a href="http://developer.android.com/sdk/ndk">http://developer.android.com/sdk/ndk</a><br><br>Lizenz: Freeware [7] |
| Verfügbare Emulatoren  | Windows, Linux<br><a href="http://developer.android.com/guide/developing/tools/emulator.html">http://developer.android.com/guide/developing/tools/emulator.html</a>   |
| Zukunftserwartung      | Hoch  |
| Developer Network      | <a href="http://developer.android.com">http://developer.android.com</a>   |
| Community-Stärke       | Hoch  |
| Gerätevielfalt         | Hoch. Viele unabhängige Gerätehersteller.<br>Beispiele: <ul style="list-style-type: none"> <li>• HTC Desire</li> <li>• Google Nexus One</li> <li>• Motorola Milestone 2</li> </ul>  |
| Verfügbare Sensoren    | Geräteabhängig, siehe Gerätevielfalt<br>Meistens: <ul style="list-style-type: none"> <li>• GPS</li> <li>• Bewegungssensor</li> <li>• Digitaler Kompass</li> <li>• Lichtsensor</li> </ul>  |
| Hardware-Ansteuerung   | Per SDK: Begrenzt<br>Per NDK: Unbegrenzt  |
| Geräte-Anschlüsse      | Geräteabhängig, siehe Gerätevielfalt.<br>Meistens: <ul style="list-style-type: none"> <li>• Bluetooth</li> <li>• USB</li> </ul>   |

|                             |   |
|-----------------------------|---|
| Sources-Verfügbarkeit       | Vollständig ( <a href="http://source.android.com">http://source.android.com</a> )<br><br>Lizenz:<br>Apache Software License 2.0 (mit Ausnahmen) [8] |
| Distributionswege           | Eigenes Paketformat, Zentrale Verteilung (Market)   |
| Abhängigkeit vom Hersteller | Gering  |
| Einstiegsbarrieren          | Gering. Java, C/C++ weit verbreitet   |
| Akkulaufzeit                | Geräteabhängig, siehe Gerätevielfalt.<br>Generell gilt: Wenige Stunden bei Volllast.  |
| Distributionskosten         | einmalig 25\$, um sich als Entwickler zu registrieren   |
| Entwicklungskosten          | Kostenlos   |
| Versionsfragmentierung      | Hoch  |

**Tabelle 1: Eigenschaften Android**

| Vorteile  | Nachteile   |
|---|---|
| <ul style="list-style-type: none"> <li> Vollständig Open-Source</li> <li> Native Entwicklung durch NDK möglich</li> <li> Weit verbreitete Programmiersprachen</li> <li> Anwendungspakete manuell installierbar</li> <li> Großer Markt mit über 100.000 Apps</li> <li> Größtes Wachstum, Marktführer</li> <li> Entwicklung auf Mac, Windows, Linux</li> <li> Rasant wachsende Community</li> </ul> | <ul style="list-style-type: none"> <li> Hohe Versionsfragmentierung</li> <li> Betriebssystemupdates herstellerabh.</li> </ul> |

**Tabelle 2: Vor- und Nachteile des Betriebssystems Android.**

### 3.2.2 Windows Phone 7

Das Microsoft Phone 7 ist das neueste Betriebssystem für mobile Geräte, welches in die Evaluierung einbezogen wird. Nachdem Google und Apple ihre Produkte mit zwei Jahren Vorsprung auf dem Markt etablieren konnten, gab Microsoft erst im Oktober 2011 die komplett neu entwickelte Plattform frei [12]. Als einer der Vorreiter im mobilen Sektor gab Microsoft die Weiterentwicklung von Windows Mobile, mit dem Projektnamen "Photon" auf und begann 2008 die Entwicklung von Windows Phone 7 [13]. Die Verkäufe von Geräten mit Windows Phone 7 verliefen mit zwei Millionen Verkäufen in den ersten drei Monaten eher schleppend, obwohl Microsoft andere Erfolgsindikatoren, wie Kundenzufriedenheit und der Anzahl der Entwickler in den Vordergrund stellt [14].

Im Laufe dieser Arbeit wurde zudem bekannt, dass Nokia zukünftig Geräte mit Windows Phone ausstatten wird und dafür die eigene Plattform MeeGo einstellt. Nokia verkaufte im Jahr 2010 ungefähr 461 Millionen mobile Geräte, was einer Steigerung von 20 Millionen Geräten im Vergleich zum Vorjahr bedeutet. Bei einem allgemeinen Wachstum von 72,1% im Jahr 2010 bedeutet diese Steigerung jedoch einen starken Verlust von Marktanteilen [4]. Durch Kooperation mit Microsoft können beide Parteien profitieren. Nokia wird voraussichtlich 2012 mit der Auslieferung von Geräten mit Windows Phone beginnen [16]. Es ist daher davon auszugehen, dass sich Microsoft als dritter Big Player im mobilen Sektor etablieren kann.

Um Anwendungen für das Windows Phone zu entwickeln, gilt es zwischen zwei verschiedenen Technologien zu entscheiden, abhängig davon, ob man klassische "point-and-click"-Anwendungen oder Spiele entwickeln möchte. Für Spiele gibt es das, auf DirectX basierende, XNA-Framework und eine Einbindung in das Xbox Live System. Um Anwendungsoberflächen zu entwickeln, steht eine abgeänderte Silverlight-Plattform zur Verfügung [17].

Silverlight ist ein in den letzten Jahren gereiftes Framework zur Entwicklung von vektorbasierten Oberflächen und unterstützt unter anderem 3D-Effekte, Animationen und verschiedene Videoformate. Wie im Desktopbereich kommen auch bei der Entwicklung für Windows Phone mit Silverlight die Entwicklungsumgebungen Visual Studio für die Programmlogik und Expression Blend für die Oberfläche zum Einsatz, also ausgereifte und in der Praxis bewährte Lösungen.

Oft aufgeführte Kritikpunkte am Windows Phone sind die fehlende Multitasking-Unterstützung für Anwendungen von Drittanbietern, die fehlende Copy-Paste-Unterstützung und der Internet Explorer 7, der aufgrund seines Alters viele Funktionen, wie z.B. HTML5 noch nicht unterstützt. Zumindest die Copy-Paste-Unterstützung wurde in der aktuellen SDK nachgereicht [18]. Multitasking und der Internet Explorer 9 sind für kommende Versionen von Windows Phone angekündigt [19].

Im Gegensatz zu Android können Anwendungen nur über den Marketplace von Windows Phone verteilt werden. Um eine XAP-Datei, das ist das Paketformat für Silverlightanwendungen, zu installieren, muss das Gerät für das Debugging freigeschaltet sein [20].







Des Weiteren wird Software vom Marketplace ausgeschlossen, welche nur mit dem Quellcode ausgeliefert werden darf, nur zur Weiterentwicklung gedacht ist oder kostenlos weiterverteilt werden darf. Das trifft unter anderem auf die GPLv3-Lizenz zu [21].

Um eine Versionsfragmentierung der Windows Phone Versionen zu vermeiden, werden an die Hersteller von Windows Phone Smartphones gewisse Anforderungen gestellt. Außer hohen Anforderungen an die Hardware ist es den Herstellern nicht gestattet, eine eigene Oberfläche über die Windows Phone Oberfläche zu legen [22]. Daher können bei einem Software-Update die Geräte aller Anbieter gleichzeitig aktualisiert werden, was auch zu Problemen führen kann, wie im Fall des Samsung Omnia, das nach einem Update nicht mehr funktionierte [23].

| Merkmal                     | Daten  |
|-----------------------------|--|
| Hersteller                  | Microsoft  |
| Verkaufszahlen [9]          | 2009: 7,2 % (3,9 Millionen Geräte)<br>2010: 3,1 % (3,1 Millionen Geräte)   |
| IDE                         | Visual Studio, Expression Blend (Windows)  |
| Entwickler-Plattformen      | SDK (.NET, C# oder VB.NET):<br>Link: <a href="http://create.msdn.com">http://create.msdn.com</a><br>NDK (C/C++):<br>nicht verfügbar  |
| Verfügbare Emulatoren       | Im Windows Phone Developer Kit enthalten   |
| Zukunftserwartung           | Hoch   |
| Developer Network           | <a href="http://msdn.microsoft.com/de-de/windowsphone">http://msdn.microsoft.com/de-de/windowsphone</a>  |
| Community-Stärke            | Mittel   |
| Gerätevielfalt              | Hoch   |
| Verfügbare Sensoren         | Geräteabhängig, siehe Gerätevielfalt<br>Meistens: <ul style="list-style-type: none"> <li>• GPS</li> <li>• Bewegungssensor</li> <li>• Digitaler Kompass</li> <li>• Lichtsensor</li> </ul>   |
| Hardware-Ansteuerung        | Per SDK: Begrenzt  |
| Geräte-Anschlüsse           | Geräteabhängig, siehe Gerätevielfalt.<br>Meistens: <ul style="list-style-type: none"> <li>• Bluetooth</li> <li>• USB</li> </ul>  |
| Sources-Verfügbarkeit       | Keine. Closed Source.  |
| Distributionswege           | Nur über einen zentralen Markt möglich   |
| Abhängigkeit vom Hersteller | Mittel. Microsoft schränkt die Anforderungen an die Software, welche über den App-Market verteilt werden darf über das Application Provider Agreement [21] ein und behält sich das Recht auf Änderungen vor. Im Gegensatz zum AppleStore sind diese Einschränkungen aber noch relativ geringfügig. |

|                        |  |
|------------------------|--|
| Einstiegsbarrieren     | Gering. Am Anfang reicht das, auch für den kommerziellen Einsatz, kostenlose "Visual Studio Express". Für den professionellen Einsatz entstehen hohe Kosten für die Entwicklungsumgebung.<br>C# und .NET sehr weit verbreitet => Gute Dokumentation und Community.   |
| Akkulaufzeit           | Geräteabhängig, siehe Gerätevielfalt.<br>Generell gilt: Wenige Stunden bei Volllast.   |
| Distributionskosten    | Nutzung des App-Market zum Verteilen eigener Anwendungen: <ul style="list-style-type: none"> <li>• 99\$ pro Jahr</li> <li>• Kostenlos für DreamSpark Studenten (Uni Stuttgart ist Mitglied des DreamSpark Programms)</li> </ul>  |
| Entwicklungskosten     | Expression Studio 4 Ultimate: <ul style="list-style-type: none"> <li>• 599\$</li> <li>• kostenlos für MSDNAA Studenten (Uni Stuttgart ist Mitglied des MSDNAA-Programms)</li> </ul> Visual Studio 2010: <ul style="list-style-type: none"> <li>• kostenlos für MSDNAA Studenten</li> <li>• Visual Studio Express: kostenlos</li> <li>• Visual Studio Professional: 799\$</li> <li>• Visual Studio Ultimate: 11899\$</li> </ul> (Stand: 24.02.2011) |
| Versionsfragmentierung | Keine  |

**Tabelle 3: Eigenschaften Windows Phone 7**

| Vorteile   | Nachteile  |
|--|--|
| <ul style="list-style-type: none"> <li> einheitliches Look &amp; Feel</li> <li> moderne GUI-Technologie (Silverlight)</li> <li> ausgereifte Entwicklungsumgebung</li> </ul> | <ul style="list-style-type: none"> <li> noch kein Multitasking für Drittanbieter</li> <li> teure Entwicklungsumgebung</li> <li> noch relativ kleiner Marktanteil</li> </ul> |

**Tabelle 4: Vor- und Nachteile Windows Phone 7**

### 3.2.3 iOS 4.x

Das iOS Betriebssystem gehört zu den derzeit am Markt führenden mobilen Betriebssystemen. Apple hat in diesem Betriebssystem viele Erfahrungen aus ihrem PC-Betriebssystem Mac OS X übernommen. Das Betriebssystem ist anfangs sehr dicht an der Entwicklung des ersten iPhones angelehnt gewesen. Es hieß daher in den früheren Versionen auch iPhone OS. Erst mit dem Erscheinen der Version 4.0 wurde der Name auf iOS geändert.

Im Vergleich zu anderen Plattformen zeichnet sich das iOS vor allem dadurch aus, dass es immer noch sehr eng an der Hardware entwickelt wird. Während Systeme wie Android (siehe 3.2.1) oder Symbian (siehe 3.2.4) für eine Vielzahl von Geräten verwendet werden kann, ist das iOS beschränkt auf wenige Geräte des Herstellers Apple.

Apple hat mit der Markteinführung des iPhones im Jahr 2007 den Markt für Smartphones ein gutes Stück weiter nach vorne gebracht. Die Geräteklasse, die bis dahin vornehmlich im Business-Bereich angesiedelt war, fand von da an auch Anwender, die das Smartphone für private Zwecke einsetzte. Apple konnte daher zu Beginn einen raschen Anstieg seiner Absatzzahlen bekannt geben.

Dass die Absatzzahlen von Apple nicht weiter gestiegen sind, lässt sich durch zwei einfache Gegebenheiten erklären. Zum einen sind die Gerätekosten im Verhältnis zu anderen Herstellern wie z.B. Android hoch, zum anderen polarisiert Apple durch seine Firmenpolitik immer wieder den Markt.

Apple behält sich für die Vermarktung aller Apps eine sehr restriktive Politik vor. So können Apps ausschließlich über den offiziellen App Store der Firma Apple heruntergeladen werden. Es gibt keine andere Möglichkeit eine App auf ein Gerät zu bekommen ohne dies vorher in dem App Store einzustellen. Allerdings kann man nur als angemeldeter User eine App einstellen. Jede App, die man einstellen möchte wird allerdings zuerst von Apple geprüft, bevor diese veröffentlicht wird. [25] Hinzu kommen Kosten in Höhe von 99 \$ jährlich als Fixbeitrag und 30 % des App-Umsatzes gehen an Apple. Dies sehen viele Entwickler eher kritisch als vorteilhaft. [26]

Generell gestaltet sich das Entwickeln von Apps für das iOS als wesentlich schwieriger als beispielsweise für Android oder Windows Phone 7. So kann man Apps nur in Objective C erstellen. Dafür wird die Entwicklungsumgebung Xcode benötigt, welche nur unter Mac OS X zur Verfügung steht. In Zusammenhang mit dem schwierigen Distributionsweg ergeben sich für diese Plattform die wohl höchsten Einstiegsbarrieren und auch Folgekosten.






Es ist allerdings aus heutiger Sicht davon auszugehen, dass das iOS auch in Zukunft einen sicheren Marktanteil haben wird. Die Plattform hat sich als eine der Großen etabliert und wird von Apple ständig weiterentwickelt. Es ist nicht davon auszugehen, dass diese Plattform auf lange Sicht völlig vom Markt verschwindet, denn Apple ist bekannt dafür, durch cleveres Marketing die Kunden zu begeistern und an sich zu binden [51, 52].



| Merkmal                     | Daten   |
|-----------------------------|---|
| Hersteller                  | Apple Inc.  |
| Verkaufszahlen [9]          | 2009: 16,3 % (8,7 Millionen Geräte)<br>2010: 16,0 % (16,2 Millionen Geräte)   |
| IDE                         | Xcode (Mac)   |
| Entwickler-Plattformen      | SDK (Objective-C):<br>Link: <a href="http://developer.apple.com/programs/ios/develop.html">http://developer.apple.com/programs/ios/develop.html</a><br>Lizenz: proprietär   |
| Verfügbare Emulatoren       | Mac<br><a href="http://developer.apple.com/programs/ios/">http://developer.apple.com/programs/ios/</a>  |
| Zukunftserwartung           | Hoch  |
| Developer Network           | <a href="http://developer.apple.com/devcenter/ios/index.action">http://developer.apple.com/devcenter/ios/index.action</a>   |
| Community-Stärke            | Hoch  |
| Gerätevielfalt              | Niedrig <ul style="list-style-type: none"> <li>• iPhone</li> <li>• iPhone 3G / 3GS</li> <li>• iPhone 4</li> </ul>   |
| Verfügbare Sensoren [1]     | <ul style="list-style-type: none"> <li>• 3-Achsen Gyrosensor</li> <li>• Beschleunigungssensor</li> <li>• Annäherungssensor (schaltet das Display ab, wenn man das Gerät zum telefonieren ans Ohr hält)</li> <li>• Umgebungslichtsensor</li> <li>• GPS-Sensor</li> <li>• Digitaler Kompass (3D)</li> </ul> |
| Hardware-Ansteuerung        | Per SDK: Begrenzt   |
| Geräte-Anschlüsse [24]      | <ul style="list-style-type: none"> <li>• Bluetooth</li> <li>• 30-poliger Dock-Anschluss</li> <li>• 3,5 mm Stereo-Minianschluss</li> <li>• Wi-Fi</li> </ul>  |
| Sources-Verfügbarkeit       | Closed Source   |
| Distributionswege           | Ausschließlich Zentrale Verteilung (App-Market)   |
| Abhängigkeit vom Hersteller | Hoch  |

|                        |  |
|------------------------|--|
| Einstiegsbarrieren     | hoch Objective-C plus Zugriff auf ein Mac OS X-System                        |
| Akkulaufzeit [24]      | Bis zu 300 Stunden Standby,<br>aber unter Voll-Last nur ca. 6 bis 10 Stunden |
| Distributionskosten    | 99 \$ Jährlich + 30 % des App-Umsatzes                                       |
| Entwicklungskosten     | Xcode ist kostenlos (gibt es aber nur unter Mac OS X)                        |
| Versionsfragmentierung | Kaum vorhanden, da nur geringe Gerätevielfalt                                |

**Tabelle 5: Eigenschaften iOS**

| Vorteile   | Nachteile  |
|--|--|
| <ul style="list-style-type: none"> <li> Übersichtliche Bedienoberfläche</li> <li> Intuitive Bedienung</li> <li> Großes Angebot im App-Market</li> </ul> | <ul style="list-style-type: none"> <li> Hersteller kontrolliert Apps und Daten</li> <li> Daten nur via iTunes</li> </ul> |

**Tabelle 6: Vor- und Nachteile von iOS.**

### 3.2.4 Symbian^3

Wie man an den Absatzzahlen erkennt (Tab.8 bzw. 3), gehört Symbian zu den führenden mobilen Plattformen. Es ist ebenfalls eine der Plattformen mit der größten Erfahrung. Allerdings liegt hier auch ein großes Manko. Die Plattform war anfangs für einfache Handysysteme gedacht. Im Vergleich zu Lösungen wie Android oder iOS ist die Plattform nicht für Smartphones entwickelt sondern für Handys aller Art.






Das größte Problem an der Plattform liegt darin, dass Nokia diese im Bereich für Smartphones nicht aktiv weiterentwickeln wird [29]. Nokia hat sich zu einer Kooperation mit Microsoft entschlossen (siehe 3.2.2). Zukünftig werden die Smartphones von Nokia mit Windows Phone 7 ausgestattet.

Auf der Basis der Symbian Plattform gibt es viele verschiedene Versionen, die auf den unterschiedlichsten Geräten laufen. Dies wirkt sich für das Entwickeln einer Anwendung auf Basis dieser Plattform eher kontraproduktiv aus. Möchte man viele Geräte dieser Plattform erreichen, so muss man ggf. verschiedene, speziell angepasste Versionen seiner Anwendung vorhalten. Im Vergleich zu anderen Plattformen wie dem des iOS, kann man bei Symbian seine Anwendung auch fernab des OVI-Stores als einfaches jar-Archiv verteilen. Die meisten Geräte dieser Plattform sind in der Lage einfache jar-Archive zu installieren und auszuführen. Die Einstiegsbarrieren für die Entwicklung von Anwendungen für Geräte auf Symbianbasis sind als gering einzustufen. Alle Tools zu deren Entwicklung werden kostenlos zur Verfügung gestellt. Anwendungen können unter verschiedenen Sprachen wie C++, QT oder Java entwickelt werden. Die Kosten um sich als Publisher eintragen zu lassen, liegen mit einem Euro deutlich unter denen der Konkurrenz.

| Merkmal                | Daten   |
|------------------------|---|
| Hersteller             | Nokia   |
| Verkaufszahlen [9]     | 2009: 44,4 % (23,9 Millionen Geräte)<br>2010: 30,6 % (31,0 Millionen Geräte)  |
| IDE                    | Carbide.c++ (Nur Windows)<br>Link:<br><a href="http://www.forum.nokia.com/Library/Tools_and_downloads/Other/Carbide.c++/">http://www.forum.nokia.com/Library/Tools_and_downloads/Other/Carbide.c++/</a>                         |
| Entwickler-Plattformen | SDK (Qt, C++, Java):<br>Link:<br><a href="http://www.forum.nokia.com/Library/Tools_and_downloads/Other/Symbian_SDKs/">http://www.forum.nokia.com/Library/Tools_and_downloads/Other/Symbian_SDKs/</a><br><br>Lizenz: Open Source |
| Verfügbare Emulatoren  | Windows<br><a href="http://www.forum.nokia.com/Library/Tools_and_downloads/Other/Symbian_SDKs/">http://www.forum.nokia.com/Library/Tools_and_downloads/Other/Symbian_SDKs/</a>  |
| Zukunftserwartung      | keine [27]  |
| Developer Network      | <a href="http://www.forum.nokia.com/Develop/">http://www.forum.nokia.com/Develop/</a>   |
| Community-Stärke       | Mittel<br>aber auf Grund der Zukunftserwartung eher abnehmend   |
| Gerätevielfalt         | Hoch<br>aber nur ein Hersteller (Nokia)   |
| Verfügbare Sensoren    | Geräteabhängig, siehe Gerätevielfalt<br>Meistens: <ul style="list-style-type: none"> <li>• GPS</li> <li>• Bewegungssensor</li> </ul>  |
| Hardware-Ansteuerung   | Per SDK: sehr hoch  |
| Geräte-Anschlüsse      | Geräteabhängig, siehe Gerätevielfalt.<br>Meistens: <ul style="list-style-type: none"> <li>• Bluetooth</li> <li>• USB</li> </ul>   |
| Sources-Verfügbarkeit  | Angeblich vollständig [27]<br><br>Lizenz:   |

|                             |   |
|-----------------------------|---|
|                             | Apache Software License 2.0   |
| Distributionswege           | Zentrale Verteilung (App-Market)<br>jar-Archive   |
| Abhängigkeit vom Hersteller | Aufgrund der Tatsache, das Symbian so nicht mehr existiert, ist die Abhängigkeit von Nokia sehr stark                     |
| Einstiegsbarrieren          | Gering  |
| Akkulaufzeit                | Geräteabhängig, siehe Gerätevielfalt.<br>Grob: Wenige Stunden bei Voll-Last   |
| Distributionskosten         | Anmeldegebühr als Entwickler: Einmalig EUR 1<br>Kosten je App-Download werden nach interner Liste vergütet (ca. 30%) [28] |
| Entwicklungskosten          | Alle Tools stehen kostenlos zur Verfügung   |
| Versionsfragmentierung      | Hoch  |

**Tabelle 7: Eigenschaften Symbian**

| Vorteile  | Nachteile  |
|---|--|
| <ul style="list-style-type: none"> <li> Noch sehr großer Marktanteil</li> <li> Große Erfahrungswerte</li> </ul> | <ul style="list-style-type: none"> <li> Keine Zukunftserwartung</li> <li> Menüs wirken oft nicht aktuell</li> <li> Teilweise sehr komplexe Menüführung</li> </ul> |

**Tabelle 8: Vor- und Nachteile von Symbian**

### 3.2.5 Blackberry OS 6

Die Blackberry Smartphone-Serie von RIM [30] gehört nach wie vor zu einer der großen Mobile-Plattformen auf dem Markt. Obwohl die Verkaufszahlen rückläufig sind und Marktanteile an Konkurrenten wie Android oder iPhone verloren wurden, kann sich die Blackberry-Plattform vor allem noch im traditionell besetzten Business-Sektor und auf dem amerikanischen Markt behaupten.

Die Tatsache, dass Blackberry für die in den Startlöchern stehende, nächste Generation seines Betriebssystems Blackberry OS 7 einen vollständigen Wechsel des Betriebssystems auf einen QNX-Kern und eine Adobe AIR Laufzeitumgebung vorsieht, macht aktuell eine Neuentwicklung, speziell für das momentane Blackberry OS 6, nicht gerade interessant.

Zur Distribution der Software setzt Blackberry, ähnlich wie seine Konkurrenten, auf einen zentralen App-Market. Über diesen kann der Benutzer direkt kostenlose sowie auch kostenpflichtige Software auf dem Gerät installieren.

Da Blackberry gleichzeitig als Betriebssystem- und Gerätehersteller auftritt, schafft dies zwar Kompatibilität und Stabilität der Software zwischen den Blackberry-Geräten untereinander, sorgt aber auch für weniger öffentliche und gut dokumentierte Software-Hardware Schnittstellen und für geringere Gerätevielfalt für den Nutzer.

Blackberry bietet sämtliche Werkzeuge zur Entwicklung kostenlos. Dazu gehören ein SDK für Windows oder Mac und eine Vielzahl an Emulatoren für die verschiedenen Geräte. Blackberry setzt dabei auf die besonders für Java bewährte Eclipse-Entwicklerplattform.







Zur Entwicklung von Anwendersoftware wird ein spezielles Java-Framework genutzt, welches über das Blackberry SDK mit einem weitgehend gekapselten Zugriff auf Betriebssystemfunktionen wie die Hardware-Ansteuerung versorgt wird.

Der Betriebssystem-Quellcode ist nicht Open-Source und damit nicht verfügbar. Auf Grund dieser Tatsache und dem Fehlen eines nativen Entwickler-Frameworks mit den nötigen Compilern ist die Modifikation des Betriebssystems, z.B. mit zusätzlichen oder modifizierten Gerätetreibern, praktisch ausgeschlossen.

| Merkmal                     | Daten  |
|-----------------------------|--|
| Hersteller                  | RIM Company  |
| Verkaufszahlen [9]          | 2009: 20,0 % (10,7 Millionen Geräte)<br>2010: 14,4 % (14,6 Millionen Geräte)   |
| IDE                         | Eclipse-Plugin oder Standalone (Windows+Mac)<br>Link: <a href="http://us.blackberry.com/developers/javaappdev/">http://us.blackberry.com/developers/javaappdev/</a>          |
| Entwickler-Plattformen      | Java   |
| Verfügbare Emulatoren       | Nur Windows  |
| Zukunftserwartung           | Mittel, Plattformwechsel (QNX) für Blackberry 7 angekündigt  |
| Developer Network           | <a href="http://us.blackberry.com/developers/">http://us.blackberry.com/developers/</a>  |
| Community-Stärke            | Mittel   |
| Gerätevielfalt              | Mittel (OS Hersteller = einziger Gerätehersteller)<br>Beispiele: <ul style="list-style-type: none"> <li>• BlackBerry Storm2 9550</li> <li>• BlackBerry Style 9670</li> </ul> |
| Verfügbare Sensoren         | Geräteabhängig, siehe Gerätevielfalt<br>Meistens: <ul style="list-style-type: none"> <li>• GPS</li> <li>• Bewegungssensor</li> </ul>   |
| Hardware-Ansteuerung        | <i>Daten konnten nicht ermittelt werden</i>  |
| Geräte-Anschlüsse           | Geräteabhängig, siehe Gerätevielfalt.<br>Meistens: <ul style="list-style-type: none"> <li>• Bluetooth</li> <li>• USB</li> </ul>  |
| Sources-Verfügbarkeit       | Closed-Source  |
| Distributionswege           | Eigenes Paketformat, Zentrale Verteilung (Market)  |
| Abhängigkeit vom Hersteller | Hoch   |
| Einstiegsbarrieren          | <i>Daten konnten nicht ermittelt werden</i>  |
| Akkulaufzeit                | <i>Daten konnten nicht ermittelt werden</i>  |
| Distributionskosten         | <i>Daten konnten nicht ermittelt werden</i>  |
| Entwicklungskosten          | <i>Daten konnten nicht ermittelt werden</i>  |

|                        |   |
|------------------------|---|
| Versionsfragmentierung | <i>Daten konnten nicht ermittelt werden</i> |
|------------------------|---|

**Tabelle 9: Eigenschaften BlackBerry**

| Vorteile  | Nachteile  |
|---|--|
| <p> Viel Know-How im Business-Sektor</p> | <ul style="list-style-type: none"> <li> Plattformwechsel für OS 7 angekündigt</li> <li> Closed-Source Betriebssystem</li> <li> Keine native Entwicklung (NDK) möglich</li> <li> Emulator nur für Windows</li> <li> Eclipse(Plugin) nur Windows/Mac</li> </ul> |

**Tabelle 10: Vor- und Nachteile von BlackBerry.**

### 3.2.6 Sonstige

Hier sollen kurz die Betriebssysteme dargestellt werden, die aufgrund ihres geringen Marktanteils oder ihrer Unausgereiftheit nicht in die Evaluierung mit einbezogen werden.

Ein sehr vielversprechender Kandidat wäre das von Nokia und Intel entwickelte MeeGo gewesen, welches in Zukunft unter anderem das Betriebssystem Symbian für Smartphones in der oberen Preisklasse ablösen sollte. MeeGo wird als Multiplattformbetriebssystem entwickelt und unterstützt die Intel Atom- und die ARMv7-Architektur. Dadurch ergibt sich ein breites Einsatzspektrum, z.B. in Netbooks, Navigationssystemen, Smartphones und Internetfernsehern [31]. Aufgrund der Kooperation von Nokia und Microsoft (siehe Abschnitt [3.2.2](#)) ist ein starkes und dauerhaftes Engagement seitens Nokia an MeeGo eher zweifelhaft. Nokia beteuert zwar die Entwicklung fortzuführen und will mit dem N950 das erste MeeGo Produkt auf den Markt bringen [32], allerdings soll MeeGo eher zum Austesten neuer Funktionen und "um damit herumzuspielen" verwendet werden [33].

Intel zeigte sich sehr enttäuscht über die neue Strategie von Nokia, möchte aber an der Entwicklung von MeeGo festhalten [34]. Für diese Arbeit spielt MeeGo aufgrund der aktuellen Entwicklungen keine beachtenswerte Rolle mehr.

Hewlett-Packard hat vor in den mobilen Markt vorzudringen und kaufte im April 2010 Palm auf. Im Vordergrund steht dabei das mobile Betriebssystem webOS 2.0 [35], welches auf Smartphones und Tablets zum Einsatz kommen wird. Für die Portierung der Campus Navigation wäre das PDK (Plugin-Development-Kit) interessant gewesen, da dieses die Portierung von C/C++ Anwendungen ermöglicht. Im Februar hat HP, auf seiner eigenen Messe "ThinkBeyond", einige vielversprechende Smartphones und ein Tablet vorgestellt [36]. Ob sich diese im Verlaufe des Jahres auf dem Markt etablieren können, bleibt abzuwarten. Das webOS-System wird aufgrund der sehr geringen Verbreitung nicht in die Evaluierung mit einbezogen.



### 3.3 Multi-Plattform Lösungen

#### 3.3.1 Übersicht

|             | Android 2.x | iOS 4  | Windows Phone 7 | Blackberry 6 |
|-------------|-------------|--------|-----------------|--------------|
| Java ME     | Orange      | Red    | Red             | Green        |
| Flashplayer | Green       | Red    | Orange          | Orange       |
| .net / mono | Yellow      | Yellow | Green           | Red          |
| Silverlight | Yellow      | Yellow | Green           | Red          |
| QT          | Yellow      | Orange | Red             | Red          |

Legende:

|                       |        |
|-----------------------|--------|
| Vollständiger Support | Green  |
| Support verfügbar     | Yellow |
| Support geplant       | Orange |
| Kein Support          | Red    |

Tabelle 11: Gegenüberstellung der Multi-Plattformlösungen

Quellen: [37, 38, 39, 40, 41, 42, 44, 45]

#### 3.3.2 Java ME

JAVA ME (MicroEdition) [46, 47] ist eine von SUN bereits 1999 veröffentlichte, abgespeckte JAVA-Laufzeitumgebung für "mobile, kleine Systeme". Besonders bekannt wurde JAVA ME durch die seit der Jahrtausendwende verstärkt auftretenden "Handygames". Das Framework enthält nur eine Teilmenge der im „großen Bruder“ JAVA SE (StandardEdition) veröffentlichten Funktionen und Bibliotheken, war aber auf älteren Handys (vor allem Nokia) weit verbreitet. SUN veröffentlichte immer wieder neue Versionen der Laufzeitumgebung, bis der Source-Code 2006 schließlich unter GPL-Lizenz veröffentlicht wurde und die Verantwortung und Weiterentwicklung damit de facto abgegeben wurden. Bereits 2007 erklärte SUN die Ära Java ME für in absehbarer Zeit beendet und so gibt es auch heutzutage keine direkte Java ME Laufzeitumgebung mehr auf Android, iPhone und andere. Allerdings ist es z.B. auf Android-Phones möglich, ältere Java-ME Programme mit Hilfe eines Wrappers direkt auszuführen. Wer jedoch Neuentwicklungen im Fokus hat, wird Java ME kaum mehr als Plattform in Betracht ziehen.

### **3.3.3 Flash / Adobe AIR**

Der weit verbreitete Flash-Player von Adobe hat inzwischen auch seinen Weg auf Teile der Smartphones gefunden. Aktuelle Android Versionen werden bereits mit installiertem Flash Player ausgeliefert. Eine Windows Phone und Blackberry Variante sind von offizieller Seite her angekündigt, aber noch nicht verfügbar. Apple hingegen verweigert sich dem FlashPlayer konsequent und will diesen, auch in Zukunft, ausdrücklich nicht unterstützen.

### **3.3.4 .net Framework / mono**

Das .net Framework wird von Microsoft oft als plattformübergreifend bezeichnet. In der Realität bietet Microsoft aber selbst nur für die hauseigenen Systeme, wie z.B. Windows Phone 7 oder das klassische Windows, eine eigene Laufzeitumgebung an. Das Mono-Framework, unterstützt und finanziert von Novell, bietet hier Abhilfe. Es versucht die nicht von Microsoft unterstützten Plattformen für .net zugänglich zu machen. In Bezug auf Smartphones sind inzwischen zwei Unterversionen des Mono-Frameworks entstanden. MonoTouch, welches .net auf das iPhone bringt und MonoDroid, das beabsichtigt .net auch auf Android lauffähig zu machen. Beide Projekte stammen aus gleichem Hause, sind jedoch kommerziell und nicht kostenlos verfügbar. Auch muss beachtet werden, dass zwar prinzipiell mit einer gemeinsamen Sprache (C#) eine Vielzahl von Plattformen abgedeckt werden kann. Gerade die Hardware- und GUI-Ansteuerung funktionieren aber von Plattform zu Plattform unterschiedlich und müssten jeweils separat implementiert werden.

### **3.3.5 Qt**

Das auf C++ basierende QT-Framework [48, 49] wird allgemein als "GUI-Toolkit" bezeichnet. Es erweitert den C++ Dialekt um neue Konstrukte und erleichtert das Erzeugen grafischer Oberflächen. QT wurde 2008 von Nokia gekauft und weitergeführt. Inzwischen ist QT für eine ganze Reihe von Plattformen veröffentlicht, darunter Windows, Linux und Mac. Vergleichsweise neu sind die Portierungen für mobile Plattformen. Steckt die QT Portierung für das iPhone noch völlig in den Kinderschuhen und wird auf absehbare Zeit wohl keine nutzbare Versionen hervorbringen, so sind für Android erste Erfolge in Form einer lauffähigen und einsatzbereiten Vorabversion zu verzeichnen. Einen Rückschlag erhielt der Smartphone-Siegeszug von QT, als Nokia vor kurzem bekannt gab, seine eigene neue geplante Smartphone Plattform (MeeGoo), welche großteils auf QT basiert, zu Gunsten von Microsofts Windos Phone 7 aufzugeben.

## **3.4 Verfügbare Sensoren für Smartphones**

Die gängigen Smartphones unterscheiden sich im Bereich der eingebauten Sensoren nur gering. Zum Standardumfang gehören in der Regel GPS-, Bewegungs-, Kompass- und Lichtsensoren in mehrheitlich vergleichbaren Qualitätsstandards. Keines der von uns untersuchten Smartphones hatte einen integrierten RFID-Reader im Angebot.

## **4 Analyse der Anforderungen**

### **4.1 Allgemeine Anforderungen**

Die allgemeinen Anforderungen an ein Navigationssystem für den Uni-Campus haben wir mittels einer kleinen Umfrage unter 15 Studenten, männliche (8) wie weibliche (7) Teilnehmer aus verschiedenen Semestern und verschiedenen Studiengängen, ermittelt. Bei dieser Umfrage haben wir die Studenten nach Schwierigkeiten zu Beginn ihres Studiums befragt. In vertiefenden Gesprächen haben wir dann weiter ermittelt, welche Unterstützung ihnen hier ein Programm auf einem Smartphone bieten könnte.

Die Umfragebögen wurden ausgewertet und dabei kam folgender Trend zum Vorschein. Die meisten Studenten würden ein System mit einer sozialen Komponente, in denen Interaktion untereinander möglich ist, z.B. anhand von Kommentaren oder Bewertungen bevorzugen.

Der aufgezeigte Trend führte dazu, dass die Anforderungen an das System in zwei Teile getrennt wurden. Es gibt zum einen die Kernkomponenten. Diese beinhalten alle grundlegenden Funktionen, die das System anonymen Anwendern für die Navigation anbieten sollte. Zum anderen gibt es den Teil der sozialen Erweiterungen, in welchem alle Funktionen betrachtet werden, die das System angemeldeten Benutzern bieten könnte, um den Nutzwert des Systems zu erhöhen. Diese Anforderungen sind im Vergleich zu den Kernkomponenten allerdings vorerst nicht Teil der Evaluierung und werden hier nur der Vollständigkeit halber erwähnt.

Eine Anforderung an das neue System ist, dass sehende Benutzer selbstständig Daten in das System einpflegen können sollen. Der Vorteil daran wäre, dass die von sehenden Studenten gepflegten Daten dann auch den blinden Studenten zu Gute kämen. Wenn eine größere Menge an Benutzern das System aktiv verwendet, kommt dies allen zu Gute. Dies heißt im Umkehrschluss, dass eine Anforderung an das System die Möglichkeit zur Datenpflege sein muss. Dies gilt sowohl für die Benutzerseite (z.B. Einpflegen von Zusatzinfos per Smartphone), als auch für die zentrale Verwaltung (z.B. Einpflegen von Raumbelagungen).

Für all die hier genannten Aspekte sind im folgenden Use-Cases abgebildet, um eine spätere Implementierung zu unterstützen.

### **4.2 Use Cases**

Aus den Umfragen heraus stellten sich schnell einige Use Cases heraus, die im Folgenden näher betrachtet werden. Für die Use Cases des Kernsystems und der sozialen Erweiterung ist dazu jeweils ein Use Case Diagramm angegeben und dazu eine vereinfachte Form der Use Case Beschreibung. Bei der Beschreibung werden auf Vor- und Nachbedingungen und auf Sonderfälle verzichtet. Diese Details würden den Rahmen der Fachstudie sprengen und sollten Teil der Dokumentation einer möglichen Implementierung sein.

## 4.2.1 Use Cases des Kernsystems

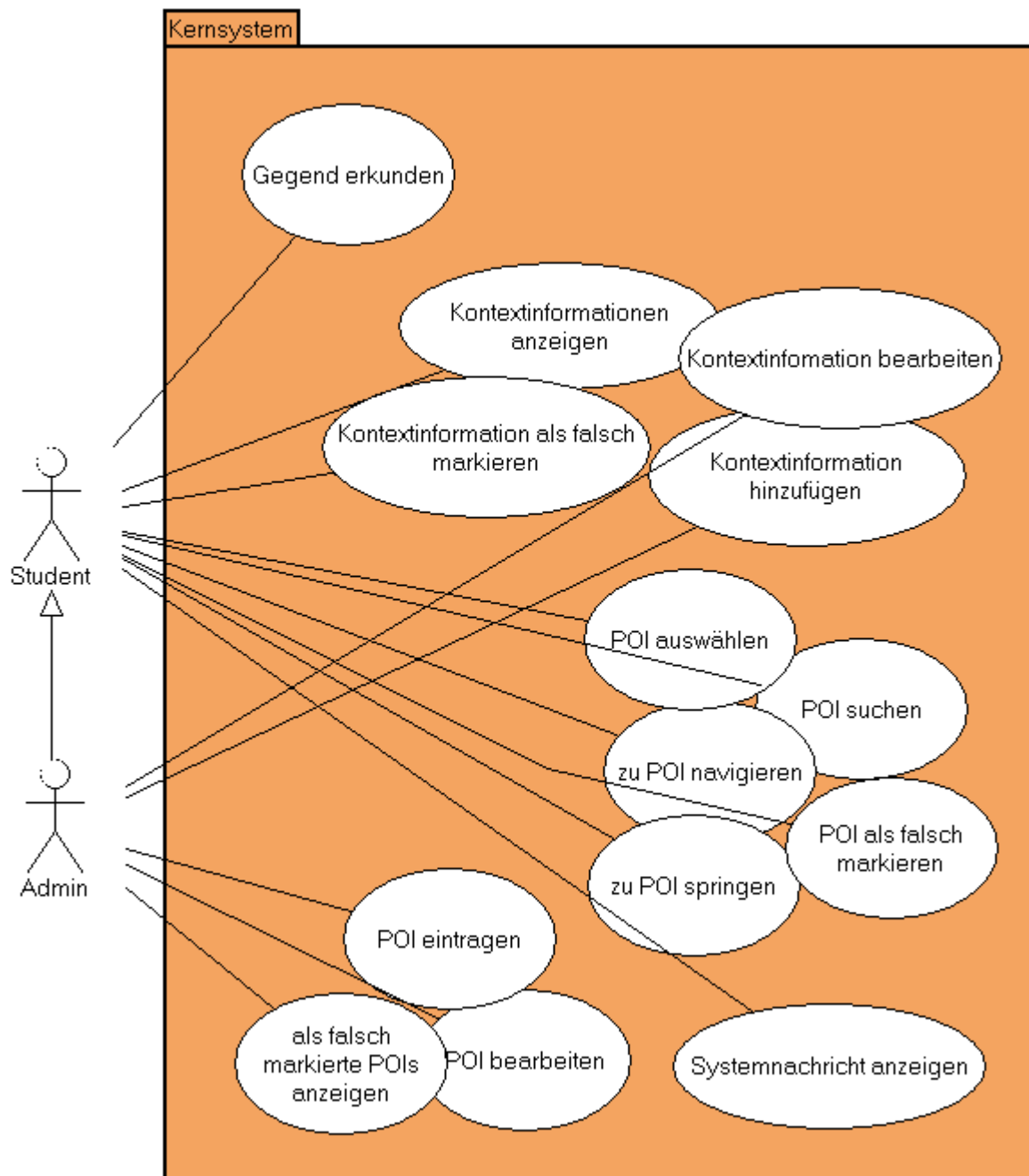


Abbildung 4: Use Case Diagramm des Kernsystems

#### 4.2.1.1 Gegend erkunden

|                     |   |
|---------------------|---|
| <b>Akteur</b>       | Anwender/-in  |
| <b>Beschreibung</b> | In der Kartenansicht kann der Benutzer von seiner eigenen Position aus die Karte scrollen, indem er mit dem Finger über das Touchpad wischt. Die POIs sowie ihre Kontextinformationen werden dynamisch nachgeladen und dargestellt. |

#### 4.2.1.2 Kontextinformationen anzeigen

|                     |  |
|---------------------|--|
| <b>Akteur</b>       | Anwender/-in   |
| <b>Beschreibung</b> | Zu einer bestimmten Kategorie eines POIs können jeweils unterschiedliche Typen von Kontextinformationen angezeigt werden. Die Kontextinformationen werden aus verschiedenen externen Datenquellen integriert und könnten um eigene Informationen, zu POI-Kategorien ohne externe Daten, erweitert werden. Zum Beispiel können zu einem Vorlesungssaal die Vorlesungen der nächsten Woche angezeigt werden oder zur Mensa der Speiseplan. |

#### 4.2.1.3 Kontextinformationen bearbeiten

|                     |  |
|---------------------|--|
| <b>Akteur</b>       | Admin  |
| <b>Beschreibung</b> | Selbst verwaltete Kontextinformationen können vom Admin geändert oder gelöscht werden. |

#### 4.2.1.4 Kontextinformationen hinzufügen

|                     |  |
|---------------------|--|
| <b>Akteur</b>       | Admin  |
| <b>Beschreibung</b> | Zu den POI-Kategorien, zu denen es keine externen Datenquellen gibt, können zu den einzelnen POIs eigene Informationen hinzugefügt werden. |

#### 4.2.1.5 Kontextinformationen als falsch markieren

|                     |   |
|---------------------|---|
| <b>Akteur</b>       | Anwender/-in  |
| <b>Beschreibung</b> | Jeder Student der Kontextinformationen eines POIs findet, die seiner Meinung nach nicht der Realität entsprechen, kann diese als falsch markieren. Der Akteur Admin bekommt dann eine Benachrichtigung über die Markierung. |

#### 4.2.1.6 POI suchen

|                     |   |
|---------------------|---|
| <b>Akteur</b>       | Anwender/-in  |
| <b>Beschreibung</b> | Über eine Suchmaske können POIs im System gesucht werden. Dabei kann nach Name, Kategorie des POIs und die Entfernung zur aktuellen Position gefiltert werden. Es werden die POIs aufgelistet, welche dem eingestellten Filter entsprechen. |

#### 4.2.1.7 POI auswählen

|                     |  |
|---------------------|--|
| <b>Akteur</b>       | Anwender/-in   |
| <b>Beschreibung</b> | Ein POI kann ausgewählt werden, indem er angeklickt wird. Es stehen dem Benutzer dann folgende Möglichkeiten zur Auswahl: <ul style="list-style-type: none"><li>• zum POI navigieren (4.3.1.8)</li><li>• zum POI auf der Karte springen (4.3.1.9)</li><li>• die Kontextinformationen dieses POIs anzeigen (4.3.1.2)</li><li>• den POI als falsch markieren (4.3.1.2)</li></ul> |

#### 4.2.1.8 Zu POI navigieren

|                     |   |
|---------------------|---|
| <b>Akteur</b>       | Anwender/-in  |
| <b>Beschreibung</b> | Zeigt eine Route auf der Karte an, welcher man nachlaufen kann. Per Sprachausgabe wird der Student beim Folgen der Route unterstützt. |

#### 4.2.1.9 Zu POI springen

|                     |  |
|---------------------|--|
| <b>Akteur</b>       | Anwender/-in   |
| <b>Beschreibung</b> | Die Anzeige springt zum ausgewählten POI. Es wird in der Ansicht ein Menüpunkt eingeblendet mit dem man wieder zur Position springen kann, an der man sich befindet. |

#### 4.2.1.10 POI als falsch markieren

|                     |  |
|---------------------|--|
| <b>Akteur</b>       | Anwender/-in   |
| <b>Beschreibung</b> | Ein Student kann einen POI als falsch markieren. Der Admin kann die als falsch markierten POIs anzeigen (4.3.1.13) und gegebenenfalls anpassen (4.3.1.12). |

#### 4.2.1.11 POI eintragen

|                     |   |
|---------------------|---|
| <b>Akteur</b>       | Admin   |
| <b>Beschreibung</b> | Ein Admin kann neue POIs eintragen. Dazu muss mindestens ein Name, eine Kategorie und eine Position angegeben werden. Des Weiteren kann eine Verknüpfung zu Kontextinformationen hergestellt werden, z.B. über eine ID. |

#### 4.2.1.12 POI bearbeiten

|                     |                                   |
|---------------------|-----------------------------------|
| <b>Akteur</b>       | Admin                             |
| <b>Beschreibung</b> | Bereits angelegte POI bearbeiten. |

#### 4.2.1.13 Als falsch markierte POIs anzeigen

|                     |   |
|---------------------|---|
| <b>Akteur</b>       | Admin   |
| <b>Beschreibung</b> | von Studenten als falsch markierte POIs werden in einer Liste angezeigt und können vom Admin bearbeitet werden (4.3.1.12). Alternativ kann der Admin die Kennzeichnung auch löschen wenn er der Meinung ist, dass der POI nicht fehlerhaft ist. |

#### 4.2.1.14 Systemnachricht anzeigen

|                     |   |
|---------------------|---|
| <b>Akteur</b>       | Anwender/-in  |
| <b>Beschreibung</b> | <p>Wenn ein Student eine Benachrichtigung vom System erhält, z.B. über aktuelle Ereignisse oder Treffpunkte, kann er sich Detailinformationen zu dieser Benachrichtigung anzeigen. Dadurch wird die Nachricht von der Liste der Systemnachrichten gelöscht.</p> <p>In den Detailinformationen können sehr unterschiedliche Arten von Informationen stehen, z.B. ist es möglich einen POI anzuhängen, mit dem der Student direkt interagieren kann (siehe 4.3.1.7)</p> |



## 4.2.2 Use Cases "Soziale Erweiterung"

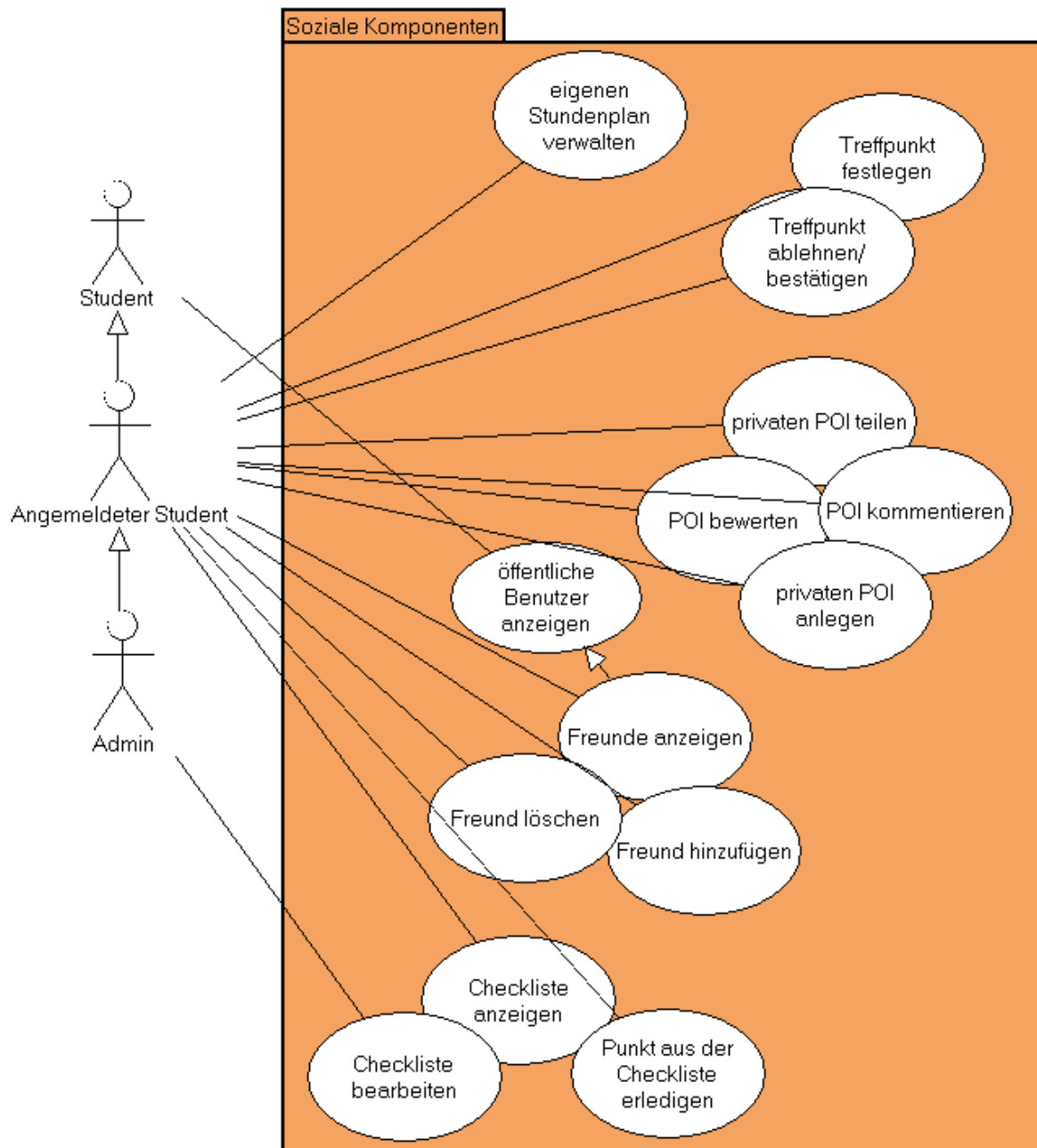


Abbildung 5: Use Case Diagramm der sozialen Erweiterungskomponente

#### 4.2.2.1 Eigenen Stundenplan verwalten

|                     |   |
|---------------------|---|
| <b>Akteur</b>       | Angemeldeter Anwender/-in   |
| <b>Beschreibung</b> | Beinhaltet das Anzeigen und das Bearbeiten des eigenen Stundenplans. Hier können einzelne Vorlesungen, Seminare oder auch regelmäßige Meetings hinzugefügt werden. Diese Objekte sind mit POIs (z.B. Vorlesungssälen) verknüpft. Das heißt es ist möglich direkt zu den im Stundenplan enthaltenen Objekten, bzw. deren POIs zu navigieren. |

#### 4.2.2.2 Treffpunkt festlegen

|                     |  |
|---------------------|--|
| <b>Akteur</b>       | Angemeldeter Anwender/-in  |
| <b>Beschreibung</b> | Ein Student kann andere im System registrierte Benutzer zu einem Treffen einladen. Dazu muss er eine Liste von Benutzern wählen und Koordinaten, Zeitpunkt und Betreff angeben. Alternativ zu den Koordinaten kann auch ein POI als Treffpunkt gewählt werden. |

#### 4.2.2.3 Treffpunkt bestätigen/ablehnen

|                     |  |
|---------------------|--|
| <b>Akteur</b>       | Angemeldeter Anwender/-in  |
| <b>Beschreibung</b> | Erhält ein Student eine Anfrage zu einem Treffpunkt kann er zu diesem seine Teilnahme bestätigen oder ablehnen. Die anderen Teilnehmer dieses Treffens können den Teilnahmestatus der anderen Teilnehmer einsehen (offen/bestätigt/abgelehnt). |

#### 4.2.2.4 Privaten POI anlegen

|                     |   |
|---------------------|---|
| <b>Akteur</b>       | Angemeldeter Anwender/-in   |
| <b>Beschreibung</b> | Ein angemeldeter Student kann eigene POIs anlegen, die dann nur für ihn oder für Studenten sichtbar sind, denen dieser POI freigegeben wurde (siehe 4.2.2.5). |

#### 4.2.2.5 Privaten POI teilen

|                     |  |
|---------------------|--|
| <b>Akteur</b>       | Angemeldeter Anwender/-in  |
| <b>Beschreibung</b> | Ein privater POI kann einem anderen angemeldeten Studenten freigegeben werden. |

#### 4.2.2.6 POI bewerten

|                     |   |
|---------------------|---|
| <b>Akteur</b>       | Angemeldeter Anwender/-in                             |
| <b>Beschreibung</b> | Ein POI kann auf einer Skala von 1-5 bewertet werden. |

#### 4.2.2.7 POI kommentieren

|                     |  |
|---------------------|--|
| <b>Akteur</b>       | Angemeldeter Anwender/-in  |
| <b>Beschreibung</b> | Es können Kommentare zu einem POI hinzugefügt werden. Diese sind für alle Studenten einsehbar. |

#### 4.2.2.8 Öffentliche Benutzer anzeigen

|                     |  |
|---------------------|--|
| <b>Akteur</b>       | Anwender/-in   |
| <b>Beschreibung</b> | Jeder Student hat die Möglichkeit sein Profil offenzulegen. Es kann dann auch von Studenten angezeigt werden, die nicht angemeldet sind. |

#### 4.2.2.9 Freunde anzeigen

|                     |   |
|---------------------|---|
| <b>Akteur</b>       | Angemeldeter Anwender/-in   |
| <b>Beschreibung</b> | Angemeldete Studenten haben die Möglichkeit, eine Freundesliste mit angemeldeten Studenten zu pflegen, um die Interaktion mit diesen zu vereinfachen. |

#### 4.2.2.10 Freund hinzufügen

|                     |   |
|---------------------|---|
| <b>Akteur</b>       | Angemeldeter Anwender/-in   |
| <b>Beschreibung</b> | Angemeldete Studenten können zur eigenen Freundesliste hinzugefügt werden. Dazu wird erst eine Anfrage versendet, welche dann vom eingeladenen Studenten bestätigt werden muss. |

#### 4.2.2.11 Freund löschen

|                     |   |
|---------------------|---|
| <b>Akteur</b>       | Angemeldeter Anwender/-in                                     |
| <b>Beschreibung</b> | Ein Freund kann wieder von der Freundesliste gelöscht werden. |

#### 4.2.2.12 Checkliste anzeigen

|                     |  |
|---------------------|--|
| <b>Akteur</b>       | Angemeldeter Anwender/-in  |
| <b>Beschreibung</b> | Um neuen Studenten den Start an der Uni zu vereinfachen gibt es eine Checkliste, welcher der Student durchlaufen kann. Diese beinhaltet z.B. wichtige Gebäude oder Verwaltungseinrichtungen. Die Checkliste sollte von der Fachschaft gepflegt werden. Unten ist ein Vorschlag für eine Checkliste angegeben (siehe 4.4).<br>Der Student kann sich die Punkte der Checkliste anzeigen und diese einzeln abarbeiten. Er muss dabei nicht in der empfohlenen Reihenfolge vorgehen. |

#### 4.2.2.13 Checkliste bearbeiten

|                     |  |
|---------------------|--|
| <b>Akteur</b>       | Admin  |
| <b>Beschreibung</b> | Ein Admin kann die Checkliste pflegen. Ein Punkt der Checkliste enthält folgende Elemente:<br>- Überschrift<br>- Beschreibung<br>- POI |

#### 4.2.2.14 Punkt aus der Checkliste erledigen

|                     |   |
|---------------------|---|
| <b>Akteur</b>       | Angemeldeter Anwender/-in   |
| <b>Beschreibung</b> | Ein Punkt der Checkliste kann als erledigt markiert werden. Er wird dann unauffälliger dargestellt. |

### 4.3 POI Kategorien

Einzelne POIs müssen einer Kategorie angehören. Diese Kategorien dienen unter anderem der Darstellung auf der Karte anhand eines Icons und können zum Filtern der Suchergebnisse benutzt werden.

Im Folgenden sind Kategorien angegeben, welches ein Navigationssystem einer Uni unterstützen sollte:

- Toiletten
- Behindertengerechte Toiletten
- Ruheräume / Stillräume
- Kopierer
- Lernplätze
- Mensa
- Cafeteria
- Supermarkt
- Vorlesungssaal
- Übungsraum/Seminarraum
- Sekretariat
- Fachschaft
- Mitarbeiterraum
- Bahnhof/Bushaltestelle
- Bibliothek
- Bankautomat
- Wohnheim
- Prüfungsamt

## 4.4 Checkliste

Im Navigationssystem erhält jeder Student eine eigene Checkliste, welche er abarbeiten kann. Welche Punkte in seiner Checkliste auftauchen hängt dabei von den Daten ab, welche er in seinem Profil hinterlegt, z.B. Studienrichtung, Studiengang, Erststudium oder ähnlichem.

Im Folgenden ist ein Vorschlag für eine Checkliste für den Studiengang Softwaretechnik angegeben. Diese wurde anhand der Umfrageergebnisse und eigenen Erfahrungen mit den typischen Problemen als Erstsemester erstellt.

Folgende Checkliste könnte unter dem Use-Case „Checkliste anzeigen“ (4.2.2.12) präsentiert werden:

- Empfohlene Vorlesungen
- Bibliothek erkunden
- Mensa
- Prüfungsamt
- Vorkurse
- Erstsemester Rallye (Treffpunkt teilnehmen)
- Studentenwohnheim finden (Anlaufstellen)
- RUS/Informatik Pool Account

## 4.5 Zusammenfassung der Umfrage

Die Ergebnisse der Umfrage wurden im Folgenden zusammengestellt und aufbereitet. Diese Punkte haben letztendlich zu den oben klassifizierten Use-Cases geführt und werden deshalb ohne Wertung kurz aufgelistet. Einige der Punkte werden nicht direkt durch die Use-Cases abgedeckt. Diese Probleme der Studenten können nicht sinnvoll im Kontext einer Navigationssoftware umgesetzt werden.

Wünsche der an der Umfrage teilnehmenden Studenten an ein Informationssystem für Studenten:

- **Kategorie Lernplätze**  
Oft bekommt man erst in späteren Semestern mit, wo es ruhige Plätze zum Lernen und Arbeiten gibt. Zu Beginn des Studiums fehlt einem oft dieses Wissen.
- **Gebäude Stadtmitte/Vaihingen haben mehrere Bezeichnungen**  
Studenten, die Vorlesungen in Vaihingen und in der Stadtmitte haben, wissen oft nicht sofort, welche Gebäude auf welchem Campus liegen.

- **Erstsemester-Einführungen**  
Häufig bekommt man zu spät mit, dass von der Fachschaft Einführungsveranstaltungen angeboten werden. Hier wäre es gut, explizit darauf hingewiesen zu werden.
- **Benachrichtigungen, bzw. zeitnahe Informationen**  
Bei Ausfall oder Raumänderung einer Veranstaltung wäre es gut, ein zentrales und kurzfristiges Benachrichtigungssystem zu haben.
- **Angebotene Tutorien/Vorkurse**  
Ähnlich wie bei den angebotenen Veranstaltungen der Fachschaften wäre hier eine Unterstützung durch Erinnern und Informieren wünschenswert.
- **Räume an der Uni, die häufig nicht direkt gefunden werden**  
Mensa, Copy-Shop, Vorlesungssäle, Seminarräume, Toiletten, Cafeteria, Prüfungsamt, Sekretariat
- **Öffnungszeiten**  
Infos zu den Öffnungszeiten der verschiedenen Gebäude werden von vielen als große Hilfe angesehen.
- **Prüfungsordnung**  
Oft finden es die Befragten sehr schwierig verlässlich die zu ihnen gehörende Prüfungsordnung ausfindig zu machen.
- **Seminarauslastung**  
Hier wäre es schön, eine zentrale Stelle zu haben, an der man erfährt, welche Veranstaltungen zu wie viel Prozent ausgebucht sind um hier nicht ständig in ausgebuchte Seminare zu laufen.
- **Orientierungshilfe für V38 1. / 2. - Stock**  
Das Informatikgebäude ist mit seiner symmetrischen Form des 1. und 2. Stockwerkes für viele Studenten anfangs sehr undurchsichtig.
- **Offizieller Ansprechpartner**  
An wen kann man sich mit welchen Fragen wenden.
- **Studentenwohnheime**  
Hier ist es schwierig ein Zimmer zu bekommen. Eine bessere Unterstützung wäre hier für viele wünschenswert.

## 5 Evaluierung

### 5.1 Bewertungsrichtlinien

#### 5.1.1 Kriterien der Bewertung

Für die Bewertung und Evaluierung wurden aus den oben betrachteten Kriterien (Kapitel 4) nicht alle zur Bewertung herangezogen. Kriterien, die für die Evaluierung keinen Mehrwert haben wurden nicht aufgenommen. Zum Beispiel ist es so, dass alle Plattformen über leicht unterschiedliche Sensoren verfügen, die Unterschiede aber zu gering sind und nicht von der Plattform sondern vom eigentlichen Gerät abhängen. Im Folgenden finden sich deshalb nur die Kriterien, die für eine Bewertung im Sinne dieser Arbeit nützlich sind.

#### Auflistung der Kriterien:

##### 1. Verkaufszahlen/Verbreitung

Hierbei werden verschiedenen Faktoren beobachtet. Zum einen werden die Verkaufszahlen aus 2010 betrachtet. Zusätzlich werden aber auch die Zahl der verfügbaren Apps und der Anteil kostenpflichtiger und kostenfreier Apps für diese Bewertung heran gezogen.

- Dabei entfallen 7 Punkte auf die Verkaufszahlen (unter 10% entspricht 0 Punkte, 10%: 1Punkt, je weitere 5%: 1 Punkt, >= 40%: 7 Punkte).
- Weitere 2 Punkte entfallen auf die Anzahl der verfügbaren Apps.
- Einen Punkt gibt es, wenn die Mehrheit der Apps (>50%) kostenlos und frei verfügbar ist.

##### 2. Finanzieller Aufwand

Bei den Kosten werden nur zu zahlende Beträge an die Hersteller betrachtet. Kosten im Sinne von Entwicklerstunden oder Aufwände zum Einrichten werden an einer anderen Stelle bewertet.

Als Kosten werden dabei sowohl die einmaligen Anschaffungskosten als auch die pro App-Verkauf anfallenden Abgaben betrachtet. Zu den Anschaffungskosten können neben denen für den Erwerb eines Accounts auch Kosten für den Erwerb von Software oder benötigten Lizenzen fallen.

- Kosten für die Registrierung als Entwickler (1 Punkt)
- Abgaben pro App (1 Punkt)
- Kosten für Entwicklungsumgebung (Hardware/Software) (3 Punkte)



### 3. Einstiegsbarriere für Entwickler

Für dieses Kriterium müssen viele Faktoren berücksichtigt werden. Hierbei kommen aus Entwicklersicht folgende Aspekte zum Tragen:

- Dokumentation (2 Punkte)  
Ist eine ausreichende Dokumentation vorhanden um sich mit der Materie vertraut zu machen? Gibt es eine starke Community, die in Foren alle Probleme diskutiert?
- Entwicklungsumgebung (3 Punkte)  
Welcher Aufwand muss betrieben werden um die benötigten Tools zu installieren und zu konfigurieren? Für welche Plattformen sind diese verfügbar? Wie sieht es mit Emulatoren aus?
  - Emulator ja/nein (1 Punkt)
  - Plattformunabhängigkeit (1 Punkt)
  - Funktionsumfang (1 Punkt)
- Verbreitung der Programmiersprache (2 Punkte)  
Im Falle von weit verbreiteten Programmiersprachen kann man auch Hilfe aus nicht plattformspezifischen Foren oder Communities bekommen.

### 4. Offenheit des Systems (4 Punkte)

Bei der Bewertung wird in diesem Fall betrachtet, ob das System von einem Hersteller alleine unter Verschluss gehalten wird oder ob auch bei der Entwicklung des Systems auf Open Source gesetzt wird. Ebenfalls betrachten sollte man hier, wie gut die Schnittstellen offengelegt sind.

- Betriebssystemquellcode verfügbar (3 Punkte)
- Native Compiler / Gerätetreiberentwicklung (1 Punkt)

### 5. Distributionsweg (3 Punkte)

Hier wird bewertet, auf welchem Wege es möglich ist, eine entwickelte App zu verteilen. Dazu gehören Fragen nach einem bestehenden Market, andere Vertriebswege und Kontrollen, bzw. Zensur durch den Hersteller.

- Manuelle Paketinstallation auch ohne Market (1 Punkt)
- Zensurverhalten des Betreibers (1 Punkt)
- Vorhandensein eines zentralen Distributionsweg (1 Punkt)

**6. Zukunftsaussicht (10 Punkte)**

Plattformen, hinter denen große Unternehmen stehen und die am Markt mit aktuellen oder gar zukunftsweisenden Techniken bestehen, können hierbei mehr Punkte holen als veraltete Systeme, denen eine baldige Ablösung droht. Hintergrund hierbei ist es, keinen hohen Aufwand zu betreiben, wenn das System nach nur kurzer Zeit obsolet wird und durch einen Nachfolger unter anderer Technik abgelöst werden muss.

- Wachstum der Verkaufszahlen 2009 → 2010 (5 Punkte)  
( $\leq 0\%$  - 0 Punkte / pro angefangene 5 % - 1 Punkt)
- Stärke der Konzerne im Hintergrund (4 Punkte)
- Entwicklungseinstellung geplant? (1 Punkt)

**7. Versionfragmentierung (3 Punkte)**

Die Anzahl der Punkte ist hierbei hoch, wenn eine Plattform eine niedrige Versionsfragmentierung hat. Je weniger verschiedene Versionen einer Plattform existieren, desto mehr Geräte erreicht man, wenn man seine Software für eine Version optimiert.

**8. Qt**

Dieses Kriterium beurteilt nur, ob eine Plattform über Qt verfügt oder nicht. Dies ist später ausschlaggebend dafür, ob die bestehende Codebasis weiter verwendet werden kann oder nicht. Da dieses Kriterium nur bedingt von Bedeutung ist, wird dieses später nicht mit den obigen Kriterien verrechnet, sondern es werden zwei Szenarien der Bewertung vorgenommen. In einem werden die ersten acht Kriterien ohne dieses Kriterium betrachtet und es wird eine Empfehlung geben, beim anderen wird betrachtet, ob es einen Unterschied macht, wenn dieses Kriterium mit hineinspielt.

## 5.1.2 Das Bewertungsschema

Um eine aussagekräftige Bewertung machen zu können müssen die gewonnen Ergebnisse quantifiziert werden. Folgendes Vorgehen findet dabei Anwendung. Den oben genannten Kriterien werden maximal erreichbare Punkte zugeordnet. Die Menge der zu erreichenden Punkte spiegelt dabei eine Gewichtung der Kriterien wieder.

Nachfolgend werden die vorgesehenen Maximalpunktzahlen aufgelistet:

|                                      |           |
|--------------------------------------|-----------|
| 1. Marktanteil/Verbreitung           | 10 Punkte |
| 2. Finanzieller Aufwand              | 05 Punkte |
| 3. Einstiegsbarrieren für Entwickler | 07 Punkte |
| 4. Offenheit des Systems             | 04 Punkte |
| 5. Distributionsweg                  | 03 Punkte |
| 6. Zukunftsaussichten                | 10 Punkte |
| 7. Versionfragmentierung             | 03 Punkte |
| 8. QT                                | 00 Punkte |
| =====                                |           |
| Summe:                               | 42 Punkte |

Für das QT Kriterium werden hierbei keine Punkte vergeben, da dieser später gesondert betrachtet wird.

## 5.2 Evaluierung der Plattformen im Kontext der Anforderungen

Im Weiteren werden nun für jede der Plattformen die Kriterien einzeln bewertet und die Punkte aller Kriterien addiert. Die so ermittelte Summe der erreichten Punkte wird dann in ein prozentuales Verhältnis zu den maximal erreichbaren Punkten gebracht. Dadurch erreicht jede Plattform eine Prozentzahl, die angibt, zu wie viel Prozent sie die Vorgaben erreicht hat. Diese Einteilung erlaubt es am Ende ein Ranking der verschiedenen Plattformen vorzunehmen.

Im ersten Szenario gibt es eine Betrachtung der Ergebnisse nur auf Basis der oben genannten prozentualen Wertung. Im zweiten Teil wird geschaut, ob es Plattformen gibt, die QT unterstützen und so eine Weiterverwendung des existierenden Codes erlauben. Sollte dies der Fall sein, so werden diese Plattformen in einem neuen separaten Ranking verglichen.

Nachdem beide Szenarien betrachtet wurden, wird im Anschluss eine Empfehlung der Autoren abgegeben.

## **5.2.1 Android 2.x**

### **5.2.1.1 Marktanteil/Verbreitung**

**Ergebnis: 9 Punkte (von 10 Punkten)**

**Begründung:**

Android hat im Bezugsjahr 2010 einen Anteil von 32% an den Verkaufszahlen erreicht. Nach der oben genannten Rechnung ergibt dies sechs Punkte, die sich wie folgt zusammensetzen: Ein Punkt für das Erreichen von zehn Punkten, plus fünfmal einen Punkt je angefangene weitere 5%. Zusätzlich verfügt Android mit 275.000 Apps über sehr viele Apps. Außerdem gibt es einen weiteren Punkt dafür, dass mit 66% über die Hälfte der Apps kostenlos sind.

### **5.2.1.2 Finanzieller Aufwand**

**Ergebnis: 3 Punkte (von 5 Punkten)**

**Begründung:**

Das Android SDK und auch das NDK sind völlig kostenlos für nahezu alle Betriebssysteme erhältlich. Dies ergibt drei Punkte. Für Apps werden wie bei den anderen führenden Herstellern allerdings 30% der Einnahmen vom Entwickler verlangt. Zusätzlich liegen die einmaligen Registrierungskosten mit 25 \$ doch deutlich über denen von beispielsweise Symbian.

### **5.2.1.3 Einstiegsbarrieren für Entwickler**

**Ergebnis: 7 Punkte (von 7 Punkten)**

**Begründung:**

Hier ist Android in allen geforderten Sparten führend. So gibt es auf den Seiten des Herstellers mehr als genug frei und leicht zugängliche Dokumentationen. Die Entwicklungsumgebung kann für nahezu alle Betriebssysteme kostenlos heruntergeladen werden und ist in Sekunden fertig konfiguriert. Zusätzlich sind verschiedene freie Emulatoren bereits in diese Plattformen integriert. Für die Programmiersprache gibt es zwei Punkte. Java gehört mit zu den derzeit am weitesten verbreiteten Sprachen.

### **5.2.1.4 Offenheit des Systems**

**Ergebnis: 4 Punkte (von 4 Punkten)**

**Begründung:**

Android wird als Open Source vertrieben und der Quellcode kann jederzeit kostenlos auf den Seiten des Herstellers bezogen werden. Zusätzlich ist es möglich über den nativen Compiler des NDKs auch auf der Ebene von Gerätetreibern weitere Entwicklungen zu betreiben.

### **5.2.1.5 Distributionsweg**

**Ergebnis: 3 Punkte (von 3 Punkten)**

**Begründung:**

Man kann Apps für Android sowohl über einen zentralen Market als auch durch freie Paketinstallation zum Endverbraucher bringen. Zusätzlich gibt es im Market selbst keine so strengen Kontrollen wie bei einigen anderen Herstellern.

### **5.2.1.6 Zukunftsaussicht**

**Ergebnis: 10 Punkte (von 10 Punkten)**

**Begründung:**

Android weist mit 24,2% im Vergleich der beiden Jahre 2009 und 2010 das stärkste Wachstum auf. Zusätzlich gehört die Firma Google zu den umsatzstärksten Konzernen in ganz Amerika. Aber nicht nur Google hat ein Interesse am Fortbestand des Systems, auch große Firmen wie Vodafone, HTC oder Sony Ericsson wollen ein Fortbestehen des Systems. Deshalb gibt es für die Zukunftsaussicht die maximale Punktzahl.

### **5.2.1.7 Versionsfragmentierung**

**Punkte: 1 (von 3 Punkten)**

**Begründung:**

Der Hersteller ist zwar bestrebt auf vielen Endgeräten die gleiche Version zum Laufen zu bringen. Doch gibt es auch hier einige Geräte, die noch auf älteren Versionen laufen. Die meisten dieser Versionen sind allerdings fast vollständig kompatibel zu neueren Versionen.

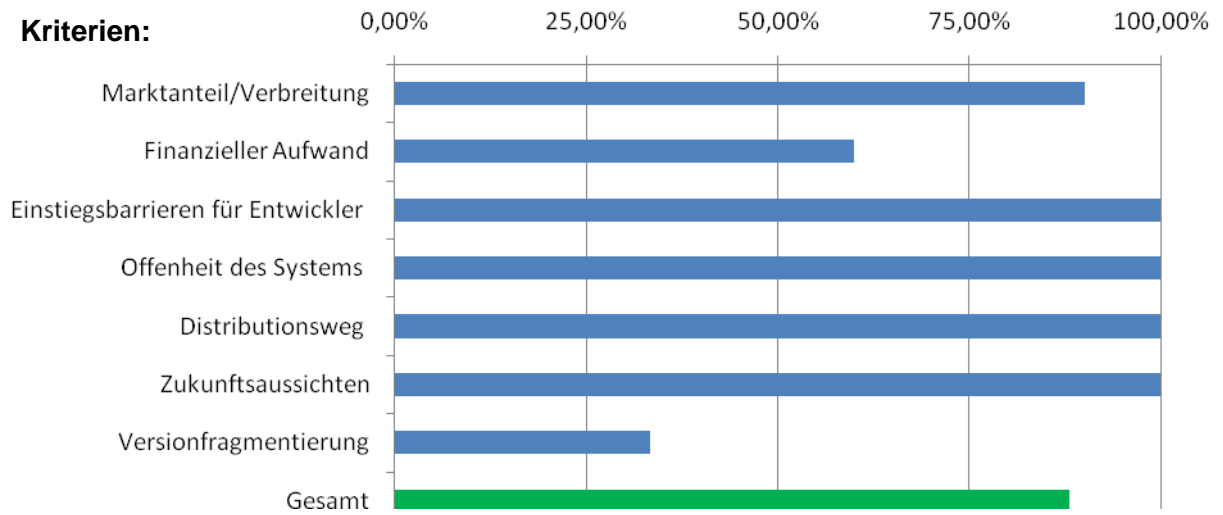
### **5.2.1.8 Qt**

Qt ist für Android standardmäßig nicht verfügbar. Es gibt allerdings bereits verschiedene Ansätze Qt verfügbar zu machen.

### **5.2.1.9 Gesamtergebnis für Android**

Android erhält in Summe 37 Punkte, dies entspricht 88% der Gesamtpunktzahl von 42 Punkten.

## Android 2.x



**Abbildung 6: Prozentuale Darstellung der Ergebnisse von Android**  
Wie zu erkennen ist erreicht Android bis auf die Versionsfragmentierung sehr gute Ergebnisse.

## 5.2.2 Windows Phone 7

### 5.2.2.1 Marktanteil/Verbreitung

**Ergebnis: 0 Punkte (von 10 Punkten)**

**Begründung:**

Mit 3,1% Anteil an den Verkaufszahlen in 2010 bekommt die Plattform keine Punkte bei diesen Kriterien. Die Anzahl der Apps liegt mit ca. 6.000 ebenfalls weit hinter den Zahlen der Konkurrenz zurück. Hinzu kommt, dass mit ca. 72% der Großteil dieser Apps kostenpflichtig sind.

### 5.2.2.2 Finanzieller Aufwand

**Ergebnis: 0 Punkte (von 5 Punkten)**

**Begründung:**

Um für das Windows Phone entwickeln zu können müssen 99\$ jährlich bezahlt werden, um sich als Entwickler zu registrieren. Hinzu kommen ggf. bis zu 12.000\$ für die Entwicklungsumgebung. Diese entfallen zwar, wenn man die eingeschränkte Express-Edition von Visual Studio nimmt, allerdings hat diese einen erheblich eingeschränkten Funktionsumfang. 30% Abgaben pro App sind identisch zu anderen Herstellern.

### 5.2.2.3 Einstiegsbarrieren für Entwickler

**Ergebnis: 7 Punkte (von 7 Punkten)**

**Begründung:**

Die Dokumentation des Herstellers ist sehr gut und umfangreich. Die Entwicklungsumgebung kann zwar extrem teuer werden, lässt sich dafür aber leicht installieren und handhaben. Die .Net-Sprachen C# und VB sind bereits weit verbreitet und viel im Einsatz.

### 5.2.2.4 Offenheit des Systems

**Ergebnis: 0 Punkte (von 4 Punkten)**

**Begründung:**

Das Betriebssystem wird als Close Source vertrieben. Zusätzlich gibt es derzeit keinerlei Chance, mittels eines nativen Compilers auf Gerätefunktionen direkt Einfluss zu nehmen.

### 5.2.2.5 Distributionsweg

**Ergebnis: 2 Punkte (von 3 Punkten)**

**Begründung:**

Für den Vertrieb der Apps steht ein zentraler Market zur Verfügung. Es sind derzeit keine Berichte im Umlauf, nach denen der Hersteller diesen Market zensiert. Allerdings hat man keine andere Möglichkeit Apps zu vertreiben, außer über den zentralen Market.

### **5.2.2.6 Zukunftsaussicht**

**Ergebnis: 5 Punkte (von 10 Punkten)**

**Begründung:**

Die aktuell vorliegenden Zahlen zeigen für Windows Mobile für 2009 und 2010 einen rückläufigen Trend. Allerdings beziehen sich diese Zahlen auf das alte Betriebssystem von Microsoft. Aktuell investiert Microsoft viel Aufwand und Gelder in das neue Windows Phone, was sie auch mit der Partnerschaft mit Nokia bewiesen haben. Diese für Microsoft teure Partnerschaft zeigt, wie wichtig Microsoft den mobilen Markt für die Zukunft einschätzt.

### **5.2.2.7 Versionsfragmentierung**

**Ergebnis: 3 Punkte (von 3 Punkten)**

**Begründung:**

Es gibt derzeit keine Versionsfragmentierung. Dies bedeutet, dass alle Geräte mit der gleichen Version ausgestattet sind. Außerdem sorgt Microsoft mittels vertraglich vereinbarter Spezifikationen dafür, dass ein leichtes Aktualisieren der Software auf allen Geräten gewährleistet sein sollte.

### **5.2.2.8 Qt**

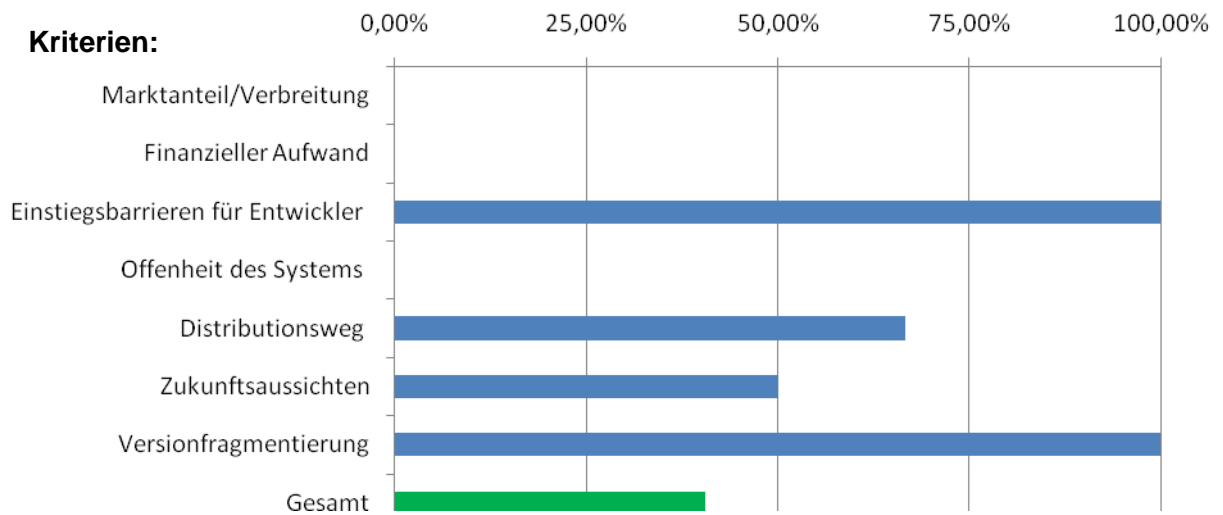
Qt steht für das Windows Phone derzeit nicht zur Verfügung.

### **5.2.2.9 Gesamtergebnis für Windows Phone 7**

Windows Phone 7 erhält in Summe 17 Punkte, dies entspricht 40,5% der Gesamtpunktzahl von 42 Punkten.



## Windows Phone 7



**Abbildung 7: Prozentuale Darstellung der Ergebnisse von Windows Phone 7**  
Windows Phone 7 ist die jüngste Plattform, die Zukunft wird die Entwicklung hier zeigen.

## 5.2.3 iOS 4.x

### 5.2.3.1 Marktanteil/Verbreitung

**Ergebnis: 5 Punkte (von 10 Punkten)**

**Begründung:**

Das iOS 4.x verfügt in 2010 über 16% Marktanteil, dies bedeutet drei Punkte. Zusätzlich verfügt diese Plattform mit 350.000 über die mit Abstand meisten Apps. Da aber 67 % dieser Apps kostenpflichtig sind, wird ein Zusatzpunkt nicht vergeben.

### 5.2.3.2 Finanzieller Aufwand

**Ergebnis: 0 Punkte (von 5 Punkten)**

**Begründung:**

Das iOS 4.x erhält keine Punkte. Dies liegt daran, dass man sich mit 99\$ im Jahr als Entwickler registrieren muss. Pro App werden 30% Abgaben verlangt. Doch der höchste Kostenfaktor liegt in der Entwicklungsumgebung. Diese ist zwar kostenlos, es wird aber eine (teure) Mac-Umgebung benötigt.

### 5.2.3.3 Einstiegsbarrieren für Entwickler

**Ergebnis: 4 Punkte (von 7 Punkten)**

**Begründung:**

Die Dokumentation ist auf den Webseiten des Herstellers ausreichend und gut zugänglich erreichbar. Es existiert bereits eine große und aktive Community. Die Plattform liefert einen guten Emulator und hat einen sehr großen Funktionsumfang. Da es sich bei der Programmiersprache um eine ausschließlich von der Herstellerfirma verwendete Version handelt, erhält die Plattform keine weiteren Punkte für dieses Kriterium.

### 5.2.3.4 Offenheit des Systems

**Ergebnis: 0 Punkte (von 4 Punkten)**

**Begründung:**

Der Quellcode ist Closed Source und nicht verfügbar. Man kann über die SDK keine eigenen nativen Funktionen ergänzen. Daher gibt es hier keine Punkte für das iOS 4.x.

### 5.2.3.5 Distributionsweg

**Ergebnis: 1 Punkt (von 3 Punkten)**

**Begründung:**

Man kann Apps nur über den zentralen Market verteilen. Der Hersteller prüft jede Anwendung ausführlich. Es gibt im Internet häufig Berichte über die Zensur von Apps ohne erkennbaren Grund.

### 5.2.3.6 Zukunftsaussicht

**Ergebnis: 5 Punkte (von 10 Punkten)**

**Begründung:**

Von 2009 nach 2010 waren die Verkaufszahlen rückläufig. Deshalb wurden dafür keine Punkte vergeben. Apple gehört wirtschaftlich zu den Top fünf Firmen der USA [60]. Eine Einstellung der Entwicklung ist auf absehbare Zeit nicht geplant.

### 5.2.3.7 Versionsfragmentierung

**Ergebnis: 3 Punkte (von 3 Punkten)**

**Begründung:**

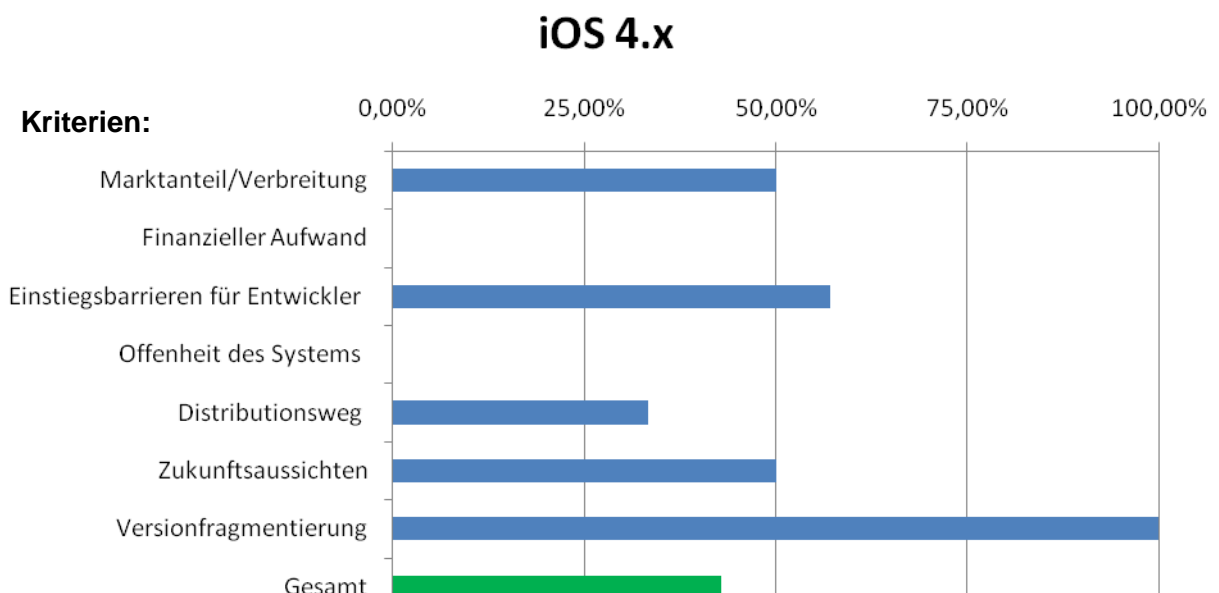
Hier ist der Hersteller sehr bestrebt, nur eine gültige Version auf allen Smartphones vorzuhalten. Man kann sich daher immer an der aktuellen Version orientieren. Als zusätzlicher Vorteil ist hier zu sehen, dass beim Erscheinen einer neuen Version per Knopfdruck die Installation vorgenommen werden kann.

### 5.2.3.8 Qt

Qt ist für das iOS 4.x standardmäßig nicht verfügbar.

### 5.2.3.9 Gesamtergebnis für iOS 4.x

iOS 4.x erhält in Summe 18 Punkte, dies entspricht 43% der Gesamtpunktzahl von 42 Punkten.



**Abbildung 8: Prozentuale Darstellung der Ergebnisse von iOS.**

**Auffallend ist der hohe finanzielle Aufwand und die fehlende Offenheit des Systems.**

## 5.2.4 Symbian^3

### 5.2.4.1 Marktanteil/Verbreitung

**Ergebnis: 9 Punkte (von 10 Punkten)**

**Begründung:**

Symbian verfügte 2010 über 30,6% der Verkaufszahlen. Dies entspricht sechs Punkten. Da es für Symbian sehr viele auf Java basierende Apps gibt, die größtenteils kostenlos außerhalb des Market vertrieben werden, ergeben sich für Symbian 2 Punkte für die Vielzahl der Apps und einen für die Tatsache, dass mehr als 50% kostenlos sind.

### 5.2.4.2 Finanzieller Aufwand

**Ergebnis: 4 Punkte (von 5 Punkten)**

**Begründung:**

Mit einem Euro Registrierungskosten als Entwickler ist Symbian der mit Abstand günstigste Anbieter unter allen Konkurrenten. Es gibt keinen Punkt dafür, da im Market wie bei den andern Herstellern ebenfalls 30 % pro App berechnet wird. Die Entwicklungsumgebung mit allem Zubehör ist allerdings völlig kostenfrei. Dies ergibt zwei Punkte.

### 5.2.4.3 Einstiegsbarrieren für Entwickler

**Ergebnis: 5 Punkte (von 7 Punkten)**

**Begründung:**

Die Dokumentation ist auf den Webseiten des Herstellers noch ausreichend und gut zugänglich erreichbar. Die Dokumentation bekommt nur einen Punkt, da diese von den Seiten von Nokia verschwinden wird. Die Entwicklungsumgebung verfügt über Emulatoren und einen umfangreichen Funktionsumfang, ist aber nur unter Windows erhältlich. Dies führt damit zu zwei Punkte. Für Symbian kann man in Java C++ und Qt entwickeln. Dies sind relativ verbreitete Sprachen und bringt zwei weitere Punkte ein.

### 5.2.4.4 Offenheit des Systems

**Ergebnis: 2 Punkte (von 4 Punkten)**

**Begründung:**

Der Quellcode ist laut Angaben des Herstellers Open Source. Es gibt derzeit aber keinen offiziellen Link um die Sources herunterzuladen. Symbian verfügt zudem nicht über einen nativen Compiler.

#### **5.2.4.5 Distributionsweg**

**Ergebnis: 3 Punkte (von 3 Punkten)**

**Begründung:**

Symbian verfügt mit dem OVI-Store von Nokia über einen zentralen Distributionsweg. Zusätzlich können aber auf den meisten Geräten auch einzelne jar-Pakete installiert werden. Im Market wird durch den Betreiber nicht zensiert.

#### **5.2.4.6 Zukunftsaussicht**

**Ergebnis: 0 Punkte (von 10 Punkten)**

**Begründung:**

Von 2009 nach 2010 waren die Verkaufszahlen rückläufig. Deshalb wurden hier keine Punkte vergeben. Symbian verfügt leider nicht mehr über eine zentrale Firma, die ein Interesse daran hat, diese Plattform weiter voranzutreiben. Nach dem Zusammenschluss von Nokia und Microsoft ist die Plattform nicht weiter relevant.

#### **5.2.4.7 Versionsfragmentierung**

**Ergebnis: 0 Punkte (von 3 Punkten)**

**Begründung:**

Für nahezu jede Handygeneration der Firma Nokia gibt es eine eigene modifizierte Version der Software. Hier lässt sich kaum etwas erreichen, wenn nur für eine Version etwas entwickelt wird.

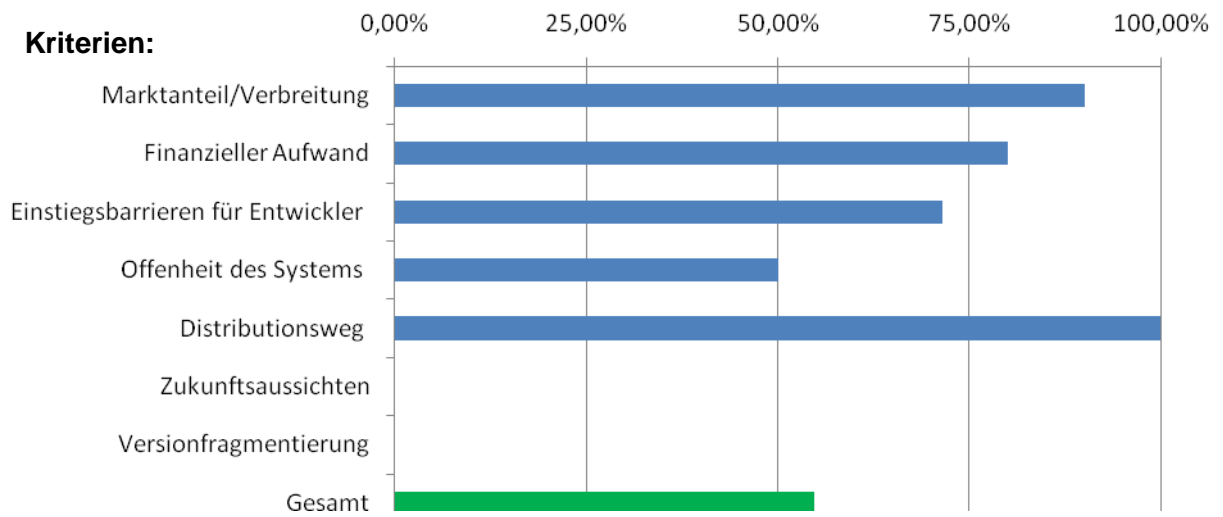
#### **5.2.4.8 Qt**

Qt ist für ist unter Symbian standardmäßig verfügbar und einsetzbar. Hier gibt es keine Einschränkungen.

#### **5.2.4.9 Gesamtergebnis für Symbian**

Symbian erhält in Summe 23 Punkte, dies entspricht 55% der Gesamtpunktzahl von 42 Punkten.

## Symbian^3



**Abbildung 9: Prozentuale Darstellung der Ergebnisse von Symbian  
Weit verbreitet aber keine Zukunftsaussichten.**

## **5.2.5 Blackberry OS 6**

### **5.2.5.1 Marktanteil/Verbreitung**

**Ergebnis: 2 Punkte (von 10 Punkten)**

**Begründung:**

Das Blackberry OS 6 verfügt 2010 über einen Marktanteil von 14,4%. Dies entspricht zwei Punkten. Da über die Anzahl der verfügbaren Apps und deren Aufteilung in „kostenlos“ und „kostenpflichtig“ keine Angaben ermittelt werden konnten, werden dafür keine Punkte vergeben.

### **5.2.5.2 Finanzieller Aufwand**

**Ergebnis: 0 Punkte (von 5 Punkten)**

**Begründung:**

Hier fehlen auf Seiten des Herstellers Angaben.

### **5.2.5.3 Einstiegsbarrieren für Entwickler**

**Ergebnis: 5 Punkte (von 7 Punkten)**

**Begründung:**

Es gibt für Entwickler eine gute und vollständige Dokumentation. Zusätzlich ist Java als Programmiersprache weit verbreitet. Zwei Punkte Abzug gibt es dafür, dass die IDE nicht in allen Systemen verfügbar ist und der Emulator nur unter Windows läuft.

### **5.2.5.4 Offenheit des Systems**

**Ergebnis: 0 Punkte (von 4 Punkten)**

**Begründung:**

Das Betriebssystem ist Closed Source und über einen nativen Compiler wurden keine Angaben gemacht.

### **5.2.5.5 Distributionsweg**

**Ergebnis: 3 Punkte (von 3 Punkten)**

**Begründung:**

Anwendungen können über einen zentralen App-Market oder per manuelle Paketinstallation vertrieben werden. Zusätzlich gibt es keine Einschränkungen seitens des Herstellers.

### 5.2.5.6 Zukunftsaussicht

**Ergebnis: 2 Punkte (von 10 Punkten)**

**Begründung:**

Von 2009 zu 2010 gab es einen Verlust bei den Verkaufszahlen. Der Hersteller RIM kann als solides Unternehmen mit großer Erfahrung im Business-Bereich betrachtet werden. Dies bringt zwei Punkte. Nach offiziellen Angaben sind allerdings die Einstellung der derzeitigen Plattform und deren Austausch durch ein neues System geplant.

### 5.2.5.7 Versionsfragmentierung

**Ergebnis: 0 Punkte (von 3 Punkten)**

**Begründung:**

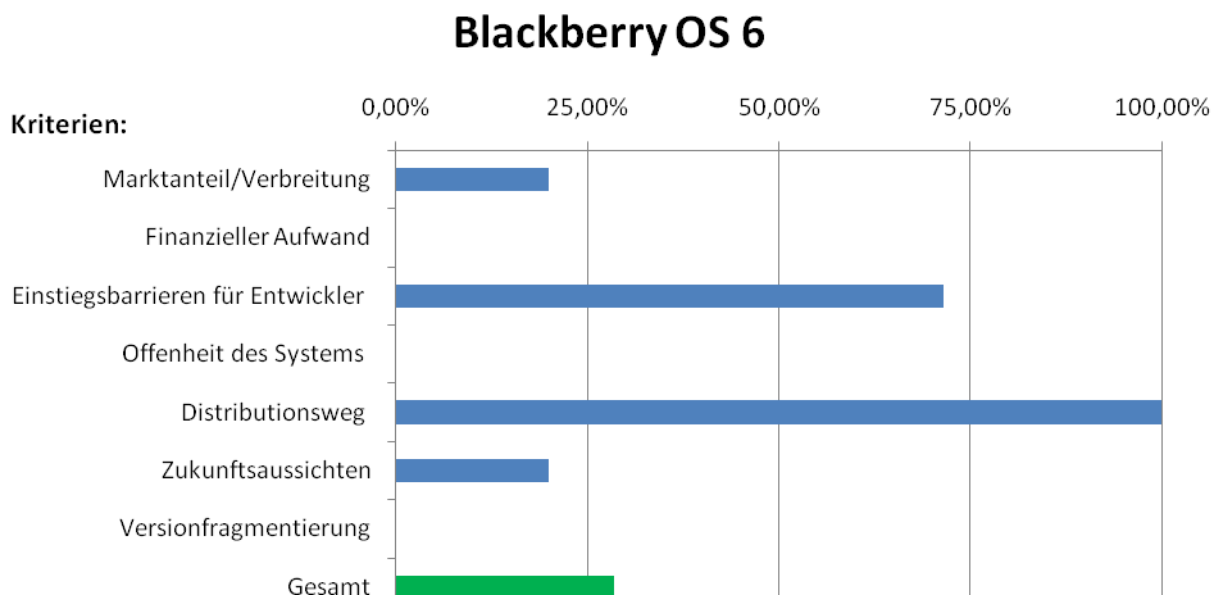
Hierzu fehlen die Angaben vom Hersteller.

### 5.2.5.8 Qt

Qt ist für das Blackberry OS 6 nicht verfügbar, weder native noch durch Drittanbieter.

### 5.2.5.9 Gesamtergebnis für Blackberry

Blackberry erhält in Summe 7 Punkte, dies entspricht 16,7% der Gesamtpunktzahl von 42 Punkten.



**Abbildung 10: Prozentuale Ergebnisse Blackberry OS**  
**Geringe Zukunftsaussichten trotz sehr guter Distribution.**



## 5.3 Ergebnis

Das Ergebnis dieser Fachstudie ist, wie unten aus der Tabelle ersichtlich, eindeutig. Android dominiert mit 88% der Gesamtpunktzahl als zu favorisierende Plattform dieses Ranking. Gründe für dieses Ergebnis gibt es viele.

Im Folgenden gibt es eine kurze Auflistung der wichtigsten Erkenntnisse:

- Großer Marktanteil
- Stabiles Wachstum
- Quelloffen
- Gut dokumentiert

Details zu den genannten Punkten finden sich in den vorangegangenen Kapiteln.

Die obigen Punkte sind vornehmlich generelle Vorzüge von Android gegenüber der Konkurrenz. Zusätzlich gab es noch verschiedene Vorteile, die Android in Bezug auf die Zielsetzung dieser Fachstudie hat:

- **Apps können in Java (SDK) oder C (NDK) geschrieben werden.**

Der Vorteil dabei ist, dass diese Sprachen weit verbreitet sind und die Einstiegsbarriere für die Entwickler damit nicht so hoch ist. Zusätzlich können Informationen auch aus anderen unabhängigen Quellen bezogen werden.

- **Die Verteilung ist unabhängig vom App-Market möglich.**

Es ist nicht notwendig die App zentral und weltweit zugänglich zu machen. Die Entwickler können sich dafür entscheiden ihre App unabhängig zu verteilen.

- **Qt ist per Erweiterung verfügbar.**

Sollte an dem ursprünglichen Ziel festgehalten werden, die Portierung oder Neuentwicklung mit der bekannten Programmiersprache fortzusetzen, gibt es bereits existierende Ansatzpunkte. Drittanbieter haben große Teile des Qt-Frameworks bereits auf Android portiert, und es wäre denkbar hier Qt einzusetzen.

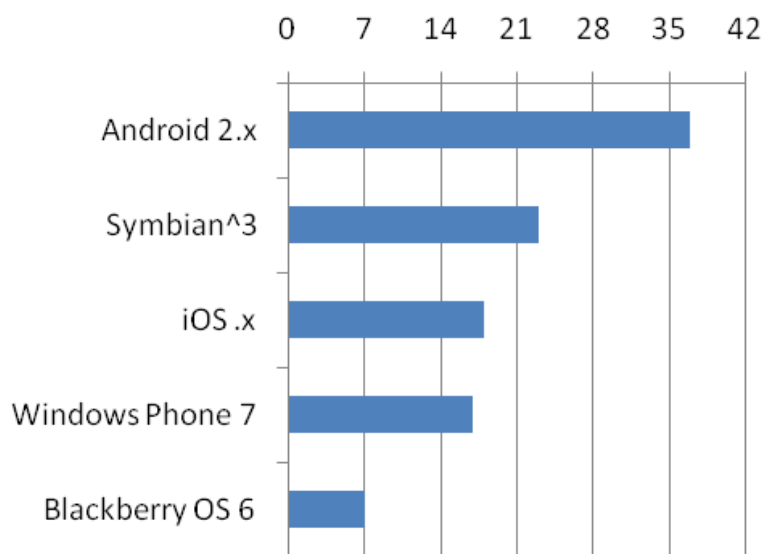
Aufgrund dieser Erkenntnisse ergibt sich, dass **Android** das Ranking dieser Fachstudie als für sich entschieden hat.

Nachfolgend (Abb. 11) ist die tabellarische und grafische Auswertung aller Ergebnisse im Vergleich dargestellt. Daraus kann man eindeutig ableiten wie die aktuelle Bewertung der Plattformen aussieht.

|                        | Punkte    | Prozent | Ranking |
|------------------------|-----------|---------|---------|
| <b>Android 2.x</b>     | 37 Punkte | 88%     | 1.      |
| <b>Symbian^3</b>       | 23 Punkte | 55%     | 2.      |
| <b>iOS 4.x</b>         | 18 Punkte | 43%     | 3.      |
| <b>Windows Phone 7</b> | 17 Punkte | 42%     | 4.      |
| <b>Blackberry OS 6</b> | 7 Punkte  | 17%     | 5.      |

**Tabelle 12: Die Auswertung der Punktevergabe führt zu Android 2.x als geeignetste Plattform, gefolgt von Symbian, iOS 4.x und Windows Phone 7. Blackberry OS 6 kommt nach der Bewertung nur eine geringe Bedeutung zu.**

## Punkteverteilung



**Abbildung 11: Grafische Darstellung des Gesamtergebnisses der Punkte im Balkendiagramm, aus welcher der Große Abstand von Android zu den anderen Plattformen gut zu erkennen ist.**

## 6 Beispielhafte Portierung

### 6.1 Übersicht

In der beispielhaften Portierung soll das bestehende Kartenmaterial, welches in einem XML-Format ähnlich dem CityGML-Standard vorliegt, auf einer Karte angezeigt werden. Dabei soll es möglich sein in der Karte zu zoomen, die Karte zu verschieben und die Stockwerke zu wechseln. Des Weiteren soll der GPS-Sensor ausgelesen werden, um die Karte auf die aktuelle Position zu zentrieren und der Kompass dazu verwendet werden um dem Benutzer die Blickrichtung als Dreieck anzuzeigen.

Als Testgerät verwenden wir ein HTC Desire mit 1Ghz Single-Core mit 576MB RAM.

Beim hier entwickelten Prototyp handelt es sich um einen Demonstrationsprototyp. Er ist als reine Machbarkeitstudie gedacht, und um die technische Realisierbarkeit zu überprüfen.

Der Quellcode des Prototyps liegt der Fachstudie auf einer CD bei.

### 6.2 Screenshots

Im Folgenden werden ein paar Screenshots des Prototyps dargestellt. Zuerst wird die GPS-Positionierung dargestellt, welche den Bildschirmmittelpunkt auf die aktuelle Position ausgibt und die Lat/Lon-Koordinaten ausgibt. Danach folgen einige Screenshots des gerenderten Kartenmaterials. In der beispielhaften Portierung werden nur die Surface-Elemente *FloorSurface*, *FloorPlane* und *FloorSteps* gezeichnet. Diese können vom Typ *Main*, *Opening*, *Multilevel* oder *Level* sein. Diese werden je nach Typ farblich wie folgt gezeichnet:

- *Main* oder kein angegebener Typ: schwarz
- *Opening*: rot
- *MultiLevel*: grün
- *Level*: grau

Auf dem folgenden Bild sieht man die aktuelle Position, welche als Pfeil dargestellt wird. Im Prototyp wird auch die aktuelle Position im Lat/Long-Koordinatensystem angezeigt, die intern als Integer Wert abgelegt wird um effizienter rechnen zu können. Dafür wird der Gleitkommawert mit  $10^6$  multipliziert.

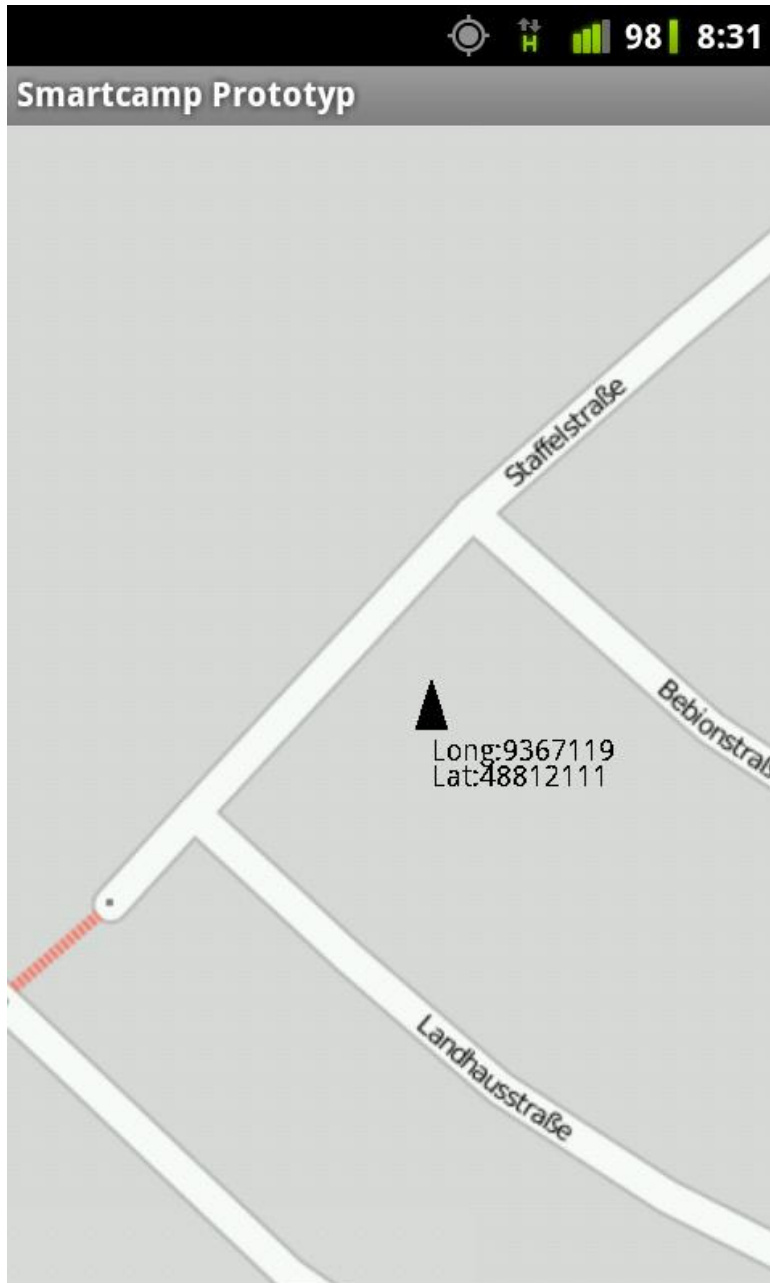


Abbildung 12: Aktuelle Position

Auf dem folgenden Screenshot sieht man einen Teil des Universitätsgeländes mit dem OpenStreetMap-Kartenmaterial über das das Kartenmaterial des TANIA-Systems gezeichnet wird. Über das Menü können folgende Aktionen aufgerufen werden:

- **Change Floor:** über diesen Button lassen sich auf einfache Weise die verschiedenen Etagen des Informatikbaus laden.
- **Toggle Overlay:** hiermit kann das Kartenmaterial des TANIA-Systems ausgeblendet werden.
- **Map neu parsen:** parst die XML-Datei erneut und schreibt die Daten in die Datenbank

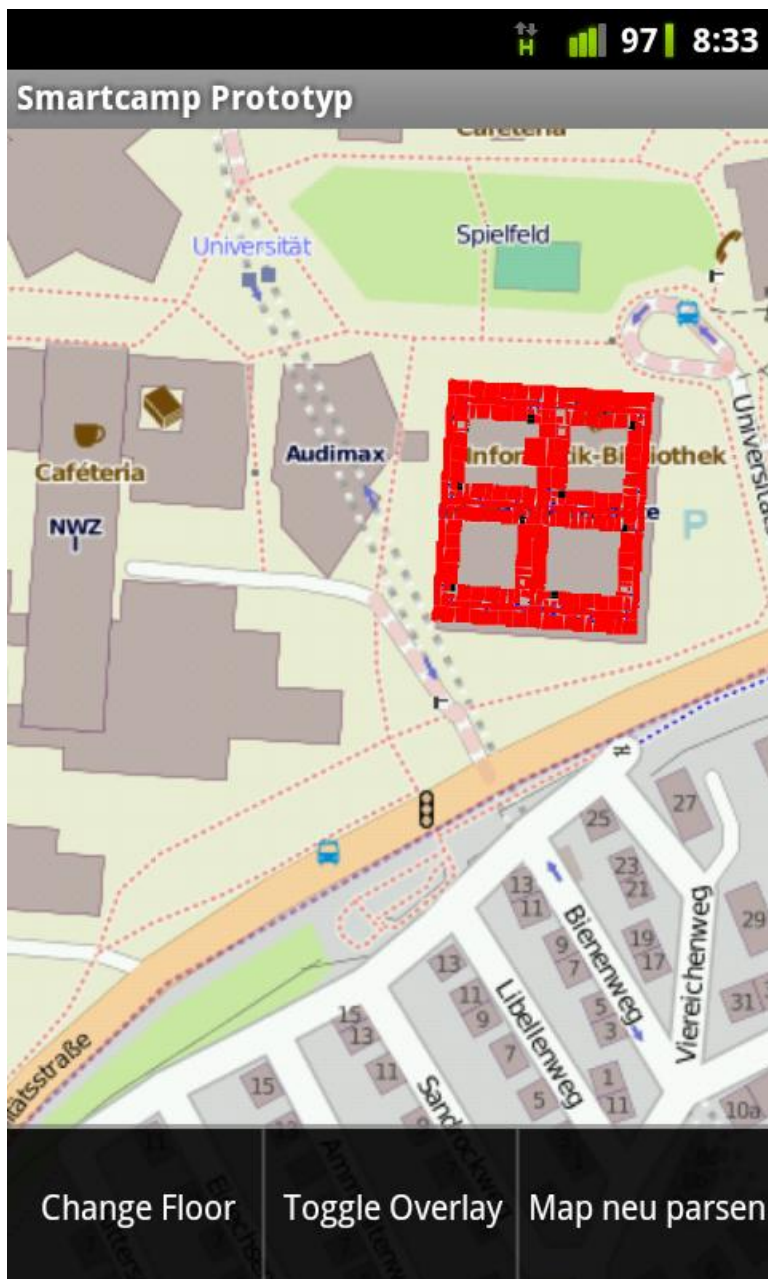


Abbildung 13: Darstellung der Karte des Universitätsgeländes

Auf den folgenden Screenshots sieht man die verschiedenen Ebenen des Informatikbaus.



Abbildung 14: Keller



Abbildung 15: Erdgeschoss

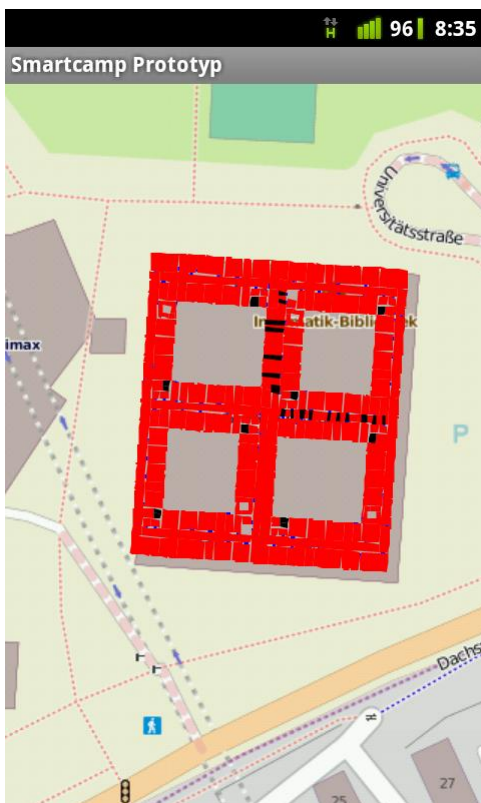


Abbildung 16: erstes Stockwerk



Abbildung 17: zweites Stockwerk

## 6.3 Ansätze, Probleme und mögliche Lösungen

In den folgenden Kapiteln wird beschrieben, welche Ansätze wir zur Deserialisierung, zum Rendern und zum Ansprechen der Sensoren gewählt haben und auf welche Probleme wir dabei gestoßen sind. Danach werden mögliche Lösungen für eine spätere Implementierung beschrieben.

### 6.3.1 Deserialisierung

Eine Deserialisierung mit einem DOM-Parser in ein Document Object Model (DOM) ist aufgrund der Kartengröße und der darausfolgenden Größe des Objektbaums nicht praktikabel, da der RAM in Android für einzelne Apps eingeschränkt zur Verfügung steht. Die Verwendung des DOM-Parsers führt beim Parsen der ungefähr 6 MB großen Datei zu einer Exception aufgrund mangelnden Speichers.

Wir sind im Prototyp daher auf einen Pull Parser ausgewichen, der jeden Tag einzeln bearbeitet. Dabei werden asynchron folgende Events ausgelöst:

1. Start des Dokuments
2. Start Tag
3. End Tag
4. Value

Der Programmieraufwand zur Verwaltung der Objekte mit deren Beziehungen, z.B. Eltern-Kind-Beziehung, erhöht sich, es ist jedoch so gut wie kein Hauptspeicher nötig.

Im Prototyp schreiben wir dabei die Objekte direkt in eine SQLite Datenbank um die Objekte später selektiv auslesen zu können. Dieser Vorgang dauert auf unserem Testgerät ein bis zwei Minuten für eine XML-Datei, das heißt eine Etage des Informatikbaus. Da das Kartenmaterial nur einmal geparkt werden muss und danach in der Datenbank zur Verfügung steht ist dieses Ergebnis für den Prototyp akzeptabel.

Das Auslesen aus der Datenbank erfolgt hierarchisch. Das heißt es werden erst die obersten Elemente ausgelesen und danach rekursiv deren Kinderelemente, was sehr ineffizient ist. Der ganze Vorgang des Auslesens dauert auf dem Testgerät ungefähr 20-30 Sekunden, was in der Praxis, zum Beispiel bei einem Etagenwechsel, nicht praktikabel ist.

Möglichkeiten zur Effizienzsteigerung wäre z.B. die Vermeidung des rekursiven Auslesens. Beispielsweise könnten alle Daten in eine Hashmap geladen werden und sind dann über die ID sehr effizient auffindbar. Eine weitere Möglichkeit wäre eine selektivere Auswahl der darzustellenden Objekte. Hier könnte man beispielsweise nur die obersten Elemente anzeigen und beim Eintreten in dieses Objekt dessen Kinder nachladen. Dabei müsste aber beim Speichern in der Datenbank aus den Kinderpolygonen ein Elternpolygon erzeugt werden. Da das Einlesen allerdings nur einmal gemacht werden muss, wäre dies ein Zugewinn an Benutzerfreundlichkeit.

### 6.3.2 Rendering

Zur Visualisierung des Kartenmaterials kommt im Prototyp in erster Linie die für Android frei verfügbare OSMDroid-Komponente [57] zum Einsatz. Dabei handelt es sich um einen Nachbau/Ersatz für Googles integrierte Karten-Komponente. Anders als die Google Komponente, die Googles Kartenmaterial nutzt, verwendet OSMDroid das frei verfügbare Kartenmaterial von OpenStreetMap [58] und erspart dem Benutzer so lästige API-Key Registrierungen oder Google-Logos im Bild. Die dem User dargestellte Karte ergibt sich im Prototyp aus der Synthese zwischen dem OpenStreetMap-Kartenmaterial (Hintergrund/Tiles) und dem darauf generierten Overlay (eigenes Kartenmaterial).

Zum Zeichnen des Kartenmaterials und des Kompasses verwenden wir Custom-Overlays welche von Overlay [59] erben und überschreiben die draw-Methode, welcher ein Canvas übergeben wird, auf dem wir zeichnen können. Dabei müssen die Kartenkoordinaten der einzelnen Elemente in die Weltkoordinaten transformiert werden.

### 6.3.3 Sensoren

Der Prototyp enthält eine exemplarische GPS-Ansteuerung. Verwendet werden dazu die vom Framework bereitgestellten Klassen, darunter der *LocationManager* [55] und das *LocationListener*-Interface [56]. Das *LocationListener*-Interface schreibt die Implementierung einer Methode *onLocationChanged* vor, welche bei entsprechenden Positionsänderungen automatisch aufgerufen wird. Das Abfrageintervall zur Bestimmung von etwaigen Positionsänderungen kann beliebig auf die eigenen Anforderungen angepasst werden.

## 6.4 Ergebnisse der Machbarkeitsstudie

Der Prototyp ermöglicht es dem Benutzer seine Position außerhalb von Gebäuden mittels GPS zu ermitteln und diese auf einer Karte von Open-Street-Map anzuzeigen. In dem Informatikgebäude werden pro Stockwerk die Informationen aus den Karten des TANIA-Systems gezeichnet und der Benutzer kann sich hier die Aufteilung der Räume und ihre Position anzeigen lassen. Es ist möglich über das Menü die einzelnen Stockwerke zu wechseln.

Im Rahmen dieser Machbarkeitsstudie sind noch einige oben genannte Konflikte aufgetreten aber das generelle Ergebnis ist überzeugend. Es ist möglich das TANIA-System für sehende Studenten/-in auf Android zu portieren.



## 7 Diskussion

Im Folgenden sollen einige diskussionswürdige Punkte aufgegriffen und näher erläutert werden.

### a) QT (Portierung vs. Neuimplementierung)

Zu Beginn dieser Fachstudie gab es die Überlegung, ob und wie man Qt berücksichtigt und damit den Wunsch zur Weiterverwendung des bestehenden Codes in die Bewertung mit einbezieht. Es wurden daher zwei grundlegende Vorgehensweisen in Betracht gezogen: Eine Neuimplementierung auf Basis des Android SDK, oder ein Portierung möglichst großer Bestandteile des heutigen Systems unter Beibehaltung der C/Qt-Technologien. Es war jedoch absehbar, dass außer Nokias Meego kaum eine der Plattformen über Qt-Unterstützung verfügen wird. Als dann im Laufe der Recherchen zu dieser Arbeit die Einstellung der Meego-Entwicklung durch Nokia bekannt gegeben wurde, schien die Abkehr von Qt endgültig besiegelt. Zwar existiert für Android seit kurzem eine erste Qt-Implementierung, diese ist jedoch bisher fast ausschließlich von einer Privatperson entwickelt worden und es bleibt abzuwarten, ob daraus eine größere Community und damit eine vertretbare Entwicklungsplattform wachsen wird. Des Weiteren existieren für die hier geplante Software deutlich geänderte Anforderungen als für die bestehende Variante (z.B. geänderte Zielgruppe), was den Portierungsaufwand weiter in die Höhe treiben würde. Aufgrund dieser Tatsachen fiel die Entscheidung letztendlich zu Gunsten einer Neuimplementierung mit Hilfe des Android-SDK und gegen eine Portierung der C/Qt-Basis aus.

### b) Windows Phone 7

Das Ergebnis aus der Arbeit beruht auf den aktuellen Daten aus dem ersten Halbjahr des Jahres 2011. Unter Berücksichtigung aktueller Entwicklungen um Windows Phone 7 ist damit zu rechnen, dass die Bedeutung dieser Plattform deutlich zunehmen wird.

### c) Prototypen-Implementierung

Schwierigkeiten und Entscheidungen bei der Implementierung des Prototyps wurden bereits direkt im entsprechenden Kapitel 6 aufgegriffen und diskutiert.

## 8 Zusammenfassung

Diese Fachstudie untersucht mögliche Szenarien, um eine am VIS entstandene Navigations-Software (TANIA-System) von einem herkömmlichen Desktop-System auf eine gängige mobile Plattform zu portieren. Dabei werden dafür relevante Merkmale der bestehenden Software erläutert und Besonderheiten der in Frage kommenden mobilen Plattformen vorgestellt. Die Zielgruppe der aktuellen Software, bestehend aus sehgeschädigten Benutzern, ist hier als besonderes Merkmal hervorzuheben, soll aber für eine mögliche Portierung nur eine untergeordnete Rolle spielen, da die portierte Variante in erster Linie für sehende Benutzer konzipiert wird.

Die Fachstudie enthält ein detailliert herausgearbeitetes Ratingsystem für die derzeit auf dem Markt konkurrierenden mobilen Plattformen und eine daraus resultierende Empfehlung für die dabei bestbewertete Android-Plattform. Im Anschluss wird der Grundstein für ein möglicherweise folgendes Entwicklungsprojekt gelegt, indem denkbare Use-Cases einer zukünftigen Software auf Grundlage von uns erhobenen Umfragewerten modelliert und ein Prototyp mit ausgewähltem Funktionsumfang implementiert und dokumentiert werden.

Im Rahmen der Use-Cases werden dabei zwei Ausbaustufen des geplanten Systems vorgestellt: Eine minimale Basisvariante, welche grundlegende Navigationsfunktionen enthält und eine erweiterte Variante, welche aus gängigen sozialen Netzwerken bekannte Funktionen zur Kommunikation der Benutzer untereinander (Gruppenbildung, Freigabe des Standorts etc.) enthält.

Zum ausgewählten Funktionsumfang des Android-Prototyps gehören die Verarbeitung des bestehenden XML-Kartenmaterials, eine dazu passende Visualisierung mit Hilfe einer frei verfügbaren OpenStreetMap-Komponente, sowie die testweise Ansteuerung des in gängigen Smartphones enthaltenen GPS-Sensors zur Markierung der eigenen Position auf der Karte.

## 9 Anhang

### 9.1 Abbildungsverzeichnis

|  |    |
|--|----|
| Abbildung 1: Prozentuale Aufteilung der Verkaufszahlen von Smartphones aus 2009 [9] .....  | 4  |
| Abbildung 2: Prozentuale Aufteilung der Verkaufszahlen von Smartphones aus 2010 [9] .....  | 5  |
| Abbildung 3: Anzahl der Apps für Android, iOS und Windows Phone [4, 5, 6] .....  | 6  |
| Abbildung 4: Use Case Diagramm des Kernsystems .....   | 26 |
| Abbildung 5: Use Case Diagramm der sozialen Erweiterungskomponente .....   | 31 |
| Abbildung 6: Prozentuale Darstellung der Ergebnisse von Android Wie zu erkennen ist erreicht Android bis auf die Versionsfragmentierung sehr gute Ergebnisse. ....                     | 44 |
| Abbildung 7: Prozentuale Darstellung der Ergebnisse von Windows Phone 7 Windows Phone 7 ist die jüngste Plattform, die Zukunft wird die Entwicklung hier zeigen. ....                  | 47 |
| Abbildung 8: Prozentuale Darstellung der Ergebnisse von iOS. Auffallend ist der hohe finanzielle Aufwand und die fehlende Offenheit des Systems. ....                                  | 49 |
| Abbildung 9: Prozentuale Darstellung der Ergebnisse von Symbian Weit verbreitet aber keine Zukunftsaussichten. ....  | 52 |
| Abbildung 10: Prozentuale Ergebnisse Blackbarry OS Geringe Zukunftsaussichten trotz sehr guter Distribution. ....  | 54 |
| Abbildung 11: Grafische Darstellung des Gesamtergebnisses der Punkte im Balkendiagramm, aus welcher der Große Abstand von Android zu den anderen Plattformen gut zu erkennen ist. .... | 56 |
| Abbildung 12: Aktuelle Position .....  | 58 |
| Abbildung 13: Darstellung der Karte des Universitätsgeländes .....   | 59 |
| Abbildung 14: Keller                      Abbildung 15: Erdgeschoss. ....  | 60 |
| Abbildung 16: erstes Stockwerk                      Abbildung 17: zweites Stockwerk .....  | 60 |

## 9.2 Tabellenverzeichnis

|  |    |
|--|----|
| Tabelle 1: Eigenschaften Android.....  | 9  |
| Tabelle 2: Vor- und Nachteile des Betriebssystems Android. ....  | 9  |
| Tabelle 3: Eigenschaften Windows Phone 7 .....   | 13 |
| Tabelle 4: Vor- und Nachteile Windows Phone 7 .....  | 13 |
| Tabelle 5: Eigenschaften iOS.....  | 16 |
| Tabelle 6: Vor- und Nachteile von iOS. ....  | 16 |
| Tabelle 7: Eigenschaften Symbian .....   | 19 |
| Tabelle 8: Vor- und Nachteile von Symbian .....  | 19 |
| Tabelle 9: Eigenschaften Blackberry .....  | 22 |
| Tabelle 10: Vor- und Nachteile von Blackberry.....   | 22 |
| Tabelle 11: Gegenüberstellung der Multi-Plattformlösungen .....  | 23 |
| Tabelle 12: Die Auswertung der Punktevergabe führt zu Android 2.x als geeignetste Plattform, gefolgt von Symbian, iOS 4.x und Windows Phone 7. Blackberry OS 6 kommt nach der Bewertung nur eine geringe Bedeutung zu..... | 56 |

### 9.3 Quellenangaben

[1] Stand 04.04.2011:

[http://de.wikipedia.org/wiki/Geography\\_Markup\\_Language](http://de.wikipedia.org/wiki/Geography_Markup_Language)

[2] Stand 04.04.2011:

<http://www.citygml.org>

[3] Stand 04.04.2011:

<http://appoftheday.com/infographic>

[4] Stand 04.04.2011:

[http://blog.androidzoom.com/docs/2010-AndroidMarketZOOM-byAndroidzoom\\_ENG.pdf](http://blog.androidzoom.com/docs/2010-AndroidMarketZOOM-byAndroidzoom_ENG.pdf)

[5] Stand 04.04.2011:

<http://wp7.apphab.com/state-windows-phone-7-infographic/>

[6] Stand 04.04.2011:

<http://www.androlib.com/appstats.aspx>

[7] Stand 17.02.2011:

<http://developer.android.com/sdk/terms.html>

[8] Stand 17.02.2011:

<http://source.android.com/source/licenses.html>

[9] Stand 24.02.2011:

<http://www.canalys.com/pr/2011/r2011013.html>

[10] Stand 04.04.2011:

[http://de.wikipedia.org/wiki/Android\\_%28Betriebssystem%29](http://de.wikipedia.org/wiki/Android_%28Betriebssystem%29)

[11] Stand 04.04.2011:

<http://www.heise.de/developer/meldung/Google-will-im-Apps-Marketplace-mitverdienen-1081623.html>

[12] Stand 04.04.2011:

<http://mashable.com/2010/10/11/microsoft-launch-windows-phone-7/>

[13] Stand 04.04.2011:

<http://pocketnow.com/thought/thoughts-on-windows-phone-7-series-btw-photon-is-dead>

[14] Stand 04.04.2011:

[http://news.cnet.com/8301-10805\\_3-20029652-75.html](http://news.cnet.com/8301-10805_3-20029652-75.html)

[15] Stand 04.04.2011:

<http://www.hna.de/nachrichten/wirtschaft-finanzen/umbruch-handy-markt-1115795.html>

[16] Stand 04.04.2011:

<http://www.heise.de/newsticker/meldung/Erste-Nokia-Handys-mit-Windows-Phone-7-sollen-2012-kommen-1189466.html>

[17] Wiley-Professional.Windows.Phone.7.Application.Development.2010

[18] Stand 04.04.2011:

[http://www.zdnet.de/news/wirtschaft\\_unternehmen\\_business\\_microsoft\\_erweitert\\_windows\\_phone\\_7\\_sdk\\_um\\_copy\\_and\\_paste\\_story-39001020-41547939-1.htm](http://www.zdnet.de/news/wirtschaft_unternehmen_business_microsoft_erweitert_windows_phone_7_sdk_um_copy_and_paste_story-39001020-41547939-1.htm)

[19] Stand 04.04.2011:

[http://www.chip.de/news/Windows-Phone-7-Update-bringt-Multitasking-und-IE9\\_47280571.html](http://www.chip.de/news/Windows-Phone-7-Update-bringt-Multitasking-und-IE9_47280571.html)

[20] Stand 04.04.2011:

[http://msdn.microsoft.com/en-us/library/ff769512\(v=vs.92\).aspx](http://msdn.microsoft.com/en-us/library/ff769512(v=vs.92).aspx)

[21] Stand 04.04.2011:

<http://create.msdn.com/downloads/?id=638>

[22] Internet Workd Business, Nr. 3/11, Seite 39

[23] Stand 04.04.2011:

<http://www.areamobile.de/news/18109-samsung-omnia-7-update-fuer-windows-phone-fuehrt-zu-totalausfall>

[24] Stand 10.03.2011:

<http://www.heise.de/newsticker/meldung/Mehr-Open-Source-im-Windows-Phone-Marketplace-1205355.html>

[24] Stand 21.02.2011:

<http://www.apple.com/de/iphone/specs.html>

[25] Stand 07.03.2011:

<http://developer.apple.com/programs/ios/>

[26] Stand 07.03.2011:

<http://www.heise.de/newsticker/meldung/Programmierer-kritisieren-Apples-App-Store-205628.html>

[27] Stand 24.02.2011:

<http://blog.symbian.org/>

[28] Stand 07.03.2011:

[http://www.forum.nokia.com/Distribute/Ovi\\_Store\\_guidelines.xhtml](http://www.forum.nokia.com/Distribute/Ovi_Store_guidelines.xhtml)

[29] Stand 10.03.2010:

<http://www.heise.de/newsticker/meldung/Erste-Nokia-Handys-mit-Windows-Phone-7-sollen-2012-kommen-1189466.html>

[30] Stand 04.04.2011:

[http://en.wikipedia.org/wiki/BlackBerry\\_OS](http://en.wikipedia.org/wiki/BlackBerry_OS)

[31] Stand 04.04.2011:

<http://meego.com/devices>

[32] Stand 04.04.2011:

<http://www.slashgear.com/nokia-n950-meego-phone-named-very-elegant-hardware-says-cto-video-22135076/>

[33] Stand 04.04.2011:

<http://www.reamobile.de/news/18011-nokia-und-die-zukunft-von-meego-und-symbian>

[34] Stand 04.04.2011:

<http://meego.com/community/blogs/imad/2011/update-intel>

[35] Stand 04.04.2011:

<http://www.golem.de/1004/74812.html>

[36] Stand 04.04.2011:

[http://www.pcworld.com/article/219115/hps\\_think\\_beyond\\_palm\\_event\\_what\\_to\\_expect.html](http://www.pcworld.com/article/219115/hps_think_beyond_palm_event_what_to_expect.html)

[37] Stand 04.04.2011:

<http://www.netmite.com/android/index.html>

[38] Stand 04.04.2011:

<http://monotouch.net>

[39] Stand 04.04.2011:

<http://monodroid.net>

[40] Stand 04.04.2011:

<http://us.blackberry.com/developers/resources/adobe.jsp>

[41] Stand 04.04.2011:

<http://www.heise.de/newsticker/meldung/Qt-Implementierung-fuer-Android-vorgestellt-1194461.html>

[42] Stand 04.04.2011:

<http://sourceforge.net/p/necessitas/home/>



[43] Stand 04.04.2011:

<http://sourceforge.net/p/necessitas/ministro/home/>

[44] Stand 04.04.2011:

<http://www.qt-iphone.com>

[45] Stand 04.04.2011:

<http://mobile360.de/kein-qt-fuer-windows-phone-7-34847.html>

[46] Stand 04.04.2011:

[http://de.wikipedia.org/wiki/Java\\_Platform,\\_Micro\\_Edition](http://de.wikipedia.org/wiki/Java_Platform,_Micro_Edition)

[47] Stand 04.04.2011:

<http://www.codepedia.com/1/J2ME>

[48] Stand 04.04.2011:

[http://de.wikipedia.org/wiki/Qt\\_%28Bibliothek%29](http://de.wikipedia.org/wiki/Qt_%28Bibliothek%29)

[49] Stand 04.04.2011:

<http://www.heise.de/newsticker/meldung/Nokia-trennt-sich-vom-Qt-Geschaeft-1203255.html>

[50] Stand 07.05.2011:

<http://cumbia.informatik.uni-stuttgart.de/ger/research/proj/sfb627/asbus/asbus-ger.html>

[51]

Dorsch, H. (2010). *iPhone 4 und Apps in der Praxis: mit iOS 4, FaceTime und Infos zur iMovie-App*. Mandl & Schwarz; Auflage: 1. Aufl. (Juli 2010).

[52]

Stäuble, M. (2010). *Programmieren für iPhone und iPad: Der Einstieg in die App-Entwicklung für das iOS 4*. Dpunkt Verlag; Auflage: 3. Aktual. (29. November 2010).

[53]

Röhlinger, F. (2010). *Das Praxisbuch zu Google Android*. Data Becker (November 2010).

[54]

Kloss, J. H. (2011). *Android-Apps - Programmierung für Einsteiger: Mobile Anwendungen entwickeln mit App Inventor*. Markt und Technik; Auflage: 1 (28. April 2011).

[55] Stand 17.05.2011

<http://developer.android.com/reference/android/location/LocationManager.html>

[56] Stand 17.05.2011

<http://developer.android.com/reference/android/location/LocationListener.html>

[57]Stand 17.05.2011

<http://code.google.com/p/osmdroid/>

[58] Stand 17.05.2011

<http://www.openstreetmap.org/>

[59] Stand 17.05.2011

<http://code.google.com/intl/de-DE/android/add-ons/google-apis/reference/com/google/android/maps/Overlay.html>

[60] Stand 23.05.2011

<http://www.aquarius.biz/de/2011/05/12/markenranking-digitale-power/>