**Zürcher Hochschule für angewandte Wissenschaften**

**Diplomstudium Informatik**

****

**Seminararbeit 5. Semester**

**Entwickeln von Anwendungen für Handheld B**

**Evaluation von 10 Frameworks für Touch Devices**

|  |  |
| --- | --- |
| Autoren | Oliver Aeschbacher  René Kamer |
| Dozent | Christian Vils |
| Projektstart | 4. Oktober 2012 |
| Projektpräsentation | 16. Januar 2013 |

Inhaltsverzeichnis

[1. Aufgabenstellung 4](#_Toc338095972)

[1.1. Ausgangslage 4](#_Toc338095973)

[1.2. Ziele der Arbeit 4](#_Toc338095974)

[1.3. Erwartetes Resultat 4](#_Toc338095975)

[1.4. Motivation 5](#_Toc338095976)

[2. Planung und Termine 5](#_Toc338095977)

[2.1. Projektplan 5](#_Toc338095978)

[2.2. Termine 5](#_Toc338095979)

[2.3. Soll-/Ist-Vergleich 6](#_Toc338095980)

[3. Einleitung 6](#_Toc338095981)

[3.1. Vorgehen 7](#_Toc338095982)

[3.1.1. Vorgehen: Evaluation der Frameworks 7](#_Toc338095983)

[3.1.2. Vorgehen: Schreiben der Lupen-App 7](#_Toc338095984)

[3.2. Aufteilen der Aufgaben 7](#_Toc338095985)

[3.2.1. Aufteilung der Evaluation 7](#_Toc338095986)

[3.2.2. Aufteilung der Programmierung 7](#_Toc338095987)

[3.3. Umgebung des Projekts 7](#_Toc338095988)

[3.3.1. Umgebung der Evaluation 7](#_Toc338095989)

[3.3.2. Umgebung der Programmierung / Framework 7](#_Toc338095990)

[3.3.2.1. Wahl des Frameworks 7](#_Toc338095991)

[3.3.2.2. Tools 7](#_Toc338095992)

[4. Quellen 7](#_Toc338095993)

[5. Abbildungsverzeichnis 8](#_Toc338095994)

# Aufgabenstellung

# Ausgangslage

Bereits die ersten Mobiltelefone enthielten kleine Anwendungen, wie Kalender,

Taschenrechner oder Spiele. Diese waren jedoch vom Hersteller exakt für die jeweiligen

Geräte konzipiert und fix installiert.  
Das erscheinen moderner Devices wie iPhone oder Android-Geräten erlaubt nun, dass man den Funktionsumfang der Geräte beinahe beliebig erweitern kann mit selbstgeschriebenen und/oder fremden Apps.  
Aus der heutigen Zeit sind Touch-Devices nicht mehr wegzudenken. Handys wie das iPhone oder das Samsung Galaxy beziehungsweise Tablets wie das iPad oder das Asus Eee Pad bieten eine schier unendliche Menge an Applikationen. Um diese zu entwickeln bedarf es der entsprechenden Frameworks.

# Ziele der Arbeit

Das Ziel dieses Projekts ist, auf der Basis von <http://woorkup.com/2010/08/25/10-useful-frameworks-to-develop-html-based-webapps-for-touch-devices/> eine Evaluation der 10 dort vorgestellten Frameworks zu vollziehen auf theoretischer Basis:

* Einfachheit
* Entwicklungsumgebung
* Support
* Tutorials
* Building
* Testing
* Verbreitung

und diese Erkenntnisse zur Verfügung zu stellen.

Auf der Basis dieser Evaluation werden 1 - 2 dieser Frameworks herangezogen, um eine kleine Lupen-Anwendung zu schreiben für Android-Telefone und darin die Erkenntnisse der Evaluation einfliessen zu lassen.   
Die Lupen-Anwendung soll eine Hilfe sein für Leute, welche nicht mehr so gut sehen können und mittels der Lupen-App kleine Schriften oder ähnliches wieder lesen können, welche zuvor von der Device-Kamera abfotografiert wurden (im Idealfall sogar eine Lupe in Realtime).

# Erwartetes Resultat

Dokument, welches die 10 Frameworks gegenüberstellt in den unter Aufgabenstellung genannten Punkten.   
Des Weiteren eine App, welche unter den 1 - 2 verheissungsvollsten Frameworks geschrieben wurden. Dazu wird ein Teaser erstellt und der ganzen Klasse zur Verfügung gestellt. Ebenfalls wird eine Präsentation von ca 40 - 60 Minuten gehalten vor der Klasse (Die Zeitdauer bestimmt sich dadurch, dass an diesem Projekt 2 Personen Arbeiten)

# Motivation

Schon länger beschäftigt uns das Thema Handhelds und die Programmierung dazu. Dieses Seminar gibt uns nun die Möglichkeit, dieses Wissen aufzubauen und zu festigen.  
Im Speziellen interessieren uns Frameworks (nicht nur in Bezug auf Handhelds) und deren Möglichkeiten, unsere Arbeit zu vereinfachen und zu vereinheitlichen. Nicht zuletzt ist auch die Möglichkeit spannend, die unter den Frameworks geschriebenen Programme auf verschiedenen Plattformen laufen zu lassen, quasi 7 Fliegen mit einer Klatsche (iOS, Android, Blackberry, Windows Phone, Palm, WebOS, Bada, Symbian).  
Daher haben wir uns entschieden, 10 dieser Frameworks einem Evaluationsprozess zu unterziehen und unter ausgewählten Frameworks eine kleine App zu realisieren.  
Da leider unsere Kriegskasse ein wenig eingeschränkt ist, wird diese Realisierung ausschliesslich auf Android stattfinden.

# Planung und Termine

# Projektplan



Der Projektplan wurde mit GanttProject entwickelt. Leider hat die aktuelle Version einen Bug und somit können die Vorgänger sowie Nachfolger eines Arbeitszeitraumes nicht visuell dargestellt werden.  
Der Projektplan in dieser Form zeigt an, in welchem Zeitraum der jeweilige Task zu erledigen ist, für die exakte Stunden-Aufteilung wird auf den Soll-/Ist-Vergleich verwiesen.  
Da der Plan aufgrund seiner Grösse in diesem Dokument etwas schlecht lesbar ist, ist dieser noch als JPG im „Documents“ Ordner des Projekts abgelegt sowie auch das Gantt-File, welches mit GanttProject geöffnet werden kann.

# Termine

Kick-Off: 19.09.2012

Projektfreigabe: 04.10.2012

Zwischenbesprechung: 12.12.2012

Präsentation: 16.01.2012

# Soll-/Ist-Vergleich

Generell werden die Projekt-Mitarbeiter mit Kürzeln versehen, René Kamer mit RKA, Oliver Aeschbacher mit OLA.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Zuständig | Soll | Ist |
| **Projektplanung** | - | **11** | **10** |
| Erstes Informieren über die Frameworks | RKA + OLA | 2 | 2 |
| Vorgehen besprechen | RKA + OLA | 1 | 2 |
| Aufteilung der Aufgaben | RKA + OLA | 2 | 1 |
| Erst-Erstellung der Dokumentation | RKA | 3 | 2 |
| Projektplan erstellen | RKA | 3 | 3 |
| **Erstellung des Teasers** | OLA | **3** |  |
| **Analyse der Frameworks** | - | **70** |  |
| Analyse von Appcelerator Titanium |  | 7 |  |
| Analyse von Sencha Touch |  | 7 |  |
| Analyse von Sproutcore Touch |  | 7 |  |
| Analyse von PhoneGap |  | 7 |  |
| Analyse von Rhodes |  | 7 |  |
| Analyse von iUi |  | 7 |  |
| Analyse von iWebKit |  | 7 |  |
| Analyse von XUI |  | 7 |  |
| Analyse von jQPad |  | 7 |  |
| Analyse von jQuery Mobile |  | 7 |  |
| **Vergleich der Frameworks** | - | **6** |  |
| Zusammenstellung der Resultate |  | 2 |  |
| Diskussion der Frameworks (inkl Doku) |  | 4 |  |
| **Projektumgebung erstellen** | - | **8** |  |
| Projektumgebung Framework 1 |  | 4 |  |
| Projektumgebung Framework 2 |  | 4 |  |
| **Lupen APP** | - | **30** |  |
| Erstellen mit Framework 1 |  | 8 |  |
| Building mit Framework 1 |  | 3 |  |
| Testing mit Framework 1 |  | 4 |  |
| Erstellen mit Framework 2 |  | 8 |  |
| Building mit Framework 2 |  | 3 |  |
| Testing mit Framework 2 |  | 4 |  |
| **Dokumentation abschliessen** | RKA + OLA | **4** |  |
| **Präsentation + Handzettel erstellen** | RKA + OLA | **10** |  |
| **Reserve** |  | **8** |  |
| **Total** |  | **150** |  |

Alle Einheiten sind in Stunden angegeben.

# Einleitung

Dieser Teil beschreibt, wie wir vorgegangen sind in der Projekt-Anfangsphase, um die detaillierte Vorgehensweise der Evaluation der Frameworks zu beschreiben sowie die detaillierte Vorgehensweise der Entwicklung der Lupen App voranzutreiben. Hierbei beschreiben wir die Kriterien und wie sie auf die Evaluation angewendet werden sowie die Gewichtung derselbigen. Des Weiteren wird der Entwicklungszyklus der Lupen-App beschrieben sowie die Anforderungen, welche an die App gestellt werden.  
Daraus resultierend werden die Aufgaben auf die jeweiligen Projektmitarbeiter aufgeteilt, was im zweiten Abschnitt dieses Kapitels dargestellt ist, sowohl für die Evaluation der Frameworks als auch für das Schreiben der Lupen-App.

# Vorgehen

# Vorgehen: Evaluation der Frameworks

Die Frameworks werden jeweils durch dieselben 8 Kriterien bewertet: Einfachheit, Entwicklungsumgebung, Support, Dokumentation, Tutorials, Building, Testing und Verbreitung. Zum Schluss werden alle Frameworks einander gegenübergestellt in einer Entscheidungsmatrix, in welcher die jeweiligen Kriterien mit Zahlen von 1 - 10 bewertet werden und danach die Summe gebildet, um die 1 - 2 geeignetsten Frameworks zu ermitteln. Die Frameworks mit der höchsten Gesamtpunktzahl werden für die Lupen-App berücksichtigt.

Die Kriterien werden nachfolgend einzeln beschrieben betreffend der Bewertung:

* Einfachheit
  + Subjektives Empfinden: Die Einfachheit des eingesetzten Frameworks ist schwerlich objektiv zu beschreiben, da unterschiedliches Vorwissen oder generelle Präferenzen stark in die Bewertung mit einfliessen können. Deshalb wird hier ausnahmsweise auch das subjektive Empfinden als Kriterium miteinbezogen. Dies ist aber das schwächere Kriterium als die Anzahl benötigter Technologien, um die nicht-technische Beschreibung des subjektiven Empfindens ein wenig abzuschwächen.
  + Anzahl benötigter Technologien: Man kann verschiedene Technologien einsetzen. Alle Frameworks basieren auf HTML, CSS und JavaScript, jedoch in unterschiedlicher Ausprägung. Weiters gibt es noch verschiedene zusätzliche Möglichkeiten wie z.B. XML im W3C Widget Specification Format oder ähnliches. Als ideal werden 2-3 Technologien angesehen, welche zum Schreiben einer App ausreichen. Zu wenige Möglichkeiten in der Basis sind schlecht sowie auch zu viele Technologien, welche in einer Basisanwendung eingesetzt werden müssen. Dem wird in diesem Kriterium Rechnung getragen.
  + Intuitive Benutzbarkeit: Werden die eingesetzten Technologien so eingesetzt, wie man sie aus dem Alltag kennt oder werden diese anders benutzt, umgeordnet, eventuell sogar eingeschränkt in der Benutzung? Ein hohes Mass an Normalgebrauch gibt eine schnelle Einarbeitungszeit, weshalb dies hoch zu gewichten ist.
* Entwicklungsumgebung
  + Aufzählung: Eine Aufzählung der Entwicklungsumgebungen. Mehr Möglichkeiten bieten Entwicklergruppen grössere Flexibilität: Der Web-Programmierer schreibt lieber mit einem Web-Programmier Tool, der alteingesessene Programmierer eher in einem Plaintext-Editor.
  + Alternativen/Erweiterungen: Gibt es Alternativen (evtl. auch etwas exotische) welche unter dem Punkt „Aufzählung“ beschriebenen Tools ergänzen/verbessern?
  + Hilfestellungen durch Framework: Wie weit geben die Frameworks Hilfestellung, bieten z.B. Text-Completion, auf das Framework zugeschnittene Fertigbausteine oder Hinweise an? Je mehr Hilfestellungen existieren, desto besser wird dieser Punkt bewertet.
* Support
  + Supporthotline: Existiert eine Supporthotline? Wie schätzt man deren Qualität ein, ist diese in Landessprache verfügbar?
  + Online-Möglichkeiten: Gibt es weitere Möglichkeiten den Support zu erreichen? Z.B. Mail, Chat, Live-Support oder ähnliches? Gibt es ein Entwicklerforum oder eine Online-Gruppe, in welchen man sich austauschen kann? Je mehr Möglichkeiten existieren, umso besser wird hier bewertet.
  + Preis: Was sind die Kosten für den Support? Wie wird dies auf die Projektgrösse berechnet, auf die Anzahl Entwickler? Was ist das Preismodell? Das Preis/Leistungsverhältnis wird bewertet.
  + Anleitungen im Netz: Gibt es anständige Anleitungen, welche im Netz verfügbar sind? Sind sie vom Anbieter erstellt oder muss auf Dritt-Anleitungen zurückgegriffen werden? Je mehr Anleitungen von guter Qualität vorhanden sind, umso besser wird dieses Kriterium bewertet.
* Dokumentation
  + Verfügbarkeit: Wie ist die Verfügbarkeit der Dokumentation? Gibt es sie nur als Buch oder sind sie online verfügbar? Je weniger Aufwand betrieben werden muss, umso besser wird dieses Kriterium bewertet.
  + Struktur: Hat die Dokumentation eine Struktur oder ist es ein simples Textfile ohne Möglichkeiten zur Suche? Ist es einfach, etwas zu finden in der Dokumentation? Je einfacher etwas zu finden ist, umso besser wird hier bewertet.
  + Ausführlichkeit: Wie ausführlich ist die Dokumentation? Sobald man eine Ausführlichkeit erreicht, welche das Verständnis erleichtert, wird hier gut bewertet. Eine zu hohe Ausführlichkeit kann jedoch kontraproduktiv sein, was ebenfalls in diesen Punkt einfliesst.
* Tutorials
  + Google Tutorial Suche: Hier wird eine simple Google-Suche gemacht der Form: „<Framework-Name>“ + „Tutorial“. Je mehr Treffer Google zurückgibt, umso mehr Punkte werden vergeben. 10 Punkte für die meisten Treffer, 1 Punkt für die wenigsten Treffer.
  + Qualität der ersten 5 Google-Hits: Es wird die Ausführlichkeit und Verständlichkeit der ersten 5 Google Treffer bewertet. Tutorials sollten einfach aufgebaut und simpel sein, jedoch trotzdem ein gewisses Mass an Komplexität aufweisen. Das richtige Mittelmass sollte angestrebt werden und wird entsprechend bewertet.
* Building
  + Wie einfach /aufwändig ist der Prozess: Ist das Building einfach zu machen? Muss man hohe Aufwände betreiben, um den Build zu erstellen? Je einfacher der Prozess umso besser wird bewertet.
  + Verschiedene Möglichkeiten: Existieren verschiedene Möglichkeiten, den Build zu erstellen? Z.B. Lokal, Online, per Post?
  + Build-Tool: Stellt der Framework-Anbieter Build-Tools zur Verfügung oder muss auf ein IDE oder ähnliches zurückgegriffen werden (Cross-Platform). Gibt es ein Build-Tool wird dies positiv bewertet.
* Testing
  + Wie kann getestet werden: Sind Testing-Tools oder Frameworks vorhanden? Kann anständig Debugging betrieben werden? Muss man auf Methoden wie Print-Outs zurückgreifen? Je mehr Möglichkeiten das Framework von sich aus bietet, desto besser wird hier bewertet.
  + Methodik: Sind die einsetzbaren Test-Möglichkeiten methodisch oder strukturiert? Konformität mit der gängigen Lehrmeinung wird hier hoch bewertet.
* Verbreitung
  + Google Suche: Hier wird eine simple Google-Suche gemacht mit dem Namen des Frameworks. Je mehr Treffer Google zurückgibt, umso mehr Punkte werden vergeben. 10 Punkte für die meisten Treffer, 1 Punkt für die wenigsten Treffer.
  + Anzahl Apps: Wie viele mit diesem Framework geschriebene Apps sind verfügbar. Je mehr Apps damit erstellt wurden, umso besser fällt die Punktzahl aus.
  + Persönliche Erfahrung: Hat man schon von diesem Framework gehört? Gibt es gute oder schlechte Kritiken, welche man schon gelesen hat? Eine Zusammenfassung der Eindrücke spiegelt sich in der vergebenen Punktzahl wieder.
* Spezielles: Gibt es noch irgendwelche speziell nennenswerte Features oder Vorteile, welche dieses Framework auszeichnen? Hiermit können maximal 2 Zusatzpunkte vergeben werden.

# Vorgehen: Schreiben der Lupen-App

# Aufteilen der Aufgaben

# Aufteilung der Evaluation

Die Aufteilung erfolgt der Logik nach mit 50/50 aufgeteilt auf die Projektmitarbeiter. Dabei wurde der Grösse der jeweiligen Frameworks einigermassen Rechnung getragen, dass die Arbeitslast in etwa gleich verteilt ist, wie in der Planung bereits berücksichtigt.

|  |  |
| --- | --- |
| Analyse von Appcelerator Titanium | RKA |
| Analyse von Sproutcore Touch | RKA |
| Analyse von PhoneGap | RKA |
| Analyse von Rhodes | RKA |
| Analyse von iUi | RKA |
| Analyse von iWebKit | OLA |
| Sencha Touch | OLA |
| Analyse von XUI | OLA |
| Analyse von jQPad | OLA |
| Analyse von jQuery Mobile | OLA |

# Aufteilung der Programmierung

# Evaluation der Frameworks

# PhoneGap

**Einfachheit**

**Entwicklungsumgebung**

**Support**

**Dokumentation**

**Tutorials**

**Building**

**Testing**

**Verbreitung**

# Gegenüberstellung der Frameworks

# Projekt Lupen-App

# Umgebung des Projekts

# Umgebung der Evaluation

# Umgebung der Programmierung / Framework

# Wahl des Frameworks

# Tools

# Quellen

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

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

<http://www.sproutcore.com/>

<http://www.phonegap.com/>

<http://rhomobile.com/products/rhodes/>

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

<http://www.iui-js.org/>

<http://iwebkit.net/>

<http://xuijs.com/>

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

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

# Abbildungsverzeichnis