



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Projekt-Bericht

Julia Pliszka

Pervasive Gaming: iLife

Julia Pliszka
Pervasive Gaming: iLife

Ausarbeitung im Rahmen der Veranstaltung Projekt
im Studiengang Master of Science Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuer: Prof. Dr. Olaf Zukunft

Abgegeben am 8. März 2009

Inhaltsverzeichnis

1 Einführung	4
2 Anforderungen an iLife	5
2.1 iCalendar	6
2.2 iOrganizer	7
2.3 iBlogger	7
3 Architektur und Realisierungablauf	8
4 Projektmanagement	13
5 Fazit und Ausblick	14
Literaturverzeichnis	15

1 Einführung

Im Wintersemester 2008/2009 fand im Rahmen des Masterstudiengangs im Fachbereich Informatik an der Hochschule für Angewandte Wissenschaften Hamburg eine Projektveranstaltung statt. Das Projekt wurde durch Herrn Prof. Dr. Zukunft unter dem Namen Pervasive Gaming betreut. *Pervasive Gaming, wörtlich übersetzt durchdringendes Spielen, bezeichnet neue Spielformen, in denen Informations- und Kommunikationstechnik genutzt werden, um Grenzen herkömmlicher Spiele in einer realen Umgebung aufzuheben und erweiterte (virtuelle) Spielwelten für Spieler und Zuschauer zu schaffen* [Fraunhofer].

Das Projektteam bestand aus acht Studenten und dem Betreuer Herrn Prof. Dr. Zukunft. Nach einigen Teammeetings wurde der Beschluss gefasst, ein Framework für Pervasive Gaming zu erstellen und zwei darauf basierende Anwendungen. Die Idee für die erste Anwendung wurde bereits sehr frühzeitig formuliert. Hier handelt es sich um das Rollenspiel. Die zweite Anwendungsidee entstand nach der Aufteilung der Teams. Es wurden innerhalb des Projektes drei Teams gebildet: das Framework Team, das Rollenspiel-Team und das Team der zweiten Anwendung, welches später den Namen iLife-Team bekam.

Das Framework Team bestand aus drei Personen; Arazm Hosieny, Tobias Hutzler und Peter Salchow. Diese Personen waren dafür zuständig, ein Framework zu entwickeln, welches Basis Funktionen für die schnelle Erstellung von Pervasive Gaming Anwendungen bietet.

Das Rollenspiel-Team bestand ebenfalls aus drei Personen; Dennis Dedaj, Sascha Knuth und Thomas Preisler. Das Ziel dieser Gruppe bestand im Erstellen einer Rollenspiel- Anwendung auf einem mobilen Telefon, welches auf dem Framework basiert.

Das iLife-Team wurde durch nur zwei Studenten gebildet; Amine El Ayadi und Julia Pliszka. Diese beiden Teammitglieder realisierten die zweite Anwendung für das mobile Telefon, welches auf den Namen iLife getauft wurde. Grob gesagt handelt es sich hierbei um einen Lebensplaner mit Historisierungsfeatures.

Die Projektleitung übernahm die Verfasserin Julia Pliszka, stellvertretend durch A. Hosieny. Als zweites wurde über die Wahl der Plattform diskutiert. Zur Wahl standen Windows Mobile, Symbian OS, die iPhone-Plattform oder Android. Die Entscheidung fiel auf Android [Android]. Der Nachteil war jedoch, dass zum Projektanfang noch keine Android unterstützende Hardware auf dem Deutschen Markt vorhanden war, und es unsicher war, ob sich dies bis zum Projektende ändern würde. Jedoch haben die Vorteile, mit Java zu programmieren, die Innovation zu testen und weitere, überwogen.

2 Anforderungen an iLife

Die Idee der iLife Anwendung ist über eine Vielzahl von Meetings zwischen der Verfasserin und dem Projektbeteiligten Amine El Ayadi entstanden. Zudem hat der Projektbetreuer Prof. Dr. Zukunft maßgeblich bei der Ideenfindung mitgewirkt. Der Grundgedanke war es, eine Anwendung zu konzipieren, mit der ein Tagesablauf historisiert werden kann, eine Art Blogger. Es sollen nicht nur einfache Tagebucheinträge publiziert werden, sondern zudem auch z.B. im Urlaub erstellte Fotos oder die erste empfangene SMS vom Liebsten, also alle auftretenden Aktivitäten im Alltag. Das mobile Telefon ist eins der meist mitgeführten mobilen Geräte und bietet in den meisten Fällen einen großen Funktionsumfang (Fotos machen, Videos aufnehmen etc.). Aus diesem Grund bietet eine derartige Anwendung auf einem mobilen Telefon einen großen Mehrwert. Die Verwaltung aller im Alltag auftretenden Aktivitäten kann mit dem mobilen Telefon am besten gehandhabt werden, da es, wie bereits erwähnt, immer in Reichweite des Anwenders ist und über eine Vielzahl von protokollierbaren Funktionalitäten verfügt (Foto, Video, Termine).

Die Blogger Anwendung und der Gedanke des beträchtlichen Potenzials des mobilen Telefons war die ausschlaggebende Inspiration für eine Vielzahl neuer Einfälle. Das Resultat bildet das Konzept von iLife, welches aus drei Komponenten besteht; dem iCalendar, dem iOrganizer und dem iBlogger. Die fachlichen Anforderungen an diese Komponenten werden in Kapitel 2.1, 2.2 und 2.3 detailliert besprochen. Zunächst folgt jedoch ein Gesamtüberblick über das iLife-Konzept und das Zusammenspiel der einzelnen Komponenten.

iLife steht für intelligent Life und soll den Anwender in vielen Bereichen des Lebens bzw. des Alltags intelligent unterstützen.

Die iLife Anwendung verwaltet alle auf dem Handy befindlichen Informationen des Alltag, wie empfangene und gesandte SMSs, MMSs, erstellte Fotos, aufgenommene Videos, Termine und Events. In einem Kalender werden diese Informationen in chronologischer Reihenfolge abgebildet. Das heißt, dass ein Video im Kalender an der Stelle veranschaulicht wird, an der es aufgenommen wurde. Dieser Bereich ist zusammengefasst unter der Komponente iCalendar.

Die zweite Komponente, auf die die Verfasserin eingeht, ist der iOrganizer. Hierbei handelt es sich um eine denkfähige Planinstanz, die den Benutzer bei dem Organisieren des Alltags unterstützt. Der iCalendar bildet hierfür die Basis, der Benutzer kann über den iCalendar die Planaktivitäten, unterstützt durch den iOrganizer, durchführen. Der iOrganizer besteht aus

einer Reihe von praktikablen Funktionen, wie z.B. dem automatisierten Einladen von Freunden zu einem Event gemäß vordefinierter Bedingungen oder das Suchen nach ähnlichen Events in den Kalendern von Freunden, um z. B. gemeinsam Sportaktivitäten auszuführen. Eine ausführliche Funktionsbeschreibung findet der Leser in Kapitel 2.2.

Die iBlogger Komponente ist für die Historisierung aller Aktivitäten aus dem iCalendar zuständig. Der Benutzer hat die Möglichkeit, über den iCalendar beliebige Einträge, vorausgesetzt, sie befinden sich in der Vergangenheit, zu selektieren, um diese anschließend zu veröffentlichen. Bei den Einträgen handelt es sich um die im iCalendar befindlichen Alltagsinformationen, wie empfangene und gesandte SMSs, MMSs, erstellte Fotos, aufgenommene Videos, Termine und Events.

Bei der Anforderungsbeschreibung der einzelnen Komponenten werden die Vorteile und Beispiele für den sinnvollen und nützlichen Gebrauch der Funktionalitäten nicht aufgeführt, da sie den Rahmen dieser Ausarbeitung sprengen würden.

2.1 iCalendar

Die Basis der Anwendung bildet der iCalendar. Hierbei handelt es sich um einen intelligenten Kalender, der abgesehen von den konventionellen, zusätzliche Funktionalitäten bietet. An dieser Stelle werden nicht die geläufigen Funktionen vorgestellt, die jeder Kalender voraussetzt. Es werden lediglich die zusätzlichen Features näher erläutert.

Die Hauptanforderung an den iCalendar ist das Darstellen der Kalendereinträge. Der Benutzer hat die Möglichkeit, über eine Monatsansicht oder direkt über die Eingabe eines speziellen Tages, alle Einträge an diesem Tag zu betrachten. Zudem kann er die Standardoperationen wie löschen, erstellen usw. nutzen, die zu den konventionellen Funktionen zählen. Der Unterschied zwischen dem iCalendar und einem herkömmlichen Kalender sind die Kalendereinträge. Einträge im iCalendar bestehen nicht nur aus Terminen und Events; vielmehr werden eine Reihe anderer Einträge im iCalendar verwaltet. Nachfolgend werden die möglichen Kalendereinträge angeführt:

- empfangende und gesendete SMSs und MMSs werden im iCalendar angezeigt, einsortiert nach Empfangs- und Sendezeit. Diese werden vom Posteingang und -ausgang des mobilen Telefons entnommen.
- Fotos, die mit Hilfe des mobilen Telefons aufgenommen wurden, werden ebenfalls im iCalendar angezeigt. Die Fotos werden nach Datum und Uhrzeit der Fotographieraktivität in den Kalender eingeordnet. Zudem werden Ortsinformationen zu jedem Foto gespeichert. Die Ortsinformationen sind ein wichtiger Bestandteil für den iBlogger, Näheres dazu in Kapitel 2.3.

- Auf dem mobilen Telefon befindliche Videos werden im iCalendar aufgeführt. Die Ein-sortierung folgt gemäß der Reihenfolge der Aufnahmeaktivität der Videos.

2.2 iOrganizer

iOrganizer informiert den Benutzer, wenn ein Bekannter ähnliche Termine um die selbe Uhrzeit hat. Als Beispiel sei angeführt: Der Benutzer hat um 18:00 20:30 Uhr den Kalendereintrag Sport. Ein Bekannter hat am selben Tag um 19:30 21:00 Uhr den Kalendereintrag Squashspielen. Der iOrganizer informiert den Benutzer über die ähnlichen Termine, sodass dieser bei Bedarf die sportlichen Aktivitäten mit dem Freund zusammen durchführen kann.

Eine weitere praktikable Funktion von iOrganizer ist das Suchen nach Begriffen in anderen Kalendern. Der Benutzer gibt ein Wort ein und kann optional bei einem bestimmten Bekannten oder einer vordefinierten Gruppe nach diesem Begriff suchen.

Der Benutzer hat durch iOrganizer Einsicht in die Kalender seiner Bekannten. Jeder Kalendereinhaber hat jedoch die Möglichkeit, seine Kalendereinträge zu verbergen. Der Benutzer kann einen Kalendereintrag auf „private“ setzen und somit für alle unsichtbar machen, oder er definiert, welche Personen oder Gruppen von dieser Ansicht ausgeschlossen sind.

2.3 iBlogger

Im Allgemeinen ist ein Blog ein öffentliches Tagebuch oder Logbuch, welches als eine Webseite dargestellt wird. Der iBlogger bei iLife stellt eine ähnliche Instanz dar. Über den iBlogger ist es möglich, Kalendereinträge zu historisieren und diese als eine Art Tagebuch zu archivieren. Der Benutzer kann optional wählen, welche Kalendereinträge publiziert werden sollen. Der iBlogger veröffentlicht im Anschluss alle selektieren Kalendereinträge, gleichgültig ob es sich um Videos, Fotos, Termine etc. handelt. Wie bei einem Blog üblich, werden auch hier die Veröffentlichungen auf einer Webseite durchgeführt. Der iBlogger bietet dem Benutzer noch weitere Funktionen. Es ist möglich, zu jedem veröffentlichten Foto oder Video die Ortsbestimmung, d.h. den Standpunkt zu publizieren, an dem das Foto oder das Video entstand. Als weiteren Zusatz bietet der iBlogger dem Anwender die Möglichkeit, zu jedem veröffentlichten Eintrag einen Kommentar hinzuzufügen.

3 Architektur und Realisierungsablauf

Basierend auf den Anforderungen wurde eine Architektur vom iLife-Team konzipiert. Nachfolgend werden Überlegungen und resultierende Entschlüsse, gegliedert nach den drei Komponenten, vorgestellt.

iCalendar

Die Android Plattform bietet bereits einen konventionellen Kalender. Demzufolge soll dieser Kalender als Basis für den iCalendar dienen. Die weiteren Funktionalitäten des iCalendars, die der Android Google Kalender nicht abdeckt, werden vom iLife Team implementiert. Durch den modularisierten und lose gekoppelten Aufbau von Android ist dies ohne weiteres möglich.

iOrganizer

Für die Realisierung der iOrganizer Komponente müsste eine Kommunikationsschicht implementiert werden, um Einträge zwischen den einzelnen Anwendern austauschen zu können. Die direkte Kommunikation zwischen den einzelnen mobilen Telefonen würde zu einer Reihe von Problemen führen. Die Informationsübermittlung mithilfe von Infrarot oder Bluetooth kommt in diesem Fall nicht in Frage, da hierfür eine festgesetzte Reichweite eingehalten werden muss. Die Anwender sollen jedoch unabhängig von Ihrem Standort Kalendereinträge miteinander synchronisieren können. Eine weitere Alternative bietet das Internet. Die Voraussetzung hierfür ist, dass das mobile Telefon 24 Stunden am Tag im Internet sein muss. Das iLife-Team hat demzufolge beschlossen, eine dritte Instanz einzuführen, eine Art Server, der über alle Kalender aller iLife-Anwender verfügt und jederzeit erreichbar ist. Somit müssen sich lediglich die einzelnen iLife-Anwendungen mit dem Server synchronisieren. Das iLife-Team ist einen Schritt weiter gegangen und hat eine umfangreiche Recherche durchgeführt, um zu prüfen, ob derzeit etwas Geeignetes auf dem Markt existiert. Dabei stieß das Team auf den Google Kalender [Google Kalender]. Google bietet über das Web, Kalender an, die bereits eine Reihe sehr nützlicher Funktionalitäten zu den konventionellen Funktionalitäten anbieten, wie z.B. das Veröffentlichen von Terminen oder Events für definierte Nutzer oder das Suchen nach Terminen oder Events in Kalendern von Bekannten. Zudem bietet der Google Kalender auch diese Funktionalitäten als API [Google Calendar API] an. Angesichts dieser Tatsachen fiel der Entschluss, als dritte immer erreichbare Instanz, welche alle Kalender verwaltet, den Google Kalender zu verwenden.

Da die meisten Anforderungen an den iOrganizer ein soziales Netzwerk beinhalten, muss dieser Aspekt ebenfalls in das Konzept mit einfließen. Soziale Netzwerke sind ebenfalls mittlerweile sehr stark verbreitet und von großer Bedeutung. Infolgedessen entschied das iLife-Team sich ebenfalls, sich in diesem Punkt des Marktes zu bedienen und ein bereits vorhandenes Social Network zu nehmen. Die umfangreiche Suche und das Austesten einer Vielzahl von Social Networks führte zu dem Entschluss, Facebook [Facebook API] zu integrieren.

iBlogger

Zur Realisierung der iBlogger Komponente wird ein Webserver benötigt, der die Blog- Einträge veröffentlicht. Da das Phänomen des Blogs seit Web2.0 sehr verbreitet ist, existiert auch in diesem Bereich eine Vielzahl an Unterstützungen. Aus diesem Grund wird kein eigener Webserver und keine Webanwendung vom Team erstellt. Es wird auf bereits existierende Blogger zurück gegriffen. Eine ausführliche Marktanalyse führte dazu, Blogger [Blogger] für das iLife zu verwenden. Blogger bietet eine API [Blogger API], über die es möglich ist, die Integration des Bloggers in iLife vorzunehmen.

Um im Blog auch Fotos, Videos und Ortsinformationen zu veröffentlichen, werden weitere existierende Systeme benötigt. Diese werden über das Mashup Prinzip in den Blog integriert. Zusammenfassend werden jene externen Systeme aufgelistet, die zur Realisierung des iBlogger verwendet werden.

- Blogger: Hierbei handelt es sich um dem Blogger.
- Picasa [Picasa API]: wird verwendet, um Fotos im Blog über das Mashup Prinzip zu veröffentlichen.
- YouTube [YouTube API]: wird verwendet, um Videos im Blog über das Mashup Prinzip zu veröffentlichen.
- GoogleMaps [Google Maps API]: wird verwendet, um Ortsinformationen im Blog über das Mashup Prinzip zu veröffentlichen. Hierbei handelt es sich um Ortsinformationen, die die Position des Aufnahmeorts des Fotos oder Videos beschreiben.

Resultierend aus den Anforderungen und Entschlüssen, entstand folgende Architektur:

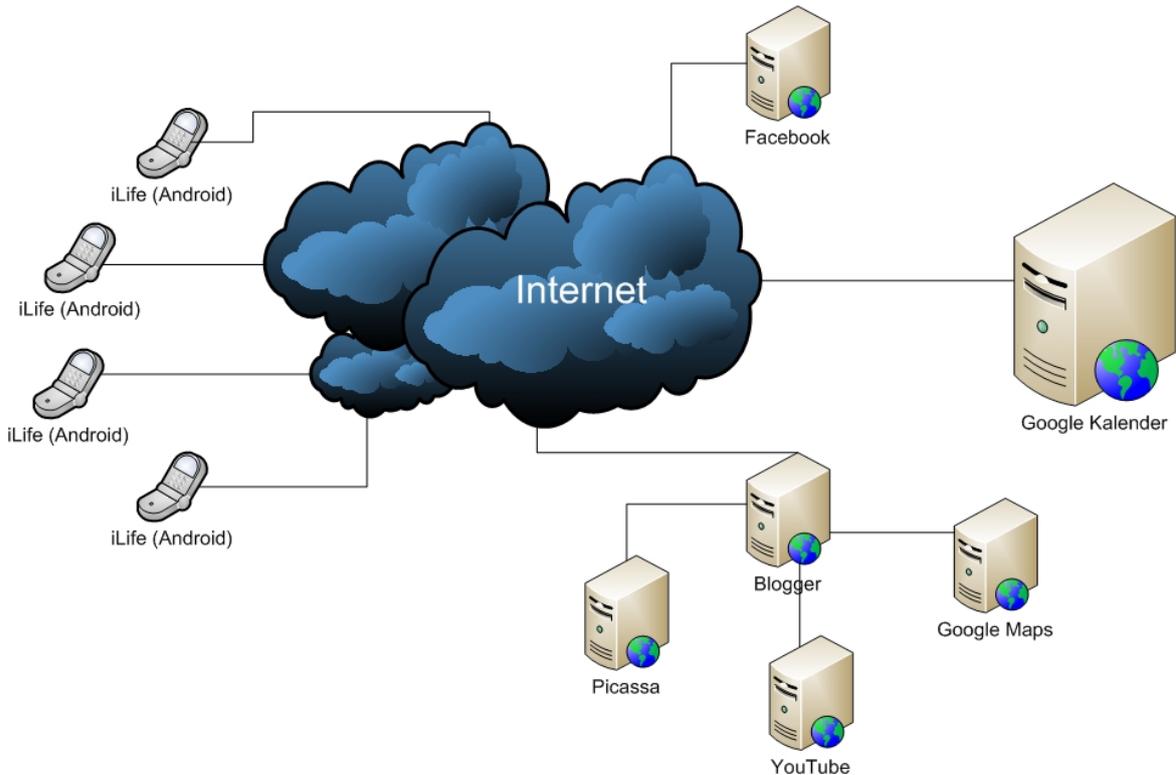


Abbildung 3.1: Architektur

Im Anschluss begann das iLife-Team mit der Realisierung. Da die Basis für die iLife Anwendung der iCalendar ist, und die Basis für den iCalendar der standard Android Calendar ist, begann die Realisierung mit der Integration des Android Calendar. Nach genauerer Recherche kam zum Vorschein, dass es sich hierbei nicht um ein OpenSource System handelt, folglich schloss das Team es aus seiner Realisierung aus.

Nach diesem Rückschlag begann eine umfangreiche Suche nach bereits existierenden Android Kalendern. Zu diesem Zeitpunkt fand das iLife-Team einige Alternativen, jedoch konnte keine der Anwendungen verwendet werden. Einige Kalender waren kommerziell, und bei den meisten war der Quellcode gar nicht erst verfügbar. Die Suche über Google Code ergab lediglich einen Android Kalender. Dieser erwies sich jedoch als nicht verwendbar, da es sich als unmöglich erwies, diesen Kalender in Betrieb zu nehmen. Die Suche und der Integrationsversuch nahmen sehr viel kostbare Zeit in Anspruch ohne Ausblick auf ein konkretes Resultat. Aus diesem Grund fiel der Entschluss, die Suche und den Integrationsversuch zu beenden und den Kalender selber zu implementieren.

An dieser Stelle wurden die Themenbereiche innerhalb des Teams grob aufgeteilt. Amine El Ayadi war zuständig für die API Anbindung an den Google Kalender, und die Verfasserin Julia Pliszka begann mit der Implementierung eines Kalenders für Android. Hierbei ist jedoch zu erwähnen, dass jede kleinste Entscheidung von beiden Teammitgliedern gemeinsam beschlossen wurde, und permanent ein Wissensaustausch über Fortschritt und Problematik stattfand.

Die Verfasserin realisierte einen konventionellen Kalender für die iCalendar Komponente. Der Kalender musste von Grund auf neu realisiert werden. Unter den konventionellen Funktionen, wie Kalendereinträge erstellen usw., wurde ebenso dem Benutzer die Wahl gelassen, über eine Monatsansicht oder über die direkte Eingabe eines Datums zur entsprechenden Tagesansicht zu gelangen. Innerhalb der Tagesansicht können Aktivitäten bezüglich der Kalendereinträge aufgeführt werden.

Parallel war Amine El Ayadi bemüht, den Google Kalender über die API zu integrieren. Zudem hat er parallel die API Anbindung des Bloggers und von Facebook exemplarisch durchgeführt. Unter anderem waren erste Versuche, Fotos über Picasa, Videos über YouTube und Ortsinformationen über Google Maps auszuweisen, erfolgreich.

Diese eben vorgestellten Tätigkeiten wurden durch enge Kooperation beider Teammitglieder durchgeführt, ungeachtet der Themenaufteilung. Die enge Zusammenarbeit führte zu einer Auflösung der Grenzlinie, insbesondere, als ein weiteres schwerwiegendes Problem auftrat. Die Anbindung der Google Kalender API über Java war erfolgreich, jedoch war es nicht möglich, über Android direkt die Google Kalender API aufzurufen. Das Android SDK ist nicht kompatibel zu Java SDK, viele Libraries sind nicht im Android SDK verfügbar. Eine Krisensitzung wurde innerhalb des iLife-Teams einberufen. Das iLife-Team fand schnell eine gute Lösung für dieses Problem. Die Lösung beinhaltet das Zwischenschalten eines Proxy-Servers. Die Anfragen werden nicht direkt vom Android an die Google Kalender API geschickt, sondern passieren den Proxy-Server. Der Proxy-Server ist nur für die Weiterleitung der Anfragen zuständig. In diesem Projekt kam der Webserver Apache 2.0 zum Einsatz, und die Kommunikation zwischen Android und dem Proxy-Server wurde über Webservices realisiert.

Nachfolgend wird die überarbeitete Architektur vorgestellt.

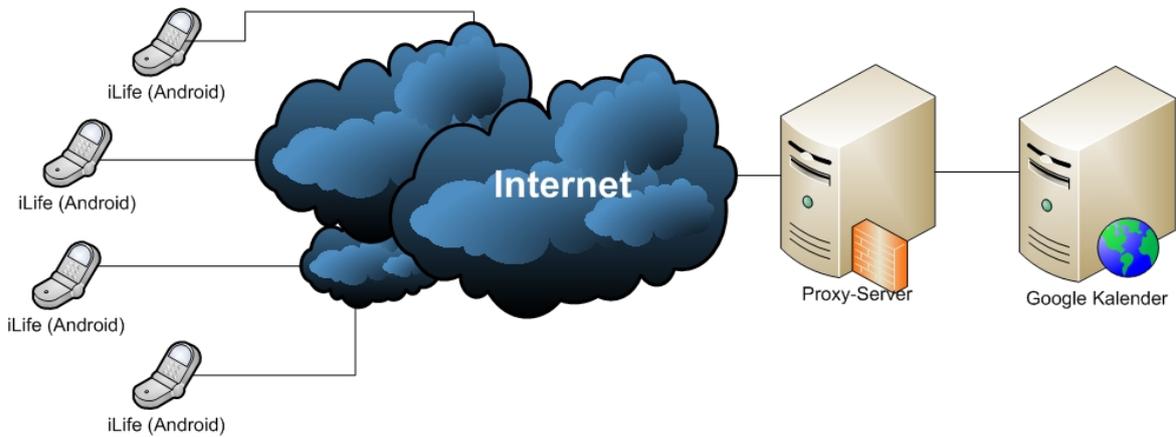


Abbildung 3.2: überarbeitete Architektur

Die Umstellung der Architektur hat viel Zeit in Anspruch genommen, jedoch führte sie zum Erfolg. Der Android Kalender interagiert mit der Google Kalender API. Es sei noch zu erwähnen, dass aus Zeitgründen bezüglich der Datenhaltung der Kalendereinträge die einfachere Variante implementiert wurde. Derzeit werden Kalendereinträge direkt an den Proxy geschickt. Termine oder Events werden nicht auf Android persistiert. Demzufolge muss bei der Erstellung von Events oder Terminen, eine Internetverbindung auf dem mobile Telefon bestehen. Bei der vorteilhafteren Variante werden erstellte Events und Termine im Android persistiert, und die Synchronisation mit dem Google Kalender kann bei der nächsten Internetverbindung durchgeführt werden.

4 Projektmanagement

Die Verfasserin hat neben ihrer Rolle als Software-Entwicklerin und Architektur-Designerin auch die Rolle der Projektleiterin übernommen.

Als erste Aufgabe hat die Projektleiterin unter Beteiligung aller Teammitglieder einen groben Projektplan erarbeitet, unterstützt durch das Programm Microsoft Project. Der Projektplan wurde im Meeting vorgestellt und war über das SVN für jedes Teammitglied jederzeit einsehbar. Jedes Teammitglied war gehalten, die Projektleiterin über Terminengpässe zu informieren. Im Laufe des Projekts wurde der Projektplan in Absprache mit allen Teammitgliedern verfeinert und überarbeitet. Alle zwei bis drei Wochen stellte die Projektleiterin die neuen Projektpläne vor. Es war deutlich zu erkennen, dass am Anfang zu viele Anforderungen in den Projektplan aufgenommen wurden. Die Zeitschätzungen der Anforderungen wurden durch alle Teambeteiligten gleichermaßen vorgenommen. Nach der zweiten Projektplanüberarbeitung konnten die Teammitglieder den realistischen Umfang der Anforderungen besser einschätzen. Demzufolge kam ein realistischer Projektplan erst in dieser Phase zustande, der auch weitgehend vom Team eingehalten wurde.

Zusätzlich zur Projektplanung führte die Projektleiterin auch ein Risikomanagement durch. Das Risikomanagement umfasste das Ermitteln, das Entschärfen und das Steuern der Risiken. Die Projektleiterin hinterlegte im SVN eine Excel-Tabelle. Versehen mit einer Deadline waren alle Teambeteiligten aufgefordert, alle von ihnen ermittelten Risiken in die Excel-Tabelle einzutragen. Nach Fristablauf sammelte und gruppierte die Projektleiterin alle Risiken. Nachfolgend wurden die Risiken nach ihrer Auftrittswahrscheinlichkeit und ihrem Schadensausmaß geschätzt. Um den Risiken entgegen zu wirken, wurden von der Projektleiterin erste Maßnahmen zur Risikominimierung ausgearbeitet. Als nächster Schritt wurden alle Risiken und Maßnahmen dem gesamten Team vorgestellt. Die Vorstellung hatte zum Zweck, die Risiken und Maßnahmen allen Teambeteiligten näher zu bringen und zusammen weitere Maßnahmen zu erarbeiten. Dieser Prozess wurde kontinuierlich fortgeführt, und alte Risiken wurden erneut aufgenommen, falls sie weiterhin eine Auftrittswahrscheinlichkeit aufwiesen. Zusammenfassend kann gesagt werden, dass das Projekt eine Vielzahl an Risiken aufwies, bedingt durch die Verwendung unbekannter Technologien, die Realisierung dreier Anwendungen, die ein Gesamtsystem bilden, Hardwarebeschaffung usw. .Trotz der Vielzahl an Risiken führte das Projekt zum Erfolg.

5 Fazit und Ausblick

Zusammenfassend kann der Projekterfolg als erzielt angesehen werden. Es wurden ein Framework und zwei Anwendungen, welche funktionsfähig sind, realisiert.

Kritisch betrachtet, ist jedoch am Ende des Projekts ein Framework mit einer darauf basierenden Anwendung und einer separaten zweiten Anwendung entstanden. Die iLife Anwendung ist zur Inselfösung mutiert. Jedoch ist der Grundgedanke, dass das iLife basierend auf dem Framework arbeitet, nicht verschwunden. Die ganzen Anbindungen an die APIs repräsentieren einen Teil des Frameworks. Aus Zeitgründen hat nicht das Framework-Team, sondern das iLife-Team, diesen Teil realisiert und diese aus dem gleichen Grund nicht mehr ins Framework integriert. Bei Fortführung des Projekts würde die API Anbindung ein Teil des Frameworks bilden

Eine Reihe weiterer Funktionalitäten, die das Framework bereits realisiert, würde in der nächsten Realisierungsphase in iLife verwendet werden. Hierbei handelt es sich z.B. um die Ortsbestimmung und die Persistierung. Zusammenfassend kann fest gestellt werden, dass das Grundkonzept eines Frameworks und zwei darauf basierender Anwendungen eingehalten wurde.

Nicht nur aus betriebswirtschaftlicher Sicht war das Projekt erfolgreich Die Verfasserin hat eine Vielzahl nützlicher Erfahrungen sammeln können; dies betrifft den technischen sowie den sozialen Aspekt. Die Realisierung eines Projekts in einem großen Team ist immer eine Herausforderung. Zudem war es wichtig, dass alle Teams kooperieren, da die Teilsysteme zukünftig eine Kooperationseinheit bilden sollen.

Ebenso war für alle Teambeteiligten der Umgang mit der Android Plattform zum Teil unvertraut. Dadurch erhöhte sich die Komplexität des Projekts noch.

Die Realisierung der iLife Anwendung kann weiter fortgeführt werden. Dieses Themengebiet bietet eine Reihe weiterer Herausforderungen, die mit Sicherheit die Kapazität einer Masterarbeit sprengt. Alleine für die Realisierung der Anforderung: nach ähnlichen Terminen suchen, ist die Auseinandersetzung mit den Themen Websemantik und Textmining notwendig.

Das Thema iLife wird mit Sicherheit noch weitere umfangreiche Forschungen erfordern.

Literaturverzeichnis

Android

<http://www.android.com/>, Abruf: 03. März 2009

Android Calendar API

<http://code.google.com/p/android-calendar/>, Abruf: 04. März 2009

Blogger

<http://www.blogger.de/>, Abruf: 28. Februar 2009

Blogger API

<http://code.blogger.com/>, Abruf: 01. März 2009

Facebook API

<http://developers.facebook.com/>, Abruf: 27. Februar 2009

Fraunhofer

<http://www.fit.fraunhofer.de/projects/mixed-reality/iperger.html>, Abruf: 01. März 2009

Google Calendar API

<http://code.google.com/intl/de-DE/apis/calendar/>, Abruf: 28. Februar 2009

Google Kalender

<http://www.google.com/calendar>, Abruf: 28. Februar 2009

Google Maps API

<http://code.google.com/intl/de-DE/apis/maps/>, Abruf: 05. März 2009

Picasa API

http://code.google.com/intl/de-DE/apis/picasaweb/developers_guide_protocol.html, Abruf: 04. März 2009

YouTube API

<http://code.google.com/intl/de-DE/apis/youtube/overview.html>, Abruf: 04. März 2009